



# *Sii-Mobility*

## **Supporto di Interoperabilità Integrato per i Servizi al Cittadino e alla Pubblica Amministrazione**

**Trasporti e Mobilità Terrestre, SCN\_00112  
Deliverable ID: DE1.2a  
Titolo: Specifica Generale,  
Specifica di Integrazione e Casi di Test**

<b>Data corrente</b>	29-09-2016
<b>Versione (solo il responsabile può cambiare versione)</b>	4.5
<b>Stato (draft, final)</b>	finale
<b>Livello di accesso (solo consorzio, pubblico)</b>	Pubblico
<b>WP</b>	WP1
<b>Natura (report, report e software, report e HW..)</b>	Report
<b>Data di consegna attesa</b>	M4, Aprile 2016
<b>Data di consegna effettiva</b>	06-06-2016
<b>Referente primario, coordinatore del documento</b>	Paolo Nesi, UNIFI DISIT Lab, <a href="mailto:paolo.nesi@unifi.it">paolo.nesi.@unifi.it</a>
<b>Contributor</b>	Tutti i partner
<b>Coordinatore responsabile del progetto</b>	Paolo Nesi, UNIFI

# Sommario

1	Executive Summary (UNIFI DISIT).....	7
2	Attori ed Enti del sistema.....	10
3	Architettura di Riferimento (DISIT).....	12
4	Regole per la Specifica Generale.....	26
4.1	Regole per gli algoritmi.....	26
4.2	Regole per gli Strumenti/tool Sii-Mobility.....	26
4.3	Esempio di Profilo descrittivo per moduli e algoritmi.....	27
5	Specifica Generale.....	29
5.1	Specifica: Sviluppo di algoritmi di data analytic e di supporto alle decisioni (Attività 2.2) 29	
5.1.1	Algoritmi e strumenti di Riconciliazione dati (A01) (UNIFI).....	31
5.1.2	Algoritmi e strumenti di Arricchimento dati (A02) (UNIFI).....	32
5.1.3	Algoritmi di ottimizzazione per la produzione suggerimenti per il parcheggio (A03) (UNIFI) 34	
5.1.4	Algoritmi e strumenti per flussi di persone e mezzi (A04) (UNIFI).....	36
5.1.5	Algoritmi di ottimizzazione anche INTERMODALE (A05, A06) (UNIFI GOL, NEGENTIS).....	38
5.1.6	Algoritmi per la produzione di percorsi per flotte merci con piu' fermate (A07) (TIME) 41	
5.1.7	Tool SmartDS (T01) (UNIFI.DISIT).....	43
5.1.8	Dashboard Builder and Manager, Monitoring: DIBUM (T02, API01, API02) (UNIFI.DISIT).....	45
5.1.8.1	Procotollo Dashboard Builder and Manager, API: DIBUM (UNIFI.DISIT).....	47
5.2	Specifica: Sviluppo del Sistema Interoperabile integrato, SII (Attività 2.1 esclusa bigliettazione 2.1.4).....	54
5.2.1	Monitoring Service Interface, e Supervisor (D01a, D01b, D02) (MIZAR).....	55
5.2.1.1	Monitoring Service Interface (D01a) (MIZAR).....	55
5.2.1.2	Monitoring Service Interface, processi interni (D01b) (UNIFI).....	57
5.2.1.3	Monitoring Process, per processi caricati (D02) (MIZAR).....	58
5.2.2	Dashboard Control Interface, DACI (T03) (UNIFI).....	59
5.2.3	Simulator Interface (T04) (UNIFI DISIT).....	61
5.2.4	Path Planner Interface (Merci e Logistica) (T05) (TIME).....	61
5.2.5	Path Planner Interface (T05) (MIZAR).....	63
5.2.6	Architettura ed Esecuzione processi (HWSW01) (UNIFI).....	65
5.2.7	Conditional Access Data, CAD (T06) (UNIFI).....	67
5.3	Specifica: Modulo per il caricamento algoritmi aggiuntivi.....	69
5.3.1	Smart City API: Process Loading (API03) (connected to DISCES) (UNIFI).....	69
5.3.1.1	Procotollo, new process loading and update.....	71
5.3.2	Smart City API Process Loaded, get data (API04) (UNIFI).....	72
5.3.2.1	Procotollo, API: dati di ritorno da processi caricati.....	73
5.3.3	Algorithm / Process Loader (T07) (TIME).....	74
5.3.4	Gestione moduli e alg, monitoraggio, accesso ai dati (T08) (UNIFI).....	76
5.4	Specifica: Infrastruttura HW del sistema integrato Interoperabile.....	78
5.4.1	Sii-Mobility Cloud HW (HWSW02) (IN20, UNIFI).....	78
5.4.2	Monitoring Services Interface, level IAAS, PAAS (HWSW03) (IN20).....	80

5.5	Specifica: prototipi applicativi verticali, sensori e attuatori.....	81
5.5.1	Apparato Bike (KIT01) (ECM).....	84
5.5.2	MobileAppBike (APP01) (UNIFI) .....	86
5.5.3	Apparato Car Bus (KIT02) (ECM) .....	87
5.5.4	Piattaforma GUIDO (T09) (ECM).....	89
5.5.5	Piattaforma Veicolare Evoluta (KIT03) (ECM) .....	91
5.5.6	Mobile App Car Bus (APP02) (UNIFI) .....	93
5.5.7	Protocollo di comunicazione fra Sensor Server and Manager ed il singolo KIT veicolare (API05) (UNIFI).....	94
5.5.8	Data Harvester dei Sensori Bluetooth dei Kit Veicolari sui mobile (MAPP16) (UNIFI DISIT) 96	
5.5.9	Processo ETL Sensor Server and Manager e Sii-Mobility (PETL24) (UNIFI).....	97
5.6	Specifica: Sensori innovativi.....	100
5.6.1	Parking Sensor (SN01) and Parking Manager (T10) (LIBEROLOGICO).....	102
5.6.2	Critical Situation Monitoring (MAPP01) (MIZAR).....	105
5.6.3	Traffic Flow Monitoring (SN03a, SN03b, T11a, T11b) (MIZAR) .....	106
5.6.3.1	Traffic and People Flow Monitoring Sensors (SN03a) (MIZAR).....	106
5.6.3.2	Traffic Flow Monitoring Sensor data (SN03b) (UNIFI).....	108
5.6.3.3	Traffic and People Flow Monitoring Tool (T11a) (MIZAR).....	109
5.6.3.4	Traffic Flow Monitoring Tool (T11b) (UNIFI).....	111
5.6.4	People Flow Monitor (T12) (UNIFI) .....	112
5.6.5	Smart Node (SN04) e Manager (T13) (PROJECT) .....	114
5.6.6	Sensori Ottici e Analisi Video (SN05) (PROJECT) .....	117
5.6.7	Sensori ambientali (SN06) e Manager (T14) (MIDRA).....	118
5.7	Specifica: Attuatori integrati per controllo accessi, direzione e velocità.....	123
5.7.1	Piattaforma Attuatori Integrati, Procedura di Segnaletica (ECM) .....	125
5.7.1.1	Piattaforma Attuatori Integrati (EHW01) (ECM) .....	125
5.7.1.2	App con Procedura di Segnaletica (APP04) (UNIFI).....	126
5.7.2	Video decisore (SN07, EHW02) (PROJECT) .....	128
5.7.3	Piattaforma Direzioneamento e Procedura Direzioneamento (EHW03, APP05) (ECM) 131	
5.7.3.1	App con Procedura Direzioneamento (APP05) (UNIFI) .....	132
5.7.4	Piattaforma rastrelliera (EHW05, APP06) (ECM) .....	134
5.7.4.1	App con Procedura Gestione Rastrelliera (APP06) (UNIFI).....	135
5.7.5	Protocollo di comunicazione Attuatore-Centrale Direzioneamento (API21) (UNIFI DISIT) 136	
5.8	Specifica: Integrazione dei sensori e attuatori verso il SII, ottimizzazione .....	137
5.8.1	Protocollo di comunicazione tra Centrale Direzioneamento ed SII (API07) (UNIFI) 138	
5.8.2	API per integrazione di nuove sorgenti dati (API08) (MIZAR).....	140
5.8.2.1	Procotollo Supervisore-SII, API: API08 (MIZAR).....	142
5.8.3	API REST di Knowledge Base su base Geolocation (API09) (UNIFI) .....	143
5.8.3.1	Protocollo esposto.....	145
5.8.4	API SPARQL di Knowledge Base (API10) (UNIFI) .....	162
5.8.4.1	Procotollo, API: SPARQL .....	163

5.8.5	API for rendering deductions on the basis of the whole ontological model of SmartDS (API11) (UNIFI) .....	163
5.8.6	API Query ID di Knowledge Base (API12) (UNIFI) .....	165
5.8.6.1	Prodotto, API: Query ID sulla KB .....	167
5.8.7	API User Crowd Sourcing (API13) (UNIFI) .....	168
5.8.7.1	Protocollo esposto, API: Crowd Sourcing .....	170
5.8.8	API User Engagement (API14) (UNIFI) .....	177
5.8.8.1	Protocollo: User Engagement .....	179
5.8.9	API User Profiling and Recommendations (API15) (UNIFI) .....	179
5.8.9.1	Protocollo esposto: Recommendations .....	180
5.8.10	API request planning (TPL e Logistic) (API16) (MIZAR) .....	186
5.8.10.1	Protocollo SII-PathPlanner, API: PathPlannerRequest .....	188
5.8.11	API Pubblicazione Dati Supervisore (API17) (MIZAR) .....	188
5.8.11.1	Protocollo Supervisore-SII, API17 – (MIZAR) .....	190
5.8.12	API verso altre smart city (API18) (UNIFI) .....	193
5.9	Specifica: Acquisizione dati da sensori .....	194
5.9.1	Acquisizione dati da Sensori ed invio a Sensor Server e Manager con call REST, Rx Comandi (MAPP02) (UNIFI.DISIT) .....	195
5.9.2	Acquisizione dati da Mobile come Sensori: Mobile APP con Wi-Fi e BT (API19) (UNIFI, ) .....	197
5.9.2.1	Protocollo, API: mobile_save_status .....	199
5.9.3	ETL Acquisizione dati da sensori ambiente (MAPP03) (MIDRA) .....	199
5.10	Specifica: applicazioni fisse e mobili campione .....	201
5.10.1	Mobile App Sii-Mobility (APP03) (UNIFI) .....	208
5.10.2	Web Server User Profiling (T15) (UNIFI) .....	210
5.10.3	Registrazione e Autenticazione e licensing per App (T16) (UNIFI) .....	212
5.10.4	Mobile APP: Modulo itinerario, percorso (MAPP04) (GEOIN) .....	214
5.10.5	Mobile APP: modulo profiling, comportamento virtuoso (MAPP05) (SOFTEC) ....	216
5.10.6	Mobile APP: modulo suggestion del tempo e nello spazio (MAPP06) (GEOIN) ....	216
5.10.7	Mobile APP: modulo parcheggio easy (MAPP07) (LIBEROLOGICO) .....	218
5.10.8	Mobile APP: modulo come ti sposti, “ma ti sposti ?” (MAPP08) (GEOIN) .....	218
5.10.9	Mobile APP: modulo tieniti in forma (MAPP09) (UNIFI) .....	219
5.10.10	Mobile APP: modulo lascia un commento, “che ne pensi.....” (MAPP10) (UNIFI) .....	221
5.10.11	Mobile APP: modulo servizi commerciali, “ma lo sai che ?” (MAPP11) (UNIFI) .....	222
5.10.12	Mobile APP: modulo partecipativo, “sarebbe opportuno che ...” (MAPP12) “sapevi che nel....” (MAPP13) (SOFTECH) .....	223
5.10.13	Mobile APP: modulo merci logistica (MAPP14) (TIME, GEOIN) .....	223
5.10.14	Mobile APP: Altri moduli per app mobile (MAPP15) (LIBEROLOGICO) .....	225
5.11	Specifica: moduli acquisizione dati e gestione ingestion process .....	226
5.11.1	Data Ingestion manager, DIM (T17) (UNIFI) .....	227
5.11.2	Sensor Server and Manager (storage and interface) (T18) (UNIFI) .....	229
5.11.2.1	Protocollo API Mobile_save_status .....	232
5.11.2.2	Protocollo API sensors_get_info .....	232
5.11.2.3	Protocollo API actuator_get_status .....	233

5.11.2.4	Procotollo API actuator_set_status.....	234
5.11.3	Distributed Smart City Engine Scheduler, DISCES (T19) (UNIFI).....	235
5.11.3.1	Protocollo esposto.....	236
5.11.4	Knowledge Base RDFStore (dati statici e real time) (T20) (UNIFI).....	256
5.11.5	Knowledge Base Statistic RDFStore (dati statistici) (T21) (UNIFI).....	257
5.11.6	RDF Indexer Manager, RIM (T22) (UNIFI).....	259
5.11.7	Moduli e processo di aggregazione / riconciliazione (A08) (UNIFI).....	260
5.12	Specifica: Acquisizione dati da enti gestori, LOD, sorgenti esterne.....	263
5.12.1	Processi ETL Static Data Ingestion, grafo e altri static data, Lista Dati da Ingerire ed integrare (PETL01) (UNIFI).....	265
5.12.2	Processi ETL per: Quality improvement (PETL02) (MIZAR).....	267
5.12.3	Processi ETL per acquisizione dati da: gestori TPL vari, web service (PETL03) (MIZAR).....	268
5.12.4	Processi ETL per acquisizione dati da: OD comuni vari, ETL (PETL04) (UNIFI) ..	270
5.12.5	Processi ETL per acquisizione dati da: Enrichment via LOD, RDF end point (PETL05) (UNIFI).....	271
5.12.6	Processi ETL per acquisizione dati da: Lamma (PETL06) (UNIFI).....	273
5.12.7	Processi ETL per acquisizione dati da: Ambiente, ARPAT (PETL07) (UNIFLLSTT) 275	
5.12.8	Processi ETL per acquisizione dati da: Autostrade (PETL08) (MIDRA).....	276
5.12.9	Processi ETL per acquisizione dati da: Ferrovie (PETL09) (MIDRA).....	278
5.12.10	Processi ETL per acquisizione dati da: Connessione con protezione civile, emergenza (PETL10) (UNIFI).....	279
5.13	Specifica: CrowdSourcing, Qualificazione delle sorgenti, integrazione multilingua....	282
5.13.1	Social Media Analysis Interface (T23) (UNIFI).....	283
5.13.2	Social Media Crawler and manager (T24) (UNIFI).....	285
5.13.3	User Crowd Sourcing Manager, UCSMANE (T25) (UNIFI).....	287
5.13.4	User Engager, UCSMANE (T26) (UNIFI).....	288
5.13.5	User Profiler, UPROSOD (T27) (UNIFI).....	290
5.13.6	Suggestion on Demand, UPROSOD (T28) (UNIFI).....	291
5.14	Specifica: strumenti di pubblicazione dati.....	294
5.14.1	SPARQL RDF store interface (T29) (UNIFI).....	296
5.14.2	Linked Open Graph, LOG (T30) (UNIFI).....	297
5.14.3	Service Map, ServiceMap (T32) (UNIFI).....	298
5.14.4	Statistic Data Map, StatisticMap (T32) (UNIFI).....	301
5.14.5	Rendering dati con varie view non solo geo grafiche (T33) (GEOIN).....	303
5.14.6	Web App Sii-Mobility per internet browser (T34) (GEOIN).....	304
5.14.7	Generazione view interattive per totem, web, e mobile (T35) (SOFTECH).....	306
5.14.8	Web App Sii-Mobility per Totem (T36) (SOFTECH).....	307
5.14.9	monitoraggio azioni utente, integrazione profiling, lato totem (T37) (UNIFI.DISIT) 309	
5.14.10	Authentication Support Tool (T38) (TIME).....	310
5.15	Specifica: Integrazione con sistemi esterni monitoraggio (MIZAR).....	312
5.16	Specifica: Sistema di partecipazione e sensibilizzazione del cittadino (GEOIN).....	313
5.16.1	Participation and Sensibilization Platform, Parti di base e presentazione informazioni (T40) (SOFTECH).....	314
5.16.2	Participation and Sensibilization Platform, raccolta suggerimenti e contributi (T39) (GEOIN).....	315

5.16.3	Participation and Sensibilization Platform, Engager, strumento di coinvolgimento (T26bis)(UNIFI DISIT, UNIFI CSL) .....	317
5.16.4	Dashboard of Participation and Sensibilization Platform (D03) (SOFTECH) .....	318
5.16.5	Participation Platform, Totem (EHW04) (TIME).....	320
5.17	Specifica: sistema di bigliettazione integrato e sua integrazione .....	322
5.17.1	Ticketing interoperabile anche in funzione dei cambiamenti (T41) (NEGENTIS)...	323
5.17.2	bigliettazione integrato: Booking (MBT01) (NEGENTIS) .....	325
5.17.3	Bigliettazione integrato: TPL e sistema toscano (MBT02) (NEGENTIS) .....	326
5.17.4	bigliettazione integrato: parcheggi vari (MBT03) (NEGENTIS) .....	328
5.17.5	Bigliettazione integrato: bike e car sharing (MBT04) (NEGENTIS) .....	329
5.17.6	Bigliettazione integrato: treno, ferrovie (MBT05) (NEGENTIS) .....	331
5.17.7	Bigliettazione integrato: ZTL e bonus (MBT06) (NEGENTIS).....	332
5.17.8	Bigliettazione integrato: Autostrada (MBT07) (MIDRA) .....	334
5.17.9	Bigliettazione integrato: integrazione su user profiling (MBT08) (NEGENTIS) ....	335
5.17.10	Bigliettazione integrato: tariffazione dinamica (MBT09) (NEGENTIS).....	337
5.18	Specifica: Integrazione con sistemi di gestione traffico (MIZAR) .....	339
5.18.1	Processo ETL: Integrazione con gestore traffico Firenze (PETL11) (MIZAR) .....	340
5.18.2	Processo ETL: Integrazione con gestore traffico Prato, Pistoia (PETL12) (MIZAR) 342	
5.18.3	Processo ETL: Integrazione con gestore traffico Siena Arezzo (PETL13) (MIZAR) 343	
5.18.4	Integrazione con gestore traffico Autostrada (PETL14) (MIDRA).....	344
5.18.5	API verso gestori traffico (API20) (IN20) .....	346
5.18.5.1	Prodotto, API: verso gestori mobilità .....	347
5.19	Specifica: Integrazione con sistemi di gestione flotte (MIZAR).....	348
5.19.1	Processo ETL: sistemi di gestione flotte: Firenze ATAF (PETL15) (MIZAR) .....	350
5.19.2	Processo ETL: sistemi di gestione flotte: Firenze provincia, BUSITALIA (PETL16) (MIZAR) .....	351
5.19.3	Processo ETL: gestione flotte: Prato, Pistoia, Pisa, CTTNORD (PETL17) (MIZAR) 353	
5.19.4	Processo ETL: sistemi di gestione flotte: Arezzo, Siena (PETL18) (MIZAR) .....	354
5.19.5	Processo ETL: gestione flotte: Merci e logistica (PETL19) (TIME).....	355
5.20	Specifica: integrazione con parcheggi, ed altri gestori servizi .....	357
5.20.1	Processo ETL: gestori servizi: parcheggi vari (PETL20) (LIBEROLOGICO).....	359
5.20.2	Processo ETL: gestori servizi: car and bike sharing (PETL21) (QUESTIT) .....	360
5.20.3	Processo ETL: gestori servizi: taxi, car service (PETL22) (QUESTIT).....	362
5.20.4	Processo ETL: gestori servizi: Nettezza e servizi (PETL23) (QUESTIT) .....	363
6	Specifica e scenari della Sperimentazione: Validazione e Sperimentazione sul campo delle applicazioni specifiche e del SII (TIME) .....	366
6.1	Azioni della sperimentazione (DISIT) .....	366
6.2	Luoghi e scenari potenziali della sperimentazione.....	370
6.3	Specifica: dimostratori sul campo, attuatori e sensori, etc. HW/SW (TIME, MIZAR)....	372
6.3.1	Specifica delle sperimentazioni come scenari e sotto scenari di sperimentazione (MIZAR, UNIFI) .....	374
6.3.2	Specifica della procedura per lo sviluppo delle sperimentazioni (MIZAR, UNIFI) .	377
6.4	Specifica: Connessione verso il SII, validazione di integrazione (MIZAR).....	378
6.4.1	Specifica: Prototipi applicativi verticali nei vari comuni (ATAF) .....	378
6.5	Specifica: usabilità e percezione delle soluzioni da parte del cittadino (UNIFI CSL).....	379
7	Acronimi .....	388

## 1 Executive Summary (UNIFI DISIT)

**Sii-Mobility** intende creare una soluzione che possa abilitare un'ampia gamma di servizi al cittadino e commerciali in connessione e integrati con il sistema di mobilità: collezionando dati puntuali e aggiornati in tempo reale da varie fonti; analizzando i flussi di dati con varie tipologie di algoritmi producendo azioni e informazioni tramite applicazioni web e mobili, totem informativi, ecc.; mettendo a disposizione dati elaborati e puntuali, che potranno essere usati da PA, gestori, e imprese per produrre servizi più efficaci ed efficienti, e anche nuovi servizi integrati. Permettendo a PA e PMI di caricare ulteriori algoritmi sul sistema per erogare servizi verso gli utenti finali e verso le PA. Per esempio algoritmi di routing, di valutazione e predizione di condizioni critiche, di ottimizzazione delle risorse, di personalizzazione dei percorsi, di guida connessa, etc.

**Sii-Mobility pone le basi per:**

- (i) **ridurre i costi sociali tramite strumenti e applicazioni innovative,**
- (ii) **semplificare l'uso dei sistemi di mobilità,**
- (iii) **contribuire verso organi di standardizzazione** come CEN e stabilire relazioni con altri sistemi di gestione Smart City, per definire insieme modelli di interoperabilità e federazione, e collaborando in modo sinergico con progetti nazionali ed Europei.

**Sii-Mobility** mira a risolvere le seguenti problematiche generali dei sistemi di mobilità terrestre:

- soluzioni di guida/percorso connesso (connect drive, smart drive o walk);
- Piattaforma di partecipazione e sensibilizzazione;
- gestione personalizzata delle politiche di accesso;
- interoperabilità ed integrazione dei sistemi di gestione;
- integrazione di metodi di pagamento e di identificazione;
- gestione dinamica dei confini delle aree ZTL;
- gestione rete condivisa di scambio dati;
- monitoraggio della domanda e dell'offerta di trasporto pubblico in tempo reale;

Il progetto **Sii-Mobility** si prefigge di produrre risultati concreti, come un **SII, Supporto di Interoperabilità Integrato**: che include:

- supporto alle decisioni, interfaccia di monitoraggio, dashboard, bigliettazione, etc.,
- supporto per la parte di Data Analytics e di intelligence sulla base dei dati raccolti dai gestori, dalle PA (Pubbliche Amministrazioni), dal social media, portali partecipativi, e dal crowd sourcing, etc.
- la possibilità di caricare (da parte di PA e PMI) algoritmi specifici addizionali che possono produrre nuovi servizi verso le applicazioni, etc.

Questo documento formalizza l'architettura di Sii-Mobility andando a specificare: la sua struttura, i singoli componenti e come saranno studiati e sviluppati, come questi si connettono fra di loro, chi li dovrà sviluppare e come saranno validati e testati tramite gli scenari di integrazione e sperimentazione.

Il documento ha preso in ingresso il documento dei requisiti DE1.1a, e come tale documento anche questo DE1.2a è la prima versione del documento di specifica di integrazione che sarà soggetto a revisione nel corso del progetto. Il documento presente è strutturato come segue:

- Sezione 2 presenta le definizioni degli attori del sistema
- Sezione 3 presenta l'architettura di riferimento
- Sezione 4 il modello con il quale è stato deciso di formalizzare questa specifica di integrazione

- Sezione 5 riporta la specifica generale scomposta in vari moduli primari, hardware e software che dovranno essere sviluppati per la realizzazione del sistema generale Sii-Mobility e per la sperimentazione nei vari scenari.
- Sezione 6 riposta la specifica della validazione e sperimentazione, che sono i casi di test.

E' parte integrante di questo deliverable il Foglio Excel Sii-Mobility-moduli-v1-6.xlsx che contiene alcuni fogli di calcolo, ed in particolare:

- Lista moduli che include per ogni modulo:
  - tipologia del modulo (utilizzato anche in questo documento);
  - codice identificativo univoco del modulo (utilizzato anche in questo documento);
  - codice identificativo univoco del task del modulo;
  - codice identificativo univoco dell'attività nel quale il modulo è sviluppato;
  - percentuale di completamento;
  - responsabile del modulo;
  - section in cui il modulo è specificato in questo stesso documento De1.2;
  - Partner responsabile per il coordinamento di sviluppo del modulo;
  - Serie di colonne che descrivono le relazioni con altri moduli e sottosistemi
  - Coinvolgimento del modulo rispetto agli scenari di sperimentazione, come descritti in section 6 di questo documento;
- Legenda con la tipologia del modulo utilizzata anche in questo documento:
  - A: Algoritmi, JAVA o ETL o C++, leggono da DB e scrivono su DB;
  - API: Application Program Interface, chiamate REST, con autenticazione per aperture del canale;
  - D: Dashboard configuration tramite il Dashboard Builder tool;
  - EHW: Embedded Hardware per direzionatore, decisore, totem, etc.;
  - HWSW: elemento integrato hardware software in cloud;
  - KIT: KIT Veicolare, hardware embedded, compatibile Android se deve eseguire App di Sii-Mobility
  - MAPP: Modulo per App di Sii-mobility, sviluppato in Cordoba, può implicare avere un Algoritmo in esecuzione su Sii di Sii-Mobility;
  - MBT: Modulo per il sistema integrato di bigliettazione e booking;
  - OD: Lista Open data come da foglio excel integrato nel documento dei requisiti De1.1a;
  - PETL: processo sviluppato con tecnologia ETL, che viene messo in esecuzione nel back office sotto la gestione del tool DISCES;
  - SN: Sensore, modulo che permette di gestire un sensore e/o relativa sensor network; sensori che vengono anche integrati direttamente dentro sistemi embedded, kit veicolari, etc.;
  - T: Tool, tipicamente sviluppato come Web Application, accessibile da web, tecnologia Java o PHP che accede a dati del SII di Sii-Mobility, produce visualizzazioni a schermo o dati su DB sempre su Sii. Può essere un servizio Web su un Web Server che espone semplicemente API, che raccoglie dati, che elabora dati, etc. Tipicamente deployato su Tomcat sempre su Sii o su Gestore del Traffico.
- Lista dei requisiti che include per ogni requisito:
  - Codice del requisito che identifica l'area;
  - Identificatore univoco del requisito;
  - Tool/Level: tool o modulo o area nella quale il requisito è centrale o comunque deve essere rispettato;
  - Ruolo Utente: tipologia di utente al quale è riferito (sviluppatore, applicazione web, applicazione mobile, utente app, gestore, amministratore, operatore mobility, ...
  - Nome del requisito, come descrizione breve;



- Declaratori, come descrizione lunga formale;
- Priorità rispetto agli obiettivi generali;
- F/NF: Funzionale o Non Funzionale;
- Lista degli Scenari di sperimentazione (questa stessa tabella viene riportata anche in Section 6) che include per ogni scenario:
  - Scenario e suo nome;
  - Coordinatore dello scenario;
  - Section in cui viene descritto nel documento dei requisiti De1.1;
  - Lista dei sottoscenari che include per ogni sottoscenario:
    - Nome come descrizione breve;
    - Identificativo univo del sottoscenario come riferito anche nella tabella Moduli nella relativa colonna;
    - Partner di riferimento per il sottoscenario;
    - Partner primari coinvolti. Tutti i partner coinvolti nel sottoscenario sono quelli i cui moduli sono coinvolti come da colonna corrispondente nella Tabella Moduli
    - Tool principali;
    - Area o aree di sperimentazione;
    - Key Indicator: indicatori primari per la valutazione quantitativa dello sviluppo e della sperimentazione dello scenario;
    - Periodi di svolgimento della sperimentazione

## 2 Attori ed Enti del sistema

Nome attore	descrizione
Analista Sii-Mobility	Esperto che sviluppa algoritmi per la piattaforma Sii-Mobility e li carica sul sistema
Decisore Sii-Mobility	Decisore politico o meno che accede al sistema di supporto alle decisioni di Sii-Mobility
Operatore ARPAT	Agenzia ambiente
Operatore Commerciali	Aziende che hanno attività commerciali in città
Operatore Ecologico, OE	Operario di azienda prelievi
Operatore Energia	Dipendente erogatore o gestore energia
Operatore LAMMA	Agenzia per il meteo in toscana
Operatore Mobilità	Operatori che lavorano in ambito mobilità, guidatori, controllori, etc.
Operatore pubblica amministrazione, OPA	Operatori della pubblica amministrazione
Operatore Protezione Civile, OPC	Dalla protezione civile della città
Operatore Sanitario, OS	Medici, ambulanze, misericordie, etc.
Operatore Sii-Mobility, OSM	Operatore che utilizza strumenti Sii-Mobility nella Centrale Sii-Mobility
Operatore UNIFI, OU	Dipendenti di UNIFI, docente, amministrativo
Sviluppatore Sii-Mobility, Developer Sii-Mobility	Sviluppatore di App o Web App sulla base di API di Sii-Mobility
Utente Finale, City User	Qualsiasi utente del sistema come city user: cittadini, pendolari, turisti, studenti, etc.
Vigile del Fuoco	Dai Vigili del Fuoco della città
Vigile Urbano	Dai Vigili Urbani della città

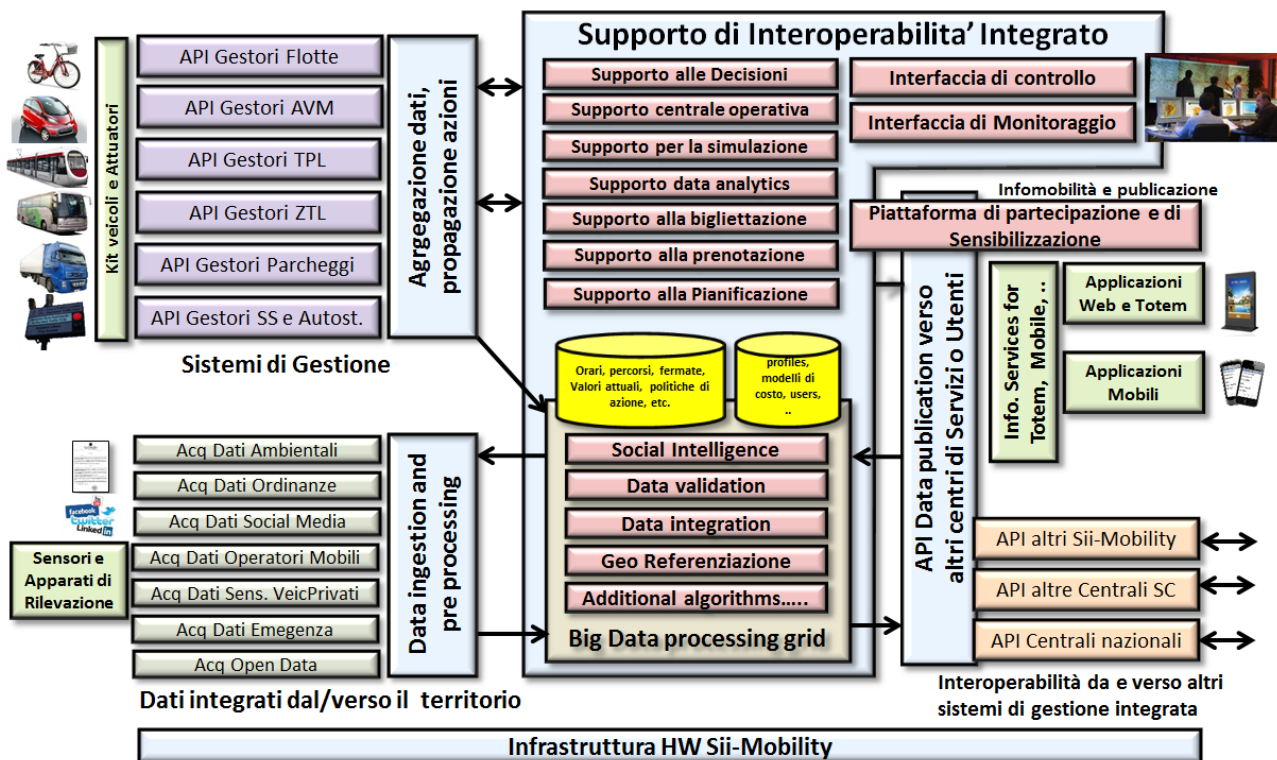
Nome ente	descrizione
ARPAT	Agenzia ambiente
Centrale Sii-Mobility	Centrale Sii-Mobility
Centrale Smart City	Centrale di una smart city
Città o Comune	Città o città metropolitana o Comune
Control Room	Centrale Smart City della Città
Gestore del traffico	Gestore del traffico, per esempio il MIIC
Gestore Sensori	Collettore di dati provenienti da Sensori
Gestore WiFi	Collettore di dati provenienti da WiFi
Gestore Energia	Ente Gestore di Energia
Gestore Parcheggio	Ente Gestore di parcheggi
Gestore SmartCity	Ente Gestore della Smart City
Gestore Telecomunicazioni	Ente Gestore Telecomunicazioni, per esempio TELECOM, Fastweb, etc.
ITS-TPL	Ente Gestore della TPL
LAMMA	Agenzia per il meteo in toscana
Gestore Raccolta Rifiuti	Gestore Raccolta Rifiuti
Protezione Civile	Protezione Civile
Social Media Service	Collettore e analista di dati provenienti da Social Network
Ticketing Service	Ente/Azienda che gestisce il pagamento dei ticket

TPL	Ente/Azienda di trasporto pubblico locale
UNIFI	Università
VVFF	Vigili del fuoco
VVUU	Vigili Urbani
Scuola Media	Scuola Media
Scuola Elementare	Scuola Elementare

### 3 Architettura di Riferimento (DISIT)

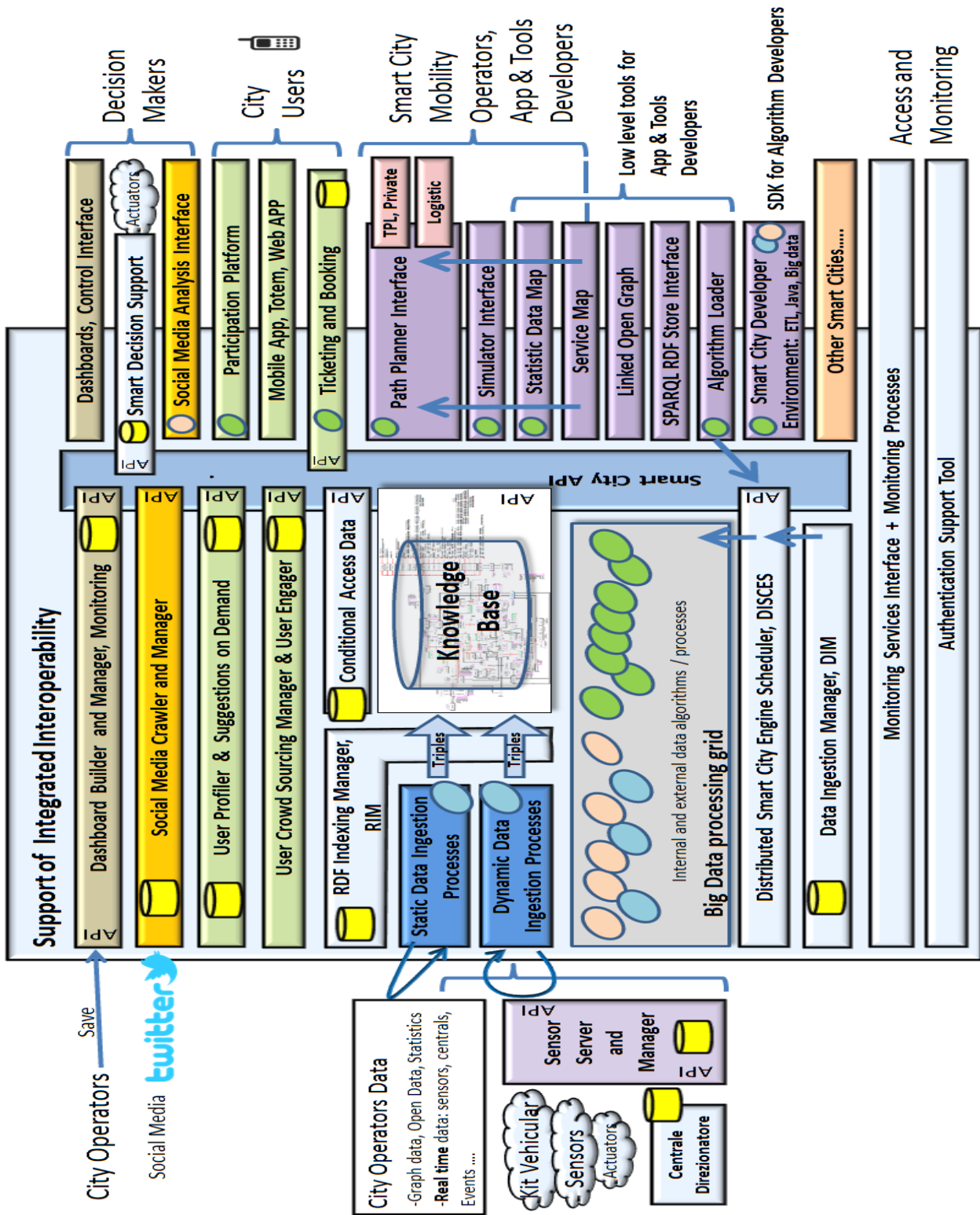
**Sii-Mobility** intende creare una soluzione che possa abilitare un'ampia gamma di servizi al cittadino e commerciali in connessione e integrati con il sistema di mobilità: collezionando dati puntuali e aggiornati in tempo reale da varie fonti; analizzando i flussi di dati con varie tipologie di algoritmi producendo azioni e informazioni tramite applicazioni web e mobili, totem informativi, ecc.; mettendo a disposizione dati elaborati e puntuali, che potranno essere usati da PA, gestori, e imprese per produrre servizi più efficaci ed efficienti, e anche nuovi servizi integrati. Permettendo a PA e PMI di caricare ulteriori algoritmi sul sistema per erogare servizi verso gli utenti finali e verso le PA. Per esempio algoritmi di routing, di valutazione e predizione di condizioni critiche, di ottimizzazione delle risorse, di personalizzazione dei percorsi, di guida connessa, etc.

Nell'architettura del progetto **Sii-Mobility** si possono notare le interfacce per la connessione con altri sistemi di Smart city, con il sistema di mobilità nazionale, la rilevazione dati ambientali, le ordinanze, etc.






**Figura 1: L'architettura di Sii-Mobility, con i dettagli interni di del SII, Supporto di Interoperabilità Integrato.**

Una versione di maggior dettaglio dell'architettura è riportata in **Figura 2** seguente che rappresenta la versione aggiornata dell'architettura completa di Sii-Mobility.



Legenda:

-  Processi ETL o Java di: Data Analytics, aggregazione ed integrazione dati, riconciliazione a posteriori, Verifica e Validazione, quality improvement a posteriori, enrichment, etc.
-  Processi ETL o Java di: acquisizione dati, quality improvement, triplicazione, riconciliazione a priori, etc.
-  Processi ETL o Java caricati tramite AlgorithmLoader per supportare processi di: routing, bigliettazione, planning (privata, pubblica, multipunto), logistica, participation, smart decision system, etc.

Le componenti primarie che saranno sviluppate per realizzare la soluzione **Sii-Mobility** sono descritte di seguito. Il sistema **Sii-Mobility** è composto da vari elementi. Le ellissi/cerchi colorate/i in verde, rosa e celeste sono processi ETL o Java che vengono messi in esecuzione nel back office e gestiti come processi a cura del **Distributed Smart City Engine Scheduler, DISCES**. Lo stesso strumento è a servizio di vari tool anche di front end che ne governano il comportamento permettendo la messa in esecuzione di processi e la lettura del suo stato. La descrizione che segue permette di mappare i nomi e le descrizioni dei moduli proposti nel capitolato rispetto ai nomi dei sistemi e sottosistemi finali descritti nei requisiti e in questo documento di specifica.

- **SII, Supporto di Interoperabilità Integrato:** che include:
  - **supporto alle decisioni**, formalizzazione di criteri di allarme e di alert che possono identificare condizioni critiche, correlazioni non attese, etc.
    - **Sottosistema: Smart Decision Support** che adotta un modello System Thinking per il supporto alle decisioni e accede ai dati della knowledge base (KB con le informazioni della città) come di altri database, per esempio quelli della dashboard, del sistema di social media, etc.
  - **interfacce di controllo e di monitoraggio**, per poter supervisionare il sistema in termini di flussi dati, calcolo delle regole decisionali e di allarme, etc.
    - **Sottosistema: Monitoring Services Interface + Monitoring Processes.** Questo è il sistema di monitoraggio dell'intera soluzione smart city Sii-Mobility. Tale Sottosistema controlla l'esecuzione ed il funzionamento delle macchine virtuali e dei processi. Riporta alcune informazioni direttamente sulla Dashboard tecnica della smart city, per esempio il numero di colpi di clock usati al secondo, lo stato della memoria, il numeri di processi eseguiti, il numero di macchine virtuali attive rispetto a quello configurato, lo stato di salute del sistema, etc.
  - **Supporto alla centrale operativa:** per la visualizzazione dei dati e degli eventi primari in termini di grafici e modelli sinottici che possono offrire una visione complessiva dello stato della area. Uso di soluzioni di grafica multischermo.
    - **Sottosistema: Dashboard Builder and Manager, Monitoring.** Soluzione che permette la costruzione di DashBoard specifiche ad utenti come PA, PMI, operatori. Tali Dashboard sono quelle gestite dalla DashBoard Control Interface che le rende disponibili con accesso controllato e ne gestisce ticket, valori, etc.
    - **Sottosistema: Dashboard, Control Interface.** Modulo che permette l'accesso alle Dashboard prodotte dal Dashboard Builder e Manager.
    - Questi sottosistemi permettono di accedere e collezionare dati di alto livello da vari database e API per mostrarli con sinottici consuntivi configurabili. Inoltre accetta che alcuni operatori comunichino in push alcuni loro dati invece che esporre i loro dati su servizi API. Il Dashboard Manager presenta della API per poter ricevere tali dati in PUSH con chiamate REST anche autenticate. Il collezionare dati può essere visto che uno spreco ma nella realtà permette di accumulare dati di alto livello che appartengono a più domini e da questi poter fare considerazioni e prendere decisioni tramite altri strumenti. Lo strumento di Dashboard deve permettere anche il replay di situazioni collezionate in passato.
  - **Supporto alla simulazione** per verificare gli effetti di cambi nei sistemi di controllo traffico, negli orari dei servizi. Etc.
    - **Sottosistema: Simulator Interface.** La simulazione viene intesa come simulazione del comportamento di traffico o del sistema in certe condizioni. La stessa simulazione può consistere nel replay di situazioni di flusso e nella rivalutazione di strategie diverse da quelle attive al momento della raccolta dati.

- **Sottosistema: Distributed Smart City Engine Scheduler, DISCES.** Questo tool svolge la funzione di scheduler distribuito, attivando processi su vari macchine virtuali del **Big Data Processing Grid**. I processi vengono monitorati e controllati, i processi possono essere attivati in cascata ed attivare segnali di allarme e/o condizioni, i processi possono essere periodici o aperiodici.
- **supporto per la parte di Data Analytic** e di intelligence sulla base dei dati raccolti dai gestori, dalle PA, dal crowd sourcing, etc.
  - **Algoritmi vari che stanno sul Big Data Processing Grid** che vengono messi in esecuzione dal DISCES: acquisizione dati, calcolo suggerimenti, calcolo traiettorie, calcolo matrici origine destinazione, calcolo cluster di tracce, regolarizzazione velocità delle persone e flussi, verifiche di consistenza, etc. etc. La visualizzazione di alcuni effetti può essere effettuata sfruttando le caratteristiche e le funzioni di ServiceMap.
  - **Sottosistema: Distributed Smart City Engine Scheduler, DISCES.** Si veda sopra.
- **supporto alla bigliettazione integrata**, etc., integrazione dei sistemi di bigliettazione integrata della regione e delle TPL.
  - **Sottosistema: Ticketing and Booking.** Questo sottosistema permette di produrre biglietti in modo intermodale e multivendor. Queste funzioni sono anche esportate come servizi che possono essere usati da Web e Mobile App. Gli algoritmi di computo intermodale come quelli che calcolano i bonus dovrebbero poter essere messi in esecuzione su Big Data processing Grid. Il sistema di bigliettazione integrata dovrebbe poter integrare sistemi di pagamento diversi, e non solo per il trasporto pubblico, ma anche per il parcheggio, pedaggi, entrate in ZTL, etc.
- **Supporto alla prenotazione:** per fornire informazioni di previsione su certi eventi pianificati e inattesi. Per esempio per la prenotazione dei prelievi, per la prenotazione delle consegne.
  - **Sottosistema: Ticketing and Booking.** Il sistema di Booking dovrebbe fare riferimento esplicito ai sistemi di prenotazione per parcheggi, prelievi di merci, entrate in ZTL a numero chiuso, e per il trasporto pubblico.
- **supporto alla pianificazione** per avere informazioni statistiche dei flussi di persone nelle varie zone e sui vari mezzi. **Sottosistema: Path Planner Interface,**
  - **TPL e Privati.** Questo modulo permette la definizione di percorsi da-a per privati e TPL in modo che l'operatore possa ottenere la soluzione ottima o sub-ottima. La pianificazione delle traiettorie per il trasporto pubblico deve essere multimodale più tipi di trasporto pubblico (diversi operatori bus, operatori treno, etc.) ed inserimento di tratte a piedi. Per il trasporto privato si assume auto, moto, furgone, si massimizza il raggiungimento della meta, si possono avere pedaggi di vario tipo ma vanno evidenziati, etc. Per il trasporto privato si può avere anche il trasporto via bike. In questo caso si dovrebbero preferire le ciclabili.
  - **Logistic.** Questo modulo permette la definizione di percorsi di prelievo in modo che l'operatore possa ottenere la soluzione ottima o subottima, l'operatore può chiedere percorsi che prevedono fermate per prelievi e consegne multiple lungo per percorso, e anche punti di scarico convenzionati.
  - **Questi modulo sfrutta Smart City API, ServiceMap, etc.** e sviluppa questo servizio come algoritmi caricati sul Big Data Processing Grid che espongono i servizi come risultati tramite le API. La stessa chiamata di questi algoritmi viene a fare parte delle Smart City API di Sii-Mobility e potrà essere usata da applicazioni web, mobile e kit veicolari.
- **Big data processing grid tool per la gestione e il semantic computing sui contenuti big data che saranno sia puntuali che storici:**

- **Database di conoscenza, Knowledge base**, sviluppato utilizzando tecniche di Big Data tramite in NoSQL database in modo da garantire elevate prestazioni anche con volumi di dati estremamente grandi;
  - **Sottosistema: Knowledge Base.** La Knowledge Base di Sii-Mobility si basa su tecnologia Km4City, ed integra aspetti innovativi non presenti in Km4City come dettagli sugli aspetti di mobilità per esempio: dettagli stradali per il routing (direzioni, svolte, velocità, lunghezze, tipologia strada, tipologia mezzi consentiti), orari TPL, orari treni, licensing, sensori evoluti, etc.
- **Social Intelligence:** strumenti per l'analisi degli umori e dei commenti dei cittadini, questi saranno recuperati dai social media come Facebook, Twitter, etc. tramite soluzioni di natural language processing;
  - **Sottosistema: Social Media Crawler and Manager.** Strumento di acquisizione post su social media, per esempio da Twitter. Sulla base di certe chiavi di ricerca vengono scaricati i post. Le chiavi possono essere inerenti al servizio TPL, ai problemi di traffico, nelle aree di interesse, etc.
  - **Sottosistema: Social Media Analysis Interface.** Strumento per l'analisi dei post collezionati dal Social Media Crawler and Manager, con analisi statistica, NLP e sentiment analysis in automatico. Anche in tempo reale per certi dati. L'analisi può essere mirata alla valutazione della qualità del servizio, all'identificazione di problematiche anche in tempo reale, alla rilevazione delle opinioni degli utenti, correlazione fra questi aspetti e gli eventi in città.
  - Questo sottosistema acquisisce informazioni da canali social media come Twitter (principalmente, ma può essere esteso) per capire gli umori della città rispetto ai servizi di mobilità ma anche per capire se questi strumenti possono essere usati per effettuare misure di traffico, misure di flusso delle persone, etc. Sono parametri utili anche per la Control Interface (dashboard).
- **Data validation and integration:** algoritmi di riconciliazione dati, soluzioni di validazione e verifica dei dati ricevuti, ricerca ed identificazione di inconsistenze nei dati ricevuti e accumulati, algoritmi di clusterizzazione dati per profili e tracce, etc.;
  - **Sono Algoritmi sviluppati come processi ETL e/o java in esecuzione su Big Data processing grid**, che accedono a Knowledge Base ed altri dati e DB, e che sono gestiti dal DISCES. Lo sviluppo di questi algoritmi ETL viene effettuato tramite un ambiente di sviluppo messo a disposizione chiamato: **Smart City Developer Environment** per processi: ETL, Java, BigData.
- **Data Integration:** algoritmi per la riconciliazione dei dati, completamento dei dati, razionalizzazione dei dati che provengono da sorgenti diverse per uniformare date, nomi, luoghi, aggregazione dati, per generare matrici OD, per creare cluster, etc.;
  - **Sono Algoritmi sviluppati come processi ETL e/o java in esecuzione su Big Data processing grid**, che accedono a Knowledge Base ed altri dati e DB, e che sono gestiti dal DISCES. Lo sviluppo di questi algoritmi ETL viene effettuato tramite un ambiente di sviluppo messo a disposizione chiamato: **Smart City Developer Environment** per processi: ETL, Java, BigData.
- **Geo referenziazione.** La regione Toscana ha un data base dove sono riportate in modo puntuale tutte le informazioni relative alle strade della toscana, ed attività commerciali, posizione dei numeri civici, etc. Alcuni dati provenienti da open data, oppure da altre sorgenti non sono riferiti in modo preciso con coordinate GPS; pertanto il processo di associazione di questi dati a coordinate viene detto di geo referenziazione.
  - **Sono Algoritmi sviluppati come processi ETL e/o java in esecuzione su Big Data processing grid**, che accedono a Knowledge Base ed altri dati e DB, e che sono gestiti dal DISCES. Lo sviluppo di questi algoritmi ETL viene effettuato tramite un



ambiente di sviluppo messo a disposizione chiamato: **Smart City Developer Environment** per processi: ETL, Java, BigData.

- **Additional Algorithm:** Questo modulo include la possibilità di caricare (da parte delle PA e delle PMI) algoritmi specifici che possono produrre nuovi servizi verso le applicazioni, verso altre centrali di Smart City, verso la piattaforma nazionale trasporti.
  - **Sottosistema: Smart City Developer Environment** per processi: ETL, Java, BigData. Questo sottosistema consiste in una sorta di SandBox per lo sviluppatore cioè di un Sistema di sviluppo integrato per sviluppare algoritmi in ETL, java, etc. Questi algoritmi, una volta sviluppati e provati in questo ambiente, possono essere sottomessi per l'esecuzione su **Big Data Processing Grid** tramite l'**Algorithm Loader**. La **Smart City Developer Environment** viene fornita con una VM (Virtual Machine, <http://www.disit.org/6690> ), include documentazione ed alcuni esempi.
  - **Sottosistema: Algorithm Loader.** Questo tool permette la descrizione, il caricamento, la pianificazione della sua esecuzione, il monitoraggio e l'accesso ai risultati di algoritmi caricati da PMI e terze parti sul sistema Sii-Mobility della Smart City. Lo sviluppatore realizza questi algoritmi, li sottomette, vengono presi in carico, validati e verificati dal personale di Sii-Mobility, ed infine messi in esecuzione e gestiti dal DISCES. Tali algoritmi devono presentare due moduli: (i) il modulo di esecuzione che include accesso ai dati, elaborazione, e scrittura dati; (ii) il modulo di accesso ai risultati come Smart City API aggiuntive e specifiche per l'algoritmo caricato.
- **Moduli di Aggregazione e Propagazione Azioni per sistemi di gestione:** in grado di:
  - acquisire data da vari tipi di gestori come: flotte, AVM, TPL, ZTL, Parcheggi, autostrade, etc., (questi dati vengono integrati insieme a quelli provenienti da sensori, metro/tramline, geografici delle PA, etc.);
    - I processi di acquisizione dati, sono sviluppati come programmi ETL e messi in esecuzione su Big Data processing grid. Questi processi ETL accedono a file (statici) o a dati tramite chiamate REST of WS (dati real time), come processo finale producono triple per la Knowledge Base. Tali processi sono gestiti dal DISCES. Lo sviluppo di questi algoritmi ETL viene effettuato tramite un ambiente di sviluppo messo a disposizione chiamato: **Smart City Developer Environment** per processi: ETL, Java, BigData.
  - modelli e soluzioni di comunicazione per collezionare dati provenienti da sensori che si muovono sui veicoli nella città;
    - **Sottosistema: Sensors Server and Manager:** questo sottosistema permette di collezionare dati da kit veicolali, da sensori avanzati, etc. Il sottosistema riceve dati da tali apparati in push, li colleziona ed Esporta delle API per poter fornire gli ultimi valori di tali dati. In certe occasioni il Sensors Server and Manager deve poter accettare anche dati in uscita. Cioè deve essere possibile sempre tramite chiamate REST poter chiamare il Sensors Server and Manager per comunicare valori di setting a certi sensori specifici (lampioni, direzionatori, video decisori, etc.). La stessa modalità permette di scrivere valori di controllo.
    - protocolli di comunicazione per accedere ai concentratori di dati provenienti da sensori. Un esempio di concentratore è il **Sensor Server and Manager** che espone delle API per i sensori evoluti di Sii-Mobility, e per altri dispositivi.
  - restituire ai gestori (della mobilità come ad altri operatori in città) informazioni integrate, deduzioni, andamenti che possono produrre per loro valore aggiunto ai loro servizi: conoscere l'arrivo di flussi puntuali, conoscere l'andamento dei flussi medi (per veicoli e persone), ricevere allarmi/notifiche sulla base di accadimenti e condizioni complesse. Produrre questi dati

non è facile né possibile per i singoli gestori, mentre è fattibile per un controllore a livello smart city come Sii-mobility. I dati saranno accessibili per i gestori (operatori smart city, city operators) tramite varie forme e canali. Ed in particolare tramite:

- **Smart City API:** servizi, ricerca, dati per ogni servizio, dati statici, storici e real time, etc.
- **Smart Decision Support:** dati consuntivi di processi di decisione;
- **Dashboard Control Interface:** dati di andamento ed eventuali allarmi/notifiche degli stessi rispetto a soglie;
- **Service map:** come funzioni di embedding dello stato della Knowledge base;
- **Mobile e Web Application:** dati di varia forma accessibili da web e mobile, suggerimenti, etc., sempre tramite Smart City API;
- **Path planner interface:** percorsi pianificati punto, pianificati con sequenze di punti per movimenti di TPL, di privati e per la raccolta di materiale, sempre tramite Smart City API.
- **Social Media Analysis Interface:** strumento di monitoraggio per eventi e trend nel social media connessi ad aspetti Smart City e Mobilità.
- **Azionare sistemi di attuazione:** abilitazione porte di ingresso (ZTL, etc.), gestione direzione in strade, abilitazione porte in parcheggi.
  - **Sensor Server and Manager** espone delle API anche per attuare azioni come azionare porte ZTL, gestire la direzione, abilitare parcheggi, etc. Le azioni possono essere (i) direttamente definite con regole sulla **Centrale Direzione** o sul **Sensor Server and Manager** (per esempio per concertare alcuni direzionatori), oppure (ii) da parte di processi definiti con il sistema di **Smart Decision Support**. Si noti che il Sensor Server and Manager può ricevere consensi anche da sensori evoluti come video camera (Video Decisori).
- **Gestione avanzata di mezzi** come bici, auto, e bus, bici ed auto private; produzione di informazioni integrate per tali mezzi e gestori; veicolo come sensore, controllori veicolari:
  - Il veicolo fornisce informazioni di dettaglio sulle condizioni del traffico (XFCD), es: velocità, destinazione, intenzione di svolta, tipo di veicolo, coda, ....
  - Al veicolo si può fornire informazioni di guida connessa in base alla velocità, posizione, obiettivi di percorso, stato dei semafori, sensori avanzati come: GPS, CANBUS, meteo, velocità, punto di destinazione, etc.
  - Sul veicolo si possono utilizzare sistemi a microprocessore con Android, con App, sensori, e capacità di geolocalizzazione e comunicazione, tipo veicolo, etc..
  - Si possono chiedere al conducente informazioni di stato tramite tool di **Engagement**: che coda vedi, quanto tempo stai aspettando, perché sei fermo, etc.
  - I driver possono ricevere suggerimenti e raccomandazioni (**Suggestion on Demand**), percorsi ottimi (**Path Planners algorithm**), rallenta per ..., ti conviene parcheggiare ora...
- **Moduli di Acquisizione e preProcessing:** vengono utilizzati per acquisire i dati che provengono da ambiente, ordinanze, social media, operatori mobili, sensori su veicoli pubblici e privati, stazioni e canali dell'emergenza, vari open data, etc. In alcuni casi, questi dati devono essere preprocessati per poter essere incamerati ed integrati nel modo corretto. Questo strumento include un supporto all'acquisizione di feedback e commenti dei cittadini verso aspetti connessi alle tematiche di Sii-Mobility e alle applicazioni caricate su Sii-Mobility. Dati che provengono da:
  - **Ambiente,**
    - Sottosistema: **Static Data Ingestion** processes e **Dynamic Data Ingestion** processes. La parte statica riferisce alle posizioni GPS di misura, alla struttura della

misura in termini di campi e significato. Mentre la parte dinamica alle misure vere e proprie: pioggia, inquinamento, vento, neve, umidità, pressione, etc.

- Algoritmi sviluppati come **processi ETL**, come descritto in precedenza;
- **Ordinanze della pubblica amministrazione**,
  - Sottosistema: **Static Data Ingestion processes e Dynamic Data Ingestion processes**. Le ordinanze dovrebbero essere prodotte in formati processabili dal calcolatore per esempio shape. L'ordinanza può prevedere cambi di viabilità che devono essere descritti in modo accurato e non ambiguo, con tanto di data di inizio e data presunta di conclusione.
  - Algoritmi sviluppati come **processi ETL**, come descritto in precedenza.
- **Sensori su Kit Veicolari**: sensori su veicoli pubblici e privati tramite kit standard e kit avanzati sviluppati per Sii-Mobility,
  - Sottosistema: **Sensors Server and Manager** e processi di **Static Data Ingestion e Dynamic Data Ingestion**. I kit veicolari sviluppati in Sii-Mobility e descritti in seguito, possono avere a bordo una serie di sensori di vari natura (ambiente, velocità, connessione al CANBUS del veicolo, ingombro, conta persone, temperatura interna ed esterna, etc.). Questi devono essere letti dal mobile e inviati alla centrale **Sensors Server and Manager** che gli aggrega e li rende disponibili per Sii-Mobility che li legge come dati statici e dinamici. La parte statica è quella relativa all'anagrafica di tali sensori e kit mentre quella dinamica è relativa ai valori, la posizione, etc.
  - Algoritmi sviluppati come **processi ETL**, come descritto in precedenza per la lettura di dati statici e dinamici dal **Sensors Server and Manager**. Mentre sono gli stessi kit veicolari a notificare i nuovi loro dati verso lo stesso **Sensors Server and Manager**, in modo asincrono e periodico, in push.
- **Social Media**
  - Questo strumento permette di collezionare e trarre vantaggio delle informazioni di dettaglio e commenti dai blog sul territorio come da social network e di comprenderle per trarre deduzioni e feedback nel sistema centrale di conoscenza. Per questo saranno utilizzati strumenti di crawling e natural Language processing.
  - Sottosistema: **Social Media Crawler and Manager** verso **Social Media Analysis interface**. Questi sottosistemi permettono di rendere disponibili per la soluzione Sii-mobility delle informazioni di contesto importanti riguardo all'uso dei servizi, all'apprezzamento di questi, ad eventuali eventi burst, etc. L'acquisizione di questi dati viene effettuata direttamente con un accesso a delle chiamate REST verso il sottosistema **Social Media Crawler and Manager**. Le chiamate possono essere per ottenere gli ultimi Tweet come per avere dati relativi ai volumi di Tweet, retweet, etc. per un singolo canale di ascolto, per una singola ricerca, o per gruppi, etc.
- **Crowd Sourcing**:
  - Sottosistema: **User Crowd Sourcing Manager and User Engager**, in connessione alla piattaforma di partecipazione e sensibilizzazione. Questo sottosistema dovrebbe permettere di collezionare commenti, voti e contributi (immagini per esempio) dai singoli cittadini.
  - **User Crowd Sourcing Manager** è: (i) un canale di ascolto dei cittadini per le loro problematiche e notifiche di problemi riscontrati sui servizi, (ii) un canale per collezionare nuove informazioni su servizi attivi e anche nuovi. Il caricamento di queste informazioni deve essere permesso in modo moderato tramite chiamate che fanno riferimento alle Smart City API.
  - L' **User Engager**: presenta problematiche di maggiore complessità. L'obiettivo è poter informare ma anche coinvolgere il city user con dei compiti specifici: fai una foto del punto di interesse (o del servizio) POISRVxxx, come vedi la coda dove sei,

quanto è pieno l'autobus, che valutazione di qualità daresti sul servizio, etc. Una vera e propria spinta a coinvolgere il city user a fornire delle opinioni e partecipare e fornire informazioni tramite il loro mobile. Le informazioni sono ovviamente sul sistema di mobilità, sui servizi di mobilità ma anche sui servizi integrati di smart city. Allo stesso tempo, e sullo stesso canale e mobile, il cittadino può ricevere informazioni utili come: non parcheggiare qui perché non è consentito, parcheggia allo scambiatore che spenderesti meno, usa i mezzi pubblici che spendi meno energia, domani ti suggerisco di fare una strada diversa, etc. Se il city user raccogli i suggerimenti inviategli il sistema deve poterlo verificare.

- **Open Data** (dati geografici della regione Toscana, GRAFO regionale), etc.
  - Sottosistema: **Static Data Ingestion** processes e **Dynamic Data Ingestion** processes. I dati statici fanno riferimenti a dati delle regione , delle province e dei comuni coinvolti come da **Allegato 1 al documento dei requisiti D1.1a**. I dati dinamici fanno riferimento a dati che possono essere reperiti dall'osservatorio trasporti della Città Metropolitana di Firenze (ex provincia), dal MIIC della regione Toscana (grafo regionale), dalle varie TPL, dal LAMMA, da Arpat, Open Street map, Google, etc. Alcuni di questi dati dinamici non sono in effetti Open Data, cioè sono coperti da license simili ad Open Data ma non sono completamente open, possono avere delle restrizioni che ne impediscono un utilizzo commerciale (No Commercial dei modelli di licenza CC, Creative Commons), per esempio.
  - Algoritmi sviluppati come **processi ETL**, come descritto in precedenza.

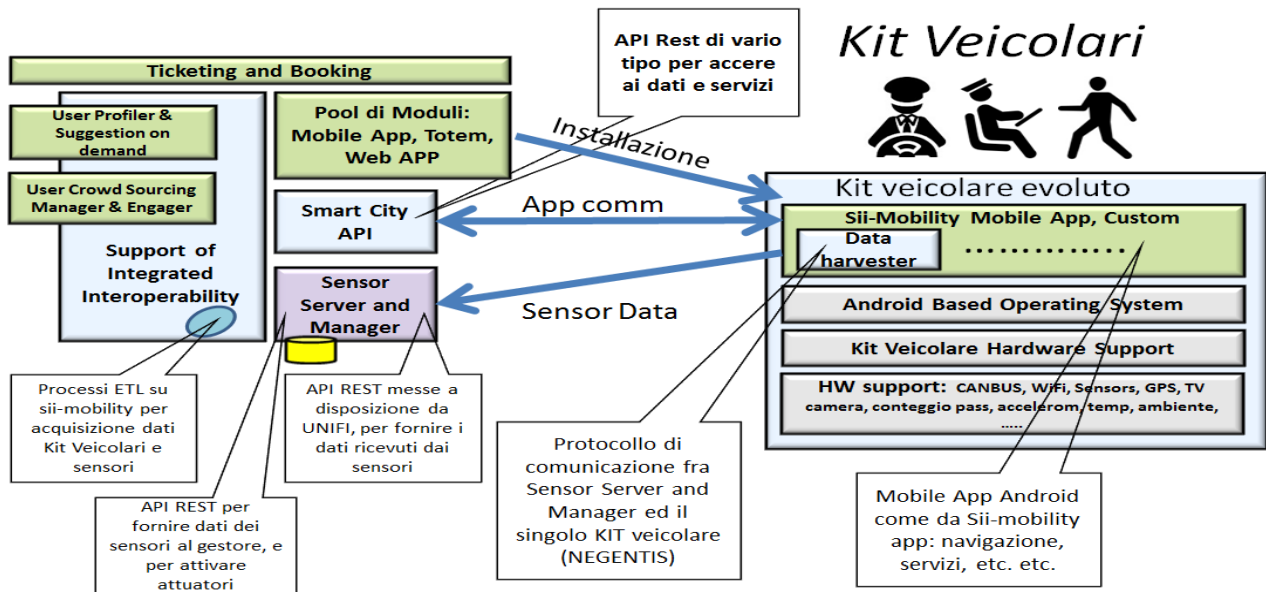
**Kit Veicolari e Apparati di Rilevazione:** in Sii-Mobility saranno sviluppati kit ed apparati di rilevazione avanzati in connessione con la loro corrispondente centrale di controllo e da questa con Sii-Mobility. In particolare:

- **Kit Veicolari per bike:** da installare su bike sharing, per la rilevazione di posizioni, di parametri ambientali e di inquinamento, di velocità, etc.
- **Kit Veicolari per CAR e BUS:** da installare su Bus di TPL e/o su veicoli privati, per la rilevazione di posizioni, on/off del veicolo, tempi di sosta, vari altri parametri del mezzo, etc. (questi kit saranno utilizzati per le sperimentazioni su auto pubbliche e BUS di TPL) . Tipicamente soluzioni Android in modo che possa ospitare/installare una sorta di App simile a quella dei sistemi mobili.
- **Kit Veicolari per CAR e BUS evoluti:** Alcuni questi kit saranno dotati di ampia capacità computazionale per sperimentare e sviluppare algoritmi sofisticati: conteggio passeggeri, acquisizione e gestione telecamere, etc., con elevata capacità di comunicazione con il conducente per fornire direttive, percorsi, suggerimenti, etc. Tipicamente soluzioni Android in modo che possa ospitare/installare una sorta di App simile a quella dei sistemi mobili.

**I Kit Veicolari** fanno riferimento alla seguente architettura di massima, che li vede interagire con Sii-Mobility tramite:

- **Smart City API** in modo del tutto simile alle applicazioni da Mobile prodotte per Android, iOS e MS phone. L'interazione è tramite chiamate REST da parte del mobile verso le Smart City API per acquisire in modo sincrono dati, suggerimenti, engagement, assistenza, etc. *Questo processo può implicare anche la notifica della posizione GPS, le rilevazioni di sensori Wi-Fi e Bluetooth, etc.*
- **Sensor Server and Manager API** per notificare (salvare, inviare) allo stesso valori rilevati sul campo in modo periodico riguardo allo stato del veicolo, alle rilevazioni di sensori, etc. Queste informazioni possono essere utilizzate per completare il profiling del mezzo, il profiling della persona, il profiling collettivo, ed in ultimo per identificare condizioni di stato ed evoluzione di stato per scatenare processi di Allerta, Notifica, Engagement, Assistenza, Raccomandazione,

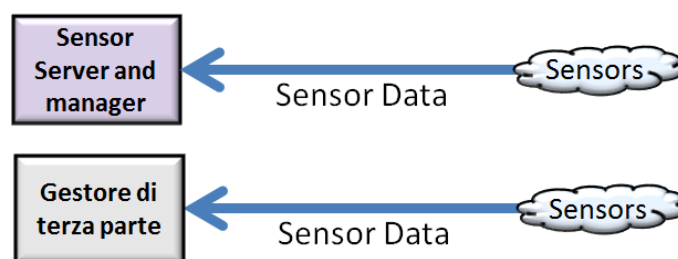
etc. dai vari tool descritti in precedenza come: **User Crowd Sourcing Manager and User Engager, Smart Decision Support, User Profiler & Suggestion on Demand, etc.**



#### Altri Sensori e Sottosistemi di Sensori specifici:

- **Sensori per parcheggi:** rilevazione della presenza dell'auto, rilevazione di accesso, pagamento automatico, etc. Utilizzanti in connessione con applicazioni mobili e soluzioni per il pagamento automatico. I sottosistemi di gestione parcheggio dovrebbero poter comunicare o rendere disponibile per Sii-Mobility le informazioni statiche e dinamiche. Sono informazioni statiche la posizione del parcheggio, la capienza, gli orari tipici. Sono informazioni dinamiche il numero di posti liberi, le variazioni sugli orari di apertura, etc. I parcheggi sono tipicamente con controllo accessi (interrati, con silos, con sbarra, etc.), oppure a raso, come normalmente quelli lungo strada, etc. I primi sono abbastanza facili da controllare, mentre i secondi sono molto difficili da controllare in modo preciso.
- **Sensori per l'identificazione di condizioni critiche** configurabili e flessibili per la rilevazione di condizioni ambientali, stradali e di flusso, per la misurazione delle condizioni, tipicamente installati presso sistemi di illuminazione, totem, etc. Questi apparati sono tipicamente dotati di telecamere e dispongono anche di altri sensori specifici.

Si fa riferimento a soluzioni architetture come quelle rappresentate nella seguente Figura, dove i sistemi che rappresentano i sensori sono sistemi a microprocessore autonomi, che inviano dati verso i gestori con chiamate REST. Fra questi il Sensor Server and Manager (di Sii-mobility) come Gestori di Terze Parti. Per esempio, il gestore dei parcheggi, il gestore delle informazioni meteo, il gestore di informazioni ambientali, etc. Questi gestori sono tipicamente realizzati e gestiti in modo autonomo da operatori della città e sul territorio.

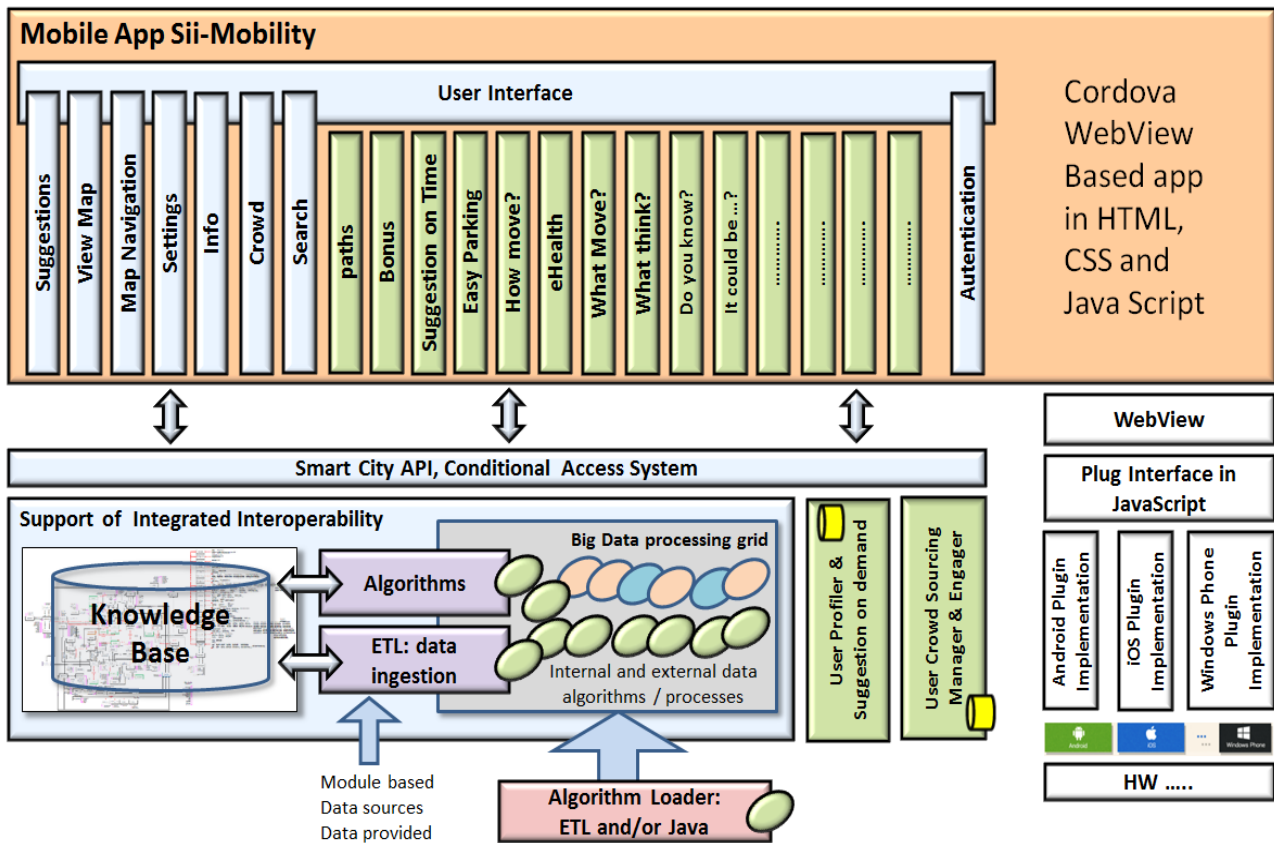


- **Piattaforma di Partecipazione e Sensibilizzazione:** per coinvolgere il cittadino a partecipare ma anche per informare e formare il cittadino, tramite totem, applicazioni mobili, web application, etc. In questo caso, le informazioni sensibili per il cittadino e per la persone specifica vengono prodotte in automatico e/o manualmente. Le considerazioni e rappresentazioni generali dei dati e delle situazioni sono prodotte in automatico: suggerimenti per evitare situazioni critiche (e.g., non uscite tutti dallo stadio dalla parte A visto che oggi vi sarà una manifestazione, tipicamente non vi sono posti liberi in questi parcheggi a queste ore, etc.), suggerimenti per comportamenti personali specifici (e.g., fai ancora due passi fino alla prossima fermata, hai il tempo di farlo senza perdere il bus, e ti fa bene).
- **Modelli partecipativi e per la formazione dei cittadini,** innesco di atteggiamenti virtuosi, modelli e strumenti che possano motivare il cittadino all'uso di mezzi pubblici, meccanismi di premiazione dei comportamenti virtuosi, modelli e soluzioni di crowd-sourcing;
  - **Sottosistema: User Engager:** informare e coinvolgere il city user con dei compiti specifici: fai una foto del punto di interesse (o del servizio) POISRV.xxx, come vedi la coda dove sei, quanto è pieno l'autobus, che valutazione di qualità daresti sul servizio, etc. Una vera e propria spinta a coinvolgere il city user a fornire delle opinioni e partecipare e fornire informazioni tramite il loro mobile. Le informazioni sono ovviamente sul sistema di mobilità, sui servizi di mobilità ma anche sui servizi integrati di smart city. Allo stesso tempo, e sullo stesso canale e mobile, il cittadino può ricevere informazioni utili come: non parcheggiare qui perché non è consentito, parcheggia allo scambiatore che spenderesti meno, usa i mezzi pubblici che spendi meno energia, domani ti suggerisco di fare una strada diversa, etc. Se il city user raccogli i suggerimenti inviategli il sistema deve poterlo verificare.
  - **Sottosistema: User Profiler & Suggestion on Demand** colleziona le richieste che provengono dalle App Mobile e dai Kit Veicolari e pertanto conosce il comportamento degli utenti, le loro preferenze, i loro movimenti, le loro velocità, i punti in cui sostano, etc. E' in grado, (utilizzando le Smart City API verso la Knowledge Base) di produrre suggerimenti di vicinanza su: servizi, eventi, twitter, etc., tenendo conto anche del profilo del city user (cittadino, pendolare, turista, studente) e del comportamento specifico della persona. Fornisce anche informazioni di guida connessa, o per quanto riguarda ai bonus che possono derivare dal comportamento virtuoso dei cittadini.
- **Partecipare tramite Applicazioni WEB e Totem.** Posizionati nei punti cruciali della città, per esempio in prossimità di stazioni, nei punti ad elevato transito di persone in mobilità, ma dove queste possono avere del tempo da dedicare al sistema per comunicare con il sistema.
  - **Sottosistema: Participation Platform** ha il compito di informare i City User riguardo a nuove informazioni, tendenze non corrette, tendenze collettive. Queste valutazioni sono prodotte da processi/algoritmi in esecuzione sul Big Data Processing Grid, che portano a produrre mappe e statistiche che possono essere visualizzate e comprese in modo diretto su: Totem, Pagine Web, Pannelli a Messaggio Variabile, etc. Per esempio: punti caldi della città, zone dove è difficile parcheggiare, aree ad elevata densità di traffico veicolare e pedonale, etc.
  - Questo tipo di consultazione web è tipicamente offline e non in movimento pertanto si presta meno a fornire suggerimenti geolocalizzati, engagement contestuali al comportamento della persona, etc.
- **Integrazione di dati e informazioni provenienti dagli utenti in mobilità** che possono produrre informazioni geolocalizzate de tipo: commenti, velocità, casi critici, rilevazioni puntali dei servizi, valutazione della soddisfazione dell'utente, utenti e cittadini come sensori;

- **Sottosistema: Mobile App** permette di collezionare il comportamento dei city user e di poterlo analizzare per comprenderlo. Queste analisi si possono fare nel rispetto della privacy ottenendo un consenso informato per l'accesso ai dati. Fra questi dati forniti dagli utenti anche i commenti, i voti, le foto che gli utenti stessi possono fornire tramite le App Mobile ed i Kit Veicolari Evoluti.
- **Sottosistema: Mobile App.** Le applicazioni mobile possono misurare le potenze di WiFi e Bluetooth ma anche il semplice comportamento degli utenti.
- **Sottosistema: User Profiler & Suggestion on Demand** colleziona le richieste che provengono dalle App Mobile e dai Kit Veicolari e pertanto conosce il comportamento degli utenti, le loro preferenze, i loro movimenti, le loro velocità, i punti in cui sostano, etc.
- **Sottosistema: User Engager:** informare e coinvolgere il city user con dei compiti specifici: fai una foto del punto di interesse (o del servizio) POISRVxxx, come vedi la coda dove sei, quanto è pieno l'autobus, che valutazione di qualità daresti sul servizio, etc. Una vera e propria spinta a coinvolgere il city user a fornire delle opinioni e partecipare e fornire informazioni tramite il loro mobile.
- **Sottosistema Sensori: Gli Access Point** potranno misurare i flussi di persone, come altri sensori quelli dei mezzi
- **Partecipare/coinvolgere:** la partecipazione dei cittadini viene favorita tramite varie soluzioni: la realizzazione di soluzioni mobili che possano permettere al cittadino registrato di comunicare situazioni di allarme o criticità, ma anche l'acquisizione di humor e trend tramite l'analisi dei social media come Twitter, Facebook, etc. Queste sorgenti di informazioni devono ovviamente essere validate per poterne tenere in considerazione. Gli utenti saranno per questo profilati e valutati con modelli di reputation. Questo strumento include un supporto all'acquisizione di feedback e commenti dei cittadini verso aspetti connessi alle tematiche di Sii-Mobility.
  - Si veda sopra per la **Mobile App**. Similmente anche per **Totem** e Web app.
- **Partecipare/coinvolgere:** Questo strumento permette di collezionare e trarre vantaggio delle informazioni di dettaglio e commenti dai blog sul territorio come da social network e di comprenderle per trarre deduzioni e feedback nel sistema centrale di conoscenza. Per questo saranno utilizzati strumenti di crawling e natural language processing.
  - Sottosistema: **Social Media Crawler and Manager, si veda sopra**
  - Sottosistema: **Social Media Analysis Interface, si veda sopra.**

Partecipare tramite **Applicazioni Mobili**. Queste applicazioni potranno fornire informazioni e ricevere informazioni similmente ai totem. Inoltre dalle applicazioni mobili ci si aspetta la possibilità di avere una comunicazione diretta con il sistema di gestione per comunicare condizioni di servizio, criticità, elementi di valorizzazione, necessità, etc. Queste stesse applicazioni potranno erogare, in cambio di tali informazioni, dei bonus che potranno essere utilizzati per accedere alla città e/o ad altri servizi di mobilità in modo scontato e/o gratuito (accesso al parcheggio, varco porte controllate, etc.). L'idea è premiare il comportamento virtuoso e il contributo al sistema.

L'architettura proposta permette di aggiungere moduli e costruire un'applicazione integrata al posto di 15 o più applicazioni verticali. Che potranno essere comunque fatte ma che tipicamente non permettono di offrire di più di quello che già si trova sul mercato delle app.



Questo implica che lo sviluppo di un modulo per l'applicazione mobile può implicare lo sviluppo:

- **ed il caricamento di dati** tramite la creazione di un processo ETL che legge a sua volta un file dati Open o Private via http/FTP;
- **ed il caricamento di uno più algoritmi/processi da mettere in esecuzione su DISCES**, per esempio di routing, intermodale, che possono trarre vantaggio dall'accesso ai dati della Knowledge Base, del sistema di User Profiler, del sistema di User Crowd Sourcing, del sistema di User Engager, etc. per produrre informazioni per l'utente sulla sua Mobile App o Kit Veicolare. Tali informazioni dovranno essere memorizzate su database per essere richiamate con chiamate REST sincrone.
- **di chiamate REST sincrone che estendono le Smart City API** in modo da poter accedere ai risultati prodotti dagli algoritmi del punto precedente. Per esempio una nuova Smart City API per risultati di algoritmi di routing particolari.
- **di un modulo Cordoba** da integrare nell'applicazione generale secondo direttive specifiche e da concordare con UNIFI che opera l'integrazione e la pubblicazione della APP sui vari portali di market e anche verso hardware specifico purché compatibile Cordova.

Potenzialmente si potranno anche sviluppare applicazioni completamente nuove. Sono però da privilegiare moduli che abilitano nuove funzionalità sul back office di Sii-Mobility per un'ampia gamma di applicazioni.

- **Interfacce API di accesso a servizi di Sii-Mobility** per le PA, PMI, per gli sviluppatori e per altri gestori, centrali e gestori di altri sistemi di SmartCity.
  - **Interfacce API per l'accesso a dati ed informazioni** sia storici che puntuali realtime, per poter:
    - abilitare lo sviluppo di applicazioni mobili e web sviluppate dalle PMI anche attraverso convegni di Hackathon.



- pubblicare dati/risultati provenienti da algoritmi di data analytics caricati sulla piattaforma da parte di PA e PMI.
- **Interfacce di ingresso ed uscita, API**, per acquisire e fornire dati che provengono da **altre**:
  - Istanze di Sii-Mobility focalizzate su altri territori.
  - centrali di Smart City, derivate da progetti Smart City SUD come da progetti Smart City Nord e realizzazioni internazionali.
  - centrali di mobilità e/o trasporti di livello nazionale.

**Overview sulle Smart City API** (per maggiori dettagli si veda anche il documento dei requisiti D1.1a di Sii-mobility Tabella 5.5.1.1:

- Accesso a servizi di ricerca per trovare servizi geolocalizzati: full text, intorno ad un punto, lungo un linea, dentro un area.
- Acquisizione dati che sono gestiti nella Dashboard
- Acquisizione dati che sono collezionati dalla piattaforma di Social Media Crawler
- Interazioni per User Profiling e Suggestion on Demand
- Collezionamento informazioni e contributi dall'utente tramite User Crowd Sourcing, ma anche azioni di stimolo che si vuole fare sull'utente. Sono sempre API richieste visto che deve essere il mobile a chiedere come client.
- Accesso condizionato ai dati del sistema
- Accesso per il caricamento, gestione e monitoraggio di algoritmi di elaborazione
- Accesso ai servizi di ticketing e booking integrato
- Collezionamento di dati che provengono da gestori, city operator e che sono destinati alla DashBoard
- Collezionamento di dati che provengono da Sensori che sono destinati alla Knowledge Base
- Acquisizione dati di operatori e/o Pubbliche amministrazioni tramite protocolli vari, con processi ETL
- Azionamento di sensori tramite la scrittura di valori nel Sensor Server and Manager, da questo il sensore andrà a leggere il valore che deve assumere...
- ....

## 4 Regole per la Specifica Generale

### 4.1 Regole per gli algoritmi

- dovranno usare dati accessibili sul sistema Sii-Mobility accedendo direttamente ai dati con delle chiamate SPARQL, MySQL o noSQL asseconda dei casi.
- dovranno essere sviluppati con linguaggio Java o ETL (con codice sorgente open) in modo da poterli mettere in esecuzione sull'architettura parallela di elaborazione su sistemi LINUX (processi in C++ ). Viene fornito da UNIFI un kit di sviluppo, una SandBox per lo sviluppo di processi ETL. Questa soluzione è presente in versione draft al seguente indirizzo <http://www.disit.org/6690> .
- non devono dipendere da librerie commerciali o proprietarie o a pagamento e comunque a codice sorgente non accessibile. Nel caso sia necessario utilizzare librerie di terzi queste dovranno essere autorizzate dal PDB.
- (alcuni algoritmi) dovranno essere sviluppati utilizzando tecnologia bigdata come MapReduce per esigenze di prestazioni.
- dovranno produrre risultati scrivendo direttamente in un data base indicato sulla stessa piattaforma oppure in XML o altri formati aperti: JSON, RDF file di triple in vari formati, etc. I file dovranno essere scritti in apposite cartelle su disco condiviso in accordo all'architettura Sii-mobility.
- dovranno avere tutti i parametri su linea di comando del processo eseguibile oppure questi dovranno essere letti dal database condiviso delle configurazioni, in caso di algoritmi che lavorino in modo incrementale su gruppi di dati condivisi.
- dovranno essere consegnati in codice sorgente sul GitHub del progetto.

### 4.2 Regole per gli Strumenti/tool Sii-Mobility

- dovranno essere sviluppati per essere messi in esecuzione sulla piattaforma centrale Sii-Mobility.
- Se forniscono accesso ai servizi in modo autenticato dovranno usare il sistema di Authentication Support centralizzato di Sii-Mobility.
- Potranno accedere ai dati della: Knowledge Base, come su altri database di Sii-Mobility tramite chiamate locali e al limite remote ma autenticate (le chiamate remote a database non locati su macchine virtuali Sii-Mobility sono sconsigliate). I protocolli di accesso saranno quelli autorizzati.
- Potranno mettere in esecuzione processi sulla piattaforma Big Data Processing Grid di Sii-Mobility tramite DISCES.
- Dovranno essere sviluppati come applicazioni web, ed in certi casi anche per sistemi mobili.
- Dovranno essere accessibili come codice sorgente per essere caricati nel sito del ministero come da capitolato.

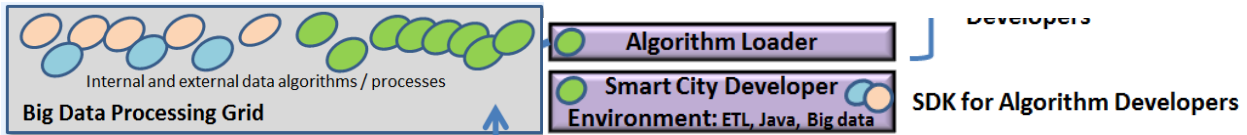
### 4.3 Esempio di Profilo descrittivo per moduli e algoritmi

<b>Profilo Tool/Algoritmo</b>		
.....		
Nome responsabile	.....	
Partner responsabile	.....	
Descrizione Tool/algoritmo	.....	
Dati primari in ingresso	.....	
Dati prodotti in Uscita	.....	
Principi (metodo, basi teoriche, etc.)	.....	
Casi di test (presenti/assenti)	.....	
Posizione casi di test	.....	
Principali problemi non risolti	.....	
Principali requisiti pendenti	.....	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	.....	
Implementato/non implementato	.....	
Stato implementazione, percentuale	.....	
Eseguibile/libreria/web app	.....	
Single thread/Multithread	.....	
Linguaggio di sviluppo (java, Php, ETL)	.....	
Piattaforme supportate	.....	
Posizione del codice sorgente	.....	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	.....	
Indirizzo/i accesso via web (se presenti)	.....	
<b>Nomi tool/moduli usati</b> <small>(aggiungere una riga per ogni tool/modulo usato)</small>	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> <small>(aggiungere una riga per ogni formato usato)</small>	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> <small>(aggiungere una riga per ogni protocollo usato)</small>	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> <small>(aggiungere una riga per ogni db usato)</small>	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b>	<b>Modello sviluppo,</b>	<b>Libreria usata per UI</b>

(web/applicazione)	linguaggio	
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

## 5 Specifica Generale

### 5.1 Specifica: Sviluppo di algoritmi di data analytic e di supporto alle decisioni (Attività 2.2)

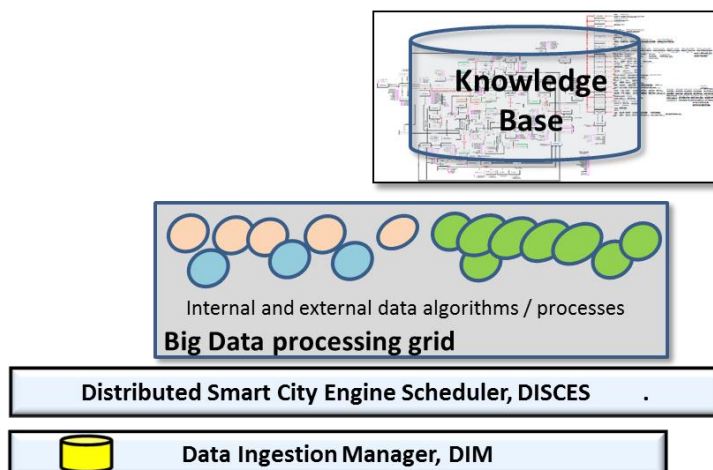


		ECM	Mizar	UNIFI	IN20	TIME	Negentis	EffKnow	liberologico	ataf	busitalia	cttnord	tiemme	areos	efi	calamai	Midra	Proiect	GEOJIN	OUESTIT	SOETEC	FWINGS	
2.2.1	Sviluppo di algoritmi di data analytic e di supporto alle decisioni: Aggregazione e riconciliazione dei dati		X	X				X															
2.2.2	Sviluppo di algoritmi di data analytic e di supporto alle decisioni: supporto alle decisioni, modello di definizione action e conditions, programmazione smart city			X				X															
2.2.3	Sviluppo di algoritmi di data analytic e di supporto alle decisioni: modello di definizione action e conditions, programmazione smart city			X				X															
2.2.4	Sviluppo di algoritmi di data analytic e di supporto alle decisioni: <b>Algoritmi di ottimizzazione</b>			X			X																
2.2.5	Sviluppo di algoritmi di data analytic e di supporto alle decisioni: <b>Algoritmi di instradamento dei veicoli</b>						X																

Principali Sottosistemi da descrivere in questa Subsubsection, sono i sottosistemi primari della specifica					
	Sotto Attività	Partner coord	Sottosistema	Tool/Level	Descrizione sommaria
2.2.1	Sviluppo di algoritmi di data analytic e di supporto alle decisioni: Aggregazione e riconciliazione dei dati	UNIFI	Data Analytics	Algoritmi e strumenti di Riconciliazione (A01)	Algoritmi e strumenti per la riconciliazione semantica di parti della knowledge base (quando alcuni dati non sono connessi in modo corretto al modello concettuale, alcune triple addizionali devono essere prodotte).
		UNIFI	Data Analytics	Algoritmi e strumenti di Arricchimento (A02)	Algoritmi e strumenti per l'Arricchimento semantico di concetti e/o attributi parti della knowledge base (quando alcuni dati come stringhe potrebbero portare ad effettuare un collegamento con personaggi famosi o con località geografiche, alcune triple addizionali devono essere prodotte).
2.2.2	Sviluppo di algoritmi di data analytic e di supporto alle decisioni:	UNIFI	Smart Decision Support	SmartDS (T01)	sistema collaborativo di supporto alle decisioni basato su System Thinking e modelli Baesiani.

	supporto alle decisioni,	UNIFI	Data Analytics	Algoritmi di ottimizzazione per la produzione suggerimenti per il parcheggio (A03)	Algoritmi per la produzione suggerimenti per il parcheggio su base sensori e realtime oppure su base statistica settimanale: per ridurre il tempo per trovare parcheggio, dando evidenza che hai scelto la durata minore, etc.
		UNIFI	Data Analytics	Algoritmi e strumenti per flussi di persone e mezzi (A04)	Algoritmi e strumenti, per la stima dei flussi di persone tramite Wi-Fi, oppure da operatori di telecomunicazione, oppure tramite applicazioni mobili o via sensori BT di rilevazione. I flussi mezzi sono rilevabili tramite sensori vari come spire, telecamere, o kit veicolari, etc.
2.2.3	Sviluppo di algoritmi di data analytic e di supporto alle decisioni: modello di definizione action e condition, programmazione smart city	UNIFI	Control Interface	Dashboard Builder and Manager, Monitoring: DIBUM (T02, API01, API02)	Strumento di produzione delle Dashboard sulla base di metriche formalizzate e criteri di allarme. DIBUM colleziona dati da sorgenti multiple. Accetta anche dati con chiamate REST autenticate da parte di centrali operative che possono fornire coppie chiave-valore.
2.2.4	Sviluppo di algoritmi di data analytic e di supporto alle decisioni: Algoritmi di ottimizzazione	UNIFI	Data Analytics	Algoritmi di ottimizzazione (percorsi con più fermate, cambi, etc.) (A05)	Algoritmi di ottimizzazione per il compimento di percorsi con più fermate, cambi, etc. in modalità multimodale,
2.2.5	Sviluppo di algoritmi di data analytic e di supporto alle decisioni: Algoritmi di instradamento dei veicoli	UNIFI GOL	Data Analytics	Algoritmi di instradamento (veicoli e persone) (A06)	Algoritmi per la navigazione di veicoli e persone da un punto iniziale ad uno target. Nel caso delle persone si dovrebbe prevedere due casi: a piedi, oppure con TPL. In questo ultimo caso è possibile la navigazione multimodale (segmenti misti di vari tipi di mezzi pubblici, a piedi, etc.). Il sistema di navigazione dovrebbe tenere in considerazione: la presenza di ordinanze, blocchi del traffico, incidenti, aspetti ambientali, etc. La modalità di navigazione con un veicolo dovrebbe proporre parcheggi vini all'arrivo, magari tenendo conto del tipo di veicolo che l'utente ha.
		NEGE NTIS	Data Analytics	Algoritmi per la produzione di percorsi per flotte merci. (A07)	Algoritmi per la navigazione di veicoli che nel loro giro di consegne o prelievi devono toccare una serie di punti dove scaricare/caricare merci. La soluzione deve tenere conto anche della capienza del messo, degli scarichi e carichi.

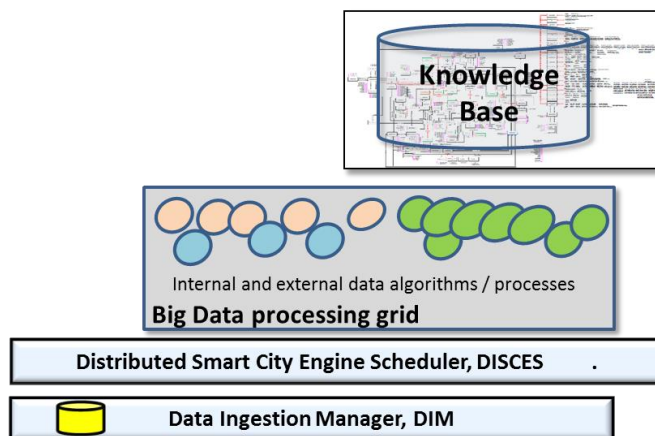
### 5.1.1 Algoritmi e strumenti di Riconciliazione dati (A01) (UNIFI)



Profilo Tool/Algoritmo		
Algoritmi e strumenti di Riconciliazione dati		
Nome responsabile		
Partner responsabile	UNIFI	
Descrizione Tool/algoritmo	Algoritmi e strumenti per la riconciliazione semantica di parti della knowledge base (quando alcuni dati non sono connessi in modo corretto al modello concettuale, alcune triple aggiuntive devono essere prodotte).	
Dati primari in ingresso	RDF store, o data set da ingerire	
Dati prodotti in Uscita	Triple RDF relative ai dataset ingeriti	
Principi (metodo, basi teoriche, etc.)	Gli ETL vengono messi in esecuzione da DISCES in modo periodico o aperiodico.	
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	n/d	
Principali problemi non risolti	Assenti	
Principali requisiti pendenti	Nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0.0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	ETL	
Piattaforme supportate	Unix, Windows	
Posizione del codice sorgente	svn	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato

Ingestion	n/a	REST API, CSV, KMZ, SHP
Quaity Improvement	n/a	n/a
Triplification	n/a	n/a
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
CSV		comma-separated values
KMZ/KML		Keyhole Markup Language
SHP		Shape format file
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
HTTP		HyperText Transfer Protocol
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
MySql	Database relazionale	
HBase	Database NoSql distribuito su filesystem Hadoop	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
n/a		
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
n/a		

### 5.1.2 Algoritmi e strumenti di Arricchimento dati (A02) (UNIFI)



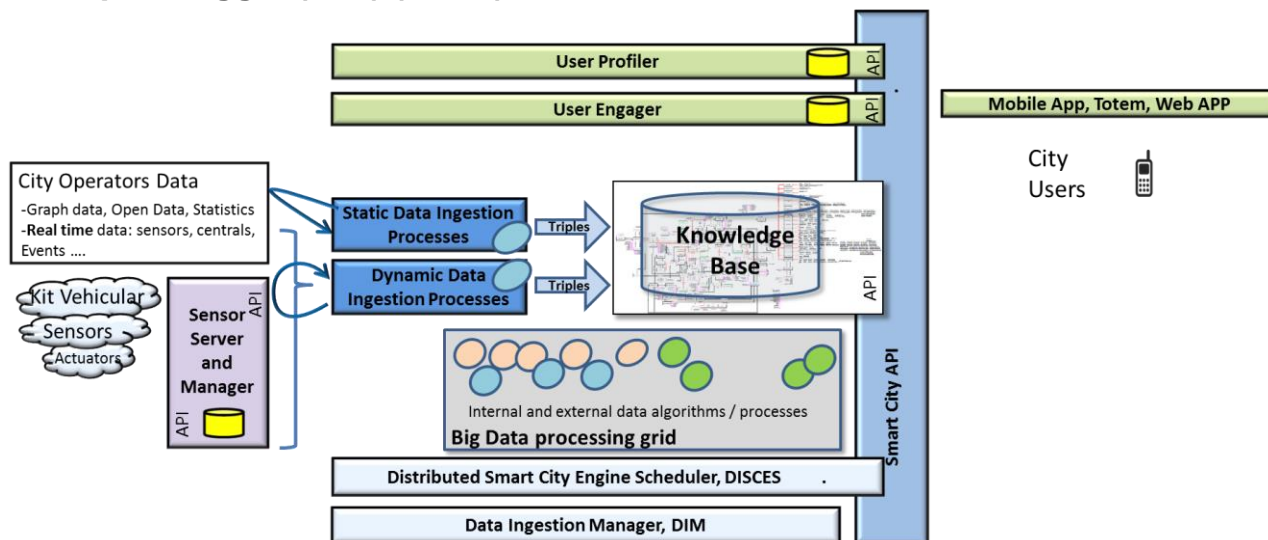
<b>Profilo Tool/Algoritmo</b>	
Algoritmi e strumenti di Arricchimento dati	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Algoritmi e strumenti per l'Arricchimento semantico di concetti e/o attributi parti della knowledge base (quando alcuni dati come stringhe potrebbero portare ad effettuare un collegamento con personaggi famosi o con località geografiche, alcune triple addizionali



	devono essere prodotte).	
Dati primari in ingresso	RDF store, o data set da ingerire	
Dati prodotti in Uscita	Triple RDF relative ai dataset ingeriti	
Principi (metodo, basi teoriche, etc.)	Gli ETL vengono messi in esecuzione da DISCES in modo periodico o aperiodico.	
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	n/d	
Principali problemi non risolti	Assenti	
Principali requisiti pendenti	Nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0.0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	ETL	
Piattaforme supportate	Unix, Windows	
Posizione del codice sorgente	svn	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
Ingestion	n/a	REST API, CSV, KMZ, SHP
Quaity Improvement	n/a	n/a
Triplification	n/a	n/a
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
CSV		comma-separated values
KMZ/KML		Keyhole Markup Language
SHP		Shape format file
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
HTTP		HyperText Transfer Protocol
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
MySql	Database relazionale	
HBase	Database NoSql distribuito su filesystem Hadoop	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
n/a		
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

n/a		

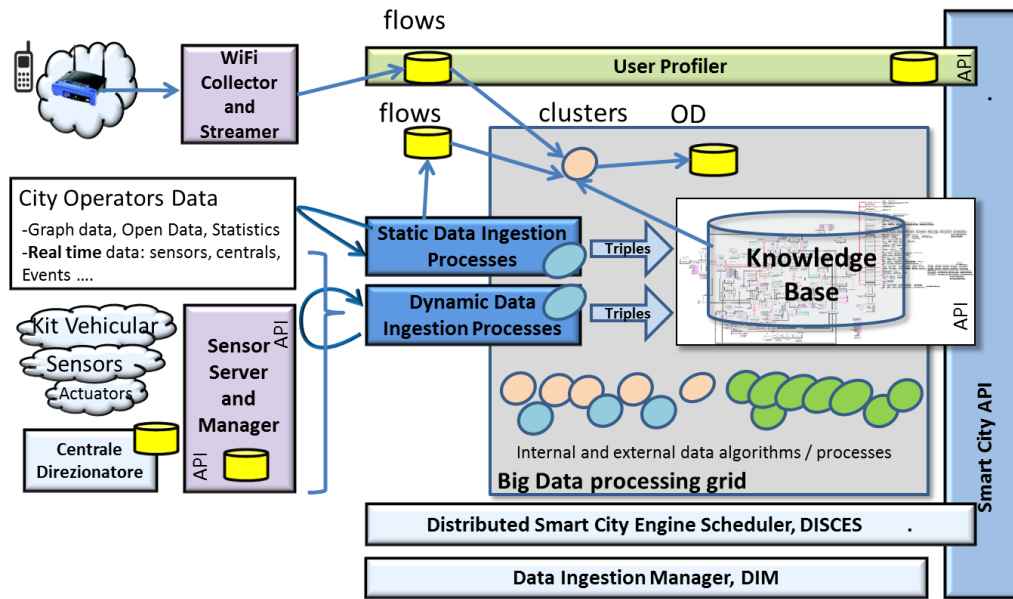
### 5.1.3 Algoritmi di ottimizzazione per la produzione suggerimenti per il parcheggio (A03) (UNIFI)



Profilo Tool/Algoritmo	
Algoritmi di ottimizzazione per la produzione suggerimenti per il parcheggio	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Algoritmi per la produzione suggerimenti per il parcheggio e mobilità su base sensori e realtime oppure su base statistica settimanale, sulla base del comportamento delle persone, della loro residenza, etc.: per ridurre il tempo per trovare parcheggio, dando evidenza che hai scelto la durata minore, etc.
Dati primari in ingresso	RDF store, o data set da ingerire
Dati prodotti in Uscita	Triple RDF relative ai dataset ingeriti
Principi (metodo, basi teoriche, etc.)	Gli ETL vengono messi in esecuzione da DISCES in modo periodico o aperiodico.
Casi di test (presenti/assenti)	Assenti
Posizione casi di test	n/d
Principali problemi non risolti	Assenti
Principali requisiti pendenti	Nessuno
Aspetti Tecnologici	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0.0%
Eseguibile/libreria/web app	Eseguibile
Single thread/Multithread	Multithread
Linguaggio di sviluppo (java, Php, ETL)	JAVA, PHP

<b>Piattaforme supportate</b>	Unix, Windows	
<b>Posizione del codice sorgente</b>	svn	
<b>Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)</b>	n/d	
<b>Indirizzo/i accesso via web (se presenti)</b>	n/d	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
flows		REST HTTP
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON		JavaScript Object Notation
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTP		HyperText Transfer Protocol
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
MySQL	DBMS	
SPARQL		
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
web	PHP, JAVA	
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

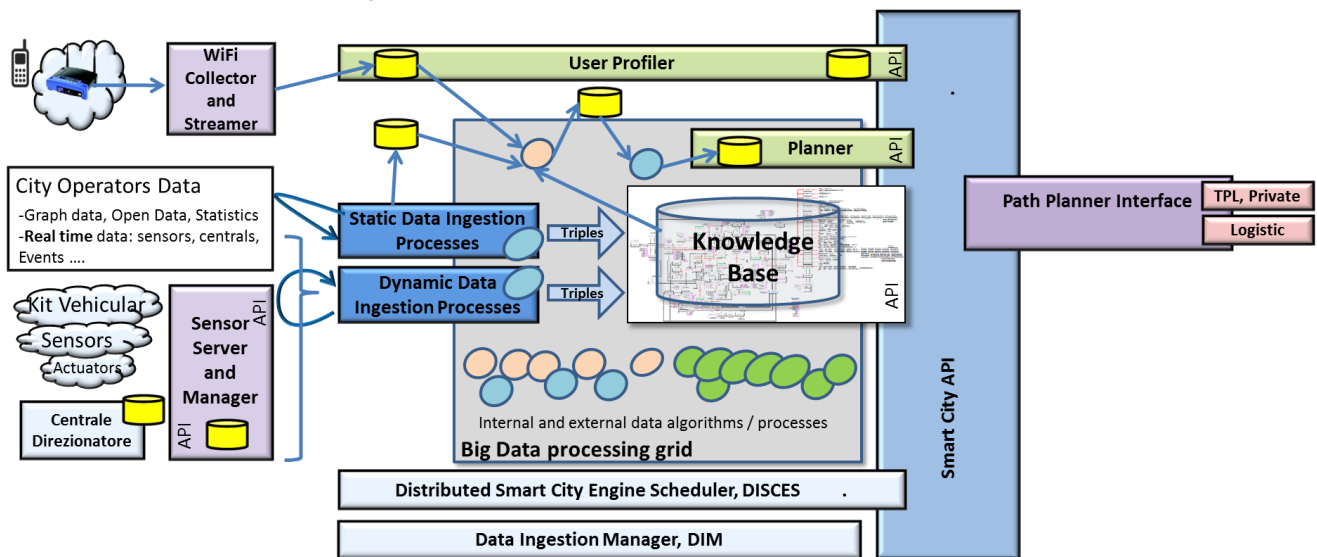
### 5.1.4 Algoritmi e strumenti per flussi di persone e mezzi (A04) (UNIFI)



<h2 style="margin: 0;">Profilo Tool/Algoritmo</h2>	
<h3 style="margin: 0;">Algoritmi e strumenti per flussi di persone e mezzi</h3>	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Algoritmi e strumenti, per la stima dei flussi di persone tramite Wi-Fi, oppure da operatori di telecomunicazione, oppure tramite applicazioni mobili o via sensori BT di rilevazione. I flussi mezzi sono rilevabili tramite sensori vari come spire, telecamere, o kit veicolari, etc.
Dati primari in ingresso	Rilevazioni di posizione degli utenti
Dati prodotti in Uscita	Stima dei flussi di persone in entrata/uscita
Principi (metodo, basi teoriche, etc.)	I flussi delle persone sono calcolati per profilo utente o aggregati
Casi di test (presenti/assenti)	assenti
Posizione casi di test	
Principali problemi non risolti	nessuno
Principali requisiti pendenti	nessuno
<b>Aspetti Tecnologici</b>	
Stato (proposto/approvato)	proposto
Implementato/non implementato	non implementato
Stato implementazione, percentuale	0%
Eseguibile/libreria/web app	web app
Single thread/Multithread	single thread
Linguaggio di sviluppo (java, Php, ETL)	Java, PHP
Piattaforme supportate	Windows/Linux/Unix
Posizione del codice sorgente	svn
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se	

necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
flows		REST HTTP
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON		JavaScript Object Notation
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTP		HyperText Transfer Protocol
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
MySQL	DBMS	
MongoDB	NoSQL	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
web		JQuery, Openstreetmap, Leaflet
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

### 5.1.5 Algoritmi di ottimizzazione anche INTERMODALE (A05, A06) (UNIFI GOL, NEGENTIS)



Gli algoritmi di instradamento o routing permettono dati due punti (DA, A) di fornire una strada per raggiungere il punto A dal punto DA. Il punto iniziale DA può essere la posizione in cui uno si trova come una qualsiasi altra posizione sulla mappa, un punto di interesse POI, un PIN, un parcheggio, etc. Gli algoritmi di instradamento possono produrre traiettorie tenendo in considerazione alcuni vincoli, e ovviamente un grafo strade annotato con tali informazioni. Il grafo strade dovrebbe riportare anche informazioni aggiuntive come: presenza di lavori, presenza di incidenti in tempo reale, flusso tipo e attuale di auto, flusso tipico e attuale di persone, etc. etc.

Nel progetto Sii-Mobility verranno studiati metodi e modelli per differenti soluzioni di instradamento, alcune delle quali saranno inserite nella piattaforma. Si riportano alcune tipologie di casi d'uso relativi a possibili tipologie di percorso – al termine di ogni sezione vengono riportati i casi che saranno implementati nella piattaforma Sii-Mobility.

- **Pedonale:** movimento pedonale fra due punti, con o senza vincoli, che ovviamente deve escludere i percorsi non accessibili alle persone come le autostrade, etc., ma può permettere i percorsi di attraversamento piazze e anche gallerie cittadine. Il percorso pedonale potrà sfruttare anche caratteristiche aggiuntive (ad esempio, preferenze per strade interne rispetto a strade di grande comunicazione o la possibilità, nel caso di percorsi turistici, di passare “vicino” a punti di interesse che non necessariamente coincidono con la destinazione). Saranno implementati: percorsi pedonali con restrizioni su alcuni archi stradali e percorsi pedonali che preferibilmente passano vicino a punti di interesse (generici, non specificati dall'utente)
- **Percorsi per utenti con handicap motorio:** potranno essere programmati percorsi che utilizzino informazioni sull'accessibilità di tratti stradali a utenti in carrozzella (proibizione di scale, tunnel, percorsi accidentati, salite/discese con pendenza elevata). Sarà implementata una soluzione bi-obiettivo nel caso di disponibilità di informazioni sufficienti sull'accessibilità.
- **Veicolare privato (auto):** movimento in auto privata con o senza vincoli. I vincoli possono essere: impossibilità di svolta a sinistra da alcune strade, esclusione di percorsi a pagamento, esclusione di percorsi pedonali, esclusione di ZTL, esclusione di aree in base agli orari, esclusione di preferenziali, etc. Nel caso ZTL si potrebbe prevedere un percorso multi-modale auto (fino al confine ZTL/zona pedonale) – piedi (fino a destinazione). I criteri, potrebbero essere quelli di raggiungere la destinazione nel minor tempo possibile oppure alla minore distanza. Per il momento verrà gestito il costo chilometrico – il costo monetario (pedaggio, biglietto TPL, ...) deve essere valutato con cura, vista la complessità (costi una tantum, costi

chilometrici, abbonamenti,...). Nel caso di altri mezzi (moto, mezzi pesanti,...) si potrebbe tenere conto della tipologia del mezzo per evitare di strade non accessibili, naturalmente a condizione che il grafo sottostante (openstreetmap, contenga le informazioni su questo tipo di restrizioni).. Saranno implementati gli algoritmi per veicolo privato e limitazioni alla svolta, veicolo privato + pedonale nel caso di ZTL / zone pedonali.

- **Ciclo privato:** movimento su bici con preferenze (massimizzare il percorso su ciclabile, oppure minimizzare la distanza, etc.), escludere segmenti non accessibili alle bici come autostrade, preferenziali, e certe ZTL, parchi, etc. Sarà implementata la versione bi-obiettivo (distanza / ciclabilità).
- **Mezzi pubblici:** bus, treno. In questo caso, occorre tenere conto degli spostamenti a piedi dal punto di inizio alla fermata, fra fermate diverse, e fra l'ultima fermata e la destinazione. Questo routing deve essere svolto in base alle traiettorie dei mezzi pubblici ma anche in base ai loro orari. Per i cambi il sistema dovrebbe tenere conto anche dei tempi di trasferimento a piedi considerando la velocità media della persona molto bassa. Alcuni mezzi pubblici potrebbero essere dotati di rastrelliera o ingresso per bici, in questo caso si dovrebbe parlare di multimodale: TPL, pedonale, bici. I vincoli, potrebbero essere quelli di raggiungere la destinazione nel minor tempo possibile oppure con il minimo numero di cambi, etc. Sarà implementata una versione pedonale+TPL+pedonale con limitazione ai trasbordi ed obiettivo di minima durata del tragitto complessivo.

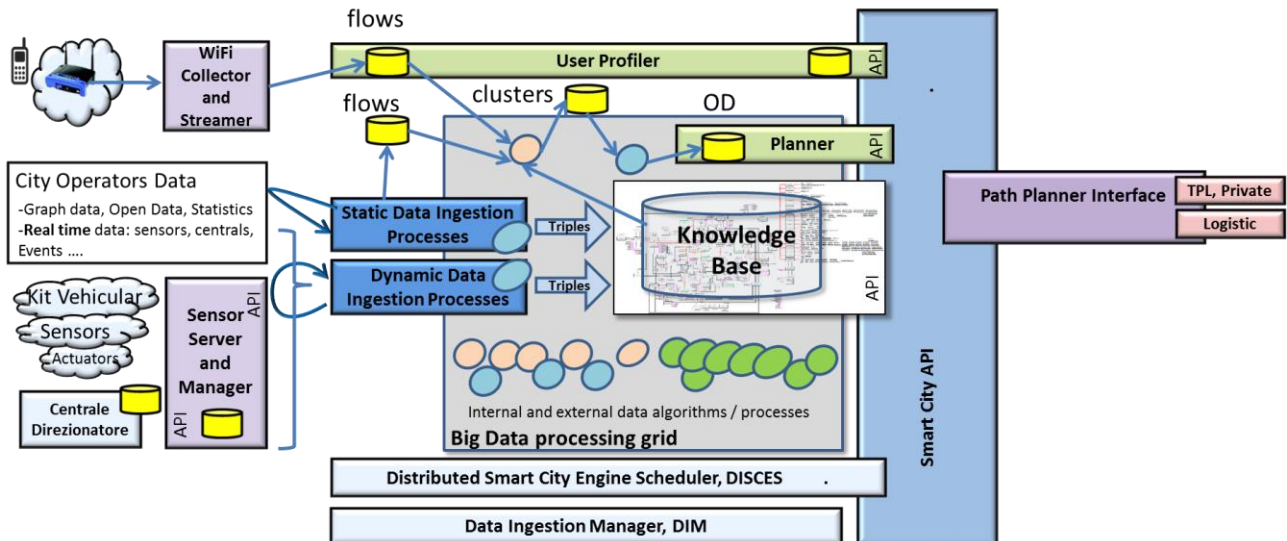
<b>Profilo Tool/Algoritmo</b>	
Algoritmi di ottimizzazione (compimento di percorsi con più fermate, cambi, etc.) INTERMODALE	
Nome responsabile	
Partner responsabile	UNIFI GOL
Descrizione Tool/algoritmo	Algoritmi di ottimizzazione per il compimento di percorsi con più fermate, cambi, etc. in modalità multimodale,
Dati primari in ingresso	Strutture dati serializzate, parametri numerici e variabili di stato che caratterizzano i singoli percorsi. Ad esempio: - definizione dei nodi origine/destinazione. - istante di tempo relativo alla richiesta di pianificazione del percorso. - vincoli sulla massima distanza percorribile a piedi verso i punti di accesso al servizio pubblico (come tra 100m e 5km).
Dati prodotti in Uscita	Elaborazione e serializzazione di strutture dati. Insieme di cammini ottimi non dominati tra cui si lascia all'utente la scelta in funzione, ad esempio, del numero massimo di trasbordi
Principi (metodo, basi teoriche, etc.)	Algoritmi euristici multimodali multi-obiettivo su grafi del tipo Multi Label Dijkstra o che traggono spunto dalla programmazione dinamica.
Casi di test (presenti/assenti)	Previsti casi di test per la validazione di algoritmi e strutture dati (da definire)
Posizione casi di test	.
Principali problemi non risolti	.

Principali requisiti pendenti	.	
Aspetti Tecnologici		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	Single thread (da definire)	
Linguaggio di sviluppo (java, Php, ETL)	C++	
Piattaforme supportate	Unix	
Posizione del codice sorgente	.	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	.	
Indirizzo/i accesso via web (se presenti)	.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Knowledge Base RDFStore	SPARQL Endpoint	SPARQL result object
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
GeoJson		Basato su JavaScript Object Notation
Binario		Boost Serialization 1.6x
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
TCP		
HTTP		
SPARQL	Knowledge Base RDFStore	
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
Libreria C++ Boost	1.6x	Boost Software License (Open Source)
Xerces-c	3.1.x	Apache Software License v. 2 (Open Source)



cURLpp	0.8.0	MIT License
--------	-------	-------------

### 5.1.6 Algoritmi per la produzione di percorsi per flotte merci con piu' fermate (A07) (TIME)



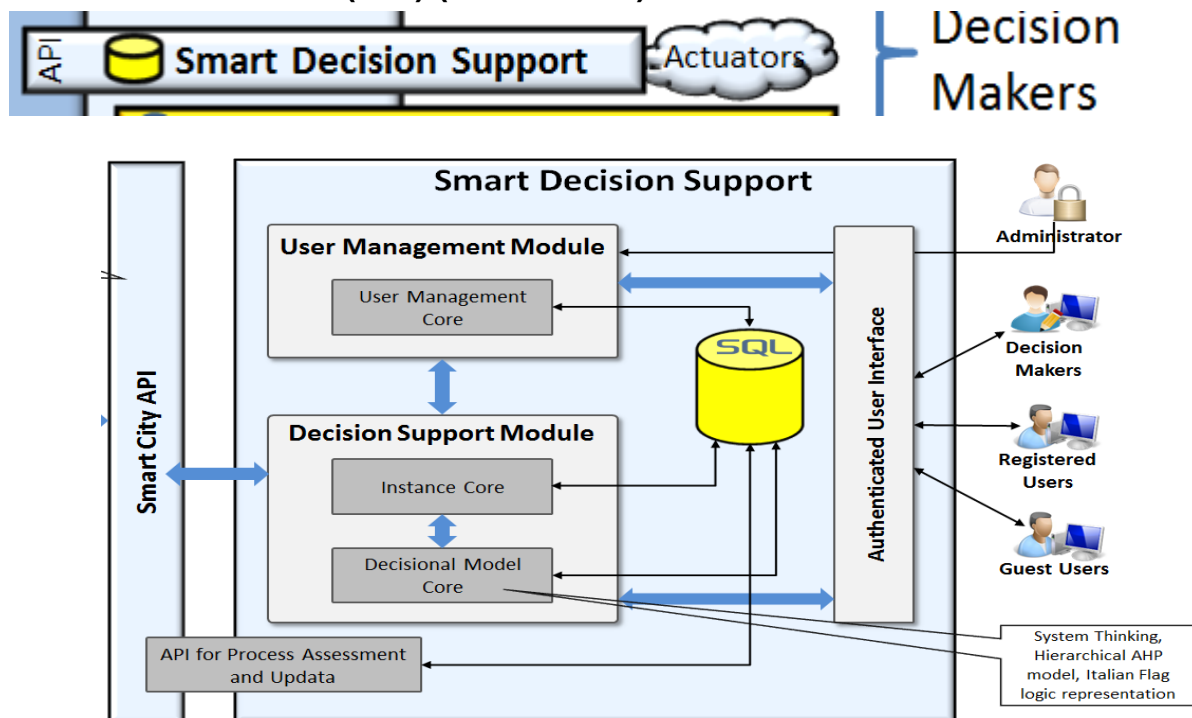
Il tool consente di definire la pianificazione di un sistema di gestione di logistica dell'ultimo miglio basata sul concetto di "truck sharing": i mezzi preposti al prelievo da magazzino ed alla consegna/ritiro di piccoli colli in ambito urbano/metropolitano effettuano tale servizio per conto di tutti i principali corrieri e viene loro assegnata una zona specifica (CAP o altra zonizzazione). Pertanto il tool deve acquisire i dati relativi alle consegne/ritiri schedulati giornalmente per ogni corriere, suddividerli per zone ed assegnare ad ogni utente (mezzo/padroncino) una lista di prelievi da magazzini corrieri e consegne/ritiri nella propria zona.

Profilo Tool/Algoritmo	
Algoritmi per la produzione di percorsi per flotte merci	
Nome responsabile	
Partner responsabile	TIME
Descrizione Tool/algoritmo	<p>Algoritmi e web services per :</p> <ul style="list-style-type: none"> <li>• acquisizione dati su consegne/ritiri dai database dei corrieri</li> <li>• creazione di una lista di prelievi di piccoli colli (limiti di dimensione/peso) dai magazzini dei corrieri</li> <li>• creazione di una lista di consegne/ritiri nella zona assegnata o gestita</li> </ul> <p>Il Tool si basa sull'utilizzo della piattaforma Pentaho Data Integration (Open Source) che consente lo scambio dati con i sistemi informativi dei corrieri, che forniscono la lista delle consegne/ritiri da effettuare e che vengono successivamente suddivisi secondo la zonizzazione preimpostata, i vincoli definiti (orari, tempi, ZTL, etc.) e predispongono la lista dei pick up e delle</p>

	consegne/ritiri corredata di destinatari, indirizzi, nr. e tipologia dei colli, per zone predefinite	
Dati primari in ingresso	Lista consegne/ritiri per zona e relativa lista dei pick up da magazzino per ogni corriere	
Dati prodotti in Uscita	Pianificazione dei ritiri da magazzino e delle consegne/ritiri per zone	
Principi (metodo, basi teoriche, etc.)	Algoritmi per ottimizzazione rotazione magazzini distribuiti	
Casi di test (presenti/assenti)	Da definire	
Posizione casi di test	Da definire	
Principali problemi non risolti	Nessuno	
Principali requisiti pendenti	Accordi di cooperazione con sistemi informativi dei Corrieri, e relativi protocolli di agreement	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Eseguibile, WEB APP	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java, ETL	
Piattaforme supportate	Windows, Linux	
Posizione del codice sorgente	n.d.	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.d.	
Indirizzo/i accesso via web (se presenti)	n.d.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Pentaho data integration	Java API	XML
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
Da definire		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
http, EDI, XML		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
MySQL	RDBMS	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>

WEB, EXE		
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
Pentaho Data Integration (Open Source)		GPLv2

### 5.1.7 Tool SmartDS (T01) (UNIFI.DISIT)

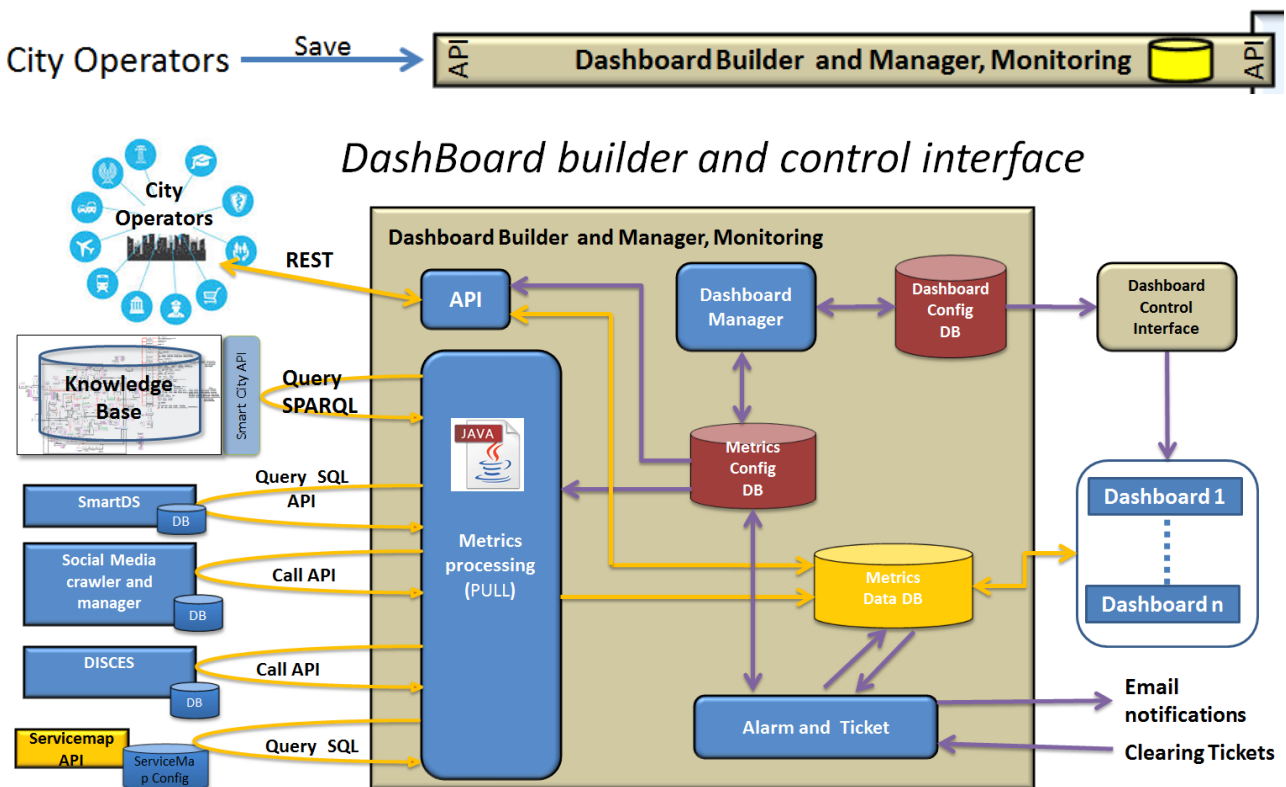


Profilo Tool/Algoritmo	
Tool SmartDS	
Nome responsabile	
Partner responsabile	UNIFI.DISIT
Descrizione Tool/algoritmo	System collaborativo di supporto alle decisioni basato su System Thinking e modelli Baesiani.
Dati primari in ingresso	- Modelli di sistemi decisionali creati e riutilizzati in ambito collaborativi da esperti; - Parametri e coefficienti numerici per il calcolo statistico dei processi decisionali. Questi dati verranno inseriti manualmente, raccolti tramite feedback utenti e interviste, o derivanti da analisi di dati provenienti dai tool da Social Media Analysis Interface (5-13-1) e Knowledge Base RDFStore (5.11.4).
Dati prodotti in Uscita	Risultati statistici del calcolo dei processi decisionali.
Principi (metodo, basi teoriche, etc.)	Modelli gerarchici di Decision Support Systems: paradigmi System Thinking, Analytical Hierarchical

	Process (AHP), distribuzione statistica a 3 valori Italian Flag (IF).	
Casi di test (presenti/assenti)	Previsti (in via di definizione), casi di test per la validazione dei modelli proposti dagli esperti in ambito collaborativo, e per valutare le modalità di collezionamento dei feedback e delle opinioni degli utenti.	
Posizione casi di test	.	
Principali problemi non risolti	.	
Principali requisiti pendenti	.	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Web app	
Single thread/Multithread	Single Thread	
Linguaggio di sviluppo (java, Php, ETL)	Java, Javascript	
Piattaforme supportate	Linux, Windows	
Posizione del codice sorgente	.	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	.	
Indirizzo/i accesso via web (se presenti)	.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Social Media Analysis Interface (5-13-1)	MySQL connector driver.	Accesso e condivisione dati su database relazionale MySQL.
Knowledge Base RDFStore (5.11.4)	SPARQL Endpoint per query su REF Store.	SPARQL result object
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
XML		
RDF		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
TCP		
HTTP		
SPARQL	Knowledge Base RDFStore (5.11.4)	
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
smartDs_database	Database MySQL per l'archiviazione dei dati risultanti dal calcolo dei processi decisionali.	

Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
Web application con accesso autenticato differenziato	Java, Javascript	jQuery
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

### 5.1.8 Dashboard Builder and Manager, Monitoring: DIBUM (T02, API01, API02) (UNIFI.DISIT)



Profilo Tool/Algoritmo	
Dashboard Builder and Manager, Monitoring: DIBUM	
Nome responsabile	
Partner responsabile	UNIFI.DISIT
Descrizione Tool/algoritmo	Strumento di creazione delle Dashboard sulla base di metriche formalizzate e criteri di allarme. DIBUM colleziona dati da sorgenti multiple: database e chiamante REST. Accetta anche dati da terzi con chiamate REST autenticate da parte di centrali operative che possono fornire coppie chiave-valore, come dati in stream da telecamere.
Dati primari in ingresso	Dati relativi alle metriche formalizzate e dati

	provenienti da chiamate REST	
Dati prodotti in Uscita	Dati relativi alle configurazioni delle dashboard prodotte	
Principi (metodo, basi teoriche, etc.)	Metriche calcolate su dati provenienti da sorgenti multiple. Accettazione di dati con chiamate REST che possono fornire coppie chiave-valore.	
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	n/d	
Principali problemi non risolti	Assenti	
Principali requisiti pendenti	Nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0	
Eseguibile/libreria/web app	Eseguibile/Web app	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java, JavaScript, PHP, HTML, CSS	
Piattaforme supportate	Unix, Windows, OS X	
Posizione del codice sorgente	svn	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Dashboard_manager	n/a	n/a
Widgets	Twitter Vigilance API, Service Map API	PUSH/PULL
Metrics processing	n/a	PULL
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON		JavaScript Object Notation
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTP		HyperText Transfer Protocol
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
MySQL	Database relazionale	

<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
Web	HTML, JavaScript, CSS	Bootstrap
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
Gridster		Licenza MIT
Bootstrap	3.3.6	Licenza MIT
JQuery	1.10.1	Licenza MIT

### 5.1.8.1 Protocollo Dashboard Builder and Manager, API: DIBUM (UNIFI.DISIT)

<b>Profilo Tool/Algoritmo</b>	
API REST fornite dal DIBUM	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Il modulo fornisce delle API REST utilizzabili da parte delle centrali operative di operatori/gestori di servizi locali per fornire/recuperare dati al/dal DIBUM come coppie chiave-valore.
Dati primari in ingresso	Dati numerici/percentuali/testuali/booleani relativi a una o più metriche.
Dati prodotti in Uscita	Elenco metriche formalizzate con le informazioni che le descrivono. Dati numerici/percentuali/testuali/booleani relativi ad una certa metrica.
Principi (metodo, basi teoriche, etc.)	Ogni operatore/gestore si deve autenticare per poter inviare dati al DIBUM tramite la API fornita dal modulo. Per ogni operatore/gestore vengono formalizzate una serie di metriche recuperabili mediante un API apposita (API REST di GET lista metriche). Uso di query SQL.
Casi di test (presenti/assenti)	Assenti
Posizione casi di test	n/a
Principali problemi non risolti	Assenti
Principali requisiti pendenti	Nessuno
<b>Aspetti Tecnologici</b>	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	non implementato
Stato implementazione, percentuale	0.0%
Eseguibile/libreria/web app	web app

Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	PHP	
Piattaforme supportate	Quelle supportate da PHP	
Posizione del codice sorgente	n/d	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
api		
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
n/a		
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

Il modulo espone una interfaccia REST (API1) su protocollo HTTP per l'invio/recupero di dati al/dal DIBUM come coppie chiave-valore.

<b>Nome:</b> Service Search
<b>URL:</b> <a href="http://dashboard.km4city.org/api/v1/metrics/add_data.php">http://dashboard.km4city.org/api/v1/metrics/add_data.php</a>
<b>Descrizione</b>
Questa API consente all'operatore/gestore di servizi di fornire dati al DIBUM relativi a metriche formalizzate. Questi dati possono essere numerici, percentuali, testuali o booleani. Il formato da utilizzare per i risultati è JSON. E' prevista anche una validazione dei dati inviati. Ad esempio, nel caso della metrica " Electricity_Consum_District" viene valutato che le chiavi "id" e "name" contengono le informazioni corrette e che sia presente la chiave "context". Come si vede dagli esempi, quest'ultima non è obbligatoria nelle altre due metriche d'esempio.



Modalità: POST	
Parametri	
-	
Esempi	
<p>Invio dei dati relativi a più metriche da parte di un operatore elettrico (tipo ENEL). In questo l'operatore può mandare dati relativi al consumo di energia elettrica per quartiere o per comune, all'energia elettrica accumulata, numeri di interruzioni del servizio, etc</p> <p><a href="http://dashboard.km4city.org/api/v1/metrics/add_data.php">http://dashboard.km4city.org/api/v1/metrics/add_data.php</a></p> <p><b>JSON payload:</b></p> <pre>{   "servicesOperatorName": "ENEL",   "metrics": [     {       "id": "3",       "name": "Electricity_Consum_District",       "data": [         {           "computationDate": "2015-10-12 15:40:13",           "context": {             "quartiere": "Quartiere1",             "comune": "FIRENZE",             "coordinates": ""           },           "value": [39.9]         },         {           "computationDate": "2015-10-12 15:40:13",           "context": {             "quartiere": "Quartiere2",             "comune": "FIRENZE",             "coordinates": ""           },           "value": [39.9]         },         {           "computationDate": "2015-10-12 15:40:13",           "context": {             "quartiere": "Quartiere3",             "comune": "FIRENZE",             "coordinates": ""           },           "value": [39.9]         }       ]     }   ] }</pre>	

```
{
  "computationDate": "2015-10-12 15:40:13",
  "context":
  {
    "quartiere": "Quartiere4",
    "comune": "FIRENZE",
    "coordinates": ""
  },
  "value": [39.9]
},
{
  "computationDate": "2015-10-12 15:40:13",
  "context":
  {
    "quartiere": "Quartiere5",
    "comune": "FIRENZE",
    "coordinates": ""
  },
  "value": [39.9]
}
]
},
{
  "id": "4",
  "name": "Stored_Electricity_Month",
  "description": "Energia elettrica accumulata mensile",
  "dataArea": "Energia",
  "sourceData": "ENEL",
  "metricType": "Numerico",
  "data": [
    {
      "computationDate": "2016-10-12 15:40:13",
      "value": [39.9]
    }
  ]
},
{
  "id": "5",
  "name": "Power_cut_Month",
  "description": "Numero di interruzioni per utente e per mese",
  "dataArea": "Energia",
  "sourceData": "ENEL",
  "metricType": "Numerico",
  "data": [
    {
      "computationDate": "2015-8-12 15:40:13",
      "value": [2.3]
    }
  ]
}
```

```

    }
  ]
}
]
}

```

NOTA: la chiave "coordinates" , se non vuota, deve contenere le coordinate latitudine e longitudine nel seguente formato [lat,long]

<b>Nome:</b> Service Search	
<b>URL:</b> <a href="http://dashboard.km4city.org/api/v1/metrics/get_list.php">http://dashboard.km4city.org/api/v1/metrics/get_list.php</a>	
<b>Descrizione</b>	
Questa API consente all'operatore/gestore di servizi di ottenere dal DIBUM le informazioni relative alle metriche formalizzate. Il formato da utilizzare per il risultato è JSON.	
Modalità: GET	
<b>Parametri</b>	
selection	Tale parametro può assumere due valori: "all" e "operator". Con "all" l'operatore/gestore ottiene tutte le metriche che sono state formalizzate, mentre con "operator" l'operatore/gestore ottiene sole le metriche che sono state formalizzate per l'operatore/gestore in questione.
<b>Esempi</b>	
Recupero di tutte le metriche che sono state formalizzate.	
<a href="http://dashboard.km4city.org/api/v1/metrics/get_list.php?selection=all">http://dashboard.km4city.org/api/v1/metrics/get_list.php?selection=all</a> <b>JSON Result is:</b> <pre> {   "metrics": [     {       "id": "1",       "name": "Bus_State_Lines",       "description": "Percentuale di ritardi/anticipi sulle linee monitorate",       "dataArea": "Mobilità",       "dataSource": "Osservatorio Regionale per la Mobilità ed i Trasporti",       "metricType": "Percentuale",       "status": "Attivo",       "frequency": "60",       "contextInformation": "no"     },     {       "id": "2",       "name": "Park_Free",       "description": "Percentuale dei posti liberi nei parcheggi di un certo comune",       "dataArea": "Mobilità",       "dataSource": "Osservatorio Regionale per la Mobilità ed i Trasporti",       "metricType": "Percentuale", </pre>	

```

    "status": "Attivo",
    "frequency": "60",
    "contextInformation": "si"
  },
  {
    "id": "3",
    "name": "Electricity_Consum_District",
    "description": "Consumo di energia elettrica per quartiere",
    "dataArea": "Energia",
    "dataSource": "ENEL",
    "metricType": "Numerico",
    "status": "Attivo",
    "frequency": "60",
    "contextInformation": "si"
  },
  {
    "id": "4",
    "name": "Stored_Electricity_Month",
    "description": "Energia elettrica accumulata mensile",
    "dataArea": "Energia",
    "dataSource": "ENEL",
    "metricType": "Numerico",
    "status": "Attivo",
    "frequency": "60",
    "contextInformation": "no"
  },
  {
    "id": "5",
    "name": "Power_cut_Month",
    "description": "Numero di interruzioni per utente e per mese",
    "dataArea": "Energia",
    "dataSource": "ENEL",
    "metricType": "Numerico",
    "status": "Attivo",
    "frequency": "60",
    "contextInformation": "no"
  }
]
}

```

<b>Nome:</b> Service Search
<b>URL:</b> <a href="http://dashboard.km4city.org/api/v1/metrics/get_data_metric.php">http://dashboard.km4city.org/api/v1/metrics/get_data_metric.php</a>
<b>Descrizione</b>
Questa API consente all'operatore/gestore di servizi di recuperare dal DIBUM dati relative a metriche formalizzate e precedentemente inviati .

Modalità: GET	
Parametri	
metric	Nome della metrica di cui si vuole recuperare i dati.
startDate	(optional) Data di inizio dell' intervallo di selezione espressa come YYYY-MM-DD.
endDate	(optional) Data di fine dell' intervallo di selezione espressa come YYYY-MM-DD.
Esempi	
<p>Recupero dal DIBUM dei dati relativi alla metrica " Electricity_Consum_District " con un intervallo di selezione compreso tra 2015-7-11 e 2015-8-14.</p> <p><a href="http://dashboard.km4city.org/api/v1/metrics/get_data_metric.php?metric=Power_cut_Month&amp;startDate=2015-7-11&amp;endDate=2015-8-14">http://dashboard.km4city.org/api/v1/metrics/get_data_metric.php?metric=Power_cut_Month&amp;startDate=2015-7-11&amp;endDate=2015-8-14</a></p> <p><b>JSON Result is:</b></p> <pre>{   "id": "5",   "name": "Power_cut_Month",   "description": "Numero di interruzioni per utente e per mese",   "dataArea": "Energia",   "sourceData": "ENEL",   "metricType": "Numerico",   "data": [     {       "computationDate": "2015-7-12 15:40:13",       "value": [2.1]     },     {       "computationDate": "2015-8-12 15:40:13",       "value": [2.3]     }   ] }</pre>	

## 5.2 Specifica: Sviluppo del Sistema Interoperabile integrato, SII (Attività 2.1 esclusa bigliettazione 2.1.4)

		FCM	Mizar	UNIFI	IN20	TIME	Negentis	Fffknow	liberologico	ataf	busitalia	cttnord	tiemme	areos	elfi	calamai	Midra	Proiert	GEOJIN	OUESTIT	SOFTEC	FWINGS
2.1.1	Sviluppo del Sistema Interoperabile integrato, SII: Controllo e monitoraggio	X	X	X	X																	
2.1.2	Sviluppo del Sistema Interoperabile integrato, SII: centrale operativa		X	X	X																	
2.1.3	Sviluppo del Sistema Interoperabile integrato, SII: simulazione	X					X															
2.1.5	Sviluppo del Sistema Interoperabile integrato, SII: Merci e logistica	X	X			X																
2.1.6	Sviluppo del Sistema Interoperabile integrato, SII: pianificazione	X	X																			
2.1.7	Sviluppo del Sistema Interoperabile integrato, SII: big data e sviluppo SII			X																		

### Principali Sottosistemi da descrivere in questa Subsubsection, sono i sottosistemi primari della specifica

	Sotto Attività	Partner coord	Sottosistema	Tool/Level	Descrizione sommaria
2.1.1	Sviluppo del Sistema Interoperabile integrato, SII: Controllo e monitoraggio	MIZAR UNIFI	Support of Integrated Interoperability	Monitoring Service Interface (D01)	Strumento di monitoraggio includendo servizi/ sistemi di Sii-Mobility. I risultati di questo strumento finiscono come KPI sul Dashboard.
		MIZAR	Support of Integrated Interoperability	Monitoring Supervisor (D02)	Strumento di monitoraggio del sistema di gestione del traffico. I risultati di questo strumento finiscono come KPI sul Dashboard.
2.1.2	Sviluppo del Sistema Interoperabile integrato, SII: centrale operativa	UNIFI MIZAR	Control Interface	Dashboard Control Interface, DACI (T03)	Un set di Dashboard per ogni area di validazione e area geografica, comune, provincia, regione e consuntive del Sistema smart city. Ogni area di sperimentazione avrà la sua DACI. Le Dashboard sono utilizzate anche nel sistema di Partecipazione e Sensibilizzazione dei cittadini.
2.1.3	Sviluppo del Sistema Interoperabile integrato, SII: simulazione	UNIFI DISIT	Sii-Mobility Developer Tools	Simulator Interface (T04)	Strumento di front end dei processi di simulazione riguardo a mobilità, etc. (sviluppato su base ServiceMap). Lo strumento visualizza le distribuzioni, i flussi, i percorsi, etc. Utilizzato anche in ambito pianificazione percorsi merci.

2.1.5	Sviluppo del Sistema Interoperabile integrato, SII: Merci e logistica	TIME	Sii-Mobility Developer Tools	Path Planner Interface(merci e logistica) (T05)	Strumento di front end (di visualizzazione e/o impostazione dati dei punti di passaggio per l'attivazione di algoritmi che generano percorsi) dei processi di ottimizzazione dei percorsi (sviluppato su base ServiceMap). Utilizzato in ambito pianificazione percorsi merci.
2.1.6	Sviluppo del Sistema Interoperabile integrato, SII: pianificazione	MIZAR	Sii-Mobility Developer Tools	Path Planner Interface (T05)	Strumento di front end (di visualizzazione e/o impostazione dati dei punti di passaggio per l'attivazione di algoritmi che generano percorsi, guida connessa, flussi persone e mezzi,) dei processi di ottimizzazione dei percorsi (sviluppato su base ServiceMap).
2.1.7	Sviluppo del Sistema Interoperabile integrato, SII: big data e sviluppo SII	UNIFI	Support of Integrated Interoperability	Architettura ed esecuzione processi (HWSW01)	Architettura parallela per l'esecuzione di processi di data analytics anche forniti da terzi. Soluzione basta su Hadoop.
		UNIFI	Support of Integrated Interoperability	Conditional Access Data, CAD (T06)	Sistema di accesso condizionato alla Knowledge Base, ai tool di Sii-Mobility, anche utilizzato per le API. Su questa base gli sviluppatori possono sapere che tipo di licenza potrebbe essere utile avere per accedere a certi dati, etc.

## 5.2.1 Monitoring Service Interface, e Supervisor (D01a, D01b, D02) (MIZAR)

**Monitoring Services Interface + Monitoring Processes**

**Sottosistema: Monitoring Services Interface + Monitoring Processes.** Questo è il sistema di monitoraggio dell'intera soluzione smart city Sii-Mobility. Tale Sottosistema controlla l'esecuzione ed il funzionamento delle macchine virtuali e dei processi. Riporta alcune informazioni direttamente sulla Dashboard tecnica della smart city, per esempio il numero di colpi di clock usati al secondo, lo stato della memoria, il numeri di processi eseguiti, il numero di macchine virtuali attive rispetto a quello configurato, lo stato di salute del sistema, etc.

### 5.2.1.1 Monitoring Service Interface (D01a) (MIZAR)

<b>Profilo Tool/Algoritmo</b>	
Monitoring Service Interface, supervisore (D01a)	
Nome responsabile	
Partner responsabile	MIZAR
Descrizione Tool/algoritmo	Strumento di monitoraggio di servizi SaaS di Sii-Mobility. I risultati di questo strumento finiscono come metriche su qualche dashboard. Anche verso lo stesso strumento di Dashboard che viene sviluppato per il servizi smart city.
Dati primari in ingresso	- flussi dati esposti per SII - stato dei moduli del supervisore (FI) con relative soglie di allarme
Dati prodotti in Uscita	KPI dei dati in ingresso
Principi (metodo, basi teoriche, etc.)	Standard per la qualità degli impianti. Il servizio

	espone le metriche di funzionamento dei servizi del Supervisore (utilizzo hardware, stato dei servizi, stato disco)	
Casi di test (presenti/assenti)	Nessuno	
Posizione casi di test	n.a.	
Principali problemi non risolti	n.a.	
Principali requisiti pendenti	n.a.	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	C#	
Piattaforme supportate	Windows	
Posizione del codice sorgente	Repository di progetto	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.a.	
Indirizzo/i accesso via web (se presenti)	n.a.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n.a.	n.a.	n.a.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
n.a.	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
n.a.	n.a.	n.a.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
SystemMonitoring	Database contenente i KPI di funzionamento del centro della mobilità	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
n.a.	n.a.	n.a.
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>



n.a.	n.a.	Proprietaria, concessa in uso gratuito per il progetto SII

### 5.2.1.2 Monitoring Service Interface, processi interni (D01b) (UNIFI)

<b>Profilo Tool/Algoritmo</b>		
Monitoring Service Interface, processi interni (D01b)		
Nome responsabile		
Partner responsabile	UNIFI	
Descrizione Tool/algoritmo	Strumento di monitoraggio di servizi, processi e flussi all'interno del sistema di controllo Sii-Mobility	
Dati primari in ingresso	<ul style="list-style-type: none"> <li>- flussi dati esposti</li> <li>- stato dei moduli</li> <li>- processi ETL e altro schedulati</li> </ul>	
Dati prodotti in Uscita	Riporto in dashboard	
Principi (metodo, basi teoriche, etc.)	Standard per la qualità degli impianti.	
Casi di test (presenti/assenti)	Nessuno	
Posizione casi di test	n.a.	
Principali problemi non risolti	n.a.	
Principali requisiti pendenti	n.a.	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	PHP, JAVA	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	PHP, JAVA	
Piattaforme supportate	Linux	
Posizione del codice sorgente	Repository di progetto	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.a.	
Indirizzo/i accesso via web (se presenti)	n.a.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n.a.	n.a.	n.a.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
n.a.	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>

n.a.	n.a.	n.a.
Nomi database usati (aggiungere una riga per ogni db usato)		Descrizione
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
n.a.	n.a.	n.a.
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
n.a.	n.a.	n.a.

### 5.2.1.3 Monitoring Process, per processi caricati (D02) (MIZAR)

<b>Profilo Tool/Algoritmo</b>	
Monitoring Supervisor (D02)	
Nome responsabile	
Partner responsabile	MIZAR
Descrizione Tool/algoritmo	Strumento di monitoraggio del sistema di gestione del traffico. I risultati di questo strumento finiscono come KPI sul Dashboard. Sempre Accessibili tramite: (i) accesso a database oppure (ii) chiamate REST.
Dati primari in ingresso	- regole decisionali (strategy manager, monitoraggio membership)
Dati prodotti in Uscita	- regole decisionali (strategy manager, monitoraggio membership)
Principi (metodo, basi teoriche, etc.)	KPI
Casi di test (presenti/assenti)	Il sistema usa soglie predefinite per il calcolo dei KPI in base ai dati di ingresso. Il servizio monitora lo stato dei servizi e dell'hardware, comparandolo con delle soglie predefinite
Posizione casi di test	Nessuno
Principali problemi non risolti	n.a.
Principali requisiti pendenti	n.a.
Aspetti Tecnologici	n.a.
Stato (proposto/approvato)	
Implementato/non implementato	Proposto
Stato implementazione, percentuale	Non Implementato
Eseguibile/libreria/web app	0%
Single thread/Multithread	Eseguibile
Linguaggio di sviluppo (java, Php, ETL)	Multithread
Piattaforme supportate	C#
Posizione del codice sorgente	Windows

Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	Repository di progetto	
Indirizzo/i accesso via web (se presenti)	n.a.	
	n.a.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n.a.	n.a.	n.a.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
n.a.	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
n.a.	n.a.	n.a.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
StrategyMonitoring	Database contenente i KPI del funzionamento del sistema di gestione del traffico	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
n.a.	n.a.	n.a.
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
		Proprietaria, concessa in uso gratuito per il progetto SII

### 5.2.2 Dashboard Control Interface, DACI (T03) (UNIFI)

#### Dashboards, Control Interface

<b>Profilo Tool/Algoritmo</b>	
Dashboard Control Interface, DACI (T03)	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Strumento di generazione delle istanze di un set di Dashboard create attraverso il DIBUM.
Dati primari in ingresso	Dati relative alle configurazioni delle dashboard create attraverso il DIBUM

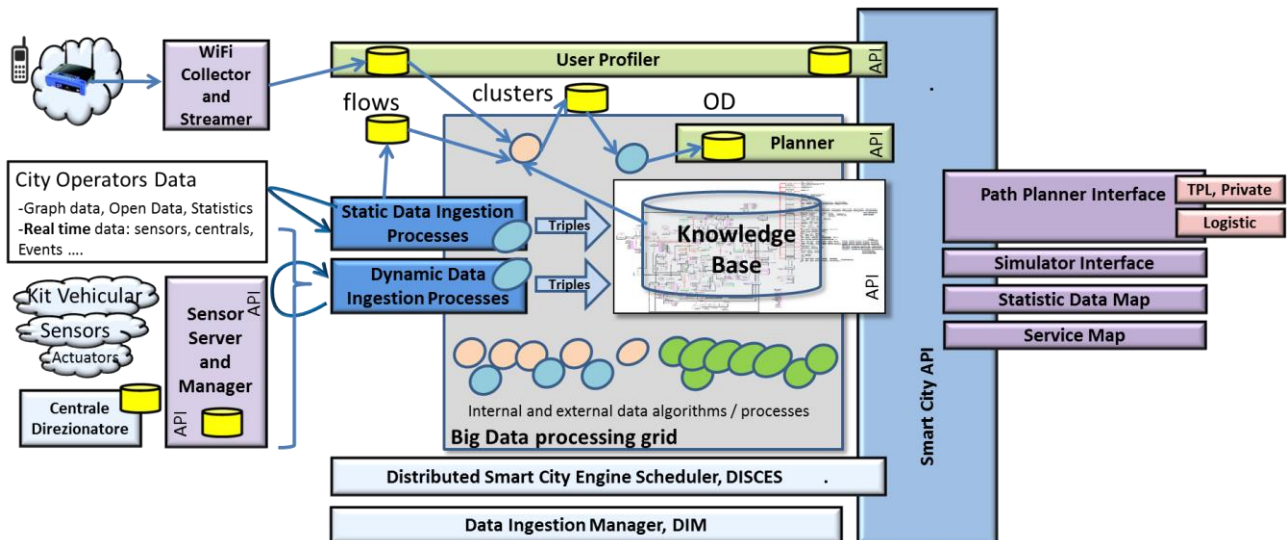
Dati prodotti in Uscita	Istanze delle dashboard	
Principi (metodo, basi teoriche, etc.)	Un set di Dashboard per ogni area di validazione e area geografica, comune, provincia, regione e consuntive del Sistema smart city. Le dashboard possono essere utilizzate anche nel sistema di Partecipazione e Sensibilizzazione dei cittadini.	
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	n/d	
Principali problemi non risolti	Assenti	
Principali requisiti pendenti	Nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0.0%	
Eseguibile/libreria/web app	Web app	
Single thread/Multithread	Single thread	
Linguaggio di sviluppo (java, Php, ETL)	PHP, JavaScript, HTML, CSS	
Piattaforme supportate	Unix, Windows, OS X	
Posizione del codice sorgente	svn	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Dashboard_manager	n/a	n/a
Widgets	Twitter Vigilance API, Service Map API	PUSH/PULL
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON		JavaScript Object Notation
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTP		HyperText Transfer Protocol
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
MySql	Database relazionale	
<b>Tipo interfaccia utente</b>	<b>Modello sviluppo,</b>	<b>Libreria usata per UI</b>

(web/applicazione)	linguaggio	
Web	HTML, JavaScript, CSS	Bootstrap
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
Gridster		Licenza MIT
JQuery	1.10.1	Licenza MIT
Bootstrap	3.3.6	Licenza MIT

### 5.2.3 Simulator Interface (T04) (UNIFI DISIT)

Strumento di front end dei processi di simulazione riguardo a mobilità, etc. (sviluppato su base ServiceMap), nel senso che puo' essere da questo derivato come semplicemente sviluppato utilizzando le SMART City API, come viene fatto con le APP HTML5. Lo strumento visualizza le distribuzioni, i flussi, i percorsi, etc. Utilizzato anche in ambito pianificazione percorsi merci.

### 5.2.4 Path Planner Interface (Merci e Logistica) (T05) (TIME)



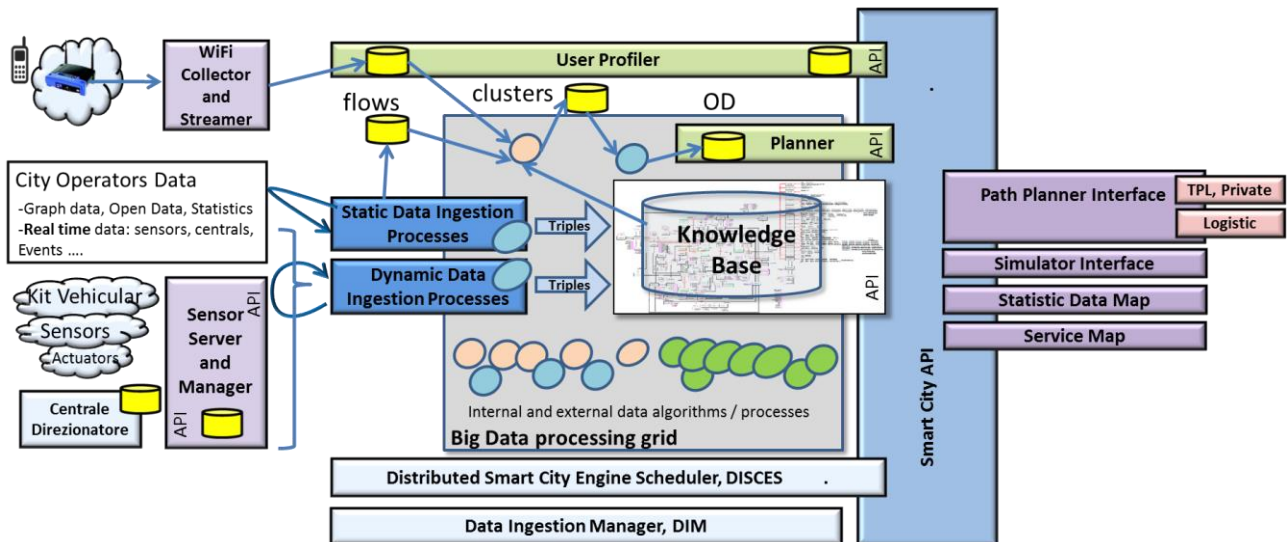
Questo tool deve consentire la produzione di percorsi ottimizzati per ciascun mezzo/utente, basati sulla lista di ritiri e prelievi e consegne/ritiri generata dal modulo di cui a 5.1.7, fornendo un percorso ottimizzato che tiene conto dei vincoli presenti nella lista di consegne/ritiri (orari, tempi, chiusure ZTL ed altri) delle condizioni dinamiche le cui informazioni sono rese disponibili da altri moduli SII MOBILITY, come ad esempio interruzioni stradali, nonché di fattori dinamici come le condizioni del traffico; in questo senso il modulo avrà una funzionalità standard che renderà disponibile all'inizio di ogni giornata i percorsi ottimizzati per ogni mezzo/utente ed una funzionalità on demand che consentirà all'utente di verificare eventuali variazioni dei percorsi correlate allo stato del traffico o altri fattori dinamici.

<h2 style="margin: 0;">Profilo Tool/Algoritmo</h2>	
Logistic Path Planner Interface (merci e logistica) (T05)	
Nome responsabile	
Partner responsabile	TIME

Descrizione Tool/algorithmo	Strumento di front end (di visualizzazione e/o impostazione dati dei punti di passaggio per l'attivazione di algoritmi che generano percorsi) dei processi di ottimizzazione dei percorsi (sviluppato su base ServiceMap). A partire dalle liste di pick up, consegna e ritiro prodotte dal modulo A07 in 5.1.7, il tool interagisce con i moduli SII Mobility relativi a traffico, interruzioni, lavori, etc.) ed attraverso algoritmi di ottimizzazione dei percorsi messi a disposizione dal modulo A06 fornisce il percorso ottimale ad ogni utente secondo regole di minor tempo/minor percorso considerando i vincoli esistenti (tempi ed orari di consegna massimi) e delle informazioni disponibili sulla piattaforma SII MOBILITY relativamente alla viabilità urbana, nonché a fattori dinamici, aggiornati in real time quali interruzioni, stato del traffico, temporanee limitazioni. Il tool può inoltre utilizzare API SII Mobility disponibili come la API Request Planning (v. 5.8.10) o gli algoritmi di instradamento di cui Al modulo A06 in par. 5.1.6	
Dati primari in ingresso	Liste prodotte da A07	
Dati prodotti in Uscita	Percorsi ottimizzati per zone/utente	
Principi (metodo, basi teoriche, etc.)	Algoritmi disponibili al modulo A06	
Casi di test (presenti/assenti)	Da definire	
Posizione casi di test	Da definire	
Principali problemi non risolti	Nessuno	
Principali requisiti pendenti	Nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implemento	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Web APP	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java	
Piattaforme supportate	Windows, Linux	
Posizione del codice sorgente	n.d.	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.d.	
Indirizzo/i accesso via web (se presenti)	n.d.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Modulo A06	Java API	XML
Modulo A07	MySQL connector driver.	Accesso e condivisione dati su database relazionale MySQL.

Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
XML		
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
http, HTTPS		
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
DA DEFINIRE		
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
WEB	JAVA	
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

### 5.2.5 Path Planner Interface (T05) (MIZAR)



Supporto **alla pianificazione** per avere informazioni statistiche dei flussi di persone nelle varie zone e sui vari mezzi.

#### Sottosistema: Path Planner Interface,

- **TPL e Privati.** Questo modulo permette la definizione di percorsi da-a per privati e TPL in modo che l'operatore possa ottenere la soluzione ottima o sub-ottima. La pianificazione delle traiettorie per il trasporto pubblico deve essere multimodale più tipi di trasporto pubblico (diversi operatori bus, operatori treno, etc.) ed inserimento di tratte a piedi. Per il trasporto privato si assume auto, moto, furgone, si massimizza il raggiungimento della meta, si possono

avere pedaggi di vario tipo ma vanno evidenziati, etc. Per il trasporto privato si può avere anche il trasporto via bike. In questo caso si dovrebbero preferire le ciclabili.

- **Logistic.** Questo modulo permette la definizione di percorsi di prelievo in modo che l'operatore possa ottenere la soluzione ottima o subottima, l'operatore può chiedere percorsi che prevedono fermate per prelievi e consegne multiple lungo per percorso, e anche punti di scarico convenzionati.

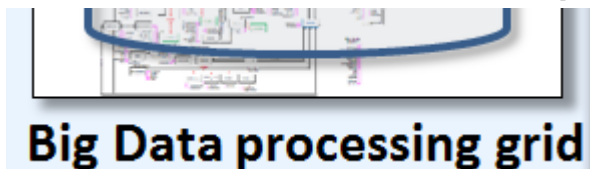
Questo modulo sfrutta Smart City API, ServiceMap, etc. e sviluppa questo servizio come algoritmi caricati sul Big Data Processing Grid che espongono i servizi come risultati tramite le API. La stessa chiamata di questi algoritmi viene a fare parte delle Smart City API di Sii-Mobility e potrà essere usata da applicazioni web, mobile e kit veicolari.

Profilo Tool/Algoritmo		
TPL Path Planner Interface (T05)		
Nome responsabile		
Partner responsabile	MIZAR	
Descrizione Tool/algoritmo	Strumento di front end (di visualizzazione e/o impostazione dati dei punti di passaggio per l'attivazione di algoritmi che generano percorsi, guida connessa, flussi persone e mezzi,) dei processi di ottimizzazione dei percorsi (sviluppato su base ServiceMap), nel senso che può essere da questo derivato come semplicemente sviluppato utilizzando le SMART City API, come viene fatto come le APP HTML5.	
Dati primari in ingresso	Punto di inizio e punto di fine del percorso	
Dati prodotti in Uscita	Lista di archi del percorso	
Principi (metodo, basi teoriche, etc.)	Dopo aver ricevuto i dati in ingresso, il sistema calcola il percorso migliore e restituisce la lista degli archi da percorrere tramite l'uso di procedure caricate sul back office di Sii-mobility	
Casi di test (presenti/assenti)	Nessuno	
Posizione casi di test	n.a.	
Principali problemi non risolti	Nessuno	
Principali requisiti pendenti	Nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	implementato	
Stato implementazione, percentuale	50%	
Eseguibile/libreria/web app	Web app	
Single thread/Multithread	Single thread	
Linguaggio di sviluppo (java, Php, ETL)	C# → PHP o direttamente in HTML5 e JavaScript	
Piattaforme supportate	Windows → tutte tramite web	
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato



n.a.	n.a.	SAOP
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
Data Objects	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
SOAP	n.a.	MPK
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	Descrizione	
n.a.	n.a.	
<b>Tipo interfaccia utente</b> (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
nessuna	n.a.	n.a.
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

### 5.2.6 Architettura ed Esecuzione processi (HWSW01) (UNIFI)

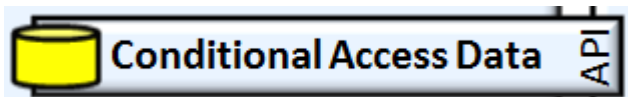


<b>Profilo Tool/Algoritmo</b>	
Architettura ed Esecuzione Processi (HWSW01)	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Architettura parallela per l'esecuzione di processi di data analytics (anche forniti da terzi.) e memorizzazione dei loro risultati. Soluzione basta su Hadoop.
Dati primari in ingresso	Stringhe
Dati prodotti in Uscita	Informazioni sul grafo stradale Algoritmi di routing e informazioni sulla mobilità Dati statistici su utenti, messaggi dei social media Stato sui processi.
Principi (metodo, basi teoriche, etc.)	I dati vengono memorizzati su database distribuito nosql HBASE come tabelle BigTable. All'occorrenza

	vengono processati mediante paradigma di computazione parallela mapreduce.	
Casi di test (presenti/assenti)	assenti	
Posizione casi di test	.	
Principali problemi non risolti	.	
Principali requisiti pendenti	.	
Aspetti Tecnologici		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java, Php, Hadoop	
Piattaforme supportate	Linux, Windows	
Posizione del codice sorgente	SVN	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	.	
Indirizzo/i accesso via web (se presenti)	.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n/a	n/a	n/a
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
n/a	n/a	n/a
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
n/a	n/a	n/a
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
n/a	n/a	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
applicazione	java	n/a
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale,</b>

		etc.
Hadoop	2.6.2	APACHE LICENCE 2.0 (compatible con GPL v3)
Hbase	1.1.2	APACHE LICENCE 2.0 (compatible con GPL v3)

### 5.2.7 Conditional Access Data, CAD (T06) (UNIFI)



<b>Profilo Tool/Algoritmo</b>	
Conditional Access Data, CAD (T06)	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Sistema di accesso condizionato alla Knowledge Base, ai tool di Sii-Mobility, anche utilizzato per le API. Su questa base gli sviluppatori possono sapere che tipo di licenza potrebbe essere utile avere per accedere a certi dati, etc.
Dati primari in ingresso	Query, richieste API e indicazione dell'utente/applicazione
Dati prodotti in Uscita	Indicazione se i dati richiesti sono accessibili dall'utente e la licenza con la quale sono accessibili e quali dataset vengono usati in una richiesta.
Principi (metodo, basi teoriche, etc.)	Vengono individuati i dataset usati dalla query trasformando la query SPARQL originaria in una che identifica i dataset usati, questo in fase di validazione della query mentre in produzione la query verrà realizzata basandosi sul profilo utente e limitando l'accesso ai soli dataset accessibili dall'utente.
Casi di test (presenti/assenti)	assenti
Posizione casi di test	n/a
Principali problemi non risolti	nessuno
Principali requisiti pendenti	nessuno
<b>Aspetti Tecnologici</b>	
Stato (proposto/approvato)	proposto
Implementato/non implementato	non implementato
Stato implementazione, percentuale	0%
Eseguibile/libreria/web app	web app
Single thread/Multithread	multithread
Linguaggio di sviluppo (java, Php, ETL)	Java, Python
Piattaforme supportate	Quelle supportate da java e Python
Posizione del codice sorgente	n/a
Indirizzo/i web services (se presenti) con	n/a

indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)	n/a	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
RDF Store		SPARQL 1.1 HTTP Protocol
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
SPARQL XML Results		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
SPARQL 1.1 HTTP Protocol		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

### 5.3 Specifica: Modulo per il caricamento algoritmi aggiuntivi

		ECM	Mizar	UNIFI	IN20	TIME	Negentis	EffKnow	liberologico	ataf	buisitalia	cttnord	fiemme	argos	elfi	calamai	Midra	Proiect	GEOIN	OUJESTIT	SOETEC	FWINGS
2.6.1	Supporto per, caricamento ed uso di moduli aggiuntivi sul SII: formalizzazione modello moduli e algoritmi, parte back end			X																		
2.6.2	Supporto per, caricamento ed uso di moduli aggiuntivi sul SII: strumento per il caricamento e installazione moduli e algoritmi		X			X																
2.6.3	Supporto per, caricamento ed uso di moduli aggiuntivi sul SII: gestione moduli e alg, monitoraggio, accesso ai dati			X		X																
2.6.4	Supporto per, caricamento ed uso di moduli aggiuntivi sul SII: API di accesso ai dati prodotti in automatico dagli algoritmi caricati			X																		

Principali Sottosistemi da descrivere in questa Subsubsection, sono i sottosistemi primari della specifica					
	Sotto Attività	Partne r coord	Sottosiste ma	Tool/Level	Descrizione sommaria
2.6.1	Supporto per, caricamento ed uso di moduli aggiuntivi sul SII: formalizzazione modello moduli e algoritmi, parte back end	UNIFI	Smart City API	Smart City API Process Loading (API03)	API per fornire al sistema DISCES un nuovo processo da eseguire: Start, periodic, duration, etc. API con accesso condizionato in base all'utente, al profilo utente, ai dati richiesti.
2.6.2	Supporto per, caricamento ed uso di moduli aggiuntivi sul SII: strumento per il caricamento e installazione moduli e algoritmi	TIME	Sii-Mobility Developer Tools	Algorithm / Process Loader (T07)	Strumento per il caricamento di algoritmi e processi. Questi vengono presi in carico e messi in esecuzione anche con il supporto di DISCES. Questi algoritmi hanno accesso ai dati delle Knowledge Base tramite le API che seguono. I risultati possono essere prodotti all'esterno via API o lasciati in un disco di share o FTP.
2.6.3	Supporto per, caricamento ed uso di moduli aggiuntivi sul SII: gestione moduli e alg, monitoraggio	UNIFI	Sii-Mobility Developer Tools	gestione moduli e alg, monitoraggio, accesso ai dati (T08)	Strumento per la gestione dei moduli aggiuntivi caricati sul sistema Sii-Mobility di esecuzione in back office. Lista dei processi, storico, utenti, stato, etc.
2.6.4	Supporto per, caricamento ed uso di moduli aggiuntivi sul SII: API di accesso ai dati prodotti in automatico dagli algoritmi caricati	UNIFI	Smart City API	API Process Loaded, get data (API04)	API con accesso condizionato in base all'utente, al profilo utente, per accedere ai dati richiesti e prodotti dagli algoritmi aggiuntivi.

#### 5.3.1 Smart City API: Process Loading (API03) (connected to DISCES) (UNIFI)

### Smart City API

## Profilo Tool/Algoritmo

Smart City API: Process Loading (conected to DISCES) (API03)

Nome responsabile		
Partner responsabile	UNIFI	
Descrizione Tool/algorithmo	<p>API per fornire al sistema DISCES un nuovo processo da eseguire: Start, periodic, duration, etc..</p> <p>API con accesso condizionato in base all'utente, al profilo utente, ai dati richiesti.</p> <p>API per:</p> <ul style="list-style-type: none"> <li>• Caricamento del processo sul sistema Sii-Mobility backoffice, il che significa metterlo in esecuzione su DISCES, dagli un periodo, etc. etc.</li> <li>• Monitorare lo stato del processo da remoto con una chiamata REST</li> </ul>	
Dati primari in ingresso	.	
Dati prodotti in Uscita	.	
Principi (metodo, basi teoriche, etc.)	.	
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	n/a	
Principali problemi non risolti	Assenti	
Principali requisiti pendenti	Nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0.0%	
Eseguibile/libreria/web app	web app	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	PHP	
Piattaforme supportate	Quelle supportate da PHP	
Posizione del codice sorgente	n/d	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
api	API DISCES per produrre ed attivare processi	http rest call.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON, ZIP		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione</b>

		definizione
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	Descrizione	
<b>Tipo interfaccia utente</b> (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
n/a		
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

### 5.3.1.1 Protocollo, new process loading and update

Si suppone che l' sviluppatore possa caricare un processo ottenendo indietro l'ID del processo anche da interfaccia utente web. Il caricamento potrebbe avvenire fornendo un ETL o un JAVA o un grappolo di file da mettere in una directory di un macchina LINUX.

Al momento del caricamento viene chiesto se il processo deve essere eseguito periodicamente, oppure on-demand sulla base di qualche richiesta, chiamata REST. Nel primo caso viene schedulato come processo DISCES. In ogni caso il processo deve essere fornito di chiamata stringa di lancio per una Shell in ambiente Linux. L'esecuzione del processo dall'esterno sarà del tipo:

<http://process.km4city.org/api/v1/process/status?id=2345&action=run>

e potrà essere replicata solo a processo terminato oppure dopo aver fermato il processo con il comando di tipo:

<http://process.km4city.org/api/v1/process/status?id=2345&action=kill>

Il caricamento del processo viene permesso solo se ci si autentifica e l'utente potrà accedere solo al monitoraggio dei suoi processi.

La soluzione, permette di caricare nuovamente una nuova versione di un certo processo fornendo nuovamente l'ID assegnato. Questo permette agli sviluppatori di aggiornare il processo di backoffice senza attendere che questo sia caricato da terzi.

<b>Nome:</b> Process status	
<b>URL:</b> <a href="http://process.km4city.org/api/v1/process/status">http://process.km4city.org/api/v1/process/status</a>	
<b>Descrizione</b>	
Questa API consente di ottenere informazioni sullo stato di esecuzione del processo indicato. L'API sarà autenticata e l'utente potrà accedere solo ai suoi processi.	
Modalità: GET	
Parametri	
id	identificatore del processo inviato nella mail di sottomissione del processo
operation	indica l'operazione da fare sul processo può essere "start", "stop", "pause"
<b>Esempi</b>	
Recupero informazioni di stato sul processo 2345.	

```

http://process.km4city.org/api/v1/process/status?id=2345
JSON Result i:
{
  "id": "2345",
  "name": "un processo",
  "description": "Processo ETL di esempio",
  "submissionTime": "2017-05-01 10:15:34",
  "processType": "ETL",
  "currentState": "RUNNING",
  "lastFireTime": "2017-06-14 10:34:12",
  ...
}
    
```

### 5.3.2 Smart City API Process Loaded, get data (API04) (UNIFI)

#### Smart City API

Profilo Tool/Algoritmo	
API Process Loading, get data (API04)	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	API con accesso condizionato in base all'utente, al profilo utente, per accedere ai dati richiesti prodotti dagli algoritmi aggiuntivi. Si assume che i processi messi in esecuzione producano dei risultati in (i) tabelle lineari su MySQL o che (ii) vadano a scrivere direttamente in RDF store Km4City. Nel caso (i) la tabella puo' essere letta da questa chiamata REST. Per risultati più complessi si assume si possa creare dei file e che questi siano indicati nella tabella MySQL, con datatime, URI/URL, parametri, etc.
Dati primari in ingresso	indicazione del processo a cui accedere per prendere i dati
Dati prodotti in Uscita	i dati del processo
Principi (metodo, basi teoriche, etc.)	.
Casi di test (presenti/assenti)	assenti
Posizione casi di test	n/a
Principali problemi non risolti	nessuno
Principali requisiti pendenti	nessuno
Aspetti Tecnologici	
Stato (proposto/approvato)	proposto
Implementato/non implementato	non implementato
Stato implementazione, percentuale	0%



Eseguibile/libreria/web app	web app	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java/PHP	
Piattaforme supportate	Quelle supportate da PHP o Java	
Posizione del codice sorgente	n/a	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)	n/a	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>external process</b>	informazioni sui processi caricati dagli utenti	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

### 5.3.2.1 Protocollo, API: dati di ritorno da processi caricati

<b>Nome:</b> Process data	
<b>URL:</b> <a href="http://process.km4city.org/api/v1/process/data">http://process.km4city.org/api/v1/process/data</a>	
<b>Descrizione</b>	
Questa API consente di ottenere i dati registrati nella tabella associata al processo.	
Modalità: GET	
<b>Parametri</b>	
id	identificatore del processo inviato nella mail di sottomissione del processo

offset	indica il numero di riga da cui partire (default 0)
limit	indica il numero di righe da restituire (default 100)
operation	può essere "select" (valore di default) per ottenere elenco dati presenti oppure "truncate" per azzerare il contenuto della tabella (l'operazione non può essere annullata)
<b>Esempi</b>	
<p>Recupero 20 righe dei dati prodotti dal processo "p2345" partendo dalla undicesima riga:</p> <p><a href="http://process.km4city.org/api/v1/process/data?id=p2345&amp;offset=10&amp;limit=20">http://process.km4city.org/api/v1/process/data?id=p2345&amp;offset=10&amp;limit=20</a></p> <p><b>JSON Result:</b></p> <pre>{   "id": "p2345",   "name": "un processo",   "processType": "ETL",   "count": "2350", #numero totale di righe   "data": [     { id:"34", "col1" : "a value"},     { id:"35", "col1" : "another value"},     ...   ] }</pre> <p>per azzerare la tabella del processo p2345</p> <p><a href="http://process.km4city.org/api/v1/process/data?id=p2345&amp;operation=truncate">http://process.km4city.org/api/v1/process/data?id=p2345&amp;operation=truncate</a></p>	

### 5.3.3 Algorithm / Process Loader (T07) (TIME)



<b>Profilo Tool/Algoritmo</b>	
Algorithm / Process Loader (T07)	
Nome responsabile	
Partner responsabile	TIME
Descrizione Tool/algoritmo	Strumento per il caricamento di algoritmi e processi. Questi vengono presi in carico e messi in esecuzione anche con il supporto di DISCES. Questi algoritmi hanno accesso ai dati delle Knowledge Base tramite le API che seguono. I risultati possono essere prodotti all'esterno via API o lasciati in un disco di share o FTP.
Dati primari in ingresso	<ul style="list-style-type: none"> <li>- API signatures/interfaces DISCES</li> <li>- Knowledge Base</li> <li>- altri dati e DB in funzione degli algoritmi e del loro ambito di utilizzo</li> </ul>

Dati prodotti in Uscita	Data flow funzionale all'ambito specifico di applicazione dell'algoritmo	
Principi (metodo, basi teoriche, etc.)	Load / unload dinamico di librerie con gestione delle dipendenze di release.	
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	n/d	
Principali problemi non risolti	Assenti	
Principali requisiti pendenti	Assenti	
Aspetti Tecnologici		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0 %	
Eseguibile/libreria/web app	Web app (lato utilizzatore) e Libreria (lato server per la gestione vera propria delle librerie)	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java, PHP	
Piattaforme supportate	Lato Server Linux, Windows; Lato client browser Firefox, Safari, Internet Explorer, Edge.	
Posizione del codice sorgente	n/d	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Eclipse	n/a	n/a
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
Json		JavaScript Object Notation
WSDL		Web Services Description Language
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
http / https		HyperText Transfer Protocol / (Secure)
XML		eXtensible Markup Language
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
MySQL	Database relazionale	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
Web	HTML, JavaScript,	Bootstrap

	CSS	
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
Bootstrap	3.3.6	Licenza MIT
JQuery	2.2.1	Licenza MIT
MySQL Connector	5.7	Licenza GPL

### 5.3.4 Gestione moduli e alg, monitoraggio, accesso ai dati (T08) (UNIFI)

Un elemento di monitoraggio dei processi caricati da terzi e accesso ai loro dati:

- di esecuzione: date and time di on/off, cpu, memoria, tempo di esecuzione, % completamento
- visualizzazione dei dati prodotti dai moduli/algoritmi su database.
- 

<b>Profilo Tool/Algoritmo</b>	
Gestione moduli e alg, monitoraggio, accesso ai dati (T08)	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Strumento per la gestione dei moduli aggiuntivi caricati sul sistema Sii-Mobility di esecuzione in back office. Lista dei processi, storico, utenti, stato, etc. Strumento di monitoraggio dei processi che vengono messi in esecuzione sul back office di Sii-Mobility.
Dati primari in ingresso	
Dati prodotti in Uscita	Informazioni sui processi in esecuzione, dati prodotti dai processi
Principi (metodo, basi teoriche, etc.)	.
Casi di test (presenti/assenti)	assenti
Posizione casi di test	n/d
Principali problemi non risolti	nessuno
Principali requisiti pendenti	nessuno
Aspetti Tecnologici	
Stato (proposto/approvato)	proposto
Implementato/non implementato	non implementato
Stato implementazione, percentuale	0%
Eseguibile/libreria/web app	webapp
Single thread/Multithread	Multithread
Linguaggio di sviluppo (java, Php, ETL)	Java/PHP
Piattaforme supportate	Quelle supportate da Java e PHP
Posizione del codice sorgente	n/d
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d
Indirizzo/i accesso via web (se presenti)	n/d

Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
JSON		
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
HTTP/HTTPS		
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
external_processes	informazioni sui processi caricati dagli utenti	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
web	PHP+HTML+JS+CSS	Bootstrap
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
Bootstrap	3.3.6	Licenza MIT
JQuery	2.2.1	Licenza MIT

## 5.4 Specifica: Infrastruttura HW del sistema integrato Interoperabile

		ECM	Mizar	UNIFI	IN20	TIME	Negentis	EffKnow	liberologico	ataf	busitalia	cttnord	fiemme	argos	elfi	calamai	Midra	Proiect	GEOIN	OUJESTIT	SOETEC	FWINGS
3.6.1	Infrastruttura del sistema integrato: Progettazione congiunta			X	X																	
3.6.2	Infrastruttura del sistema integrato: messa in opera				X																	
3.6.3	Infrastruttura del sistema integrato: Gestione e controllo			X	X																	

Principali Sottosistemi da descrivere in questa Subsubsection, sono i sottosistemi primari della specifica					
	Sotto Attività	Partner coord	Sottosistema	Tool/Level	Descrizione sommaria
3.6.1	Infrastruttura del sistema integrato: Progettazione congiunta	IN20	Sii-Mobility Cloud	Sii-Mobility Cloud (HWSW02)	Hardware e infrastruttura Analisi dei requisiti di dettaglio ed identificazione delle componenti e delle risorse necessarie per l'erogazione dei servizi Cloud Hardware e infrastruttura
3.6.2	Infrastruttura del sistema integrato: messa in opera	IN20	Sii-Mobility Cloud	Sii-Mobility Cloud (HWSW02)	Hardware e infrastruttura Installazione e configurazione delle componenti e delle risorse individuate nella fase di analisi per l'erogazione dei servizi Cloud Hardware e infrastruttura
3.6.3	Infrastruttura del sistema integrato: Gestione e controllo	IN20	Sii-Mobility Cloud	Monitoring Services Interface (HWSW03)	Strumento di monitoraggio dei processi della smart city in cloud. Monitoraggio a livello Host, VM, Servizio.

### 5.4.1 Sii-Mobility Cloud HW (HWSW02) (IN20, UNIFI)

Profilo Tool/Algoritmo	
Sii-Mobility Cloud HW (HWSW02)	
Nome responsabile	
Partner responsabile	IN20
Descrizione Tool/algoritmo	Hardware e infrastruttura Installazione e configurazione delle componenti e delle risorse individuate nella fase di analisi per l'erogazione dei servizi Cloud Hardware e infrastruttura
Dati primari in ingresso	
Dati prodotti in Uscita	
Principi (metodo, basi teoriche, etc.)	
Casi di test (presenti/assenti)	
Posizione casi di test	
Principali problemi non risolti	
Principali requisiti pendenti	
Aspetti Tecnologici	

Stato (proposto/approvato)	proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app		
Single thread/Multithread		
Linguaggio di sviluppo (java, Php, ETL)		
Piattaforme supportate	Esx VMWARE	
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)	<a href="https://asgard.in20.com/cloud/org/IAAS&lt;numero&gt;/">https://asgard.in20.com/cloud/org/IAAS&lt;numero&gt;/</a> e <a href="http://www.disit.org">http://www.disit.org</a>	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
VMware vCloud Director		
VMware vCenter		
VMware Esxi		
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
IAAS		Infrastructure as a Service – Assegnazione di risorse Ghz di CPU, GB di RAM, GB di Spazio disco, Mbit di Banda Internet e IP Pubblici
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
https		Interfaccia web di gestione
Tutti i protocolli richiesti		VM interne allo IAAS
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
SQL Server	Per la gestione dell'infrastruttura virtuale	
Tutti i db richiesti	VM interne allo IAAS previo accordo di licenze	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
web		
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

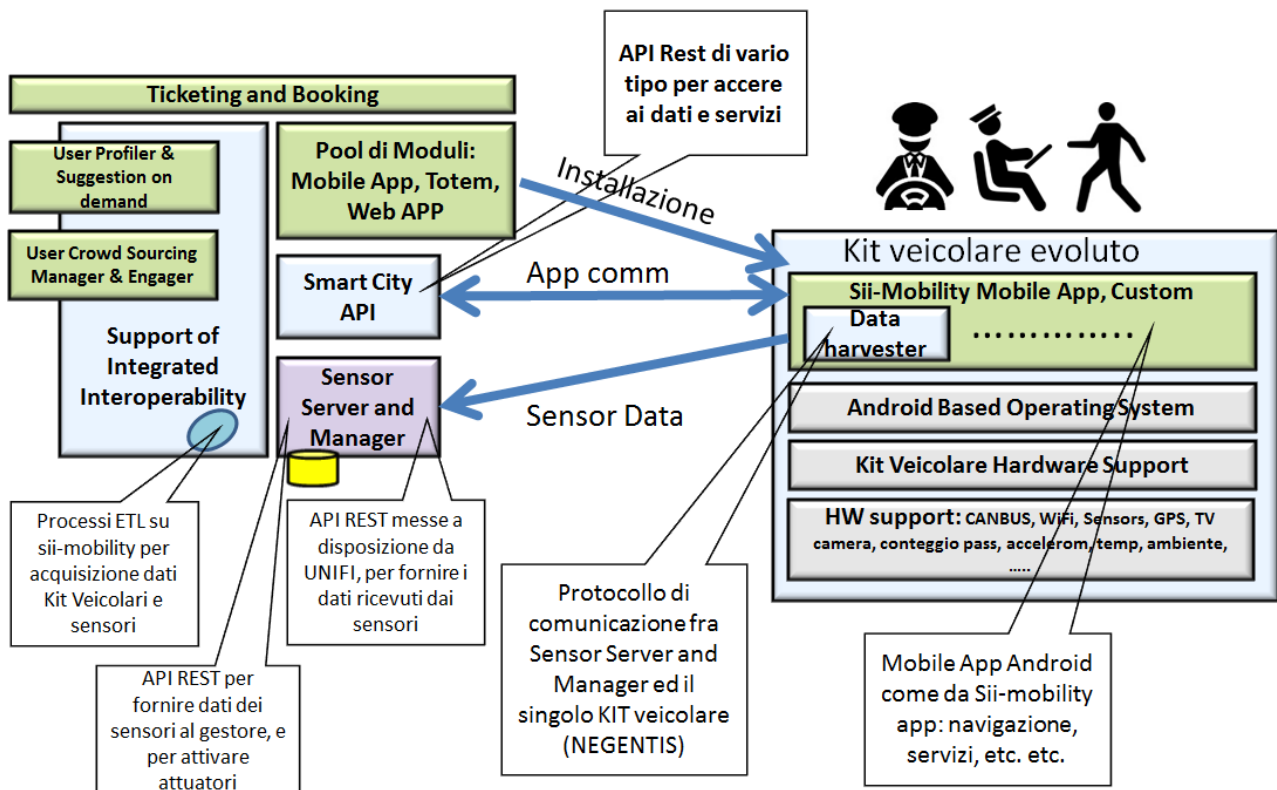

### 5.4.2 Monitoring Services Interface, level IAAS, PAAS (HWSW03) (IN20)

<b>Profilo Tool/Algoritmo</b>		
Monitoring Services Interface, level IAAS, PAAS (HWSW03)		
Nome responsabile		
Partner responsabile	IN20	
Descrizione Tool/algoritmo	Strumento di monitoraggio dei processi della smart city in cloud. Monitoraggio a livello Host, VM, Servizio. Nel dettaglio si assume possa essere realizzato con una soluzione Open Source per il monitoraggio PAAS e IAAS del cloud. Per esempio con un Nagios, o Zabbix, etc.	
Dati primari in ingresso		
Dati prodotti in Uscita		
Principi (metodo, basi teoriche, etc.)		
Casi di test (presenti/assenti)		
Posizione casi di test		
Principali problemi non risolti		
Principali requisiti pendenti		
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app		
Single thread/Multithread		
Linguaggio di sviluppo (java, Php, ETL)		
Piattaforme supportate		
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Zabbix		Monitor VM interne allo IAAS
VMware vRealize Operation Manager		Monitor infrastruttura virtuale
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>



Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
Snmp		Collezione dati relati allo stato di CPU, RAM, Spazio Disco, Carico di rete
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
SQL Server	Per la monitoraggio dell'infrastruttura virtuale	
mysql	Per software monitoraggio VM interne allo IAAS	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
web		
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

### 5.5 Specifica: prototipi applicativi verticali, sensori e attuatori

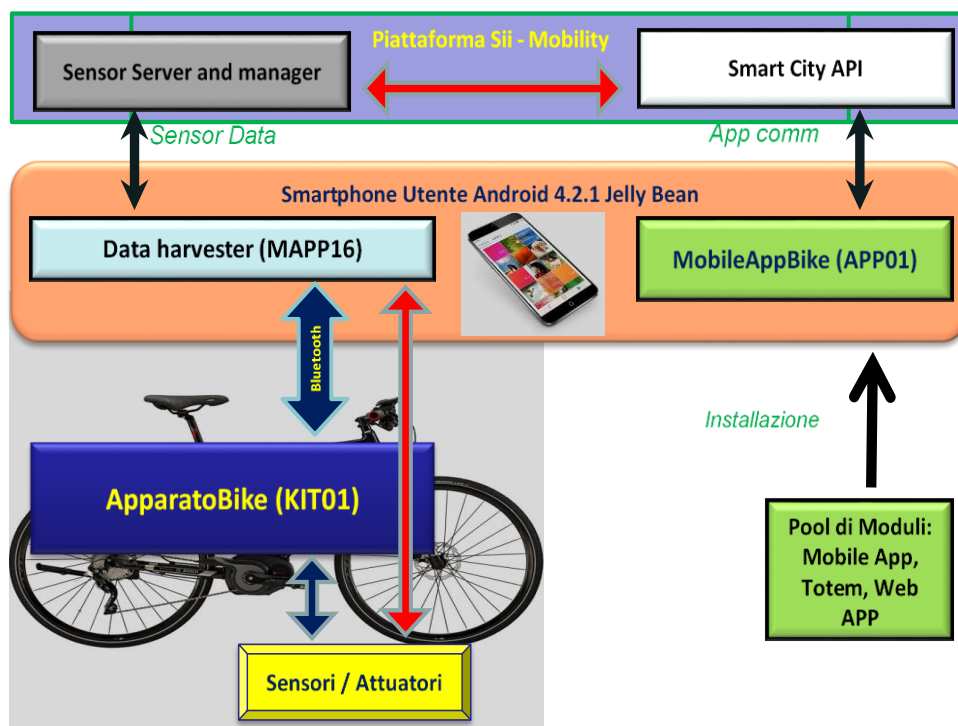


		ECM	Mizar	UNIFI	IN20	TIME	Nezentis	Effknow	liberologico	ataf	busitalia	cttnord	fiemme	argos	elfi	calamai	Midra	Proiect	GEOIN	OLJESTIT	SOITEC	FWINGS
3.1.1	Studio, definizione e sviluppo di Kit per veicoli: bici, auto e mezzi pubblici e privati: kit per bike	X		X									X	X	X							
3.1.2	Studio, definizione e sviluppo di Kit per veicoli: bici, auto e mezzi pubblici e privati: kit per car-bus	X											X	X	X							
3.1.3	Studio, definizione e sviluppo di Kit per veicoli: bici, auto e mezzi pubblici e privati: kit carbus evoluti	X											X	X	X							
3.1.4	Studio, definizione e sviluppo di Kit per veicoli: bici, auto e mezzi pubblici e privati: comunicazione con Sii			X																		

Principali Sottosistemi da descrivere in questa Subsubsection, sono i sottosistemi primari della specifica					
	Sotto Attività	Partner coord	Sottosistema	Tool/Level	Descrizione sommaria
3.1.1	Studio, definizione e sviluppo di Kit per veicoli: bici, auto e mezzi pubblici e privati: kit per bike	ECM	Kit Veicolari per Bike	ApparatoBike (KIT01)	Apparato veicolare per bike. Fornisce le interfacce HW/SW verso i sensori installabili sul veicolo. Fornisce interfaccia Bluetooth verso il dispositivo mobile dell'Utente.
		UNIFI		MobileAppBike (APP01)	Una o più applicazioni pilota da scaricare sul dispositivo mobile dell'Utente. Permettono guida connessa, prenotazione, pagamento, invio a Sii-Mobility dei dati dei sensori veicolari.
		none		Apparato mobile Utente	È lo smartphone dell'Utente, non gli vengono allocati requisiti, ma piuttosto condizioni applicative, ovvero caratteristiche che deve possedere perché le varie funzioni possano essere realizzate.
3.1.2	Studio, definizione e sviluppo di Kit per veicoli: bici, auto e mezzi pubblici e privati: kit per car-bus	ECM	Kit Veicolari per auto e mezzi pubblici	ApparatoCarBus (KIT02)	Apparato veicolare per auto e bus. Richiede l'alimentazione dal veicolo e fornisce dati di localizzazione, velocità, soste, duty cycle e allarmi diagnostici. Invia i dati via GSM alla Piattaforma GUIDO. Apparato veicolare per auto e bus. È costituito da un Tool pre-esistente (T09) formato da un dispositivo da installare a bordo (TURMS) e da una centrale di raccolta dati via GSM (GUIDO). Nel contesto di Sii-Mobility si sviluppa una versione speciale del Data Harvester (MAPP16) e la si integra al SW di GUIDO per collegarla alla piattaforma Sii-Mobility e trasferire i dati veicolari.
				Piattaforma GUIDO (T09)	Tool pre-esistente fornito in comodato d'uso a Sii-Mobility da ELFI. Composto da: Dispositivo TURMS da installare a bordo. Richiede l'alimentazione dal veicolo e fornisce dati di posizione, velocità, accelerazioni via GSM. Centrale operativa GUIDO che concentra i dati dai dispositivi veicolari, li elabora e li rende disponibili agli utenti/gestori. Localizzazione, velocità, soste, duty cycle e allarmi diagnostici.

3.1.3	Studio, definizione e sviluppo di Kit per veicoli: bici, auto e mezzi pubblici e privati: kit carbus evoluti	ECM	Kit Veicolari evoluti per auto e mezzi pubblici	Piattaforma veicolare evoluta (KIT03)	Piattaforma evoluta per auto e bus. Fornisce interfaccia HW/SW verso i sensori installabili sul veicolo. Comprende un Tablet di servizio Android adatto allo sviluppo di Applicazioni mobili pilota a cui fornisce via Bluetooth i dati dei sensori e i servizi per comandare / controllare i sensori
		UNIFI		MobileAppCarBus (APP02)	Una o più applicazioni pilota da installare/scaricare sulla piattaforma evoluta. Utilizzano i servizi della piattaforma per attuare l'invio a Sii-Mobility dei dati di localizzazione, ambientali, diagnostici etc. e per realizzare le funzioni di infomobilità, guida connessa, gestione ZTL e parcheggi etc.
3.1.4	Studio, definizione e sviluppo di Kit per veicoli: bici, auto e mezzi pubblici e privati: comunicazione con Sii	UNIFI	Kit Veicolari	Protocollo di comunicazione fra Sensor Server and Manager ed il singolo KIT veicolare. (API05)	Acquisizione di dati provenienti da Kit Car-Bus-Bike veicolari Sii-mobility, a cura del Custom Crawler and Manager. Parte server su Sensor Server and Manager a cura di UNIFI
		UNIFI DISIT	Kit Veicolari	Data harvester (MAPP16)	Acquisizione di dati provenienti da sensori su kit veicolare tramite Bluetooth.
		UNIFI	Kit Veicolari	Protocollo di comunicazione fra Sensor Server and Manager e processo ETL su Sii-Mobility (PETL24) prima era (API06)	Acquisizione di dati provenienti da Custom Crawler and Manager (riguardo a Kit Car-Bus-Bike Sii-mobility) sviluppati con ETL per portare i dati in triple dentro la Knowledge base in base al modello semantico. Il processo è gestito da DISCES

### 5.5.1 Apparato Bike (KIT01) (ECM)



<h2 style="text-align: center;">Profilo Tool/Algoritmo</h2> <h3 style="text-align: center;">Apparato Bike (KIT01)</h3>	
Nome responsabile	
Partner responsabile	ECM
Descrizione Tool/algoritmo	Apparato veicolare per bike. Fornisce le interfacce HW/SW verso i sensori installabili sul veicolo. Fornisce interfaccia Bluetooth verso l'apparato mobile dell'Utente.
Dati primari in ingresso	Valori di tensione da sensori passivi, messaggi da sensori attivi Protocollo base Bluetooth con Data Harvester (MAPP16) – messaggi di controllo Messaggi dati Bluetooth da Data Harvester (MAPP16) – comandi per sensori
Dati prodotti in Uscita	Valori di tensione per sensori passivi, messaggi per sensori attivi Protocollo base Bluetooth con Data Harvester (MAPP16) – messaggi di controllo Messaggi dati Bluetooth a Data Harvester (MAPP16) – dati da sensori
Principi (metodo, basi teoriche, etc.)	Architettura HW/SW open basata su tecnologia Arduino
Casi di test (presenti/assenti)	assenti
Posizione casi di test	N/A
Principali problemi non risolti	nessuno
Principali requisiti pendenti	nessuno
Aspetti Tecnologici	

Stato (proposto/approvato)	approvato	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0 %	
Eseguibile/libreria/web app	Piattaforma HW/SW embedded	
Single thread/Multithread	N/A	
Linguaggio di sviluppo (java, Php, ETL)	Sketches Arduino	
Piattaforme supportate	N/A	
Posizione del codice sorgente	N/A	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	N/A	
Indirizzo/i accesso via web (se presenti)	N/A	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
N/A	N/A	N/A
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
N/A	N/A	N/A
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
Protocollo base Bluetooth	MAPP16	DE3.1 – sezione 3.2.5
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
N/A	N/A	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
N/A	N/A	N/A
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
N/A	N/A	N/A

## 5.5.2 MobileAppBike (APP01) (UNIFI)



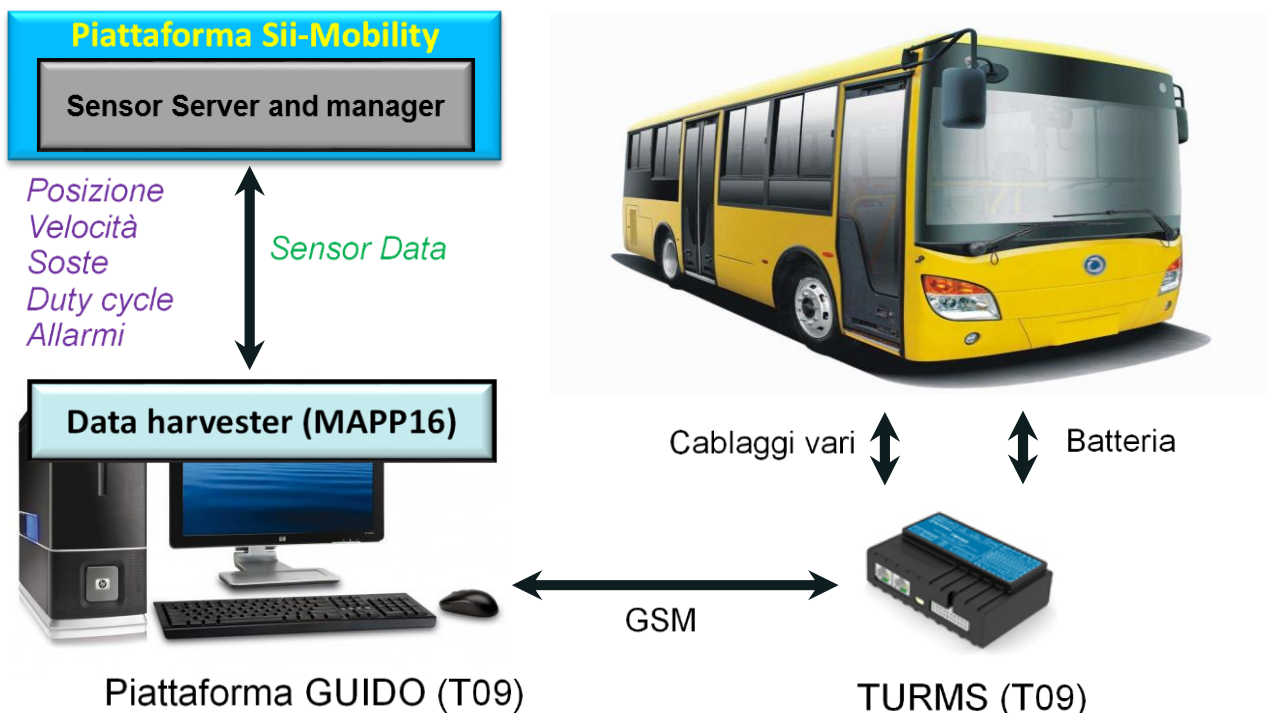
Una o più applicazioni pilota da scaricare sul dispositivo mobile dell'Utente. Tipicamente viene utilizzata la stessa APP mobile della soluzione Sii-Mobility (per esempio Firenze Dove cosa ... Km4city) anche per l'uso in bike o sul kit evoluto ma con aggiunta un processo in background che permette di leggere i dati dai sensori specifici. In questo caso il data harvester legge da Bluetooth il device della bike in altri casi sempre con lo stesso approccio dal kit veicolare.

Sull'App devono essere implementati algoritmi e soluzioni di guida connessa, prenotazione, pagamento, invio a Sii-Mobility dei dati dei sensori veicolari.

In questo modulo si prevede di adattare l'APP di Sii-mobility anche a tali processi, in modo che all'installazione possa riconoscere la condizione e installare i driver specifici, oppure con una installazione adhoc per i kit e per le bike. Vi saranno inoltre degli accorgimenti ed un vista ad hoc per la navigazione in bike: Piste ciclabili, Servizi e POI relativi, Velocità, Stato sensori, Km fatti, connessione e CANBUS, etc.; Suggerimenti ed assistance adhoc.

La soluzione per bike deve offrire una navigazione focalizzata per le piste ciclabili, con segmenti nelle ciclabili, su strada o a piedi; oppure intermodale con mezzi (treni bus) che permettono il carico della bici. Sono inoltre presenti tramite il kit sensori specifici, per esempio: stato della strada, anticollisione, etc.

### 5.5.3 Apparato Car Bus (KIT02) (ECM)

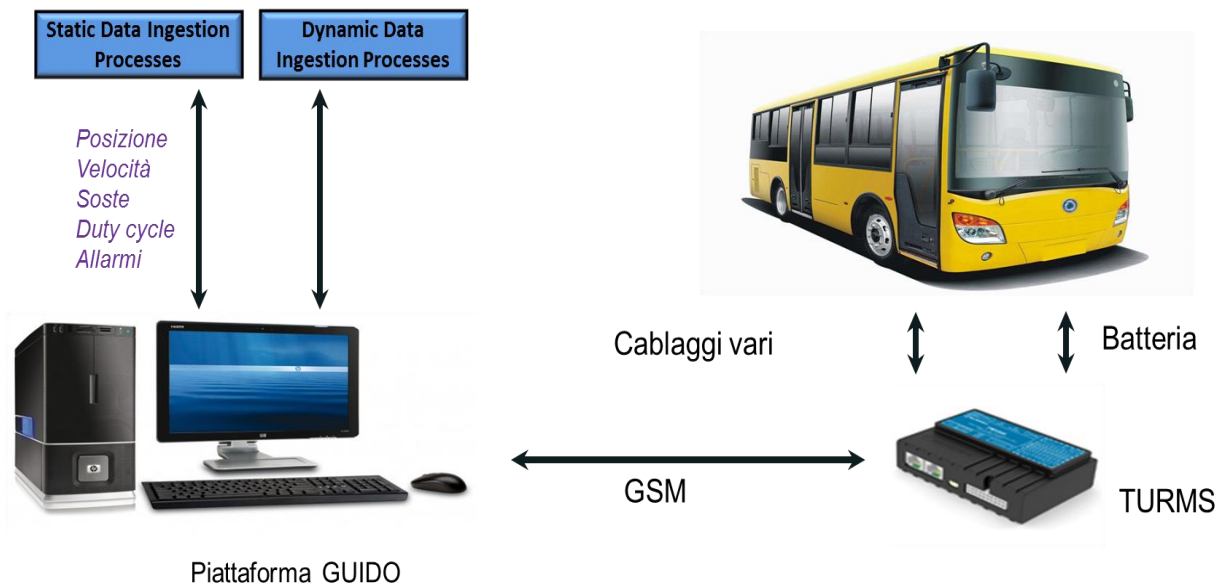


<h2 style="text-align: center;">Profilo Tool/Algoritmo</h2> <h3 style="text-align: center;">Apparato Car Bus (KIT 02)</h3>	
Nome responsabile	
Partner responsabile	ECM
Descrizione Tool/algoritmo	Apparato veicolare per auto e bus. È costituito da un Tool pre-esistente (T09) formato da un dispositivo da installare a bordo (TURMS) e da una centrale di raccolta dati via GSM (GUIDO). Nel contesto di Sii-Mobility si sviluppa una versione speciale del Data Harvester (MAPP16) e la si integra al SW di GUIDO per collegarla alla piattaforma Sii-Mobility e trasferire i dati veicolari.
Dati primari in ingresso	Dati veicolari contenuti in GUIDO e acquisiti dal Data Harvester
Dati prodotti in Uscita	Dati veicolari inviati dal Data Harvester alla piattaforma Sii-Mobility
Principi (metodo, basi teoriche, etc.)	Come Data Harvester ( vedi 5.5.8)
Casi di test (presenti/assenti)	assenti
Posizione casi di test	N/A
Principali problemi non risolti	nessuno
Principali requisiti pendenti	nessuno
Aspetti Tecnologici	
Stato (proposto/approvato)	approvato
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0 %
Eseguibile/libreria/web app	Eseguibile

Single thread/Multithread	Compatibile con ambiente SW GUIDO	
Linguaggio di sviluppo (java, Php, ETL)	Compatibile con ambiente SW GUIDO	
Piattaforme supportate	GUIDO	
Posizione del codice sorgente	GUIDO	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	Come Data Harvester ( vedi 5.5.8)	
Indirizzo/i accesso via web (se presenti)	Come Data Harvester ( vedi 5.5.8)	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Come Data Harvester ( vedi 5.5.8)		
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Formato dati GUIDO</b>	SW GUIDO (T09)	D3.1 – sezione 4.2.2
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Accesso a dati GUIDO</b>	SW GUIDO (T09)	D3.1 – sezione 4.2.2
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
Come Data Harvester ( vedi 5.5.8)		
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
N/A	N/A	N/A
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
Come Data Harvester ( vedi 5.5.8)		



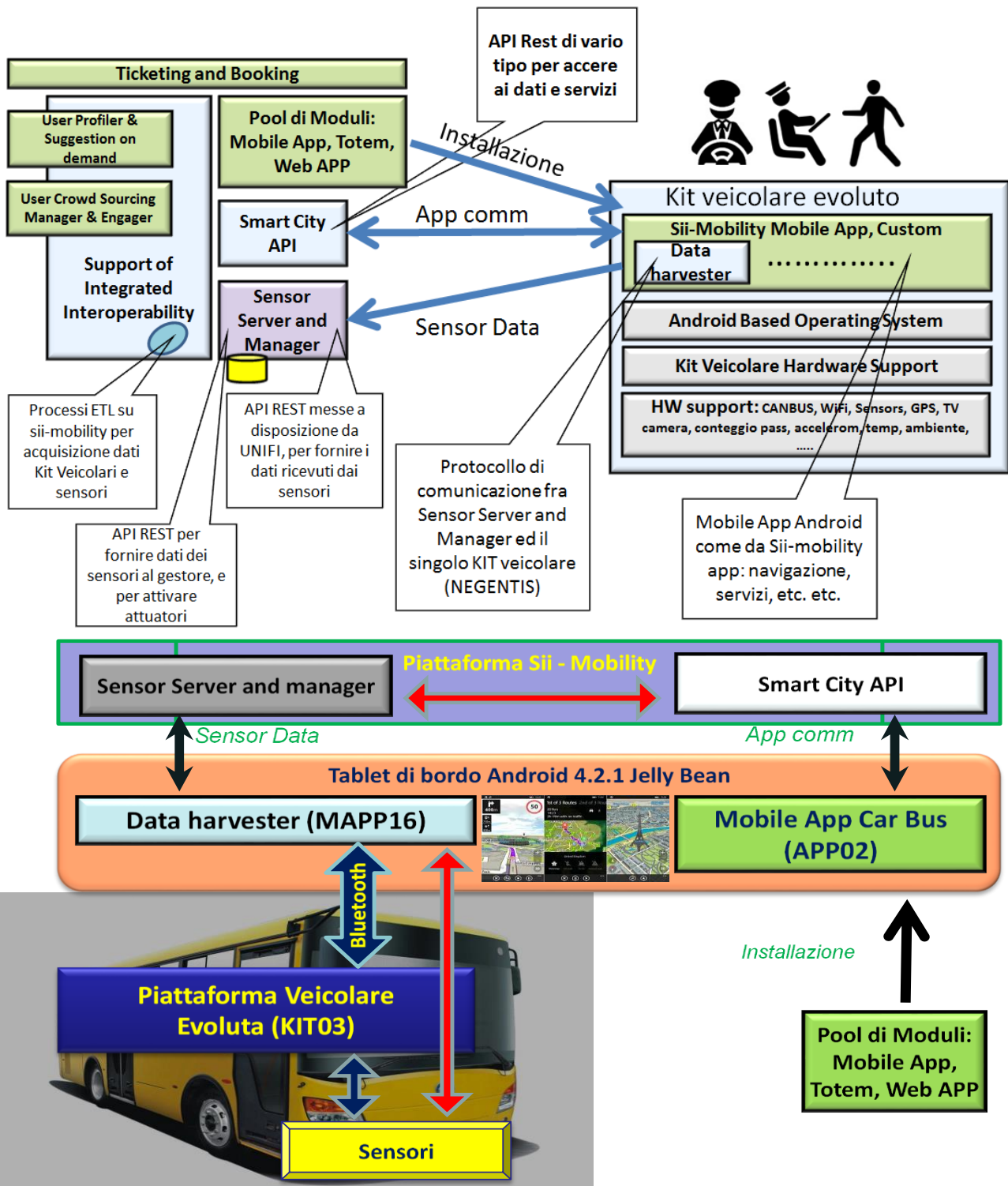
### 5.5.4 Piattaforma GUIDO (T09) (ECM)



<h2 style="text-align: center;">Profilo Tool/Algoritmo</h2> <h3 style="text-align: center;">Piattaforma GUIDO (T09)</h3>	
Nome responsabile	
Partner responsabile	ELFI
Descrizione Tool/algoritmo	Tool pre-esistente fornito in comodato d'uso a Sii-Mobility da ELFI. Composto da: Dispositivo TURMS da installare a bordo. Richiede l'alimentazione dal veicolo e fornisce dati di posizione, velocità, accelerazioni via GSM. Centrale operativa GUIDO che concentra i dati dai dispositivi veicolari, li elabora e li rende disponibili agli utenti/gestori. Localizzazione, velocità, soste, duty cycle e allarmi diagnostici.
Dati primari in ingresso	Dati da sensori di bordo connessi a TURMS. Posizione, velocità, accelerazioni. Sono trasmessi a GUIDO via GSM.
Dati prodotti in Uscita	Elaborazioni dei dati da TURMS visualizzati dalla centrale operativa. Localizzazione, velocità, soste, duty cycle e allarmi diagnostici
Principi (metodo, basi teoriche, etc.)	N/A
Casi di test (presenti/assenti)	N/A
Posizione casi di test	N/A
Principali problemi non risolti	N/A
Principali requisiti pendenti	N/A
<b>Aspetti Tecnologici</b>	
Stato (proposto/approvato)	N/A
Implementato/non implementato	N/A
Stato implementazione, percentuale	N/A
Eseguibile/libreria/web app	N/A
Single thread/Multithread	N/A

Linguaggio di sviluppo (java, Php, ETL)	N/A	
Piattaforme supportate	N/A	
Posizione del codice sorgente	N/A	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	N/A	
Indirizzo/i accesso via web (se presenti)	N/A	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
N/A	N/A	N/A
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
N/A	N/A	N/A
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
N/A	N/A	N/A
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
N/A	N/A	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
N/A	N/A	N/A
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
N/A	N/A	N/A

### 5.5.5 Piattaforma Veicolare Evoluta (KIT03) (ECM)

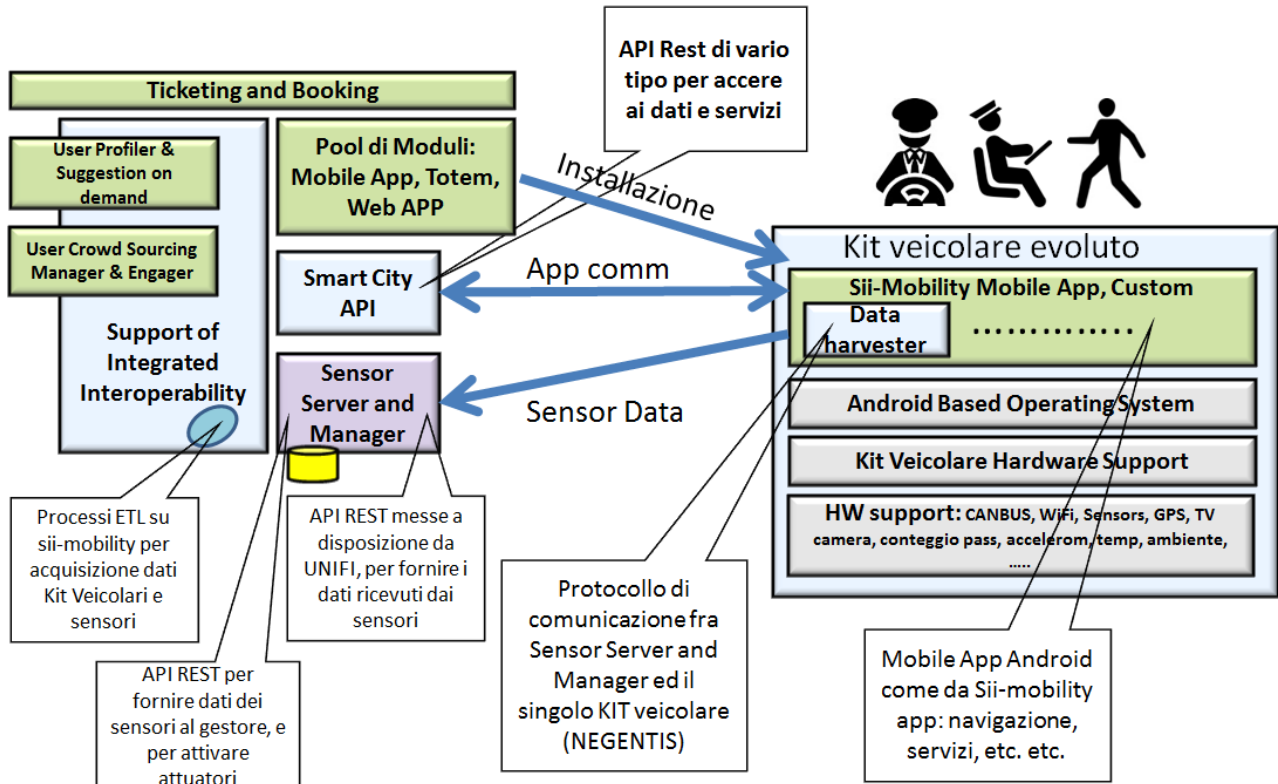


<h2 style="text-align: center;">Profilo Tool/Algoritmo</h2> <h3 style="text-align: center;">Piattaforma Veicolare Evoluta (KIT03)</h3>	
Nome responsabile	
Partner responsabile	ECM
Descrizione Tool/algoritmo	Piattaforma evoluta per auto e bus. Fornisce interfaccia HW/SW verso i sensori installabili sul veicolo. Comprende un Tablet di servizio Android adatto allo

	sviluppo di Applicazioni mobili pilota a cui fornisce via Bluetooth i dati dei sensori e i servizi per comandare / controllare i sensori	
Dati primari in ingresso	Valori di tensione da sensori passivi, messaggi da sensori attivi Protocollo base Bluetooth con Data Harvester (MAPP16) – messaggi di controllo Messaggi da Data Harvester (MAPP16) – comandi per sensori	
Dati prodotti in Uscita	Valori di tensione per sensori passivi, messaggi per sensori attivi Protocollo base Bluetooth con Data Harvester (MAPP16) – messaggi di controllo Messaggi a Data Harvester (MAPP16) – dati da sensori	
Principi (metodo, basi teoriche, etc.)	Architettura HW/SW open basata su tecnologia Arduino	
Casi di test (presenti/assenti)	assenti	
Posizione casi di test	N/A	
Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	approvato	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0 %	
Eseguibile/libreria/web app	Piattaforma HW/SW embedded	
Single thread/Multithread	N/A	
Linguaggio di sviluppo (java, Php, ETL)	Sketches Arduino	
Piattaforme supportate	N/A	
Posizione del codice sorgente	N/A	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	N/A	
Indirizzo/i accesso via web (se presenti)	N/A	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
N/A	N/A	N/A
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
N/A	N/A	N/A
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
Protocollo base Bluetooth	MAPP16	DE3.1 – sezione 5.2.4
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
N/A	N/A	
<b>Tipo interfaccia utente</b>	<b>Modello sviluppo,</b>	<b>Libreria usata per UI</b>

(web/applicazione)	linguaggio	
N/A	N/A	N/A
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
N/A	N/A	N/A

### 5.5.6 Mobile App Car Bus (APP02) (UNIFI)



Una o più applicazioni pilota da scaricare sul dispositivo mobile dell'Utente. Tipicamente viene utilizzata la stessa APP mobile della soluzione Sii-Mobility (per esempio Firenze Dove cosa ... Km4city) anche per l'uso in bike o sul kit evoluto ma con aggiunta un processo in background che permette di leggere i dati dai sensori specifici. In questo caso il data harvester legge da Bluetooth il device della bike in altri casi sempre con lo stesso approccio dal kit veicolare.

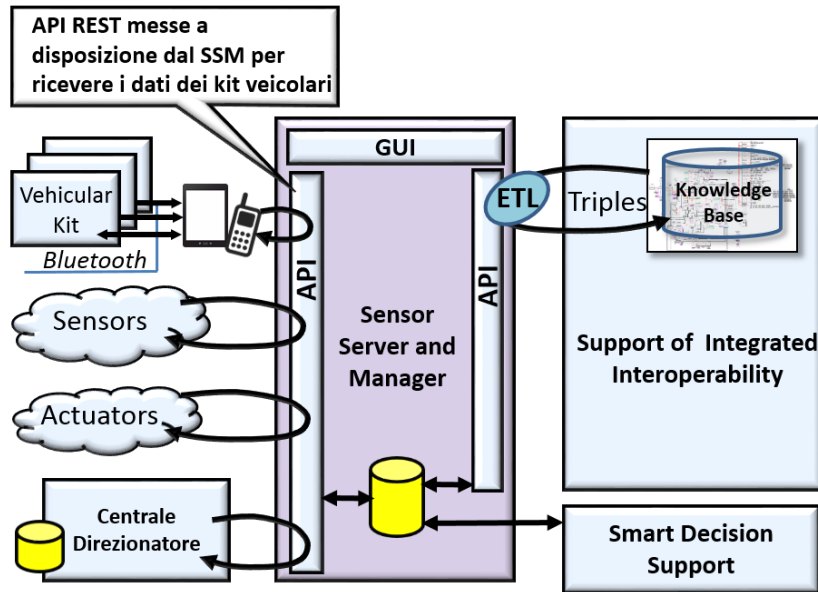
Sull'App devono essere implementi algoritmi e soluzioni di guida connessa, prenotazione, pagamento, invio a Sii-Mobility dei dati dei sensori veicolari.

In questo modulo si prevede di adattare l'APP di Sii-mobility anche a tali processi, in modo che all'installazione possa riconoscere la condizione e installare i driver specifici, oppure con una installazione ad hoc per i kit e per le bike.

Vi saranno delle viste e delle informazioni specifiche in navigazione per la guida connessa in auto come per esempio: Percorso da fare; Percorso fatto, posizione; Velocita', Km fatti, etc. statistiche se possibile; stato sensori e Can Bus; cartelli e stato delle strada; info dei flussi cittadini; info sui flussi persone in città; stato parcheggi; informazioni in anticipo su ZTL; etc. Vi saranno inoltre degli accorgimenti ed un vista ad hoc per la navigazione in auto, derivati dalla connessione con CANBUS, suggerimenti ed assistance adhoc.

La soluzione per movimento con mezzo privato deve offrire una navigazione focalizzata la mezzo.

### 5.5.7 Protocollo di comunicazione fra Sensor Server and Manager ed il singolo KIT veicolare (API05) (UNIFI)



Profilo Tool/Algoritmo	
Protocollo di comunicazione fra Sensor Server and Manager ed il singolo KIT veicolare (API05)	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	<p>1) Acquisizione di dati esponendo delle API REST sulla sinistra del disegno presentato in figura, per collezionare dati provenienti da:</p> <ul style="list-style-type: none"> <li>(i) data harvester su mobile e kit veicolare (dati che possono essere relativi a stato sensori mobili, sensori bike sensori can bus, sensori autobus, etc. Questi vanno memorizzati su database;</li> <li>(ii) sensori diretti, per esempio un embedded che chiama direttamente il SSM per salvare certi dati;</li> <li>(iii) attuatori (per esempio anche il direzionatore), questi fanno una chiamata rest per sapere se vi sono dei valori da aggiornare, delle azioni da fare, per esempio cambiare luce ad un semaforo, etc. SSM deve fornire API REST con accesso autenticato con una chiave o certificato in modo da sapere esattamente quale dispositivo sta chiamando.</li> </ul> <p>2) Fornisce un'interfaccia API rest anche per</p>

	<p>produrre dati verso lo storage generale. questo permette di usare il SSM come concentratore di dati, e per questo e' possibile replicarlo scalando ul numero di canali che inviato dati. Le API accedute da ETL sono semplici e permettono la lettura di quanto accumulato su database. In questo caso le API sono di sola lettura.</p> <p>3) API di scrittura da parte di algoritmi che producono decisioni, per esempio per decidere se il semaforo deve cambiare colore o il direzionatore cambiare senso, etc.</p> <p>4) Il database del SSM puo' essere tipicamente acceduto anche in modo diretto se questo SSM è locato in una intranet.</p> <p>5) SSM presenta un'interfaccia utente per navigare sulla tabella dei dati accumulati, e dei dati/comandi inviati come se fossero un LOG. Questa tabella deve poter essere visualizzata anche tramite filtri, ordinamenti, etc.</p> <p>Per la realizzazione si puo' vedere se vi sono componenti open source accessibili già pronti. Per esempio da Fi Ware.</p>
Dati primari in ingresso	Acquisizione di dati provenienti da Kit Car-Bus-Bike veicolari Sii-mobility tramite protocollo Bluetooth.
Dati prodotti in Uscita	Il modulo scrive i dati su Database HBase o altri NoSQL se necessario, inoltre fornisce delle API REST che consentono di esporre i dati dei kit JSON per la conseguente acquisizione verso la base di conoscenza Sii-Mobility, tramite un processo ETL (uno per ciascun tipo di sensore).
Principi (metodo, basi teoriche, etc.)	Realizzazione API REST (metodi http: POST, formato: JSON), realizzazione di specifiche per lo scambio dati.
Casi di test (presenti/assenti)	Assenti
Posizione casi di test	n/d
Principali problemi non risolti	Assenti
Principali requisiti pendenti	Nessuno
<b>Aspetti Tecnologici</b>	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0.0%
Eseguibile/libreria/web app	Realizzazione di: API REST, ETL
Single thread/Multithread	Multithread

Linguaggio di sviluppo (java, Php, ETL)	Php	
Piattaforme supportate	Non dipende dal tipo di piattaforma	
Posizione del codice sorgente	UNIFI, Laboratorio DISIT	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	Da stabilire	
Indirizzo/i accesso via web (se presenti)	Da stabilire	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Kit Car-Bus-Bike veicolari Sii-mobility	Protocollo BT	JSON
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON	Nessuno	-----
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
Protocollo HTTP	-----	Fra mobile app Data harvester e Sensor Server e Manager
Bluetooth	Data harvester	Fra Kit veicolare e Data Harvester
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
MySQL HBase	Database necessario per la memorizzazione di informazioni relative ai Kit Car-Bus-Bike veicolari.	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
nessuna		
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

API REST, usate dal Data Harvester, per l'inserimento dei dati dei kit veicolari. Si veda la descrizione in dettaglio al paragrafo 5.11.2.

### 5.5.8 Data Harvester dei Sensori Bluetooth dei Kit Veicolari sui mobile (MAPP16) (UNIFI DISIT)

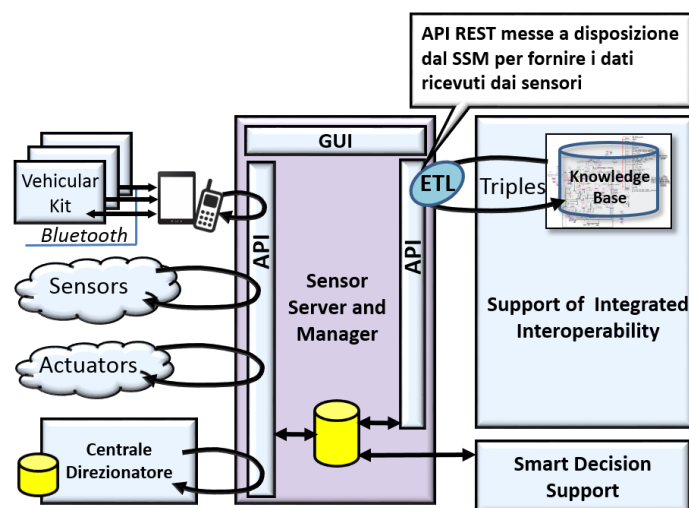
Modulo presente su mobile che acquisisce dati provenienti da kit veicolari tramite Bluetooth e gli scrive verso il Sensor Server and Manager tramite chiamate rest.



Effettua anche il salvataggio di tali dati in un parte condivisa in modo che possano essere utilizzati dall'applicazione mobile. Lo stesso modulo deve essere in grado di chiedere dati al Sensor Server manager per capire se vi sono dati per gli attuatori. In caso affermativo i dati vengono passati al kit attuatore direttamente tramite Bluetooth.

Il data Harvester riceve i dati dal Kit veicolare tramite Bluetooth invia dati al Sensor Server and Manager tramite chiamate REST che portano dati relativi ai sensori del kit veicolare, gli stessi dati sono resi disponibili per l'app stessa su mobile su una parte condivisa. Il Data Harvester ha inoltre il compito di poter ricevere comandi dalla APP e dal Sensor Server and Manager per azionare direzionamenti, segnali, etc. Il data Harvester lavora sempre in Pull verso il Sensors Server Manager: in sostanza i comandi vengono richiesti da mobile, per esempio ogni 30 secondi viene richiesto dal mobile se vi sono comandi per il Kit che potrebbe anche essere un direzionatore. La scrittura in Push da parte del SSM va evitata perché implicherebbe conoscere gli IP di tali dispositivi che sono ovviamente in reti mobili e pertanto non stabili.

### 5.5.9 Processo ETL Sensor Server and Manager e Sii-Mobility (PETL24) (UNIFI)



<b>Profilo Tool/Algoritmo</b>	
Protocollo di comunicazione fra Sensor Server and Manager e processo ETL su Sii-Mobility (PETL24)	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Acquisizione di dati provenienti da Sensor Server and Manager (SSM), riguardo a Kit Car-Bus-Bike Sii-mobility, sviluppati con ETL per portare i dati in triple dentro la Knowledge base in base al modello semantico. Il processo e' gestito da DISCES
Dati primari in ingresso	Accesso a Dati tramite chiamata REST
Dati prodotti in Uscita	Triple RDF relative ai dataset ingeriti
Principi (metodo, basi teoriche, etc.)	Gli ETL vengono messi in esecuzione da DISCES in modo periodico o aperiodico.
Casi di test (presenti/assenti)	Assenti
Posizione casi di test	n/d
Principali problemi non risolti	Assenti

Principali requisiti pendenti	Nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0.0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	ETL	
Piattaforme supportate	Unix, Windows	
Posizione del codice sorgente	svn	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Ingestion	n/a	REST API, CSV, KMZ, SHP
Quaity Improvement	n/a	n/a
Triplification	n/a	n/a
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
CSV		comma-separated values
KMZ/KML		Keyhole Markup Language
SHP		Shape format file
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTP		HyperText Transfer Protocol
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
MySQL	Database relazionale	
HBase	Database NoSql distribuito su filesystem Hadoop	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
n/a		
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
n/a		

API rest anche per produrre dati verso lo storage generale. questo permette di usare il SSM come concentratore di dati, e per questo è possibile replicarlo scalando il numero di canali che inviato

dati. Le API accedute da ETL sono semplici e permettono la lettura di quanto accumulato su database. In questo caso le API sono di sola lettura.

Si vedano le specifiche descritte al paragrafo 5.11.2, API usate dagli ETL per trasferire le informazioni dal DB del Sensor Service and Manager, nella Knowledge Base

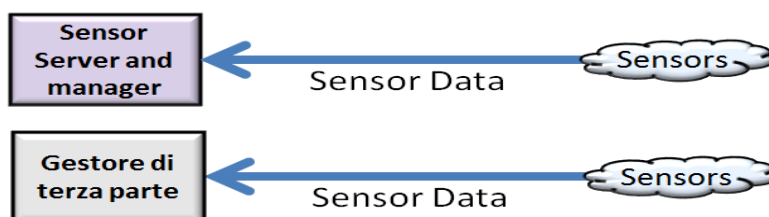
## 5.6 Specifica: Sensori innovativi

		ECM	Mizar	UNIFI	IN20	TIME	Negentis	EffKnow	liberologico	ataf	busitalia	cttnord	fiemme	argos	elfi	calamai	Midra	Proiect	GEOIN	OUJESTIT	SOJTEC	FWINGS
3.2.1	Studio, definizione e sviluppo di Sensori innovativi: Sensori per parcheggi				X				X													
3.2.2	Studio, definizione e sviluppo di Sensori innovativi: sensori per situazioni critiche, flussi, etc.		X															X				
3.2.3	Studio, definizione e sviluppo di Sensori innovativi: smart node			X														X				
3.2.4	Studio, definizione e sviluppo di Sensori innovativi: sensori video																	X				
3.2.5	Studio, definizione e sviluppo di Sensori innovativi: sensori inquinamento			X													X					

Principali Sottosistemi interessati dai requisiti da descrivere in questa Subsubsection, saranno i sottosistemi primari della specifica					
	Sotto Attività	Partner coord	Sottosistema	Tool/Level	Descrizione sommaria
3.2.1	Studio, definizione e sviluppo di Sensori innovativi: Sensori per parcheggi	LIBERLOGICO	Sensori	Parking Sensors and Parking Manager (SN01, T10)	<p>Rete di sensori WSN – Wireless sensor Network – in grado di monitorare in real time lo stato di occupazione degli stalli di sosta per parcheggi su strada (per es.: strisce blu, disabili, veicoli merci), . L’informazione acquisita permette di alimentare servizi di infoparking relativi al sistema di sosta monitorato, per:</p> <ul style="list-style-type: none"> <li>- il gestore della sosta</li> <li>- gli utenti finali/cittadini</li> <li>- il Sii-Mobility e sistemi di terze parti registrati.</li> </ul> <p>Il Gestore di questi sensori deve comunicare i dati con la centrale Sii-Mobility.</p>
3.2.2	Studio, definizione e sviluppo di Sensori innovativi: sensori per situazioni critiche, flussi, etc.	MIZAR	Monitor	Critical Situation Monitoring (MAPP01)	Sistema integrato di acquisizione a bordo veicolo (parte della OBU per il monitoraggio delle flotte di trasporto pubblico) in grado di segnalare situazioni con carattere critico come aumenti di velocità oppure frenate brusche. Il Gestore di questi sensori deve comunicare i dati con la centrale Sii-Mobility.
		MIZAR	Monitor	Traffic Flow Monitoring (SN03, T11)	Sistema di sensori e/o algoritmi che sulla base di dati di monitoraggio flussi fornisce informazioni sui flussi mezzi in città. Le informazioni di flusso possono essere derivate, attraverso appositi modelli/ algoritmi, sia in modo diretto che in modo indiretto da posizioni/ritardi TPL, da movimento mobile App o da informazioni provenienti da altri gestori. Queste informazioni possono essere utili anche per la gestione delle emergenze. Il Gestore di questi sensori deve comunicare i dati con la centrale Sii-Mobility.

		UNIFI	Monitor	People Flow Monitor (T12)	Sensori per la misura dei flussi di persone, per esempio tramite Wi-Fi, BT, o mobile App. Queste informazioni possono essere utili anche per la gestione delle emergenze. Per l'accesso e migliorare i servizi del trasporto pubblico locale, per suggerire al car e bike sharing posizioni strategiche dove lasciare le auto/bike, o dove mettere nuovi parcheggi/rastrelliere, per definire le vie di fuga da eventuali disastri, per migliorare i servizi in generale, per stimolare il commercio. Il Gestore di questi sensori deve comunicare i dati con la centrale Sii-Mobility.
3.2.3	Studio, definizione e sviluppo di Sensori innovativi: smart node	Project	Sensori	Smart Node e Manager (SN04, T13)	Il sistema di sensori, apparati ed algoritmi che sulla base di rilevamenti degli accadimenti circostanti fornisce le informazioni, agli apparati di controllo della zona, relative al traffico in entrata e uscita, autorizzato e non, allo stato di congestione e di manovra, a comportamenti anomali possibili fonti di pericolo. Questi nodi potranno essere installati anche su portali dedicati e fungeranno da collettore di informazioni varie sul traffico da comunicare alla centrale Sii-Mobility.
3.2.4	Studio, definizione e sviluppo di Sensori innovativi: sensori video	Project	Sensori	Sensori Ottici e Analisi Video (SN05)	I sensori video sono delle speciali telecamere da installare a bordo di mezzi pubblici (kit veicolari evoluti) od in tratti stradali da controllare, dedicati a molteplici impieghi legati alla gestione ed alla sicurezza a bordo e del traffico circostante. Gli algoritmi previsti consentiranno la classificazione degli eventi in modo da consentire una rapida reazione sia del mezzo che degli enti preposti alla gestione del traffico cittadino. Le informazioni in tempo reale verranno trasferite alla centrale Sii-Mobility.
3.2.5	Studio, definizione e sviluppo di Sensori innovativi: sensori inquinamento	MIDRA	Sensori	Sensori ambientali e Manager (SN06, T14)	Lo scopo è quello di sviluppare una infrastruttura di base, eventualmente espandibile, in grado di monitorare parametri ambientali che le sostanze inquinanti e/o nocive. specialmente quelle legate o correlate alla mobilità. L'infrastruttura sarà dotata di: <ul style="list-style-type: none"> <li>- Sensori per il particolato</li> <li>- Sensori di VOC (tecnologia PID)</li> <li>- Sensori di Gas</li> <li>- Sensori Microclimatici in grado di rilevare la presenza di, di altri sensori di gas come NO, SO2, CO e CO2, e di sensori microclimatici in grado di rilevare temperatura ed umidità dell'aria, velocità e direzione del vento, radiazione solare e precipitazioni atmosferiche.</li> </ul> Questi sensori devono poter essere usati da kit veicolari come da apparati indipendenti che comunicano con la centrale Sii-Mobility.

### 5.6.1 Parking Sensor (SN01) and Parking Manager (T10) (LIBEROLOGICO)



<b>Profilo Tool/Algoritmo</b>	
Parking Sensor and Parking Manager (T10) (e' un Gestore di Terza Parte)	
Nome responsabile	
Partner responsabile	LIBEROLOGICO
Descrizione Tool/algoritmo	<p>Rete di sensori WSN – Wireless Sensor Network – in grado di monitorare in real time lo stato di occupazione degli stalli di sosta per parcheggi su strada (per es.: strisce blu, disabili, veicoli merci), .</p> <p>L'informazione acquisita permette di alimentare servizi di infoparking relativi al sistema di sosta monitorato, per:</p> <ul style="list-style-type: none"> <li>- il gestore della sosta</li> <li>- gli utenti finali/cittadini</li> <li>- il Sii-Mobility e sistemi di terze parti registrati.</li> </ul> <p>Il Gestore di questi sensori deve comunicare i dati con la centrale Sii-Mobility.</p> <p>Acquisizione di Dati provenienti dai Sensor Server and Manager dei sensori innovativi installati, tramite modulo di ETL per portare i dati in triple dentro la Knowledge base in base al modello semantico. Il processo è gestito da DISCES.</p>
Dati primari in ingresso	Accesso ai Dati del gestore tramite chiamate API Rest
Dati prodotti in Uscita	Triple RDF relative ai dataset ingeriti
Principi (metodo, basi teoriche, etc.)	Gli ETL vengono messi in esecuzione da DISCES in modo periodico o aperiodico.
Casi di test (presenti/assenti)	Assenti
Posizione casi di test	n/d
Principali problemi non risolti	Assenti
Principali requisiti pendenti	Nessuno
Aspetti Tecnologici	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0%
Eseguibile/libreria/web app	Eseguibile
Single thread/Multithread	Multithread
Linguaggio di sviluppo (java, Php, ETL)	ETL
Piattaforme supportate	LINUX, Windows
Posizione del codice sorgente	git
Indirizzo/i web services (se presenti) con	n/a

indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)	n/a	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Ingestion	n/a	REST API, JSON
Triplification	n/a	n/a
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON		JavaScript Object Notation
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTPS		HyperText Transfer Protocol over Secure Socket Layer
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
MySql	Database relazionale	
HBase	Database NoSql distribuito su filesystem Hadoop	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
Web	PHP, CSS, HTML5, Java Script	n/a
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
WordPress Plugin Application Program Interface (API)	n/a	GPLv2

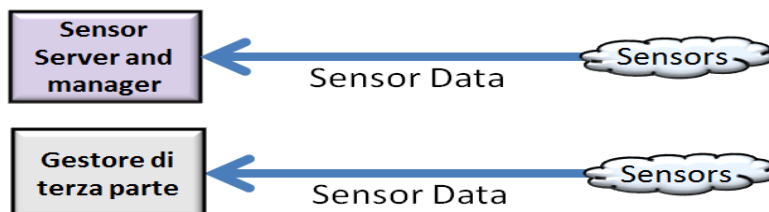
<b>Profilo Tool/Algoritmo</b>	
<b>READ Parking Status via Manager (SN01)</b>	
Nome responsabile	
Partner responsabile	LIBEROLOGICO
Descrizione Tool/algoritmo	<p>Rete di sensori WSN – Wireless sensor Network – in grado di monitorare in real time lo stato di occupazione degli stalli di sosta per parcheggi su strada (per es.: strisce blu, disabili, veicoli merci), . L’informazione acquisita permette di alimentare servizi di infoparking relativi al sistema di sosta monitorato, per:</p> <ul style="list-style-type: none"> <li>- il gestore della sosta</li> </ul>

	<ul style="list-style-type: none"> <li>- gli utenti finali/cittadini</li> <li>- il Sii-Mobility e sistemi di terze parti registrati.</li> </ul> Il Gestore di questi sensori deve comunicare i dati con la centrale Sii-Mobility.	
Dati primari in ingresso	Accesso a chiamate REST	
Dati prodotti in Uscita	Triple RDF relative ai dataset ingeriti	
Principi (metodo, basi teoriche, etc.)	Gli ETL vengono messi in esecuzione da DISCES in modo periodico o aperiodico.	
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	n/d	
Principali problemi non risolti	Assenti	
Principali requisiti pendenti	Nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0.0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	ETL	
Piattaforme supportate	Unix, Windows	
Posizione del codice sorgente	svn	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Ingestion	n/a	REST API, CSV, KMZ, SHP
Quaity Improvement	n/a	n/a
Triplification	n/a	n/a
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON		JavaScript Object Notation
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTPS		HyperText Transfer Protocol over Secure Socket Layer
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
MySQL	Database relazionale	
HBase	Database NoSql distribuito su filesystem Hadoop	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>



n/a		
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
n/a		

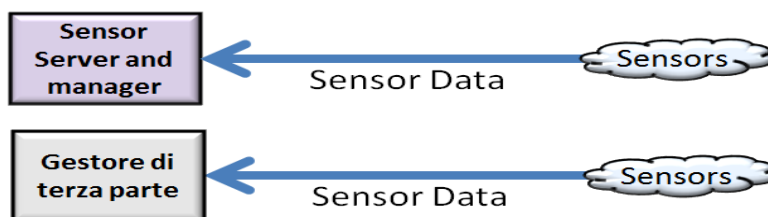
### 5.6.2 Critical Situation Monitoring (MAPP01) (MIZAR)



<h2 style="text-align: center;">Profilo Tool/Algoritmo</h2> <h3 style="text-align: center;">Sensori per Situazioni critiche (MAPP01)</h3>	
Nome responsabile	
Partner responsabile	MIZAR
Descrizione Tool/algoritmo	Sistema integrato di acquisizione a bordo veicolo (parte della OBU per il monitoraggio delle flotte di trasporto pubblico) in grado di segnalare situazioni con carattere critico come aumenti di velocità oppure frenate brusche. Il Gestore di questi sensori deve comunicare i dati con la centrale Sii-Mobility. Modulo che sfrutta i dati di accelerometro del mobile per identificare le situazioni critiche.
Dati primari in ingresso	Accelerazione
Dati prodotti in Uscita	Alert per eventi critici
Principi (metodo, basi teoriche, etc.)	In caso di brusca frenata/accelerazione il sistema genera un alert per segnalare l'evento
Casi di test (presenti/assenti)	Nessuno
Posizione casi di test	n.a.
Principali problemi non risolti	n.a.
Principali requisiti pendenti	n.a.
Aspetti Tecnologici	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non Implementato
Stato implementazione, percentuale	0%
Eseguibile/libreria/web app	
Single thread/Multithread	Multithread
Linguaggio di sviluppo (java, Php, ETL)	Java
Piattaforme supportate	Android
Posizione del codice sorgente	Repository di progetto
Indirizzo/i web services (se presenti) con	n.a.

indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)	n.a.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n.a.	n.a.	n.a.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
n.a.	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
n.a.	n.a.	n.a.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
n.a.	n.a.	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
n.a.	n.a.	n.a.
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
n.a.	n.a.	n.a.

### 5.6.3 Traffic Flow Monitoring (SN03a, SN03b, T11a, T11b) (MIZAR)



#### 5.6.3.1 Traffic and People Flow Monitoring Sensors (SN03a) (MIZAR)

<b>Profilo Tool/Algoritmo</b>	
Traffic and People Flow Monitoring (SN03a)	
Nome responsabile	
Partner responsabile	MIZAR
Descrizione Tool/algoritmo	Sistema di sensori e/o algoritmi che sulla base di dati di monitoraggio flussi fornisce informazioni sui flussi

	<p>leggeri in città. Le informazioni di flusso possono essere derivate, oltre che attraverso appositi modelli/ algoritmi, sia in modo diretto che in modo indiretto da posizioni/ritardi TPL, da movimento mobile App o da informazioni provenienti da altri gestori anche da misure dirette provenienti da telecamere.</p> <p>Il Gestore di questi sensori deve comunicare i dati con la centrale Sii-Mobility tramite API.</p> <p>Nel supervisore saranno accumulati tali dati, e saranno resi disponibili i dati puntuali in real time del numero di persone ai semafori o che altro, ogni x secondi. Da questi stessi dati MIZAR produce algoritmi e ne da evidenza all'esterno tramite api verso sii-mobility.</p>	
Dati primari in ingresso	Scena ambientale	
Dati prodotti in Uscita	Conteggio di persone ai semafori	
Principi (metodo, basi teoriche, etc.)	Il sistema analizza il flusso video della telecamera e restituisce il numero di persone presenti nella scena, per esempio ai semafori	
Casi di test (presenti/assenti)	Nessuno	
Posizione casi di test	n.a.	
Principali problemi non risolti	n.a.	
Principali requisiti pendenti	n.a.	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	libreria	
Single thread/Multithread	multithread	
Linguaggio di sviluppo (java, Php, ETL)	C	
Piattaforme supportate	Embedded	
Posizione del codice sorgente	Proprietario	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.a.	
Indirizzo/i accesso via web (se presenti)	n.a.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n.a.	n.a.	n.a.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
n.a.	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>

n.a.	n.a.	n.a.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)		<b>Descrizione</b>
n.a.	n.a.	
<b>Tipo interfaccia utente</b> (web/applicazione)		<b>Modello sviluppo, linguaggio</b>
n.a.	n.a.	n.a.
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)		<b>Nome e versione usata</b>
n.a.	n.a.	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
n.a.	n.a.	n.a.

### 5.6.3.2 Traffic Flow Monitoring Sensor data (SN03b) (UNIFI)

<b>Profilo Tool/Algoritmo</b>	
Traffic Flow Monitoring, sensor data (SN03b)	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Sistema che acquisisce da sensori di flusso traffico e ne fa delle elaborazioni. Le informazioni di flusso possono essere derivate, oltre che attraverso appositi modelli/algoritmi, sia in modo diretto che in modo indiretto da posizioni/ritardi TPL, da movimento mobile App o da informazioni provenienti da altri gestori anche da misure dirette provenienti da telecamere. Queste informazioni possono essere utili anche per la gestione delle emergenze. Il Gestore di questi sensori deve comunicare i dati con la centrale Sii-Mobility.
Dati primari in ingresso	Scena ambientale
Dati prodotti in Uscita	Conteggio di flusso
Principi (metodo, basi teoriche, etc.)	Numero di veicoli, velocità, etc. Si derivano i flussi anche in altri punti della città per propagazione statistica o modelli a code.
Casi di test (presenti/assenti)	Nessuno
Posizione casi di test	n.a.
Principali problemi non risolti	n.a.
Principali requisiti pendenti	n.a.
<b>Aspetti Tecnologici</b>	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0%
Eseguibile/libreria/web app	Web app

Single thread/Multithread	multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java	
Piattaforme supportate	linux	
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.a.	
Indirizzo/i accesso via web (se presenti)	n.a.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n.a.	n.a.	n.a.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
n.a.	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
n.a.	n.a.	n.a.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
n.a.	n.a.	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
n.a.	n.a.	n.a.
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
n.a.	n.a.	n.a.

### 5.6.3.3 Traffic and People Flow Monitoring Tool (T11a) (MIZAR)

<b>Profilo Tool/Algoritmo</b>	
Traffic Flow Monitoring (T11)	
Nome responsabile	
Partner responsabile	MIZAR
Descrizione Tool/algoritmo	Sistema di sensori e/o algoritmi che sulla base di dati di monitoraggio flussi fornisce informazioni sui flussi leggeri in città. Le informazioni di flusso possono essere derivate, oltre che attraverso appositi modelli/algoritmi, sia in modo diretto che in modo indiretto da

	posizioni/ritardi TPL, da movimento mobile App o da informazioni provenienti da altri gestori anche da misure dirette provenienti da telecamere. Nel supervisore saranno accumulati tali dati, e saranno resi disponibili i dati puntuali in real time del numero di persone ai semafori o che altro, ogni x secondi. Da questi stessi dati MIZAR produce algoritmi e ne da evidenza all'esterno tramite api verso sii-mobility.	
Dati primari in ingresso	Conteggio di persone ai semafori	
Dati prodotti in Uscita	Level of Service	
Principi (metodo, basi teoriche, etc.)	HCM2010	
Casi di test (presenti/assenti)	Nessuno	
Posizione casi di test	n.a.	
Principali problemi non risolti	n.a.	
Principali requisiti pendenti	n.a.	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	C#	
Piattaforme supportate	Windows	
Posizione del codice sorgente	Repository di progetto	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.a.	
Indirizzo/i accesso via web (se presenti)	n.a.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n.a.	n.a.	n.a.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
n.a.	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
n.a.	n.a.	n.a.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
n.a.	n.a.	
<b>Tipo interfaccia utente</b>	<b>Modello sviluppo,</b>	<b>Libreria usata per UI</b>

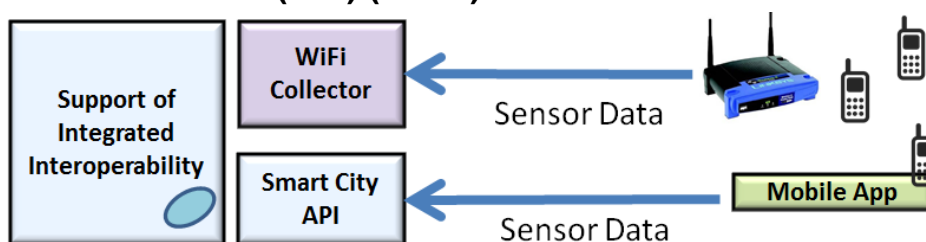
(web/applicazione)	linguaggio	
n.a.	n.a.	n.a.
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
n.a.	n.a.	n.a.

#### 5.6.3.4 Traffic Flow Monitoring Tool (T11b) (UNIFI)

<b>Profilo Tool/Algoritmo</b>	
Traffic Flow Monitoring Tool (T11b) (controparte di Tool di sintesi e monitoraggio rispetto alla parte sensori specificata nella tabella precedente)	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Sistema di sensori e/o algoritmi che sulla base di dati di monitoraggio flussi fornisce informazioni sui flussi mezzi in città. Le informazioni di flusso possono essere derivate, attraverso appositi modelli/ algoritmi, sia in modo diretto che in modo indiretto da posizioni/ritardi TPL, da movimento mobile App o da informazioni provenienti da altri gestori. Queste informazioni possono essere utili anche per la gestione delle emergenze. Il Gestore di questi sensori deve comunicare i dati con la centrale Sii-Mobility.
Dati primari in ingresso	Conteggio di mezzi, velocità, etc.
Dati prodotti in Uscita	Level of Service
Principi (metodo, basi teoriche, etc.)	
Casi di test (presenti/assenti)	Nessuno
Posizione casi di test	n.a.
Principali problemi non risolti	n.a.
Principali requisiti pendenti	n.a.
Aspetti Tecnologici	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0%
Eseguibile/libreria/web app	Web app
Single thread/Multithread	Multithread
Linguaggio di sviluppo (java, Php, ETL)	Java
Piattaforme supportate	Linux
Posizione del codice sorgente	Repository di progetto
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.a.
Indirizzo/i accesso via web (se presenti)	n.a.

<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
n.a.	n.a.	n.a.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
n.a.	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
n.a.	n.a.	n.a.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	Descrizione	
n.a.	n.a.	
<b>Tipo interfaccia utente</b> (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
n.a.	n.a.	n.a.
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
n.a.	n.a.	n.a.

### 5.6.4 People Flow Monitor (T12) (UNIFI)



## Profilo Tool/Algoritmo

Soluzione per People Flow Monitor

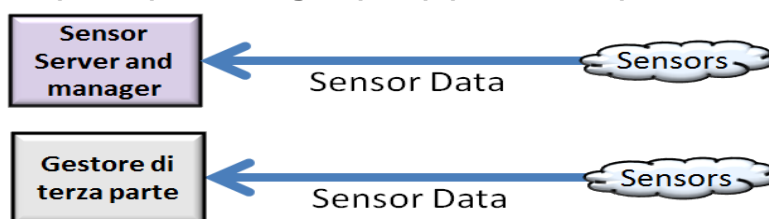
Nome responsabile	
Partner responsabile	DISIT-UNIFI
Descrizione Tool/algoritmo	Sensori per la misura dei flussi di persone, per esempio tramite Wi-Fi, BT, o mobile App. Queste informazioni possono essere utili anche per la gestione delle emergenze. Per l'accesso e migliorare i servizi del trasporto pubblico locale, per suggerire al car e bike sharing posizioni



	strategiche dove lasciare le auto/bike, o dove mettere nuovi parcheggi/rastrelliere, per definire le vie di fuga da eventuali disastri, per migliorare i servizi in generale, per stimolare il commercio. Il Gestore di questi sensori deve comunicare i dati con la centrale Sii-Mobility.	
Dati primari in ingresso	accessi agli access point (streaming)	
Dati prodotti in Uscita	Scrittura su file dei dati in ingresso	
Principi (metodo, basi teoriche, etc.)	L'applicazione è in ascolto su una porta di comunicazione TCP e riceve un flusso di dati	
Casi di test (presenti/assenti)	assenti	
Posizione casi di test	.	
Principali problemi non risolti	.	
Principali requisiti pendenti	.	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	eseguibile	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java	
Piattaforme supportate	Linux, Windows, Mac	
Posizione del codice sorgente	.	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	.	
Indirizzo/i accesso via web (se presenti)	.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
TCP		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo,</b>	<b>Libreria usata per UI</b>

	linguaggio	
applicazione	Java	
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

### 5.6.5 Smart Node (SN04) e Manager (T13) (PROJECT)



Profilo Tool/Algoritmo	
Smart Node (SN04)	
Nome responsabile	
Partner responsabile	PROJECT
Descrizione Tool/algoritmo	Il sistema di sensori, apparati ed algoritmi che sulla base di rilevamenti degli accadimenti circostanti fornisce le informazioni, agli apparati di controllo della zona, relative al traffico in entrata e uscita, autorizzato e non, allo stato di congestione e di manovra, a comportamenti anomali possibili fonti di pericolo. Questi nodi potranno essere installati anche su portali dedicati e fungeranno da collettore di informazioni varie sul traffico da comunicare alla centrale Sii-Mobility.
Dati primari in ingresso	Passaggio dei veicoli
Dati prodotti in Uscita	Classificazione dei veicoli transitati
Principi (metodo, basi teoriche, etc.)	Architettura HW/SW basata su tecnologia termica
Casi di test (presenti/assenti)	Assenti
Posizione casi di test	NA
Principali problemi non risolti	Nessuno
Principali requisiti pendenti	Nessuno
Aspetti Tecnologici	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Parzialmente implementato
Stato implementazione, percentuale	40% circa
Eseguibile/libreria/web app	N.A.
Single thread/Multithread	N.A.
Linguaggio di sviluppo (java, Php, ETL)	N.A.
Piattaforme supportate	N.A.
Posizione del codice sorgente	N.A.

Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	No	
Indirizzo/i accesso via web (se presenti)	No	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
N.A.	N.A.	N.A.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
N.A.	N.A.	N.A.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
N.A.	N.A.	N.A.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
N.A.	N.A.	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
N.A.	N.A.	N.A.
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
N.A.	N.A.	N.A.

## Profilo Tool/Algoritmo

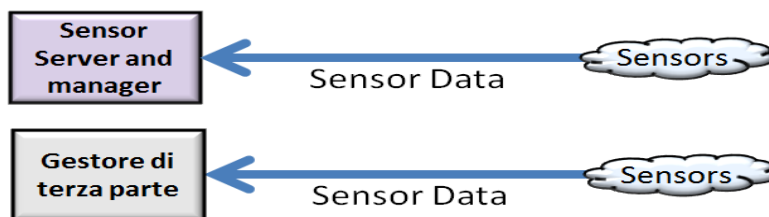
Smart Node Manager (T13)

<b>Nome responsabile</b>	
<b>Partner responsabile</b>	PROJECT
<b>Descrizione Tool/algoritmo</b>	Il sistema di sensori, apparati ed algoritmi che sulla base di rilevamenti degli accadimenti circostanti fornisce le informazioni, agli apparati di controllo della zona, relative al traffico in entrata e uscita, autorizzato e non, allo stato di congestione e di manovra, a comportamenti anomali possibili fonti di pericolo. Questi nodi potranno essere installati anche su portali dedicati e fungeranno da collettore di informazioni varie sul traffico da comunicare alla centrale Sii-Mobility.
<b>Dati primari in ingresso</b>	Classificazione veicoli transitati

Dati prodotti in Uscita	Allarmi eventi anomali	
Principi (metodo, basi teoriche, etc.)	In caso di rilevamento situazione anomala, il sistema genera un allarme	
Casi di test (presenti/assenti)	N.A.	
Posizione casi di test	N.A.	
Principali problemi non risolti	N.A.	
Principali requisiti pendenti	N.A.	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	PROPOSTO	
Implementato/non implementato	IMPLEMENTATO	
Stato implementazione, percentuale	20%	
Eseguibile/libreria/web app	ESEGUIBILE E LIBRERIE PROPRIETARIE	
Single thread/Multithread	MULTITHREAD	
Linguaggio di sviluppo (java, Php, ETL)	C#	
Piattaforme supportate	WINDOWS	
Posizione del codice sorgente	REPOSITORY DI PROGETTO	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	N.A.	
Indirizzo/i accesso via web (se presenti)	N.A.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
N.A.	N.A.	N.A.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
N.A.	N.A.	N.A.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
N.A.	N.A.	N.A.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
N.A.	N.A.	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
N.A.	N.A.	N.A.
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
N.A.	N.A.	N.A.

--	--	--

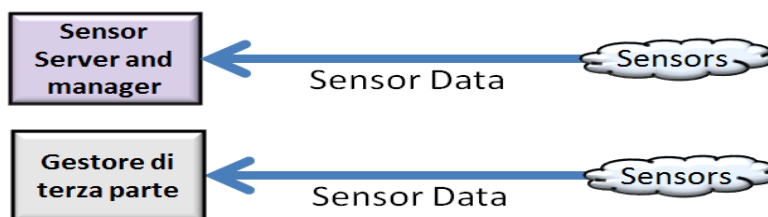
### 5.6.6 Sensori Ottici e Analisi Video (SN05) (PROJECT)



Profilo Tool/Algoritmo	
Sensori Ottici e Analisi Video	
Nome responsabile	
Partner responsabile	PROJECT
Descrizione Tool/algoritmo	I sensori video sono delle speciali telecamere da installare a bordo di mezzi pubblici (kit veicolari evoluti) od in tratti stradali da controllare, dedicati a molteplici impieghi legati alla gestione ed alla sicurezza a bordo e del traffico circostante. Gli algoritmi previsti consentiranno la classificazione degli eventi in modo da consentire una rapida reazione sia del mezzo che degli enti preposti alla gestione del traffico cittadino. Le informazioni in tempo reale verranno trasferite alla centrale Sii-Mobility.
Dati primari in ingresso	Passaggio veicoli e persone
Dati prodotti in Uscita	Classificazione veicoli, conteggio persone, rilevamento anomalie
Principi (metodo, basi teoriche, etc.)	Architettura HW/SW basata su tecnologia termica
Casi di test (presenti/assenti)	N.A.
Posizione casi di test	N.A.
Principali problemi non risolti	N.A.
Principali requisiti pendenti	N.A.
Aspetti Tecnologici	
Stato (proposto/approvato)	PROPOSTO
Implementato/non implementato	IMPLEMENTATO
Stato implementazione, percentuale	10%
Eseguibile/libreria/web app	N.A.
Single thread/Multithread	N.A.
Linguaggio di sviluppo (java, Php, ETL)	N.A.
Piattaforme supportate	N.A.
Posizione del codice sorgente	N.A.
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	N.A.
Indirizzo/i accesso via web (se presenti)	N.A.
Nomi tool/moduli usati	Interfacce API usate   Modello di comunicazione e

(aggiungere una riga per ogni tool/modulo usato)		formato
N.A.	N.A.	N.A.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
N.A.	N.A.	N.A.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
N.A.	N.A.	N.A.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	Descrizione	
N.A.	N.A.	
<b>Tipo interfaccia utente</b> (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
N.A.	N.A.	N.A.
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
N.A.	N.A.	N.A.

### 5.6.7 Sensori ambientali (SN06) e Manager (T14) (MIDRA)



<b>Profilo Tool/Algoritmo</b>	
Sensori ambientali (SN06)	
Nome responsabile	
Partner responsabile	MIDRA
Descrizione Tool/algoritmo	Unità dotata di connettività GPRS su protocollo TCP/IP (finalizzata alla connessione del sistema verso internet) dotata di: <ul style="list-style-type: none"> <li>• Sensori per il particolato</li> </ul>

	<ul style="list-style-type: none"> <li>• Sensori di VOC (tecnologia PID)</li> <li>• Sensori di Gas</li> <li>• Sensori Microclimatici</li> </ul> <p>in grado di rilevare la presenza di, di altri sensori di gas come NO, SO<sub>2</sub>, CO e CO<sub>2</sub>, e di sensori microclimatici in grado di rilevare temperatura ed umidità dell'aria, velocità e direzione del vento, radiazione solare e precipitazioni atmosferiche.</p> <p>I dati vengono trasmessi utilizzando un protocollo proprietario possono essere resi disponibili o direttamente tramite opportuna interfaccia oppure attraverso un server da cui vengono resi disponibili al sistema SII Mobility attraverso l'apertura di un socket e lo sviluppo di un interfaccia (in particolare con chiamata REST su HTTP).</p> <p>Il sistema si compone di:</p> <ul style="list-style-type: none"> <li>- Blocco per la trasmissione dati basato su tecnologia GPRS EDGE</li> <li>- Blocco per la gestione dell'Alimentazione</li> <li>- Blocco per l'Acquisizione dei dati da sensori</li> </ul> <p>Il sistema risulta alimentato a pannelli fotovoltaici</p>
Dati primari in ingresso	Nessuno
Dati prodotti in Uscita	<p>Misure dei parametri ambientali e dei principali inquinanti</p> <p>Sensori per il particolato</p> <ol style="list-style-type: none"> <li>1. PM 10</li> <li>2. PM 2,5</li> </ol> <p>Sensori di VOC (tecnologia PID)</p> <ol style="list-style-type: none"> <li>1. Benzene</li> <li>2. Xilene</li> <li>3. Tuloene</li> </ol> <p>Sensori di Gas</p> <ol style="list-style-type: none"> <li>1. NO</li> <li>2. SO<sub>2</sub></li> <li>3. CO</li> <li>4. CO<sub>2</sub></li> <li>5. O<sub>3</sub></li> </ol> <p>Sensori Microclimatici</p> <ol style="list-style-type: none"> <li>1. temperatura ed umidità dell'aria,</li> <li>2. velocità e direzione del vento</li> <li>3. radiazione solare e</li> <li>4. precipitazioni atmosferiche.</li> </ol> <p>Tempo di campionamento 5 minuti</p>
Principi (metodo, basi teoriche, etc.)	Sistema di monitoraggio espandibile
Casi di test (presenti/assenti)	Assenti
Posizione casi di test	Non Applicabile
Principali problemi non risolti	Non Applicabile
Principali requisiti pendenti	Interfaccia di scambio con il sistema SII Mobility
Aspetti Tecnologici	

Stato (proposto/approvato)	Non Applicabile	
Implementato/non implementato	In fase di implementazione	
Stato implementazione, percentuale	10%	
Eseguibile/libreria/web app	Non Applicabile	
Single thread/Multithread	Non Applicabile	
Linguaggio di sviluppo (java, Php, ETL)	Non Applicabile	
Piattaforme supportate	Non Applicabile	
Posizione del codice sorgente	Non Applicabile	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	Non Applicabile	
Indirizzo/i accesso via web (se presenti)	Non Applicabile	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

## Profilo Tool/Algoritmo



Sensori ambientali Manager (T14)	
Nome responsabile	
Partner responsabile	MIDRA
Descrizione Tool/algorithmo	<p>Unità dotata di connettività GPRS su protocollo TCP/IP (finalizzata alla connessione del sistema verso internet) dotata di:</p> <ul style="list-style-type: none"> <li>• Sensori per il particolato</li> <li>• Sensori di VOC (tecnologia PID)</li> <li>• Sensori di Gas</li> <li>• Sensori Microclimatici</li> </ul> <p>in grado di rilevare la presenza di, di altri sensori di gas come NO, SO<sub>2</sub>, CO e CO<sub>2</sub>, e di sensori microclimatici in grado di rilevare temperatura ed umidità dell'aria, velocità e direzione del vento, radiazione solare e precipitazioni atmosferiche.</p> <p>I dati vengono trasmessi utilizzando un protocollo proprietario possono essere resi disponibili o direttamente tramite opportuna interfaccia oppure attraverso un server da cui vengono resi disponibili al sistema SII Mobility attraverso l'apertura di un socket e lo sviluppo di un interfaccia.</p> <p>Il sistema si compone di:</p> <ul style="list-style-type: none"> <li>- Blocco per la trasmissione dati basato su tecnologia GPRS EDGE</li> <li>- Blocco per la gestione dell'Alimentazione</li> <li>- Blocco per l'Acquisizione dei dati da sensori</li> </ul> <p>Il sistema risulta alimentato a pannelli fotovoltaici</p>
Dati primari in ingresso	Nessuno
Dati prodotti in Uscita	<p>Misure dei parametri ambientali e dei principali inquinanti</p> <p>Sensori per il particolato</p> <ol style="list-style-type: none"> <li>3. PM 10</li> <li>4. PM 2,5</li> </ol> <p>Sensori di VOC (tecnologia PID)</p> <ol style="list-style-type: none"> <li>4. Benzene</li> <li>5. Xilene</li> <li>6. Tuloene</li> </ol> <p>Sensori di Gas</p> <ol style="list-style-type: none"> <li>6. NO</li> <li>7. SO<sub>2</sub></li> <li>8. CO</li> <li>9. CO<sub>2</sub></li> <li>10.O<sub>3</sub></li> </ol> <p>Sensori Microclimatici</p> <ol style="list-style-type: none"> <li>5. temperatura ed umidità dell'aria,</li> <li>6. velocità e direzione del vento</li> <li>7. radiazione solare e</li> <li>8. precipitazioni atmosferiche.</li> </ol> <p>Tempo di campionamento 5 minuti</p>
Principi (metodo, basi teoriche, etc.)	Sistema di monitoraggio espandibile
Casi di test (presenti/assenti)	Assenti
Posizione casi di test	Non Applicabile
Principali problemi non risolti	Non Applicabile

Principali requisiti pendenti	Interfaccia di scambio con il sistema SII Mobility	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Non Applicabile	
Implementato/non implementato	In fase di implementazione	
Stato implementazione, percentuale	10%	
Eseguibile/libreria/web app	Non Applicabile	
Single thread/Multithread	Non Applicabile	
Linguaggio di sviluppo (java, Php, ETL)	Non Applicabile	
Piattaforme supportate	Non Applicabile	
Posizione del codice sorgente	Non Applicabile	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	Non Applicabile	
Indirizzo/i accesso via web (se presenti)	Non Applicabile	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

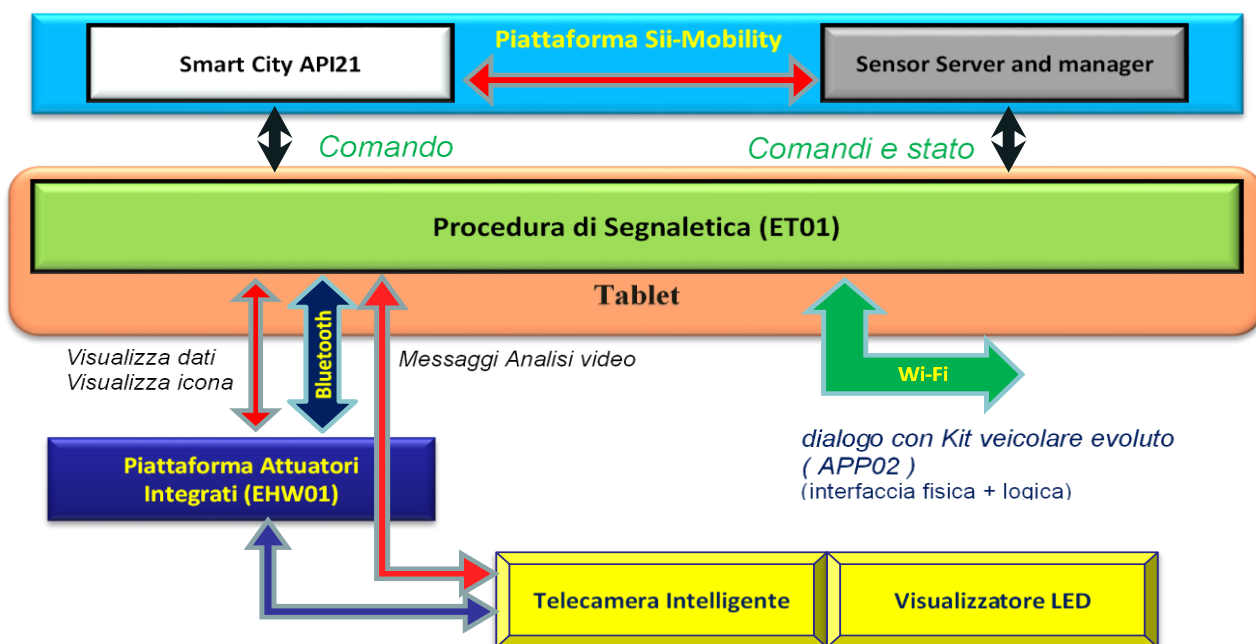
## 5.7 Specifica: Attuatori integrati per controllo accessi, direzione e velocità

		FCM	Mizar	UNIFI	IN20	TIME	Neentis	EffKnow	liberologico	ataf	bustalia	cttnord	tiemme	areos	elfi	calamai	Midra	Project	GEOIN	OJESTIT	SOFTEC	FWINGS
3.3.1	Studio, definizione e sviluppo di Attuatori integrati per controllo accessi, direzione e velocità: segnaletica remota	X		X										X	X	X						
3.3.2	Studio, definizione e sviluppo di Attuatori integrati per controllo accessi, direzione e velocità: video decisori																	X				
3.3.3	Studio, definizione e sviluppo di Attuatori integrati per controllo accessi, direzione e velocità: attuatori	X		X										X	X	X						
3.3.4	Studio, definizione e sviluppo di Attuatori integrati per controllo accessi, direzione e velocità: comunicazione						X															

Principali Sottosistemi interessati dai requisiti da descrivere in questa Subsubsection, saranno i sottosistemi primari della specifica					
	Sotto Attività	Partner coord	Sotto sistema	Tool/Level	Descrizione sommaria
3.3.1	Studio, definizione e sviluppo di Attuatori integrati per controllo accessi, direzione e velocità: segnaletica remota	ECM	Segnaletica Remota	Piattaforma Attuatori Integrati (EHW01)	Piattaforma evoluta per attuatori integrati. Fornisce interfaccia HW/SW verso sensori e attuatori. Comprende un Tablet di servizio Android adatto allo sviluppo di processi/procedure pilota a cui fornisce i dati dei sensori e i servizi per attuare il controllo degli accessi, direzione e velocità.
		UNIFI. DISIT		APP con Procedura di Segnaletica (APP04)	Processo pilota da installare/scaricare sulla piattaforma attuatori integrati. Utilizza i servizi della piattaforma per visualizzare grafica e testo secondo le indicazioni fornite dalla centrale Sii-Mobility, o altra centrale utente, per realizzare della segnaletica che può essere controllata da remoto per il controllo della direzione e della velocità.
3.3.2	Studio, definizione e sviluppo di Attuatori integrati per controllo accessi, direzione e velocità: video decisori	Project	Attuatori	video decisori (SN07, EHW02)	I video decisori sono apparsi in grado di elaborare le informazioni in formato dati per metterle a disposizione di applicazioni di attuazione e di comunicazione alla centrale Sii-mobility.

3.3.3	Studio, definizione e sviluppo di Attuatori integrati per controllo accessi, direzione e velocità: attuatori	ECM	Direzi onam ento Mezzi	Piattaforma Direzioneamento (EHW03)	Piattaforma evoluta per attuatori integrati. Fornisce interfaccia HW/SW verso sensori e attuatori. Comprende un Tablet di servizio Android adatto allo sviluppo di processi/procedure pilota a cui fornisce i dati dei sensori e i servizi per il controllo degli accessi per cambiare direzione in modo dinamico a segmenti di strade o strade.
		UNIFI. DISIT		APP con Procedura di Direzioneamento (APP05)	Processo pilota da installare/scaricare sulla piattaforma attuatori. Utilizza i servizi della piattaforma per visualizzare grafica e testo secondo le indicazioni fornite dalla centrale Sii-Mobility, o altra centrale utente, per il controllo degli accessi per cambiare direzione in modo dinamico a segmenti di strade o strade.
		ECM	rastrelliera	Piattaforma rastrelliera (EHW04)	Rastrelliera per bike sharing.
		UNIFI. DISIT		APP con gestione rastrelliera (APP06)	Gestione rastrelliera. Invio dati al SII sullo stato, integrazione e gestione rastrelliere multiple server side. Comunicazione con telefono per lo stato: attacco, stacco bici, contabilizzazione. Etc.
3.3.4	Studio, definizione e sviluppo di Attuatori integrati per controllo accessi, direzione e velocità: comunicazione	NEGENTIS → UNIFI DISIT	Attuatori	Protocollo di comunicazione Attuatore-Centrale Direzioneamento (API21) Prima era PROT01	Protocollo di comunicazione fra il sensore ed il sistema di gestione.

### 5.7.1 Piattaforma Attuatori Integrati, Procedura di Segnaletica (ECM)



#### 5.7.1.1 Piattaforma Attuatori Integrati (EHW01) (ECM)

<b>Profilo Tool/Algoritmo</b>	
Piattaforma Attuatori Integrati (EHW01)	
Nome responsabile	
Partner responsabile	ECM
Descrizione Tool/algoritmo	Piattaforma evoluta per attuatori integrati. Fornisce interfaccia HW/SW verso sensori e attuatori. Comprende un Tablet di servizio Android adatto allo sviluppo di processi/procedure pilota a cui fornisce i dati dei sensori e i servizi per attuare il controllo degli accessi, direzione e velocità.
Dati primari in ingresso	Valori di tensione da sensori passivi, messaggi da sensori attivi Protocollo base Bluetooth con Procedura di Segnaletica (APP04) – messaggi di controllo Messaggi da Procedura di Segnaletica (APP04) – comandi per attuatori Protocollo Wi-Fi con Kit veicolare evoluto (APP02) – messaggi di controllo e di dati
Dati prodotti in Uscita	Valori di tensione per attuatori passivi, messaggi per attuatori attivi Protocollo base Bluetooth con Procedura di Segnaletica (APP04) – messaggi di controllo Messaggi a Procedura di Segnaletica (APP04) – dati da sensori Protocollo Wi-Fi con Kit veicolare evoluto (APP02) – messaggi di controllo e di dati
Principi (metodo, basi teoriche, etc.)	Architettura HW/SW open basata su tecnologia Arduino
Casi di test (presenti/assenti)	assenti
Posizione casi di test	N/A

Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	approvato	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0 %	
Eseguibile/libreria/web app	Piattaforma HW/SW embedded	
Single thread/Multithread	TBD	
Linguaggio di sviluppo (java, Php, ETL)	Sketches Arduino	
Piattaforme supportate	N/A	
Posizione del codice sorgente	N/A	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	N/A	
Indirizzo/i accesso via web (se presenti)	N/A	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
N/A	N/A	N/A
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
N/A	N/A	N/A
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
Protocollo base Bluetooth	APP04	DE3.3 – sezione 3.1.4
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
N/A	N/A	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
N/A	N/A	N/A
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
N/A	N/A	N/A

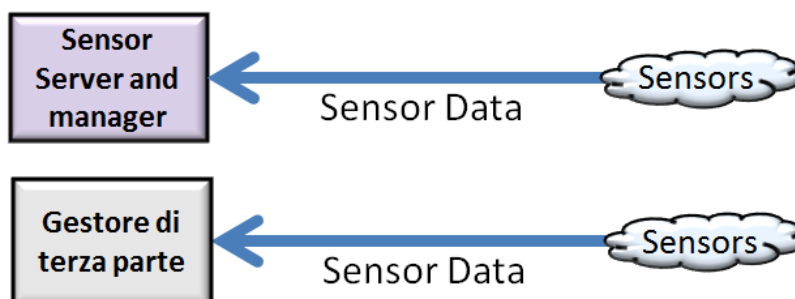
### 5.7.1.2 App con Procedura di Segnaletica (APP04) (UNIFI)

<b>Profilo Tool/Algoritmo</b>	
Procedura di Segnaletica (APP04)	
Nome responsabile	

Partner responsabile	UNIFI	
Descrizione Tool/algorithmo	Processo pilota da installare/scaricare sulla piattaforma attuatori integrati. Utilizza i servizi della piattaforma per visualizzare grafica e testo secondo le indicazioni fornite dalla centrale Sii-Mobility, o altra centrale utente, per realizzare della segnaletica che può essere controllata da remoto per il controllo degli accessi, direzione e velocità.	
Dati primari in ingresso	Protocollo base Bluetooth con Procedura Direzioneamento (APP05) – messaggi di controllo Messaggi a Procedura Direzioneamento (APP05)	
Dati prodotti in Uscita	Visualizzazione grafica dei dati provenienti dal protocollo base bluetooth	
Principi (metodo, basi teoriche, etc.)		
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	n/a	
Principali problemi non risolti	n/a	
Principali requisiti pendenti	n/a	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0.0%	
Eseguibile/libreria/web app	Mobile App	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Javascript, HTML, CSS Java	
Piattaforme supportate	Android	
Posizione del codice sorgente	SVN	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/a	
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
Protocollo base Bluetooth	EHW01	DE3.3 – sezione 3.1.4
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	

Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
Bootstrap	3.3.6	MIT
jquery	1.11.0	MIT
jQuery.mobile	1.4.5	MIT
jQuery.ui	1.11.4	MIT

### 5.7.2 Video decisore (SN07, EHW02) (PROJECT)



Profilo Tool/Algoritmo	
Video decisore (SN07)	
Nome responsabile	
Partner responsabile	PROJECT
Descrizione Tool/algoritmo	I video decisori sono apparati virtuali in grado di elaborare le informazioni in formato dati per metterle a disposizione di applicazioni di attuazione e di comunicazione alla centrale Sii-mobility.
Dati primari in ingresso	CLASSIFICAZIONE VEICOLI TRANSITATI
Dati prodotti in Uscita	ALLARMI EVENTI ANOMALI
Principi (metodo, basi teoriche, etc.)	DALL'ANALISI DEI VEICOLI RILEVATI IL SISTEMA DETERMINA IL VERIFICARSI DI EVENTI ANOMALI
Casi di test (presenti/assenti)	N.A..
Posizione casi di test	N.A.
Principali problemi non risolti	N.A.
Principali requisiti pendenti	N.A.
Aspetti Tecnologici	
Stato (proposto/approvato)	PROPOSTO
Implementato/non implementato	NON IMPLEMENTATO
Stato implementazione, percentuale	0%



Eseguibile/libreria/web app	DA DEFINIRE	
Single thread/Multithread	DA DEFINIRE	
Linguaggio di sviluppo (java, Php, ETL)	DA DEFINIRE	
Piattaforme supportate	DA DEFINIRE	
Posizione del codice sorgente	DA DEFINIRE	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	DA DEFINIRE	
Indirizzo/i accesso via web (se presenti)	DA DEFINIRE	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
N.A.	N.A.	N.A.
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
N.A.	N.A.	N.A.
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
N.A.	N.A.	N.A.
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
N.A.	N.A.	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
N.A.	N.A.	N.A.
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
N.A.	N.A.	N.A.

## Profilo Tool/Algoritmo

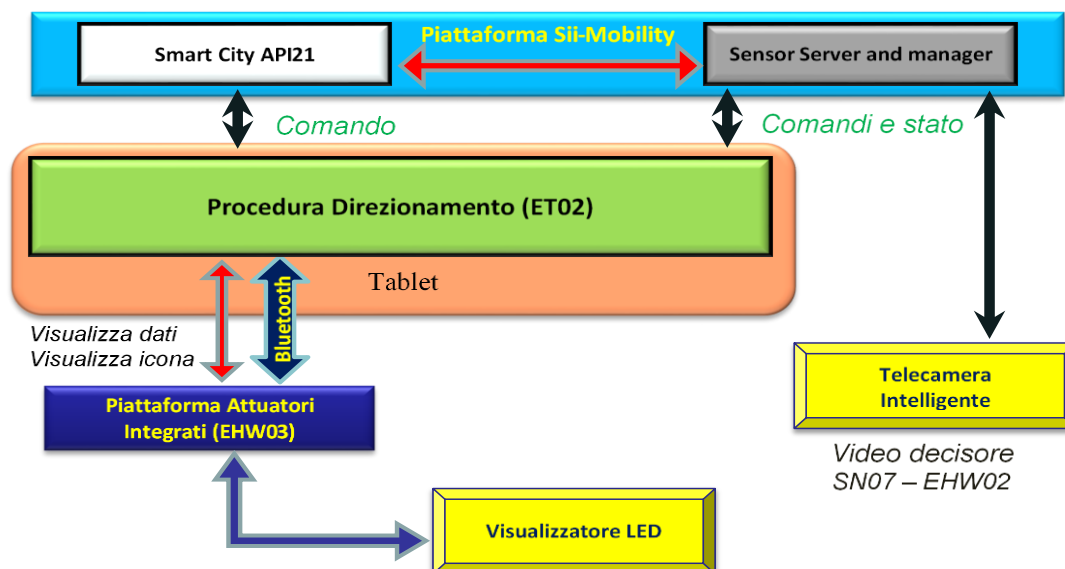
### Video decisore (EHW03)

Nome responsabile	
Partner responsabile	PROJECT
Descrizione Tool/algoritmo	I video decisori sono apparati virtuali in grado di elaborare le informazioni in formato dati per metterle a disposizione di applicazioni di attuazione e di comunicazione alla centrale Sii-mobility.
Dati primari in ingresso	I valori determinati dai videodecisori

Dati prodotti in Uscita	Valori determinati dai videodecisoro trasferiti con protocollo sensor server and manager	
Principi (metodo, basi teoriche, etc.)	ARCHITETTURA HW/SW BASATRA SU PIATTAFORMA WINDOWS	
Casi di test (presenti/assenti)	N.A.	
Posizione casi di test	N.A.	
Principali problemi non risolti	N.A.	
Principali requisiti pendenti	N.A.	
Aspetti Tecnologici		
Stato (proposto/approvato)	PROPOSTO	
Implementato/non implementato	NON IMPLEMENTATO	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	DA DEFINIRE	
Single thread/Multithread	DA DEFINIRE	
Linguaggio di sviluppo (java, Php, ETL)	<b>DA DEFINIRE</b>	
Piattaforme supportate	N.A.	
Posizione del codice sorgente	N.A.	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	N.A.	
Indirizzo/i accesso via web (se presenti)	N.A.	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
N.A.	N.A.	N.A.
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
N.A.	N.A.	N.A.
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
BLUE TOOTH	APP04	DE3.3 SEZ. 3.1.4
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
N.A.	N.A.	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
N.A.	N.A.	N.A.
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

N.A.	N.A.	N.A.

### 5.7.3 Piattaforma Direzione e Procedura Direzione (EHW03, APP05) (ECM)



<h2 style="text-align: center;">Profilo Tool/Algoritmo</h2>	
Piattaforma Direzione Attuatori Integrati (EHW03)	
Nome responsabile	
Partner responsabile	ECM
Descrizione Tool/algoritmo	Piattaforma evoluta per attuatori integrati. Fornisce interfaccia HW/SW verso sensori e attuatori. Comprende un Tablet di servizio Android adatto allo sviluppo di processi/procedure pilota a cui fornisce i dati dei sensori e i servizi per il controllo degli accessi per cambiare direzione in modo dinamico a segmenti di strade o strade.
Dati primari in ingresso	Valori di tensione da sensori passivi, messaggi da sensori attivi Protocollo base Bluetooth con Procedura Direzione (APP05) – messaggi di controllo Messaggi da Procedura Direzione (APP05) – comandi per attuatori
Dati prodotti in Uscita	Valori di tensione per attuatori passivi, messaggi per attuatori attivi Protocollo base Bluetooth con Procedura Direzione (APP05) – messaggi di controllo Messaggi a Procedura Direzione (APP05) – dati da sensori
Principi (metodo, basi teoriche, etc.)	Architettura HW/SW open basata su tecnologia Arduino
Casi di test (presenti/assenti)	assenti

Posizione casi di test	N/A	
Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	approvato	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0 %	
Eseguibile/libreria/web app	Piattaforma HW/SW embedded	
Single thread/Multithread	TBD	
Linguaggio di sviluppo (java, Php, ETL)	Sketches Arduino	
Piattaforme supportate	N/A	
Posizione del codice sorgente	N/A	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	N/A	
Indirizzo/i accesso via web (se presenti)	N/A	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
N/A	N/A	N/A
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
N/A	N/A	N/A
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
Protocollo base Bluetooth	APP05	DE3.3 – sezione 3.1.4
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
N/A	N/A	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
N/A	N/A	N/A
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
N/A	N/A	N/A

### 5.7.3.1 App con Procedura Direzioneamento (APP05) (UNIFI)

Profilo Tool/Algoritmo	
Procedura Direzioneamento (APP05)	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Processo pilota da installare/scaricare sulla piattaforma attuatori. Utilizza i servizi della piattaforma per

	visualizzare grafica e testo secondo le indicazioni fornite dalla centrale Sii-Mobility, o altra centrale utente, per il controllo degli accessi per cambiare direzione in modo dinamico a segmenti di strade o strade.	
Dati primari in ingresso	Protocollo base Bluetooth con Procedura Direzioneamento (APP05) – messaggi di controllo Messaggi a Procedura Direzioneamento (APP05)	
Dati prodotti in Uscita	Visualizzazione grafica dei dati provenienti dal protocollo base bluetooth	
Principi (metodo, basi teoriche, etc.)		
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	n/a	
Principali problemi non risolti	n/a	
Principali requisiti pendenti	n/a	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0.0%	
Eseguibile/libreria/web app	Mobile App	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Javascript, HTML, CSS Java	
Piattaforme supportate	Android	
Posizione del codice sorgente	SVN	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/a	
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Cordova 5.4.0		
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
Protocollo base Bluetooth	EHW03	DE3.3 – sezione 3.1.4
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b>	<b>Modello sviluppo,</b>	<b>Libreria usata per UI</b>

(web/applicazione)	linguaggio	
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
Bootstrap	3.3.6	MIT
jquery	1.11.0	MIT
Jquery.mobile	1.4.5	MIT
Jquery.ui	1.11.4	MIT

#### 5.7.4 Piattaforma rastrelliera (EHW05, APP06) (ECM)

<b>Profilo Tool/Algoritmo</b>		
Piattaforma rastrelliera (EHW05)		
Nome responsabile		
Partner responsabile	ECM	
Descrizione Tool/algoritmo	Piattaforma evoluta per gestione rastrelliera	
Dati primari in ingresso	Piattaforma evoluta per gestione rastrelliera	
Dati prodotti in Uscita	Piattaforma evoluta per gestione rastrelliera	
Principi (metodo, basi teoriche, etc.)	Architettura HW/SW open basata su tecnologia Arduino	
Casi di test (presenti/assenti)	assenti	
Posizione casi di test	N/A	
Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	approvato	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0 %	
Eseguibile/libreria/web app	Piattaforma HW/SW embedded	
Single thread/Multithread	TBD	
Linguaggio di sviluppo (java, Php, ETL)	Sketches Arduino	
Piattaforme supportate	N/A	
Posizione del codice sorgente	N/A	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	N/A	
Indirizzo/i accesso via web (se presenti)	N/A	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
N/A	N/A	N/A
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>

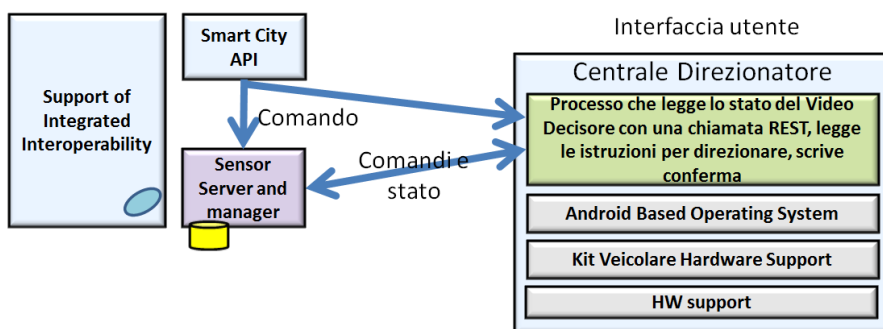
N/A	N/A	N/A
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
Protocollo base Bluetooth	APP05	DE3.3 – sezione 3.1.4
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
N/A	N/A	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
N/A	N/A	N/A
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
N/A	N/A	N/A

#### 5.7.4.1 App con Procedura Gestione Rastrelliera (APP06) (UNIFI)

<b>Profilo Tool/Algoritmo</b>	
Procedura Direzioneamento (APP05)	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Gestione rastrelliera. Invio dati al SII sullo stato, integrazione e gestione rastrelliere multiple server side. Comunicazione con telefono per lo stato: attacco, stacco bici, contabilizzazione. Etc.
Dati primari in ingresso	Protocollo base Bluetooth con Procedura Direzioneamento (APP06) – messaggi di controllo Messaggi a Procedura Direzioneamento (APP06)
Dati prodotti in Uscita	Visualizzazione grafica dei dati provenienti dal protocollo base bluetooth
Principi (metodo, basi teoriche, etc.)	
Casi di test (presenti/assenti)	Assenti
Posizione casi di test	n/a
Principali problemi non risolti	n/a
Principali requisiti pendenti	n/a
Aspetti Tecnologici	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0.0%
Eseguibile/libreria/web app	Mobile App
Single thread/Multithread	Multithread
Linguaggio di sviluppo (java, Php, ETL)	Javascript, HTML, CSS Java
Piattaforme supportate	Android
Posizione del codice sorgente	SVN
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se	n/a

necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Cordova 5.4.0		
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
Protocollo base Bluetooth	EHW04	DE3.3 – sezione 3.1.4
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
Bootstrap	3.3.6	MIT
jquery	1.11.0	MIT
Jquery.mobile	1.4.5	MIT
Jquery.ui	1.11.4	MIT

### 5.7.5 Protocollo di comunicazione Attuatore-Centrale Direzioneamento (API21) (UNIFI DISIT)



Invio di comandi da SSM verso la centrale di direzioneamento (come per lo blocco delle bici). Questa viene implementata tramite una applicazione mobile. Tale applicazione in modo periodico interroga le Sensor API e riceve in risposta eventuali azionamenti. Pertanto si faccia riferimento a



protocollo che da processo in background del amobile comunica con SSM. Le informazioni di attuazione sono inviate dal gestore del direzionamento che potrebbe essere semplicemente un bottone di attuazione su dashboard.

### 5.8 Specifica: Integrazione dei sensori e attuatori verso il SII, ottimizzazione

		ECM	Mizar	UNIFI	IN20	TIME	Negentis	EffKnow	liberologico	ataf	bustalia	cttnord	fiemme	argos	elfi	calamai	Midra	Project	GEOIN	OUJESTIT	SOETEC	FWINGS	
3.4.1	Integrazione verso il SII, ottimizzazione: IN/OUT kit veicolari, sensori, attuatori	X				X			X	X	X	X											
3.4.2	Integrazione verso il SII, ottimizzazione: API per IN/OUT con mobile e Totem			X		X																	
3.4.3	Integrazione verso il SII, ottimizzazione: API per dati PA e SME	X	X						X	X	X	X											
3.4.4	Integrazione verso il SII, ottimizzazione: API sii-mobility interop. e con altre centrali	X	X			X																	

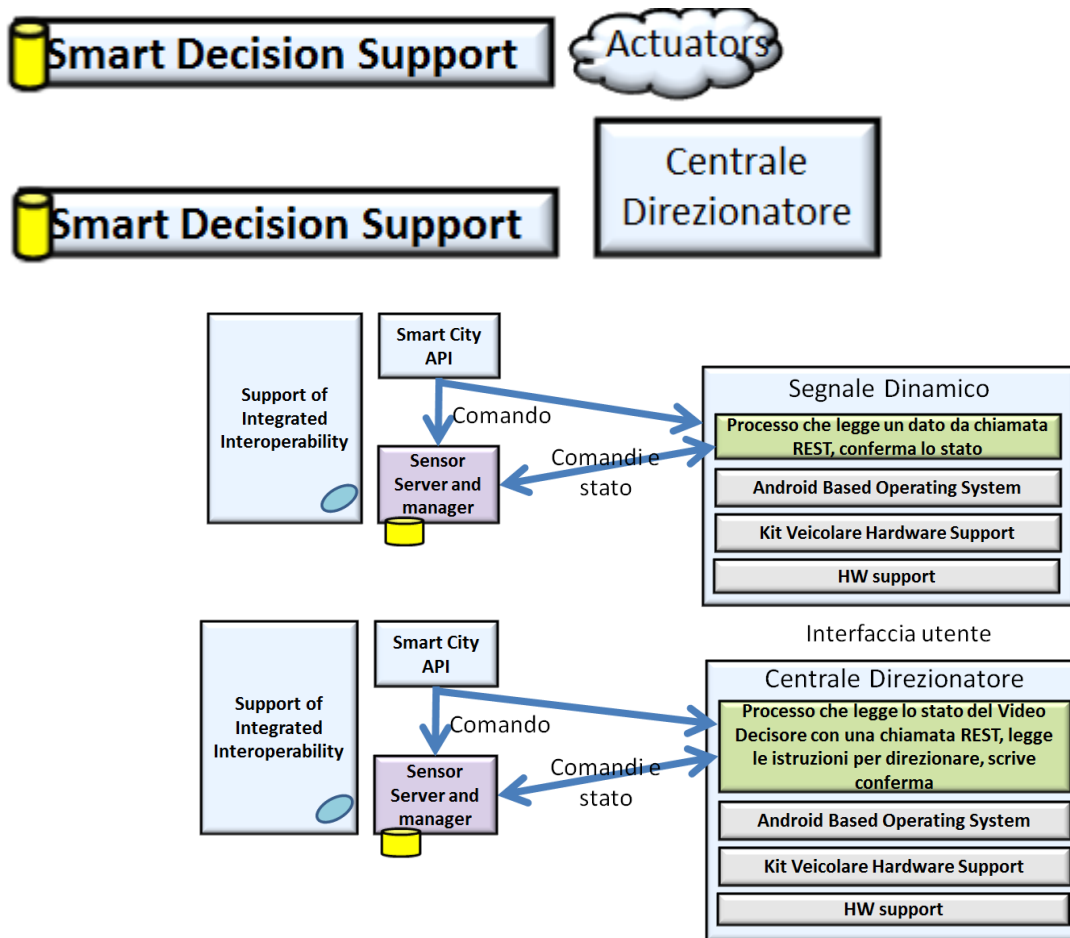
Principali Sottosistemi interessati dai requisiti da descrivere in questa Subsubsection, saranno i sottosistemi primari della specifica					
	Sotto Attività	Partner coord	Sottosistema	Tool/Level	Descrizione sommaria
3.4.1	Integrazione verso il SII, ottimizzazione: IN/OUT kit veicolari, sensori, attuatori	UNIFI.D ISIT	Attuatori Sensor Server and Manager	Protocollo di comunicazione tra Centrale Direzioneamento ed SII (API07)	Protocollo di comunicazione fra la Centrale Direzioneamento ed il sistema SII. Il sistema di gestione deve presentare delle API per: accettare comandi dall'esterno programmando l'ora del cambio direzione oppure subito, fornire dati relativi allo stato ed ai sensori ed allo storico.
		MIZAR	Monitoring Supervisor	API per integrazione di nuove sorgenti dati (API08)	API per: integrare dati provenienti da nuovi sorgenti (e.g. FCD, floating cellular data) e per fornire dati relativi allo stato ed ai sensori ed allo storico proveniente dal Gestore della mobilità.
3.4.2	Integrazione verso il SII, ottimizzazione: API per IN/OUT con mobile e Totem	UNIFI	Smart City API	API REST di Knowledge Base su base Geolocation (API09)	API per accesso ai dati della Knowledge Base tramite REST call, sulla base di una coordinata GPS, per ottenere: XML, JSON, HTML. API con accesso condizionato in base all'utente, al profilo utente, ai dati richiesti.
		UNIFI	Smart City API	API SPARQL di Knowledge Base (API10)	API per accesso ai dati di della Knowledge Base tramite SPQRQL, per ottenere: XML, JSON, HTML. API con accesso condizionato in base all'utente, al profilo utente, ai dati richiesti.

		UNIFI	Smart City API	API for rendering deductions on the basis of the whole ontological model o SmartDS (API11)	API con le quali è possibile fare query sulla knowledge base per servizi e per informazioni statistiche, ma anche sullo stato di processi decisionali dello SmartDS. Questo significa avere accesso ai dati del database del DashBoard manager. API con accesso condizionato in base all'utente, al profilo utente, ai dati richiesti.
		UNIFI	Smart City API	API Query ID di Knowledge Base (API12)	API per accesso ai dati di della Knowledge Base tramite Query ID, per ottenere: XML, JSON, HTML. API con accesso condizionato in base all'utente, al profilo utente, ai dati richiesti.
		UNIFI	Smart City API	API User Crowd Sourcing (API13)	API per fornire informazioni e dati al sistema partecipativo e comunque al Sii. Informazioni come: immagini, commenti, ranking/voti riguardo a servizi
		UNIFI	Smart City API	API User Engagement (API14)	API per inviare in push (da server verso le mobile APP e web) suggerimenti e stimoli a partecipare, compiti da svolgere, etc.
		UNIFI	Smart City API	API User Profiling and Recommendations (API15)	API per permettere alle App mobile e WEB di richiedere suggerimenti al server di user profiling and suggestion e comunicare user profile, come anche il suo menu, comportamento, etc..
3.4.3	Integrazione verso il SII, ottimizzazione: API per dati PA e SME	MIZAR	Smart City API	API request planning (API16)	API per richiedere la pianificazione del percorso fra due punti (o sequenza di punti) tramite: mezzi pubblici, auto, e a piedi. API con accesso condizionato in base all'utente, al profilo utente, ai dati richiesti.
		MIZAR	Smart City API	API pubblicazione dati (API17)	API per accedere a dati storici per PA e SME. API con accesso condizionato in base all'utente, al profilo utente, ai dati richiesti per implementazione di nuovi scenari B2B o B2C
3.4.4	Integrazione verso il SII, ottimizzazione: API sii-mobility interop. e con altre centrali	UNIFI	Smart City API	API verso altre smart city (API18)	API con le quali altre centrali smart city possono richiedere e fornire informazioni. API con accesso condizionato in base all'utente, al profilo utente, ai dati richiesti.

### 5.8.1 Protocollo di comunicazione tra Centrale Direzioneamento ed SII (API07) (UNIFI)

La centrale di direzioneamento è nella buona sostanza (come il sistema di messaggio variabile) una applicazione android su un dispositivo android gemellato con un Arduino che gestisce le parti IO. La centrale di direzioneamento va direttamente a leggere su certe API il lo stato che dovrebbe acquisire. In questo modo è il server su Sensors Server Manager che espone il servizio, la App semplicemente va a leggere con modalità polling per vedere se vi sono dei cambi di stato programmati o da eseguire in immediatamente.

Si veda la soluzione fra SmartDS, Dashboard e Sensor Server Manager. Tale soluzione permette di scrivere dentro il SSM lo stato al quale un attuatore deve conformarsi. E' l'attuatore stesso che chiama il SSM per leggere se vi sono informazioni/comandi per cambiare stato. Questo permette di avere attuatori anche in DHCP e anche su reti mobili. Il protocollo dovrebbe essere autenticato.



## Profilo Tool/Algoritmo

Protocollo di comunicazione tra Centrale Direzione ed SII (API07)

Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Protocollo di comunicazione fra la Centrale Direzione ed il sistema SII. Il sistema di gestione Sensor Server Manager deve presentare delle API per: accettare comandi dall'esterno programmando l'ora del cambio direzione oppure subito, fornire dati relativi allo stato ed ai sensori ed allo storico.
Dati primari in ingresso	Identificativi dei sensori, intervalli di tempo
Dati prodotti in Uscita	Dati dei sensori di traffico, ZTL, parcheggi
Principi (metodo, basi teoriche, etc.)	La centrale legge lo stato che deve acquisire o che altro dal Sensor Server Manager
Casi di test (presenti/assenti)	nessuno
Posizione casi di test	n.a.
Principali problemi non risolti	nessuno
Principali requisiti pendenti	nessuno

Aspetti Tecnologici		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Mobile app, lato server come PHP	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java su mobile, PHP su server	
Piattaforme supportate	Android, Linux	
Posizione del codice sorgente	repository di progetto	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	da definire	
Indirizzo/i accesso via web (se presenti)	da definire	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
n.a.	n.a.	HTTP protocol
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
JSON	n.a.	n.a.
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
HTTP	n.a.	n.a.
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
	n.a.	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
n.a.	n.a.	n.a.
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
n.a.	n.a.	n.a.

### 5.8.2 API per integrazione di nuove sorgenti dati (API08) (MIZAR)

<h2 style="text-align: center;">Profilo Tool/Algoritmo</h2>	
API per Integrazione di nuove sorgenti dati (API08)	
Nome responsabile	
Partner responsabile	MIZAR

Descrizione Tool/algorithmo	API per: integrare dati provenienti da nuovi sorgenti (e.g. FCD, floating cellular data) e per fornire dati relativi allo stato ed ai sensori ed allo storico proveniente dal Gestore della mobilità. Queste API sono esportate dal supervisore verso sii-mobility ed includono dati del supervisore attuale piu' i dati che saranno sviluppati dai punti 5.6.3	
Dati primari in ingresso	Identificativo del sensore, misure	
Dati prodotti in Uscita	Misure	
Principi (metodo, basi teoriche, etc.)	L'API riceve dal sensore le misure e le inoltra a SII	
Casi di test (presenti/assenti)	nessuno	
Posizione casi di test	n.a.	
Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Web app	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	C#	
Piattaforme supportate	Windows	
Posizione del codice sorgente	da definire	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	<a href="http://IPserver/rest/api/v1/">http://IPserver/rest/api/v1/</a> per il progetto sarà configurato un nome utente (SII_test) e password (511_t35t)	
Indirizzo/i accesso via web (se presenti)	Non applicabile	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n.a.	n.a.	HTTP
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON/XML	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTP	n.a.	n.a.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
n.a.	n.a.	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>

Nessuna	n.a.	n.a.
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
n.a.	n.a.	Proprietaria, ad uso gratuito per il progetto SII

### 5.8.2.1 Protocollo Supervisore-SII, API: API08 (MIZAR)

Per ogni oggetto che produce misure (regolatori semaforici, stazioni di misura, sensori) è possibile impostare delle misure.

Per impostare le misure è necessario invocare il metodo **POST** al seguente URL:

```
http://domain/rest/v1/objecttypes/{GUID}/measures
```

con il valore appropriato nei campi *{GUID}*

In caso di successo, il metodo restituisce il codice HTTP **200 OK**.

#### Esempio di richiesta:

```
POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{
  "Measures": [
    {
      "MeasureTimeUTC": "2016-06-08T15:50:00Z",
      "Sensors": [
        {
          "Data": [
            {
              "Key": "VOLUME",
              "Value": "20"
            },
            {
              "Key": "SPEED",
              "Value": "50"
            }
          ]
        },
        {
          "Data": [
            {
              "Key": "VOLUME",
              "Value": "30"
            },
            {
              "Key": "SPEED",
              "Value": "60"
            }
          ]
        },
        {
          "Data": [
            {
              "Key": "VOLUME",
              "Value": "20"
            },
            {
              "Key": "SPEED",

```

```

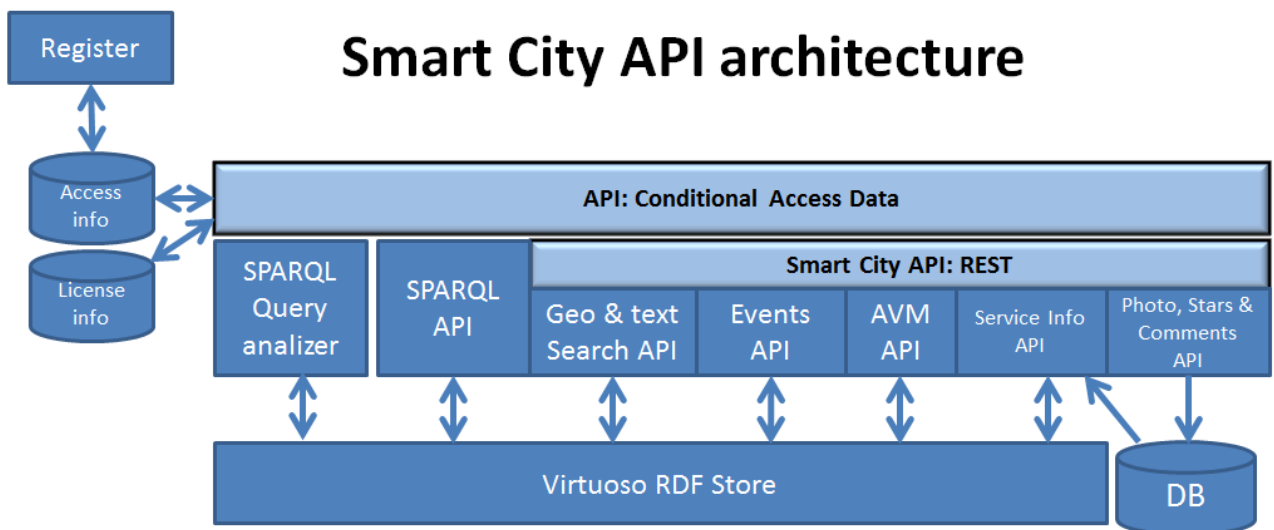
        "Value": "50"
      }
    ]
  },
  {
    "Data": [
      {
        "Key": "VOLUME",
        "Value": "30"
      },
      {
        "Key": "SPEED",
        "Value": "60"
      }
    ]
  }
]
}
]
}
]
}' 'http://192.168.55.75:8088/rest/v1/detectionunits/6f796196-5ca7-4546-ad84-17ae79224e4b/measures?token=1234'
    
```

**Esempio di risposta:**

```

RESPONSE CODE: 200
RESPONSE HEADER:
{
  "access-control-allow-origin": "*",
  "date": "Wed, 08 Jun 2016 15:56:31 GMT",
  "server": "Microsoft-HTTPAPI/2.0",
  "content-length": "0",
  "content-type": null
}
    
```

**5.8.3 API REST di Knowledge Base su base Geolocation (API09) (UNIFI)**



**Profilo Tool/Algoritmo**

API REST di Knowledge Base su base Geolocation (API09)

Nome responsabile	
Partner responsabile	UNIFI

Descrizione Tool/algorithmo	<p>Il modulo fornisce delle API REST per ottenere informazioni sui servizi e infrastrutture presenti nella Knowledge Base sulla base di informazioni di geolocalizzazione come coordinate GPS (su singolo punto, area geografica, percorso) o indicazione del comune.</p> <p>API per accesso ai dati della Knowledge Base tramite REST call, sulla base di una coordinata GPS, per ottenere: XML, JSON, HTML.</p> <p>API con accesso condizionato in base all'utente, al profilo utente, ai dati richiesti.</p>	
Dati primari in ingresso	Indicazione geografica	
Dati prodotti in Uscita	descrizione dei servizi o infrastrutture presenti nell'area indicata	
Principi (metodo, basi teoriche, etc.)	Uso di query SPARQL	
Casi di test (presenti/assenti)	assenti	
Posizione casi di test	-	
Principali problemi non risolti	Nessuno	
Principali requisiti pendenti	Nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	approvato	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	web app	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java	
Piattaforme supportate	Quelle supportate da Java	
Posizione del codice sorgente	-	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	<a href="http://servicemap.disit.org/api/...">http://servicemap.disit.org/api/...</a>	
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
ServiceMap		
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>



Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
na		
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

### 5.8.3.1 Protocollo esposto

Il modulo espone una interfaccia REST su protocollo HTTP per l'accesso ai dati presenti nella KnowledgeBase.

<b>Nome:</b> Service Search	
<b>URL:</b> <a href="http://servicemap.disit.org/WebAppGrafo/api/v1">http://servicemap.disit.org/WebAppGrafo/api/v1</a>	
<b>Descrizione</b>	
<p>This API provides results about a selection of categories of services chosen in a municipality or from a starting point (giving geolat position). It is possible to choose the number of results required, the radius of search, if the selection is a couple of latitude and longitude, and the format result.</p> <p>This case could be useful, as json results could be parsed in order to obtain single Service Uris to reuse as in case 1, calling the API <a href="http://servicemap.disit.org/api?serviceUri=[serviceUri]">servicemap.disit.org/api?serviceUri=[serviceUri]</a>. Real time pins could be actualized simply calling this uri, and showing them on the map taking the geo-located coordinates.</p>	
Modalità: GET	
<b>Parametri</b>	
Selection	<p>the start point of the query:</p> <ul style="list-style-type: none"> <li>• Could be a Latitude;Longitude coordinates (43.77078623698416; 11.256734415727848),</li> <li>• Municipality (COMUNE di FIRENZE)</li> <li>• a serviceUri (<a href="http://www.disit.org/km4city/resource/4573d318065f48a2214066f7b32d94c4">http://www.disit.org/km4city/resource/4573d318065f48a2214066f7b32d94c4</a>)</li> </ul>
categories	<p>the list of categories of services that user is interested in, they can be categories or subcategories separated with ';'. If a category is specified all the subcategories are automatically considered, the Service top category matches all services. Some of the possible categories and subcategories are (for a complete list see <a href="http://servicemap.disit.org">http://servicemap.disit.org</a>):</p> <p>Service</p> <ul style="list-style-type: none"> <li>• Accommodation <ul style="list-style-type: none"> <li>○ boarding_house; agritourism;hotel; bed_and_breakfast; camping; rest_home;</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>○ religios_guest_house; summer_residence; day_care_center; hostel; vacation_resort;</li> <li>○ farm_house; historic_residence; mountain_dew; beach_resort; holiday_village;</li> <li>● WineAndFood;             <ul style="list-style-type: none"> <li>○ highway_stop; pastry_shop; small_shop; catering; wine_shop_and_wine_bar;</li> <li>○ bakery; ice_cream_parlour; dining_hall; sandwich_shop_pub; pizzeria;</li> <li>○ restaurant; grill; sushi_bar;trattoria</li> <li>○ ...</li> </ul> </li> <li>● Sensor</li> <li>● BusStop</li> </ul>
maxResults	in the range of 100, 200, 500, or 0 for no limits. At most three values separated with ';' can be specified for the number of results: one for the service results, one for sensors and one for bus stops. If one value is missing it is assumed equal to the previous, so if one value is specified all the three values are considered equal, if the default value is set to 100.
maxDists	is active when the selection is a point on the map or a service. Indicates the maximum distance in km from the point (or service) specified as selection. Like for the number of results, at most three parameters can to be specified, one for services, one for sensors and one for bus stops. If one is missing it is assumed equal to the previous one, so if one is specified all the three values are considered equal. If omitted it is assumed equal to 0.1km. This parameter is ignored for searches in a municipality (COMUNE di..).
text	(optional) is a sequence of keywords separated by spaces to be present in any field of the service.
lang	(optional) it can be 'it' or 'en' (the default is en) the language to be used to return the typeLabel attribute
format	the format to be used for the results it can be json or html
<b>Esempi</b>	
<p>Search services in the category of Accommodation and Bus Stops around a point, giving the coordinates Latitude and Longitude, within 200m for services and 100m for bus stops.</p> <p><b><a href="http://servicemap.disit.org/WebAppGrafo/api/v1?selection=43.769346%3B11.25524&amp;categories=Accommodation%3BBusStops&amp;maxResults=100&amp;maxDists=0.2%3B0.1%3B0.1&amp;lang=it&amp;format=json">http://servicemap.disit.org/WebAppGrafo/api/v1?selection=43.769346%3B11.25524&amp;categories=Accommodation%3BBusStops&amp;maxResults=100&amp;maxDists=0.2%3B0.1%3B0.1&amp;lang=it&amp;format=json</a></b></p> <p><b>JSON Results are:</b></p> <pre>{   "BusStops":{     "type":"FeatureCollection",     "features":[       {         "geometry":{           "type":"Point",           "coordinates":[ 11.2555, 43.7702 ]         }       }     ]   } }</pre>	

```
    },
    "type":"Feature",
    "properties":{
      "name":"CONDOTTA",
      "typeLabel":"Fermata",
      "serviceType":"TransferServiceAndRenting_BusStop",
      "serviceUri":"http://www.disit.org/km4city/resource/FM5132"
    },
    "id":1
  }
]
},
"Services":{
  "type":"FeatureCollection",
  "features":[
    {
      "geometry":{
        "type":"Point",
        "coordinates":[ 11.2552, 43.7694 ]
      },
      "type":"Feature",
      "properties":{
        "name":"SOGGIORNO_ANTICA_TORRE",
        "typeLabel":"Affittacamere",
        "serviceType":"Accommodation_boarding_house",
        "serviceUri":"http://www.disit.org/km4city/resource/2a97797accf3ed11b7a2eefb635b666c",
      },
      "id":1
    },
    {
      "geometry":{
        "type":"Point",
        "coordinates":[ 11.2552, 43.7694 ]
      },
      "type":"Feature",
      "properties":{
        "name":"SOGGIORNO_ANTICA_TORRE",
        "typeLabel":"Affittacamere",
        "serviceType":"Accommodation_boarding_house",
        "serviceUri":"http://www.disit.org/km4city/resource/36780267be3f4ac30027489d347f2fc5",
      },
      "id":2
    }
    ....
  ]
}
```

}  
 }  
 (These results have been truncated.)

If HTML format is chosen, the result is like the following:



<b>Nome:</b> Service Info	
<b>URL:</b> <a href="http://servicemap.disit.org/WebAppGrafo/api/v1">http://servicemap.disit.org/WebAppGrafo/api/v1</a>	
<b>Descrizione</b>	
<p><b>This API provides results about a service, given the serviceUri. It is possible to choose the format of result, as json or html. The system query the km4city ontology taking information on the service, and depending on type, give results for static services, weather predictions or real time data. If the format chosen is html, the result will be displayed on ServiceMap portal as web page.</b></p>	
Modalità: GET	
<b>Parametri</b>	
serviceUri	URI of the service/entity (e.g. )
lang	Language of results (en or it)
Format	the format to be used for the results it can be json or html
<b>Esempi</b>	
<ul style="list-style-type: none"> <li>For a static service identified by uri <a href="http://www.disit.org/km4city/resource/76933db4b76647226ddabbb62477cd12">http://www.disit.org/km4city/resource/76933db4b76647226ddabbb62477cd12</a> call to <a href="http://servicemap.disit.org/WebAppGrafo/api/v1?serviceUri=http://www.disit.org/km4city/resource/76933db4b76647226ddabbb62477cd12&amp;format=json">http://servicemap.disit.org/WebAppGrafo/api/v1?serviceUri=http://www.disit.org/km4city/resource/76933db4b76647226ddabbb62477cd12&amp;format=json</a> provides something like:                     <pre>{       "Service":{         "type":"FeatureCollection",         "features":[</pre> </li> </ul>	

```
{
  "geometry":{
    "type":"Point",
    "coordinates":[ 11.256, 43.7752 ]
  },
  "type":"Feature",
  "properties":{
    "name":"PALAZZO_MEDICI_RICCARDI",
    "typeLabel":"Monument location",
    "serviceType":"CulturalActivity_Monument_location",
    "phone":"",
    "fax":"",
    "website":"",
    "province":"FI",
    "city":"FIRENZE",
    "cap":"50100",
    "email":"",
    "note":"",
    "linkDBpedia" : [],
    "description":"",
    "description2":"",
    "multimedia":"",

    "serviceUri":"http://www.disit.org/km4city/resource/76933db4b76647226ddabbb62477c
d12",
    "address":"VIA CAVOUR",
    "civic":"3"
  },
  "id":1
}
]
```

- For a BUS Stop with real time information:

<http://servicemap.disit.org/WebAppGrafo/api/v1?serviceUri=http://www.disit.org/km4city/resource/FM0022&format=json>

provides something like:

```
{
  "Fermata": {
    "type": "FeatureCollection",
    "features": [
      {
        "geometry": {
          "type": "Point",
          "coordinates": [
            11.249076423532143,
```

```
        43.7764653220976
      ]
    },
    "type": "Feature",
    "properties": {
      "name": "STAZIONE PENSILINA",
      "serviceUri": "http://www.disit.org/km4city/resource/FM0022",
      "typeLabel": "TransferServiceAndRenting_BusStop"
    },
    "id": 1
  }
]
},
"linee": {
  "head": {
    "BusStop": "STAZIONE PENSILINA",
    "vars": "busLine"
  },
  "results": {
    "bindings": [
      {
        "busLine": {
          "type": "literal",
          "value": "11"
        }
      },
      {
        "busLine": {
          "type": "literal",
          "value": "17"
        }
      },
      {
        "busLine": {
          "type": "literal",
          "value": "22"
        }
      },
      {
        "busLine": {
          "type": "literal",
          "value": "23"
        }
      },
      {
        "busLine": {
          "type": "literal",
```

```
        "value": "36"
      }
    },
    {
      "busLine": {
        "type": "literal",
        "value": "4"
      }
    },
    {
      "busLine": {
        "type": "literal",
        "value": "52"
      }
    },
    {
      "busLine": {
        "type": "literal",
        "value": "54"
      }
    },
    {
      "busLine": {
        "type": "literal",
        "value": "6"
      }
    }
  ]
},
"realtime": {
  "head": {
    "busStop": [
      "STAZIONE PENSILINA"
    ],
    "vars": [
      "arrivalTime",
      "busLine",
      "state",
      "direction",
      "ride"
    ]
  },
  "results": {
    "bindings": [
      {
        "arrivalTime": {
```

```

        "type": "literal",
        "value": "15:16:54"
    },
    "busLine": {
        "type": "literal",
        "value": "4"
    },
    "state": {
        "type": "literal",
        "value": "Ritardo"
    },
    "ride": {
        "type": "literal",
        "value": "5084785"
    },
    "direction": {
        "type": "literal",
        "value": "CAPPUCCINI"
    }
    ]
}
}
}
}

```

- ROAD SENSOR with realtime information:

<http://servicemap.disit.org/WebAppGrafo/api/v1?serviceUri=http://www.disit.org/km4city/resource/EM0100102&format=json>

provides something like:

```

{"Sensor":
  { "type": "FeatureCollection",
    "features": [
      { "geometry": {
          "type": "Point",
          "coordinates": [ 10.928608, 43.72319 ]
        },
        "type": "Feature",
        "properties": {
          "name": "EM0100102",
          "typeLabel": "Sensor",
          "serviceType": "TransferServiceAndRenting_SensorSite",
          "serviceUri": "http://www.disit.org/km4city/resource/EM0100102",
          "municipality": "EMPOLI",
          "address": "VIALE GIOVANNI BOCCACCIO",
          "photos": [],
          "avgStars": 0.0,
          "starsCount": 0,
          "comments": [], "id": 1 }
    ]
  }
}

```



```

,"realtime":
{
  "head": {"sensor": ["EM0100102"],"vars": ["avgDistance",
"avgTime","occupancy","concentration","vehicleFlow","averageSpeed","thresholdPerc","s
peedPercentile","instantTime"]},
"results": {
"bindings": [
{
"avgDistance": {"type": "literal","value": "120.48573" },
"avgTime": {"type": "literal","value": "15.267568" },
"occupancy": {"type": "literal","value": "0.03136767" },
"concentration": {"type": "literal","value": "7.0" },
"vehicleFlow": {"type": "literal","value": "37.0" },
"averageSpeed": {"type": "literal","value": "29.18919" },
"thresholdPerc": {"type": "literal","value": "0.0" },
"speedPercentile": {"type": "literal","value": "Not Available" },
"instantTime": {"type": "literal","value": "2015-10-09T09:30:00+02:00" }
}
]}}}
}

```

- PARKING LOT with realtime information:

[http://servicemap.disit.org/WebAppGrafo/api/v1?serviceUri=http://www.disit.org/km4city/resource/RT048014PK002EM\\_PO&format=json](http://servicemap.disit.org/WebAppGrafo/api/v1?serviceUri=http://www.disit.org/km4city/resource/RT048014PK002EM_PO&format=json)

```

{"Service":
{"type": "FeatureCollection",
"features": [

{ "geometry": {
  "type": "Point",
  "coordinates": [ 10.941049, 43.714954 ]
},
"type": "Feature",
"properties": {
  "name": "B. Buozzi",
  "typeLabel": "Car park",
  "serviceType": "TransferServiceAndRenting_Car_park",
  "phone": "05717571",
  "fax": "",
  "website": "",
  "province": "FI",
  "city": "Empoli",
  "cap": "50053",
  "email": "",
  "linkDBpedia": ["http://it.dbpedia.org/resource/Bruno_Buozzi"],
  "note": "",
  "description": "",
  "description2": "",

```

```

"multimedia": "",
"serviceUri": "http://www.disit.org/km4city/resource/RT048014PK002EM_PO",
"address": "VIALE B. BUOZZI", "civic": "",
"wktGeometry": "",
"photos": [],
"avgStars": 0.0,
"starsCount": 0,
"comments": []},
"id": 1
}
]}
,"realtime":
{"head": {"parkingArea": [ "B. BuoZZi"], "vars": [ "capacity",
"freeParkingLots", "occupiedParkingLots", "occupancy", "updating"]},
"results": {
"bindings": [
{"capacity": {"value": "146" }, "freeParkingLots": {"value": "0" }, "occupiedParkingLots":
{"value": "146" }, "occupancy": {"value": "100.0" }, "status": {"value": "carParkFull"
}, "updating": {"value": "2016-07-28T06:46:00+02:00" }
}
]}}}

```

- For a Municipality provides weather forecast

<http://www.disit.org/ServiceMap/api/v1/?serviceUri=http://www.disit.org/km4city/resource/048017&format=json>

like:

```

{"head": {"location": "FIRENZE", "vars": [ "day",
"description", "minTemp", "maxTemp", "instantDateTime"]},
"results": {
"bindings": [
{"day": {"type": "literal", "value": "Giovedì" }, "description": {"type": "literal", "value":
"pioggia e schiarite" }, "minTemp": {"type": "literal", "value": "27" }, "maxTemp": {"type":
"literal", "value": "34" }, "instantDateTime": {"type": "literal", "value": "2016-07-
28T09:02:00+01:00" }}
,
{"day": {"type": "literal", "value": "Venerdì" }, "description": {"type": "literal", "value": "poco
nuvoloso" }, "minTemp": {"type": "literal", "value": "20" }, "maxTemp": {"type":
"literal", "value": "34" }, "instantDateTime": {"type": "literal", "value": "2016-07-
28T09:02:00+01:00" }}
,
{"day": {"type": "literal", "value": "Sabato" }, "description": {"type": "literal", "value": "poco
nuvoloso" }, "minTemp": {"type": "literal", "value": "19" }, "maxTemp": {"type":
"literal", "value": "35" }, "instantDateTime": {"type": "literal", "value": "2016-07-
28T09:02:00+01:00" }}
,
{"day": {"type": "literal", "value": "Domenica" }, "description": {"type": "literal", "value":
"poco nuvoloso" }, "minTemp": {"type": "literal", "value": "" }, "maxTemp": {"type":
"literal", "value": "" }, "instantDateTime": {"type": "literal", "value": "2016-07-

```

```

28T09:02:00+01:00" }}
,
{"day": {"type": "literal", "value": "Lunedì" }, "description": { "type": "literal", "value":
"nuvoloso" }, "minTemp": {"type": "literal", "value": "" }, "maxTemp": {"type":
"literal", "value": "" }, "instantDateTime": {"type": "literal", "value": "2016-07-
28T09:02:00+01:00" }}
}}}
    
```

**Nome:** Full text search

**URL:** <http://servicemap.disit.org/WebAppGrafo/api/v1>

**Descrizione**

This API provides a way to search for keywords (separated by spaces) that should be present in any field associated with a service or any other thing that is geo-located. Optionally the search may be restricted to an area given the center with latitude and longitude as decimal degrees and the radius in km. The number of results maybe limited to a maximum.

**[http://servicemap.disit.org/WebAppGrafo/api/v1?search=...text.&\[selection=<lat>;<lng>&maxDists=dist\[&lang=it|en\]\[&format=json|html\]&maxResults=nResults](http://servicemap.disit.org/WebAppGrafo/api/v1?search=...text.&[selection=<lat>;<lng>&maxDists=dist[&lang=it|en][&format=json|html]&maxResults=nResults)**

Modalità: GET

**Parametri**

<b>search</b>	is a list of keywords to be matched in the fields associated with an entity, the keywords should be separated by space.
<b>selection</b>	(optional) if specified should be latitude;longitude, it indicates the center of the region where to find results.
<b>maxDists</b>	(optional) if selection is specified it indicates the maximum distance in Km of the entity from the point specified as selection.
<b>maxResults</b>	the maximum number of results to be returned, 0 means all results. 100 is the default value.
<b>lang</b>	(optional) is the language to be returned for the serviceLabel, it can be 'it' or 'en', if not specified 'en' is used.
<b>format</b>	the format to be used for the results it can be json or html

**Esempi**

- Search for something with words “palazzo” and “medici”  
<http://servicemap.disit.org/WebAppGrafo/api/v1/?search=palazzo%20medici&maxResults=50&format=json>  
 the results obtained are like:  

```

{
  "fullCount": 23,
  "type": "FeatureCollection",
  "features": [
    {
      "geometry":{
        "type": "Point",
        "coordinates": [11.256026,43.77523]
      },
      "type": "Feature",
    }
  ]
}
        
```

```
"properties": {
  "serviceUri":
"http://www.disit.org/km4city/resource/76933db4b76647226ddabbb62477cd12",
  "name": "PALAZZO_MEDICI_RICCARDI",
  "tipo": "servizio",
  "civic": ""
, "serviceType": "CulturalActivity_Monument_location"
, "typeLabel": "Monument location"
},
"id": 1
}, {
  "geometry":{
    "type": "Point",
    "coordinates": [11.25603,43.77524]
  },
  "type": "Feature",
  "properties": {
    "serviceUri":
"http://www.disit.org/km4city/resource/2c2219392e6063188b7217575a0b1cc4",
    "name": "PALAZZO MEDICI RICCARDI",
    "tipo": "servizio",
    "civic": ""
, "serviceType": "CulturalActivity_Museum"
, "typeLabel": "Museum"
  },
  "id": 2
}, {
  "geometry":{
    "type": "Point",
    "coordinates": [11.25591,43.774765]
  },
  "type": "Feature",
  "properties": {
    "serviceUri":
"http://www.disit.org/km4city/resource/8ac785a07dafcc2cb0a648257a5a6774",
    "name": "Palazzo Medici Riccardi",
    "tipo": "servizio",
    "civic": ""
, "serviceType": "CulturalActivity_Historical_buildings"
, "typeLabel": "Historical buildings"
  },
  "id": 3
}
...
note: The fullCount attribute indicates the total numer of results matching the query
```

<b>Nome:</b> Location Info	
<b>URL:</b> <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/location">http://servicemap.disit.org/WebAppGrafo/api/v1/location</a>	
<b>Descrizione</b>	
This API provides a way to search for address and municipality based on the GPS position. It provides as JSON the address, number, municipality name and municipality URI. The municipality URI can be used to get the meteo information using the ServiceInfo API. Moreover it is possible to get all the entities intersecting with a geometry in the provided position, for this entities the uri, class and name are provided and optionally also the whole intersecting geometry (Polygon or LineString).	
Modalità: GET	
Parametri	
position	the GPS position expressed as: <lat>;<long> (required)
intersectGeom	could be true/false/geometry, indicates to find entities with geometry intersecting with the position specified (optional, default false), if 'geometry' is provided the full geometries intersecting the point are returned
format	the format to be used for the results it can be json or html
Esempi	
<ul style="list-style-type: none"> <li>• <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/location/?position=43.797002;11.222921">http://servicemap.disit.org/WebAppGrafo/api/v1/location/?position=43.797002;11.222921</a> provides as result the nearest civic number:  <pre>{ "number": "9", "address": "VIA CARLOMAGNO",   "addressUri": "http://www.disit.org/km4city/resource/RT048017021150CV",   "municipality": "FIRENZE",   "municipalityUri": "http://www.disit.org/km4city/resource/048017" }</pre> </li> <li>• <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/location/?position=43.797002;11.222921&amp;intersectGeom=true">http://servicemap.disit.org/WebAppGrafo/api/v1/location/?position=43.797002;11.222921&amp;intersectGeom=true</a> provides the intersecting geometries:  <pre>{ "number": "9", "address": "VIA CARLOMAGNO",   "intersect": [     { "name": "ZCS_5_D", "type": "Polygon",       "uri": "http://www.disit.org/km4city/resource/956ccc79c45f34ffa5a837ad90b6e4ea",       "class": "http://www.disit.org/km4city/schema#Controlled_parking_zone" },     { "name": "percorsi_ciclabili.210", "type": "LineString",       "uri": "http://www.disit.org/km4city/resource/cea6a6e713f190c2863dddbde31c4bb9",       "class": "http://www.disit.org/km4city/schema#Cycle_paths" } ],   "addressUri": "http://www.disit.org/km4city/resource/RT048017021150CV",   "municipality": "FIRENZE",   "municipalityUri": "http://www.disit.org/km4city/resource/048017" }</pre> </li> </ul>	

<b>Nome:</b> Query TPL on an area	
<b>URL:</b> <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/tp">http://servicemap.disit.org/WebAppGrafo/api/v1/tp</a>	
<b>Descrizione</b>	
This API provides a list of transport lines passing over a geographic area, considering the bus stops present on the selected area.	
Modalità: GET	
Parametri	

selection	geographical position as <ul style="list-style-type: none"> <li>• “&lt;lat&gt;;&lt;long&gt;” for the center of the area</li> <li>• “&lt;lat1&gt;;&lt;long1&gt;;&lt;lat2&gt;;&lt;long2&gt;” for a rectangular area</li> <li>• “wkt: ...” linestring or polygon expressed as WKT text (experimental)</li> </ul>
maxDists	maximum distance expressed in Km (default 0.1) applicable only for a single point
maxResults	maximum results (default 100)
geometry	if true returns all the geometries of the routes (default false)
<b>Esempi</b>	
<ul style="list-style-type: none"> <li>• <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/?selection=43.7780;11.2668">http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/?selection=43.7780;11.2668</a> provides:                     <pre>{ "PublicTransportLine":{"type": "FeatureCollection", "features": [ { "type": "Feature", "properties": { "lineNumber": "8", "lineName": "8", "route": "553242", "routeUri": "http://www.disit.org/km4city/resource/553242", "direction": "CAREGGI DINO DEL GARBO →BAGNO A RIPOLI I PONTI", "serviceType": "PublicTransportLine" }, "id": 1 }, { "type": "Feature", "properties": { "lineNumber": "8", "lineName": "8", "route": "553241", "routeUri": "http://www.disit.org/km4city/resource/553241", "direction": "BAGNO A RIPOLI I PONTI →CAREGGI DINO DEL GARBO", "serviceType": "PublicTransportLine" }, "id": 2 } ]}}</pre> </li> </ul>	

<b>Nome:</b> Query TPL bus lines	
<b>URL:</b> <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-lines">http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-lines</a>	
<b>Descrizione</b>	
This API provides the list of bus lines	
Modalità: GET	
<b>Parametri</b>	
nessuno	
<b>Esempi</b>	
<ul style="list-style-type: none"> <li>• <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-lines">http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-lines</a> provides:                     <pre>{"BusLines": [ { "line":"S1"}, { "line":"S3"}, { "line":"SC"}, { "line":"C1"}, { "line":"C2"}, { "line":"C3"}, ... </pre> </li> </ul>	

<b>Nome:</b> Query TPL bus positions	
<b>URL:</b> <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-position">http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-position</a>	
<b>Descrizione</b>	
This API provides the list of bus position	
Modalità: GET	
<b>Parametri</b>	
nessuno	

Esempi	
<ul style="list-style-type: none"> <li> <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-position">http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-position</a> provides:                             <pre>{ "type": "FeatureCollection", "features": [ { "geometry": { "type": "Point", "coordinates": [ 11.240147, 43.779545 ] }, "type": "Feature", "properties": { "vehicleNum": "3750", "line": "17 B", "direction": "VERGA TOZZI &amp;#10132; BOITO", "tipo": "RealTimeInfo", "serviceUri": "busCode3750", "detectionTime": "8", "serviceType": "bus_real_time" }, "id": 1 }, { "geometry": { "type": "Point", "coordinates": [ 11.293703, 43.77902 ] }, "type": "Feature", "properties": { "vehicleNum": "3713", "line": "17 B", "direction": "VERGA TOZZI &amp;#10132; BOITO", "tipo": "RealTimeInfo", "serviceUri": "busCode3713", "detectionTime": "8", "serviceType": "bus_real_time" }, "id": 2 }, { "geometry": { "type": "Point", "coordinates": [ 11.278573, 43.781036 ] }, "type": "Feature", "properties": { "vehicleNum": "5301", "line": "17", "direction": "BOITO &amp;#10132; VERGA TOZZI", "tipo": "RealTimeInfo", "serviceUri": "busCode5301", "detectionTime": "8", "serviceType": "bus_real_time" }, "id": 3 }, { "geometry": { "type": "Point", "coordinates": [ 11.220392, 43.7819 ] }, "type": "Feature", "properties": { "vehicleNum": "3463", "line": "17", "direction": "KENNEDY &amp;#10132; VERGA TOZZI", "tipo": "RealTimeInfo", "serviceUri": "busCode3463", "detectionTime": "8", "serviceType": "bus_real_time" }, "id": 4 }, { "geometry": { "type": "Point", "coordinates": [ 11.253667, 43.780045 ] }, "type": "Feature", "properties": { "vehicleNum": "5341", "line": "17 C", "direction": "VERGA TOZZI &amp;#10132; KENNEDY", "tipo": "RealTimeInfo", "serviceUri": "busCode5341", "detectionTime": "13", "serviceType": "bus_real_time" }, "id": 5 }, ... ] }</pre> </li> </ul>	

Nome: Query TPL bus route of a line or bus stop	
<b>URL:</b> <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-routes">http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-routes</a>	
Descrizione	
This API provides the list of bus routes of a line or on a bus stop	
Modalità: GET	
Parametri	
line	the identifier used for the line (e.g. "11", "6", "17")
busStopName	the name of the bus stop
Esempi	
<ul style="list-style-type: none"> <li> <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-routes?line=11">http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-routes?line=11</a> provides:                             <pre>{"BusRoutes": [ { "line": "11", "route": "263884", "routeName": "11", "firstBusStop": "SALVIATINO", "lastBusStop": "LA GORA" }, {</pre> </li> </ul>	

```

"line": "11",
"route": "263886",
"routeName": "11",
"firstBusStop": "LA GORA",
"lastBusStop": "SALVIATINO"
}}

```

- <http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-routes/?busStopName=VECCHIETTI>

provides:

```

{"BusRoutes": [
{
"line": "11",
"route": "263886",
"routeName": "11",
"firstBusStop": "LA GORA",
"lastBusStop": "SALVIATINO"
},
{
"line": "6",
"route": "488613",
"routeName": "6 B",
"firstBusStop": "OSPEDALE TORRE GALLI",
"lastBusStop": "NOVELLI"
},
{
"line": "6",
"route": "488612",
"routeName": "6 A",
"firstBusStop": "OSPEDALE TORRE GALLI",
"lastBusStop": "NOVELLI"
}
]}

```

<b>Nome:</b> Query TPL bus stop of a line or route	
<b>URL:</b> <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-stops">http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-stops</a>	
<b>Descrizione</b>	
This API provides the list of bus stops of a line or a route	
Modalità: GET	
<b>Parametri</b>	
line	the identifier used for the line (e.g. "11", "6", "17")
route	the identifier of the route (e.g. 488613)
<b>Esempi</b>	
<ul style="list-style-type: none"> <li> <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-stops?line=11">http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-stops?line=11</a> </li> </ul> provides: <pre> { "type": "FeatureCollection", "features": [ { "geometry": {"type": </pre>	



```
"Point","coordinates":[11.258755,43.77866]],"type":"Feature","properties":{"popupContent":"ARAZZIERI","name":"ARAZZIERI","serviceUri":"http://www.disit.org/km4city/resource/FM0858","tipo": "fermata", "serviceType": "TransferServiceAndRenting_BusStop"},"id": 1},
{"geometry":{"type":
"Point","coordinates":[11.282586,43.784496]],"type":"Feature","properties":{"popupContent":"BALDESI","name":"BALDESI","serviceUri":"http://www.disit.org/km4city/resource/FM0493","tipo": "fermata", "serviceType": "TransferServiceAndRenting_BusStop"},"id": 2},
{"geometry":{"type":
"Point","coordinates":[11.288752,43.78387]],"type":"Feature","properties":{"popupContent":"BRONZETTI","name":"BRONZETTI","serviceUri":"http://www.disit.org/km4city/resource/FM0491","tipo": "fermata", "serviceType": "TransferServiceAndRenting_BusStop"},"id": 3},
```

...

- <http://servicemap.disit.org/WebAppGrafo/api/v1/tpl/bus-stops?route=488613>  
provides:  
{  
"Route":{  
"lineNumber":"6","lineName":"6 B","wktGeometry":"LINESTRING (11.2024974949955 43.7587277192279,....)"  
},  
"BusStops":{  
"type": "FeatureCollection", "features": [  
{"geometry":{"type":"Point","coordinates":[11.2025385,43.75867]],"type":"Feature","properties":{"popupContent": "OSPEDALE TORRE GALLI","name": "OSPEDALE TORRE GALLI","serviceUri": "http://www.disit.org/km4city/resource/FM0106","tipo": "fermata", "serviceType": "TransferServiceAndRenting\_BusStop"}, "id": 1 },  
{"geometry":{"type":"Point","coordinates":[11.205384,43.76039]],"type":"Feature","properties":{"popupContent": "OSPEDALE DON GNOCCHI","name": "OSPEDALE DON GNOCCHI","serviceUri": "http://www.disit.org/km4city/resource/FM3021","tipo": "fermata", "serviceType": "TransferServiceAndRenting\_BusStop"}, "id": 2 },  
{"geometry":{"type":"Point","coordinates":[11.206944,43.761955]],"type":"Feature","properties":{"popupContent": "DI SCANDICCI 04","name": "DI SCANDICCI 04","serviceUri": "http://www.disit.org/km4city/resource/FM0107","tipo": "fermata", "serviceType": "TransferServiceAndRenting\_BusStop"}, "id": 3 },  
{"geometry":{"type":"Point","coordinates":[11.2104845,43.764973]],"type":"Feature","properties":{"popupContent": "DI SCANDICCI 02","name": "DI SCANDICCI 02","serviceUri": "http://www.disit.org/km4city/resource/FM0108","tipo": "fermata", "serviceType": "TransferServiceAndRenting\_BusStop"}, "id": 4 },  
{"geometry":{"type":"Point","coordinates":[11.2139435,43.76578]],"type":"Feature","properties":{"popupContent": "FILARETE SCUOLE","name": "FILARETE SCUOLE","serviceUri": "http://www.disit.org/km4city/resource/FM0109","tipo": "fermata", "serviceType": "TransferServiceAndRenting\_BusStop"}, "id": 5 },  
{"geometry":{"type":"Point","coordinates":[11.217653,43.76432]],"type":"Feature","properties":{"popupContent": "FILARETE 02","name": "FILARETE 02","serviceUri": "http://www.disit.org/km4city/resource/FM0110","tipo": "fermata", "serviceType": "TransferServiceAndRenting\_BusStop"}, "id": 6 },

```

{"geometry":{"type":"Point","coordinates":[11.217652,43.76596]},"type":"Feature","properties":{"popupContent":"MATTEO DI GIOVANNI","name":"MATTEO DI GIOVANNI","serviceUri":"http://www.disit.org/km4city/resource/FM0344","tipo":"fermata","serviceType":"TransferServiceAndRenting_BusStop"},"id":7},
...
    
```

### 5.8.4 API SPARQL di Knowledge Base (API10) (UNIFI)

#### Smart City API

Profilo Tool/Algoritmo		
API SPARQL di Knowledge Base		
Nome responsabile		
Partner responsabile	UNIFI	
Descrizione Tool/algoritmo	API per accesso ai dati di della Knowledge Base tramite SPQRQL, per ottenere: XML, JSON, HTML. API con accesso condizionato in base all'utente, al profilo utente, ai dati richiesti.	
Dati primari in ingresso	Query in linguaggio SPARQL, identificatore utente o applicazione che effettua la query	
Dati prodotti in Uscita	Dati di risposta alla query in formato JSON o HTML	
Principi (metodo, basi teoriche, etc.)	API REST compatibile con protocollo SPARQL HTTP API per i dati pubblici, usa accesso autenticato per accedere ai dati privati accessibili solo ad alcune categorie di utenti	
Casi di test (presenti/assenti)	assenti	
Posizione casi di test	n/a	
Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	web app	
Single thread/Multithread	multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java JSP	
Piattaforme supportate	Java	
Posizione del codice sorgente	...	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	<a href="https://servicemap.disit.org/WebAppGrafo/sparql">https://servicemap.disit.org/WebAppGrafo/sparql</a>	
Indirizzo/i accesso via web (se presenti)		
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
KnowledgeBase		HTTP REST

<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	Descrizione	
<b>Tipo interfaccia utente</b> (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

#### 5.8.4.1 Protocollo, API: SPARQL

Il protocollo usato per effettuare query SPARQL sulla base di conoscenza sarà compatibile con il protocollo standard di accesso agli endpoint SPARQL 1.1 come specificato in <https://www.w3.org/TR/sparql11-protocol/#query-operation> (il protocollo di update non sarà accessibile dall'esterno). Per limitare l'accesso ad alcuni dati sensibili alcune parti della base di conoscenza sarà accessibile solo tramite autenticazione effettuata tramite certificato (sia lato client che lato server) quindi usando protocollo HTTPS. L'accesso a parti della base di conoscenza sarà possibile anche utilizzando connessione non autenticata ma i dati utilizzabili saranno limitati ai soli dataset senza restrizioni di uso.

#### 5.8.5 API for rendering deductions on the basis of the whole ontological model of SmartDS (API11) (UNIFI)

### Smart City API

## Profilo Tool/Algoritmo

API for rendering deductions on the basis of the whole ontological model of SmartDS (API11)

Nome responsabile

Gianni Pantaleo

Partner responsabile	UNIFI	
Descrizione Tool/algorithmo	<p>API con le quali è possibile fare query sulla knowledge base per servizi e per informazioni statistiche, ma anche sullo stato di processi decisionali dello SmartDS. Questo significa avere accesso ai dati del database del DashBoard manager, e prevedere l'implementazione di processi periodici di aggiornamento del calcolo degli alberi decisionali dello SmartDS. A tale proposito, il presente modulo sarà incaricato di eseguire le seguenti operazioni:</p> <ul style="list-style-type: none"> <li>• Lanciare processi schedulati periodici di aggiornamento del calcolo degli alberi decisionali, sulla base della variazioni dei parametri e dei dati in ingresso allo SmartDS;</li> <li>• Inviare i dati di risposta tramite chiamata REST (notifiche PUSH);</li> <li>• Rendere accessibile i dati di risposta in lettura (in PULL).</li> </ul> <p>API con accesso condizionato in base all'utente, al profilo utente, ai dati richiesti.</p>	
Dati primari in ingresso	Query SPARQL verso l'RDF Store, e richieste utente generate da processi di update di calcolo decisionale e di lettura dati dello SmartDS.	
Dati prodotti in Uscita	Dati di risposta alle query in ingresso.	
Principi (metodo, basi teoriche, etc.)	Uso di protocolli di comunicazione, chiamate REST, invio dati in modalità PUSH/PULL; uso di API REST compatibile con protocollo SPARQL 1.1;	
Casi di test (presenti/assenti)	.	
Posizione casi di test	.	
Principali problemi non risolti	.	
Principali requisiti pendenti	.	
Aspetti Tecnologici		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	.	
Linguaggio di sviluppo (java, Php, ETL)	Java	
Piattaforme supportate	Piattaforme che supportano Java	
Posizione del codice sorgente	.	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	.	
Indirizzo/i accesso via web (se presenti)	.	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
Knowledge Base RDFStore (5.11.4)	SPARQL Endpoint	SPARQL result object

	per query su RDF Store.	
Tool SmartDS (5.1.8)	MySQL connector driver.	Accesso e condivisione dati su database relazionale MySQL.
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
Formati di interscambio dati: XML, JSON		
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
HTTP REST		
SPARQL 1.1	RDF Store	
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
Knowledge Base RDFStore	Repository RDF	
SmartDS database	Database MySQL	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

Il presente modulo utilizzerà API compatibili con il protocollo SPARQL 1.1 come da specifica (<https://www.w3.org/TR/sparql11-protocol/#query-operation>) e HTTP REST.

### 5.8.6 API Query ID di Knowledge Base (API12) (UNIFI)

#### Smart City API

Profilo Tool/Algoritmo	
API Query ID di Knowledge Base	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	API per accesso ai dati di della Knowledge Base tramite Query ID, per ottenere: XML, JSON, HTML. API con accesso condizionato in base all'utente, al profilo utente, ai dati richiesti.
Dati primari in ingresso	QueryID che identifica la query da effettuare, precedentemente salvata da ServiceMap, indicazione dell'utente
Dati prodotti in Uscita	Informazioni legate alla richiesta fatta in formato

	JSON o HTML	
Principi (metodo, basi teoriche, etc.)	dato il query id vengono recuperati la API e i parametri associati e viene chiamata la relativa API	
Casi di test (presenti/assenti)	assenti	
Posizione casi di test	n/a	
Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Web app	
Single thread/Multithread	Multi thread	
Linguaggio di sviluppo (java, Php, ETL)	Java JSP	
Piattaforme supportate	Quelle supportate da Java	
Posizione del codice sorgente	n/a	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/a	
Indirizzo/i accesso via web (se presenti)	n/a	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
RDF Store		SPARQL 1.1 HTTP Protocol
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
Sesame	2.7.8	BSD-style License

--	--	--

### 5.8.6.1 Protocollo, API: Query ID sulla KB

<b>Nome:</b> Query	
<b>URL:</b> <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/">http://servicemap.disit.org/WebAppGrafo/api/v1/</a>	
<b>Descrizione</b>	
<p>This API provides results based on a query saved on ServiceMap. Different kinds of queries can be saved from the ServiceMap widget when you find the icon disk. Clicking on it, and filling a form, the system will perform two steps (1) saving the query performed in a database for your further reuse, (2) send to the user a set of links via email. Among them, two links are: one for read only (rendering only if HTML, or getting the JSON of results), the second, allows you to open again the editing web page on the ServiceMap to overwrite the previously saved query performed.</p>	
Modalità: GET	
<b>Parametri</b>	
queryId	identifier of the query saved via ServiceMap (required)
format	the format to be used for the results, it can be 'json' or 'html'
<b>Esempi</b>	
<ul style="list-style-type: none"> <li> <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/?queryId=8469888d2cf60a98f3dcae5abc2bef52&amp;format=json">http://servicemap.disit.org/WebAppGrafo/api/v1/?queryId=8469888d2cf60a98f3dcae5abc2bef52&amp;format=json</a>                      provides the services in category Accommodation or WineAndFood near to service CHIESA_DI_SANTA_MARIA_NOVELLA                      it provides as result the following:  <pre> {"Services":{   "fullCount": 96,   "type": "FeatureCollection",   "features": [     {       "geometry": {         "type": "Point",         "coordinates": [11.250053,43.773357]       },       "type": "Feature",       "properties": {         "name": "SANTA MARIA NOVELLA",         "tipo": "Hotel",         "typeLabel": "Hotel",         "serviceType": "Accommodation_Hotel",         "serviceUri":         "http://www.disit.org/km4city/resource/bf5d4aed6e9c092c2f8153861a969e32"       },       "id": 1     },     {       "geometry": { </pre> </li> </ul>	

```

        "type": "Point",
        "coordinates": [11.250053,43.773357]
    },
    "type": "Feature",
    "properties": {
        "name": "SANTA_MARIA_NOVELLA",
        "tipo": "Hotel",
        "typeLabel": "Hotel",
        "serviceType": "Accommodation_Hotel",
        "serviceUri":
"http://www.disit.org/km4city/resource/c3679b4941710ae248f367c163ea26e5"
    },
    "id": 2
},
{
    "geometry": {
        "type": "Point",
        "coordinates": [11.250053,43.773357]
    },
    "type": "Feature",
    "properties": {
        "name": "HOTEL SANTA MARIA NOVELLA S.R.L.",
        "tipo": "Hotel",
        "typeLabel": "Hotel",
        "serviceType": "Accommodation_Hotel",
        "serviceUri":
"http://www.disit.org/km4city/resource/900a0c6bf88ead66553594c7cb6f3977"
    },
    "id": 3
},
...

```

### 5.8.7 API User Crowd Sourcing (API13) (UNIFI)

#### Smart City API

<b>Profilo Tool/Algoritmo</b> API User Crowd Sourcing	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	API per fornire informazioni e dati al sistema partecipativo e comunque al Sii. Informazioni come: immagini, commenti, ranking/voti riguardo a servizi
Dati primari in ingresso	indicazione utente, immagine, commento, voto e identificatore servizio
Dati prodotti in Uscita	nessuno



Principi (metodo, basi teoriche, etc.)	Le API permettono di sottomettere valutazioni (stelle), commenti, foto dei servizi che saranno validati da un operatore (una mail viene inviata al momento della sottomissione), l'operatore può validare o scartare la sottomissione (che rimarrà comunque registrata ma non visibile)	
Casi di test (presenti/assenti)	nessuno	
Posizione casi di test	n/a	
Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	web app	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java, PHP	
Piattaforme supportate	Quelle supportate da Java e PHP	
Posizione del codice sorgente	n/a	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/a	
Indirizzo/i accesso via web (se presenti)	n/a	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
RDF Store		SPARQL 1.1 HTTP Protocol
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
SPARQL 1.1 HTTP Protocol		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
ServiceMap	informazioni aggiuntive sui servizi come valutazioni, commenti e foto	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>

Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
Sesame	2.7.8	BSD-style License

### 5.8.7.1 Protocollo esposto, API: Crowd Sourcing

Il modulo espone una interfaccia REST su protocollo HHTP per l'accesso ai dati presenti nella piattaforma di Crowd Sourcing. Nel caso che non riconosca un insieme valido di parametri restituisce HTTP response 400.

<b>Nome:</b> Tweets per channel	
<b>URL:</b> <a href="http://www.disit.org/tv/query">http://www.disit.org/tv/query</a>	
<b>Descrizione</b>	
L'API restituisce i tweet del canale selezionato in un intervallo specifico di tempo. I campi restituiti sono: message, publicationTime, twitterUser, twitterId, lang, retweetCount, favoriteCount, geo_lat, geo_long, hashtagsOnTwitter, links	
Modalità: GET	
<b>Parametri</b>	
Canale	Obligatorio
[start_date, end_date]	Se start_dat è null vengono presi tutti fino ad end_date; viceversa se end_date è null vengono presi I twee a partire da start_date
format	L'output può essere o csv o json.
<b>Esempi</b>	
Restituisce tutti I tweet relativi canale "apretoscana" a parte dal 17 dicembre 2015 fino al 19 dicembre 2015.	
<b><a href="http://disit.org/tv/query/query.php?channel=apretoscana&amp;start_date=2015-12-17&amp;end_date=2015-12-19&amp;format=json">http://disit.org/tv/query/query.php?channel=apretoscana&amp;start_date=2015-12-17&amp;end_date=2015-12-19&amp;format=json</a></b>	
<b>I risultati JSON sono:</b>	
<pre>{ "message":"Neuer Blog zum #medtech #matchmaking #event [INSIGHT] Tools, Support and Capital for Innovators ist online: https://t.co/O3403aToaD #SIP", "publicationTime":"2015-12-17 09:23:12", "twitterUser":"anitajoerg1", "twitterId":"677403660582825984", "lang":"de", "retweetCount":"0", "favoriteCount":"0", "geo_lat":"0.00000", "geo_long":"0.00000", "hashtagsOnTwitter":"#medtech #matchmaking #event #SIP",</pre>	

```

"links":"https://t.co/O3403aToaD http://bit.ly/1YY55X8"
},
"message":"Thank you @traveldavide!\n#matchmaking #love #happiness #dating #luxurylifestyl
e https://t.co/W8G2NwZTvb",
"publicationTime":"2015-12-17 09:32:54",
"twitterUser":"lvyIntMM",
"twitterId":"677406101827133440",
"lang":"en",
"retweetCount":"1",
"favoriteCount":"2",
"geo_lat":"0.00000",
"geo_long":"0.00000",
"hashtagsOnTwitter":"#matchmaking #love #happiness #dating #luxurylifestyle",
"links":"https://t.co/W8G2NwZTvb https://twitter.com/anonymagence/status/6771804
50985648128"
},
}
(I risultati sono stati troncati )
    
```

<b>Nome:</b> Tweet ID per channel	
<b>URL:</b> <a href="http://www.disit.org/tv/query">http://www.disit.org/tv/query</a>	
<b>Descrizione</b>	
L'API restituisce tutti i tweet ID dei tweets del canale selezionato in un intervallo specifico di tempo.	
Modalità: GET	
<b>Parametri</b>	
Canale	Obbligatorio
[start_date, end_date]	Se start_date è null vengono presi tutti fino ad end_date; viceversa se end_date è null vengono presi i tweet a partire da start._date
onlyID	True
<b>Esempi</b>	
Restituisce tutti i tweet ID relativi canale "apretoscana" a partire dal 17 dicembre 2015 fino al 19 dicembre 2015.	
<p><b><a href="http://disit.org/tv/query/query.php?channel=apretoscana&amp;start_date=2015-12-17&amp;end_date=2015-12-19&amp;onlyID=true">http://disit.org/tv/query/query.php?channel=apretoscana&amp;start_date=2015-12-17&amp;end_date=2015-12-19&amp;onlyID=true</a></b></p> <p><b>I risultati JSON sono:</b></p> <pre>{   { "twitterId":"677627227895431168"},</pre>	

```
{
  "twitterId":"677628560383811585"},
{
  "twitterId":"677629191706251266"},
{
  "twitterId":"677630463884836867"}
}
(These results have been truncated.)
```

<b>Nome:</b> List of active search	
<b>URL:</b> <a href="http://www.disit.org/tv/query">http://www.disit.org/tv/query</a>	
<b>Descrizione</b>	
L'API restituisce tutti le ricerca attive	
Modalità: GET	
<b>Parametri</b>	
activeSearch	activeSearch=true
<b>Esempi</b>	
L'API restituisce tutti le ricerca attive <b><a href="http://disit.org/tv/query/query.php?activeSearch=true">http://disit.org/tv/query/query.php?activeSearch=true</a></b>	
<b>I risultati JSON sono:</b>	
<pre>{   {     "ID_request":"1",     "text_req":"#meteo",     "lan_req":"it"   },   {     "ID_request":"6",     "text_req":"#ODDIT15 #Firenze",     "lan_req":"it"   },   {     "ID_request":"7",     "text_req":"#fodd",     "lan_req":"it"   },   {     "ID_request":"8",     "text_req":"#OpenDataDay #Firenze",     "lan_req":""   },   {     "ID_request":"9",     "text_req":"@flash_meteo",     "lan_req":"it"   }, }</pre>	
(I risultati sono stati troncati )	

--

<b>Nome:</b> Retweet number per tweet	
<b>URL:</b> <a href="http://www.disit.org/tv/query">http://www.disit.org/tv/query</a>	
<b>Descrizione</b>	
L'API restituisce il numero di retweet associati ad un tweet padre indicato tramite il twitterId.	
Modalità: GET	
<b>Parametri</b>	
TweetID	Obligatorio
<b>Esempi</b>	
Restituisce il numero di retweet associati a 605237373820006400 <b><a href="http://disit.org/tv/query/query.php?originalTweet=605237373820006400">http://disit.org/tv/query/query.php?originalTweet=605237373820006400</a></b> <b>I risultati JSON sono:</b> <pre>{   " MAX(retweetCount)": "9", }</pre>	

<b>Nome:</b> Tweets per search	
<b>URL:</b> <a href="http://www.disit.org/tv/query">http://www.disit.org/tv/query</a>	
<b>Descrizione</b>	
L'API restituisce i tweet della ricerca selezionata in un intervallo specifico di tempo. I campi restituiti sono: twitterId, message, publicationTime, lang, retweet, originalTweet.	
Modalità: GET	
<b>Parametri</b>	
Canale	Obligatorio
[start_date, end_date]	Se start_date è null vengono presi tutti fino ad end_date; viceversa se end_date è null vengono presi I twee a partire da start._date
format	L'output può essere o csv o json.
lang	
<b>Esempi</b>	
Restituisce tutti I tweet "italiani" relativi alla ricerca "Fedez" a parte dal 17 dicembre 2015 fino al 18 dicembre 2015.  <b><a href="http://disit.org/tv/query/query.php?search=%23fedez&amp;start_date=2015-12-17&amp;end_date=2015-12-18&amp;lang=it">http://disit.org/tv/query/query.php?search=%23fedez&amp;start_date=2015-12-17&amp;end_date=2015-12-18&amp;lang=it</a></b> <b>I risultati JSON sono:</b> <pre>{   {     "message": "Mercoled\u00ec 23 Dicembre #ChristmasParty #PinetaGarden\nSpecialGuest #FEDEZ\nPullman &amp; Prevedite da ogni zona!... https://t.co/eWM0WUWvpT",     "twitterId": "677470005567557632",     "publicationTime": "2015-12-17 13:46:50",     "lang": "it",     "originalTweet": ""   } }</pre>	

```

    "retweet": "0"
  },
  {
    "message": "Ho sempre pensato che questa canzone fosse intensa e molto forte.. Davvero, davvero bella! \n#Fedez @Fedez https://t.co/IQGwT4WRax",
    "twitterId": "677486476414869504",
    "publicationTime": "2015-12-17 14:52:17",
    "lang": "it",
    "originalTweet": "",
    "retweet": "0"
  },
  {
    "message": "Ho sempre pensato che questa canzone fosse intensa e molto forte.. Davvero, davvero bella! \n#Fedez https://t.co/IQGwT4WRax",
    "twitterId": "677488322462240768",
    "publicationTime": "2015-12-17 14:59:37",
    "lang": "it",
    "originalTweet": "",
    "retweet": "0"
  },
  },
}

```

(I risultati sono stati troncati )

<b>Nome:</b> Number of Tweets per search	
<b>URL:</b> <a href="http://www.disit.org/tv/query">http://www.disit.org/tv/query</a>	
<b>Descrizione</b>	
L'API restituisce il numero di tweet della ricerca selezionata ordinata per lingua.	
Modalità: GET	
<b>Parametri</b>	
ID_request	Obligatorio
lang	
<b>Esempi</b>	
Restituisce restituisce il numero di tweet della ricerca identificata da 101 ordinati per lingua	
<b><a href="http://disit.org/tv/query/query.php?ID_request=101">http://disit.org/tv/query/query.php?ID_request=101</a></b>	
<b>I risultati JSON sono:</b>	
<pre> {   "lang": "it",      "count": "605" },   {     "lang": "en",    "count": "130"   },   {     "lang": "und",   "count": "82"   },   {     "lang": "es",    "count": "26"   } </pre>	

```
}
}
(I risultati sono stati troncati )
```

<b>Nome:</b> First 10 hot hashtags	
<b>URL:</b> <a href="http://www.disit.org/tv/query">http://www.disit.org/tv/query</a>	
<b>Descrizione</b>	
L'API restituisce i primi 10 hashtag per numero di tweet associati al canale specificato.	
Modalità: GET	
<b>Parametri</b>	
Trends	Obligatorio
<b>Esempi</b>	
restituisce i primi 10 hashtag per numero di tweet associati al canale Firenze	
<b><a href="http://disit.org/tv/query/query.php?trends=Firenze">http://disit.org/tv/query/query.php?trends=Firenze</a></b>	
<b>I risultati JSON sono:</b>	
<pre>{   { 'value' : '#Expo2015', 'tweet_count' : 3746 },   { 'value' : '#expo2015', 'tweet_count' : 3321 },   { 'value' : '#expo2015', 'tweet_count' : 2992 },   { 'value' : '#Expo2015', 'tweet_count' : 2989 },   { 'value' : '#Expo2015', 'tweet_count' : 2180 },   { 'value' : '#noexpo', 'tweet_count' : 1891 },   { 'value' : '#Expo2015', 'tweet_count' : 1881 },   { 'value' : '#Expo2015', 'tweet_count' : 1521 },   { 'value' : '#Expo2015', 'tweet_count' : 1386 },   { 'value' : '#Expo2015Milano', 'tweet_count' : 1124 } }</pre>	

<b>Nome:</b> First 10 hot mentions	
<b>URL:</b> <a href="http://www.disit.org/tv/query">http://www.disit.org/tv/query</a>	
<b>Descrizione</b>	
L'API restituisce le prime 10 mentions per numero di tweet associate al canale specificato.	

Modalità: GET	
Parametri	
Mentions	Obligatorio
Esempi	
restituisce le prime 10 mention per numero di tweet associate al canale EXPO2015 <b>http://disit.org/tv/query/query.php?trends=EXPO2015</b> <b>I risultati JSON sono:</b> <pre> {   {     'value' : '@Expo2015Milano',      'tweet_count' : 1423   },   {     'value' : '@matteoreenzi',      'tweet_count' : 257   },   {     'value' : '@Pad_Ita2015',      'tweet_count' : 122   },   {     'value' : '@AstroSamantha',    'tweet_count' : 100   },   {     'value' : '@camilleriwrites',   'tweet_count' : 88   },   {     'value' : '@FedericaMog',      'tweet_count' : 88   },   {     'value' : '@beppeevergnini',   'tweet_count' : 77   },   {     'value' : '@EUExpo2015',       'tweet_count' : 77   },   {     'value' : '@corriereit',       'tweet_count' : 76   },   {     'value' : '@SkyTG24',          'tweet_count' : 74   }, }                 </pre>	

<b>Nome:</b> Feedback su servizio	
<b>URL:</b> <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/feedback">http://servicemap.disit.org/WebAppGrafo/api/v1/feedback</a>	
<b>Descrizione</b>	
This API provides a way to provide feedback on a service, allowing to submit stars (1-5), comments about a service	
Modalità: GET	
Parametri	
serviceUri	identifier of the service (required)
stars	number of stars associated with the service
comment	comment on the service
uid	user identifier
Esempi	

<b>Nome:</b> Invio foto di un servizio
--



<b>URL:</b> <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/photos">http://servicemap.disit.org/WebAppGrafo/api/v1/photos</a>	
<b>Descrizione</b>	
This API provides a way to provide feedback on a service, allowing to provide photos of a service (as JPEG or PNG), the photo after submission will be validated and if ok will be associated with the service.	
Modalità: POST, multipart form	
<b>Parametri</b>	
serviceUri	identifier of the service (required)
uid	user identifier (required)
file	the photo content, it should have content type image/jpeg or image/png
<b>Esempi</b>	

<b>Nome:</b> Accesso foto di un servizio	
<b>URL:</b> <a href="http://servicemap.disit.org/WebAppGrafo/api/v1/photos/&lt;idphoto&gt;.[png jpg]">http://servicemap.disit.org/WebAppGrafo/api/v1/photos/&lt;idphoto&gt;.[png jpg]</a>	
<b>Descrizione</b>	
This API provides a way to access photos of services, the <idphoto> file name indicates the photo to be retrieved. The <photoid> is accessible using the service information API.	
Modalità: GET	
<b>Parametri</b>	
nessuno	
<b>Esempi</b>	

### 5.8.8 API User Engagement (API14) (UNIFI)

#### Smart City API

Profilo Tool/Algoritmo	
API User Engagement	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	API per inviare in push (da server verso le mobile APP e web) suggerimenti e stimoli a partecipare, compiti da svolgere, etc.
Dati primari in ingresso	posizione e preferenze degli utenti, storico delle assistance e engagement fornite agli utenti
Dati prodotti in Uscita	produzione di suggerimenti di assistenza e coinvolgimento personalizzati
Principi (metodo, basi teoriche, etc.)	Il modulo produce suggerimenti date le preferenze dell'utente
Casi di test (presenti/assenti)	Corrispondenti a informazioni di assistenza per evitare multe, ridurre i tempi di parcheggio, usare lo scambiatore, consumare meno, andare in bike, etc.
Posizione casi di test	

Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	web app	
Single thread/Multithread	single thread	
Linguaggio di sviluppo (java, Php, ETL)	Java, PHP	
Piattaforme supportate	Windows, Linux, Unix	
Posizione del codice sorgente	svn	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Assistance, engagement		HTTP REST
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON		JavaScript Object Notation
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
REST HTTP		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
web		jQuery
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
spring	org.springframework 1.3.3.RELEASE	Apache License 2.0
hibernate	org.hibernate 5.1.0.Final	GNU Lesser General Public License

### 5.8.8.1 Procoltolo: User Engagement

Note di assistance o di engagement sono forniti direttamente quando l'applicazione mobile invia i dati come un evento sul Sensor Server and Manager. Il modulo di engagement, valuta in continuo l'evoluzione di stato degli utenti con i loro mobile e precalcola note di assistance e engagement che vengono mandate in risposta alla successiva chiamata di salvataggio dei dati di stato del telefono.

Questo tipo di approccio (inviare le engagement nello stesso scambio dati di salvataggio di stato) evita di avere processi multiple sul telefono che lo caricano troppo e potrebbe produrre problemi di consumo e di carico del telefono stesso, cosa per altro poco accettabile per l'utente.

### 5.8.9 API User Profiling and Recommendations (API15) (UNIFI)

#### Smart City API

Profilo Tool/Algoritmo	
API User Profiling and Recommendations	
Nome responsabile	
Partner responsabile	DISIT-UNIFI
Descrizione Tool/algoritmo	Il Recommender fornisce raccomandazioni all'utente sulla base del proprio profilo (i.e., student, commuter, tourist, citizen, all) a cui corrispondono le categorie di interesse, delle coordinate geografiche e del raggio di prossimità entro cui effettuare le raccomandazioni. È uno strumento che utilizza la Single Value Decomposition per la produzione delle raccomandazioni API per permettere alle App mobile e WEB di richiedere suggerimenti al server di user profiling and suggestion e comunicare user profile, come anche il suo menu, comportamento, etc..
Dati primari in ingresso	Azione, UserId
Dati prodotti in Uscita	Raccomandazioni in formato JSON
Principi (metodo, basi teoriche, etc.)	La rilevanza dei servizi raccomandati dipende dall'attività dell'utente e dalla prossimità ai servizi
Casi di test (presenti/assenti)	assenti
Posizione casi di test	
Principali problemi non risolti	
Principali requisiti pendenti	
Aspetti Tecnologici	
Stato (proposto/approvato)	proposto
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0%
Eseguibile/libreria/web app	Web app
Single thread/Multithread	Single thread
Linguaggio di sviluppo (java, Php, ETL)	Java, PHP
Piattaforme supportate	Linux, Mac, Windows, Unix

Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	http://www.disit.org/SmartCityRecommender/	
Indirizzo/i accesso via web (se presenti)	http://www.disit.org/SmartCityRecommender/	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
ServiceMap	Weather, EventS, Services	HTTP REST
RDF Store	SPARQL Query	HTTP REST
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
JSON	ServiceMap	
JSON	RDF Store	
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
HTTP REST	ServiceMap	
HTTP REST	RDF Store	
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
AccessLog	Log degli accessi ai servizi da parte degli utenti presi dal ServiceMap	
Recommender	Dati del raccomandatore	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
web	PHP	HighCharts
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
Apache Mahout	0.11.0	Apache License, Version 2.0

### 5.8.9.1 Protocollo esposto: Recommendations

Il modulo espone una interfaccia REST su protocollo HTTP per l'accesso alle raccomandazioni per l'utente.

<b>Nome:</b> Recommend
<b>URL:</b> http://disit.org/SmartCityRecommender/
<b>Descrizione</b>
API per richiedere raccomandazioni per un utente.
Modalità: GET, POST
<b>Parametri Obbligatori</b>

action	l'azione richiesta (in questo caso <i>recommend</i> )
user	l'hash dell'id utente calcolato dalla mobile app
profile	il profilo dell'utente (i.e., student, commuter, tourist, citizen, all)
language	la lingua dell'utente (i.e., en, it)
latitude	la latitudine in formato decimale
longitude	la longitudine in formato decimale
distance	Il raggio di ricerca in km
<b>Parametri Opzionali</b>	
mode	la modalità delle coordinate (i.e., gps, manual)
version	la versione dell'app mobile; questo campo è necessario se si vogliono ottenere tweet nei suggerimenti
aroundme	Specifica se si vogliono ricevere i suggerimenti più vicini (e.g., true) L'effetto di aroundme = true è quello di impostare svd = false e accettare suggerimenti già suggeriti, con il risultato di ottenere sempre le raccomandazioni dei servizi più vicini. Il raccomandatore svd = true (default) fa sì che l'algoritmo scelga di volta in volta quali categorie utilizzare per costruire la query SPARQL, e quindi i risultati possono variare (possono essere forniti prima raccomandazioni di servizi lontani, e successivamente più vicini).
svd	se utilizzare il raccomandatore svd (i.e., true)
<b>Esempio</b>	
<p><a href="http://disit.org/SmartCityRecommender/?action=recommend&amp;user=3043b85d23d6f4879e1765c2c2e431cbc71d393065af06b03486ba4a04642b5b&amp;profile=student&amp;language=en&amp;latitude=43.7727&amp;longitude=11.2532&amp;distance=1">http://disit.org/SmartCityRecommender/?action=recommend&amp;user=3043b85d23d6f4879e1765c2c2e431cbc71d393065af06b03486ba4a04642b5b&amp;profile=student&amp;language=en&amp;latitude=43.7727&amp;longitude=11.2532&amp;distance=1</a></p> <p><b>JSON Result:</b></p> <pre>[{"suggestions":[{"Service":{"features":[{"geometry":{"coordinates":[11.2544,43.7732],"type":"Point"},"id":1,"type":"Feature","properties":{"serviceType":"ShoppingAndService_Pharmacy","note":"farmacie.100","website":"","address":"PIAZZA DI SAN GIOVANNI","comments":[],"city":"FIRENZE","serviceUri":"http://www.disit.org/km4city/resource/50b6312db054b3dab55edf1c0aac0004","description":"","description2":"","photos":[],"linkDBpedia":[],"civic":"17R","multimedia":"","cap":"50144","avgStars":0.0,"province":"FI","starsCount":0,"phone":"","name":"Farmacia S. ANTONINO","typeLabel":"Pharmacy","fax":"","email":""}]}],"type":"FeatureCollection"}],{"Service":{"features":[{"geometry":{"coordinates":[11.2535,43.7716],"type":"Point"},"id":1,"type":"Feature","properties":{"serviceType":"ShoppingAndService_Pharmacy","note":"farmacie.94","website":"","address":"PIAZZA DELLA REPUBBLICA","comments":[],"city":"FIRENZE","serviceUri":"http://www.disit.org/km4city/resource/4846b073f8032213bf0630d747e93272","description":"","description2":"","photos":[],"linkDBpedia":[],"civic":"23R","multimedia":"","cap":"50123","avgStars":0.0,"province":"FI","starsCount":0,"phone":"","name":"Farmacia INTERNAZIONALE","typeLabel":"Pharmacy","fax":"","email":""}]}],"type":"FeatureCollection"}],{"Service":{"features":[{"geometry":{"coordinates":[11.2546,43.7735],"type":"Point"},"id":1,"type":"Feature","properties":{"serviceType":"ShoppingAndService_Pharmacy","note":"farmacie.49","website":"","address":"PIAZZA DI SAN GIOVANNI","comments":[],"city":"FIRENZE","serviceUri":"http://www.disit.org/km4city/resource/d71a90da83e4a9acdc40e58fe947d6cd","description":"","description2":"","photos":[],"link</pre>	

```

DBpedia":[],"civic":"20R","multimedia":"","cap":"50129","avgStars":0.0,"province":"FI","starsCount":0,"phone":"","name":"Farmacia ALL INSEGNA DEL
MORO","typeLabel":"Pharmacy","fax":"","email":"","},"type":"FeatureCollection"}},{"label":"Things to do","priority":1,"group":"Things to do"},{"suggestions":[{"Service":{"features":[{"geometry":{"coordinates":[11.2532,43.7732],"type":"Point"},"id":1,"type":"Feature","properties":{"serviceType":"WineAndFood_Restaurant","note":"","website":"","address":"VICOLO DI SANTA MARIA
MAGGIORE"},"comments":[],"city":"FIRENZE","serviceUri":"http://www.disit.org/km4city/resource/c4f1ebc8d44d373823173ffeb3695214","description":"IMPRESA INDIVIDUALE –
Individuali o assimilabili o non iscritti al
RI","description2":"","photos":[],"linkDBpedia":[],"civic":"1","multimedia":"","cap":"50123","avgStars":0.0,"province":"FI","starsCount":0,"phone":"","name":"BERNACCHIONI
SIMONE","typeLabel":"Restaurant","fax":"","email":"","},"type":"FeatureCollection"}},{"Service":{"features":[{"geometry":{"coordinates":[11.2535,43.772],"type":"Point"},"id":1,"type":"Feature","properties":{"serviceType":"WineAndFood_Restaurant","note":"","website":"","address":"PIAZZA DELLA
REPUBBLICA"},"comments":[],"city":"FIRENZE","serviceUri":"http://www.disit.org/km4city/resource/87486751af941d9a1d5be5297f90e3fd","description":"SOCIETA' A
RESPONSABILITA' LIMITATA – Società di
capitale"},"description2":"","photos":[],"linkDBpedia":[],"civic":"5","multimedia":"","cap":"50100","avgStars":0.0,"province":"FI","starsCount":0,"phone":"","name":"PERSEUS FIESOLE
SRL","typeLabel":"Restaurant","fax":"","email":"","},"type":"FeatureCollection"}},{"Service":{"features":[{"geometry":{"coordinates":[11.2522,43.7716],"type":"Point"},"id":1,"type":"Feature","properties":{"serviceType":"WineAndFood_Restaurant","note":"","website":"","address":"VIA DEI
PESCIANI"},"comments":[],"city":"FIRENZE","serviceUri":"http://www.disit.org/km4city/resource/e6a5976759e6ade6714ff0ce8de29876","description":"","description2":"","photos":[],"linkDBpedia":[],"civic":"2","multimedia":"","cap":"50100","avgStars":0.0,"province":"FI","starsCount":0,"phone":"05526651","name":"Hotel Helvetia & Bristol e Ristorante Hosteria
Bibendum"},"typeLabel":"Restaurant","fax":"0552399897","email":"information.hbf@royaldemeure.com"}},{"type":"FeatureCollection"}},{"label":"Wine and Food","priority":2,"group":"Wine and Food"},
...
}]
(These results have been truncated)
    
```

aroundme	alreadyRecommended	svd	action
true	true	true	restituisce sempre e soltanto i suggerimenti più vicini (alreadyRecommended e svd sono ignorati)
true	true	false/null	restituisce sempre e soltanto i suggerimenti più vicini (alreadyRecommended e svd sono ignorati)
true	false/null	true	restituisce sempre e soltanto i suggerimenti più vicini (alreadyRecommended e svd sono ignorati)
true	false/null	false/null	restituisce sempre e soltanto i suggerimenti più vicini (alreadyRecommended e svd sono ignorati)
false/null	true	true/null	può restituire i suggerimenti già suggeriti negli n giorni precedenti (attualmente n=7), ma non necessariamente quelli più vicini
false/null	true	false	restituisce sempre e soltanto i suggerimenti più vicini
false/null	false/null	true/null	restituisce soltanto i suggerimenti non suggeriti negli n giorni precedenti (attualmente n=7), ma non necessariamente quelli più vicini
false/null	false/null	false	restituisce soltanto i suggerimenti più vicini fra quelli non suggeriti negli n giorni precedenti (attualmente n=7)

<b>Nome:</b> Recommend for Group	
<b>URL:</b> <a href="http://disit.org/SmartCityRecommender/">http://disit.org/SmartCityRecommender/</a>	
<b>Descrizione</b>	
API per richiedere raccomandazioni per un utente per un gruppo.	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
action	l'azione richiesta (in questo caso <i>recommendForGroup</i> )
user	l'hash dell'id utente calcolato dalla mobile app
profile	il profilo dell'utente (i.e., student, commuter, tourist, citizen, all)
group	il nome del gruppo
language	la lingua dell'utente (i.e., en, it)
latitude	la latitudine in formato decimale
longitude	la longitudine in formato decimale
distance	il raggio di ricerca in km
<b>Parametri Opzionali</b>	
mode	la modalità delle coordinate (i.e., gps, manual)
version	la versione dell'app mobile; questo campo è necessario se si vogliono ottenere tweet nei suggerimenti
svd	se utilizzare il raccomandatore svd (i.e., true)
<b>Esempio</b>	
Search services in the category of Accommodation and Bus Stops around a point, giving the coordinates Latitude and Longitude, within 200m for services and 100m for bus stops.	
<p><b><a href="http://192.168.0.207:8080/SmartCityRecommender/?action=recommendForGroup&amp;user=3043b85d23d6f4879e1765c2c2e431cbc71d393065af06b03486ba4a04642b5b&amp;profile=student&amp;group=Bus&amp;language=en&amp;latitude=43.7727&amp;longitude=11.2532&amp;distance=1">http://192.168.0.207:8080/SmartCityRecommender/?action=recommendForGroup&amp;user=3043b85d23d6f4879e1765c2c2e431cbc71d393065af06b03486ba4a04642b5b&amp;profile=student&amp;group=Bus&amp;language=en&amp;latitude=43.7727&amp;longitude=11.2532&amp;distance=1</a></b></p>	
<b>JSON Result:</b>	
<pre>[{"suggestions":[{"realtime":{},"busLines":{"head":{"busStop":"PECORI","vars":"busLine"},"results":{"bindings":[{"busLine":{"type":"literal","value":"C1"}},{"busLine":{"type":"literal","value":"C2"}]}]},"BusStop":{"features":{"geometry":{"coordinates":[11.2533,43.7727],"type":"Point"},"id":1,"type":"Feature","properties":{"serviceType":"TransferServiceAndRenting_BusStop","address":"","avgStars":0.0,"comments":[],"starsCount":0,"serviceUri":"http://www.disit.org/km4city/resource/FM0758","name":"PECORI","typeLabel":"BusStop","photos":[]}}},"type":"FeatureCollection"}],"realtime":{},"busLines":{"head":{"busStop":"PECORI","vars":"busLine"},"results":{"bindings":[{"busLine":{"type":"literal","value":"C1"}},{"busLine":{"type":"literal","value":"C2"}]}]},"BusStop":{"features":{"geometry":{"coordinates":[11.2533,43.7727],"type":"Point"},"id":1,"type":"Feature","properties":{"serviceType":"TransferServiceAndRenting_BusStop","address":"","avgStars":0.0,"comments":[],"starsCount":0,"serviceUri":"http://www.disit.org/km4city/resource/FM0758","name":"PECORI","typeLabel":"BusStop","photos":[]}}},"type":"FeatureCollection"}],"realtime":{},"busLines":{"head":{"busStop":"VECCHIETTI","vars":"busLine"},"results":{"bindings":[{"busLine":{"type":"literal","value":"11"}},{"busLine":{"type":"literal","value":"6"}]}]},"BusStop":{"features":{"geometry":{"coordinates":[11.2528,43.7725],"type":"Point"},"id":1,"type":"Feature","properties":{"serviceType":"TransferServiceAndRenting_BusStop","address":"VIA DEI VECCHIETTI","avgStars":0.0,"comments":[],"starsCount":0,"serviceUri":"http://www.disit.org/km4city/resource/FM3004","name":"VECCHIETTI","typeLabel":"BusStop","photos":[]}}},"type":"FeatureCollection"}]},"label":"Bus","priority":2,"group":"Bus"}]</pre>	

<b>Nome:</b> Dislike	
<b>URL:</b> <a href="http://disit.org/SmartCityRecommender/">http://disit.org/SmartCityRecommender/</a>	
<b>Descrizione</b>	
API per mettere il dislike per un gruppo.	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
action	l'azione richiesta (in questo caso <i>dislike</i> )
user	l'hash dell'id utente calcolato dalla mobile app
group	il nome del gruppo da rimuovere dalle raccomandazioni
<b>Esempio</b>	
<p><b><a href="http://disit.org/SmartCityRecommender/?action=dislike&amp;user=3043b85d23d6f4879e1765c2c2e431cbc71d393065af06b03486ba4a04642b5b&amp;group=Hotel">http://disit.org/SmartCityRecommender/?action=dislike&amp;user=3043b85d23d6f4879e1765c2c2e431cbc71d393065af06b03486ba4a04642b5b&amp;group=Hotel</a></b></p> <p><b>Result:</b> true, false</p>	

<b>Nome:</b> Remove Dislike	
<b>URL:</b> <a href="http://disit.org/SmartCityRecommender/">http://disit.org/SmartCityRecommender/</a>	
<b>Descrizione</b>	
API per rimuovere il dislike per tutti i gruppi.	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
action	l'azione richiesta (in questo caso <i>removeDislike</i> )
user	l'hash dell'id utente calcolato dalla mobile app
<b>Esempio</b>	
<p><b><a href="http://disit.org/SmartCityRecommender/?action=removeDislike&amp;user=3043b85d23d6f4879e1765c2c2e431cbc71d393065af06b03486ba4a04642b5b">http://disit.org/SmartCityRecommender/?action=removeDislike&amp;user=3043b85d23d6f4879e1765c2c2e431cbc71d393065af06b03486ba4a04642b5b</a></b></p> <p><b>Result:</b> true, false</p>	

<b>Nome:</b> Log Tweet	
<b>URL:</b> <a href="http://disit.org/SmartCityRecommender/">http://disit.org/SmartCityRecommender/</a>	
<b>Descrizione</b>	
API per fare il log di un tweet visualizzato sul database del recommender	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
action	l'azione richiesta (in questo caso <i>logViewedTweet</i> )
user	l'hash dell'id utente calcolato dalla mobile app
twitterId	l'id del tweet
<b>Esempio</b>	



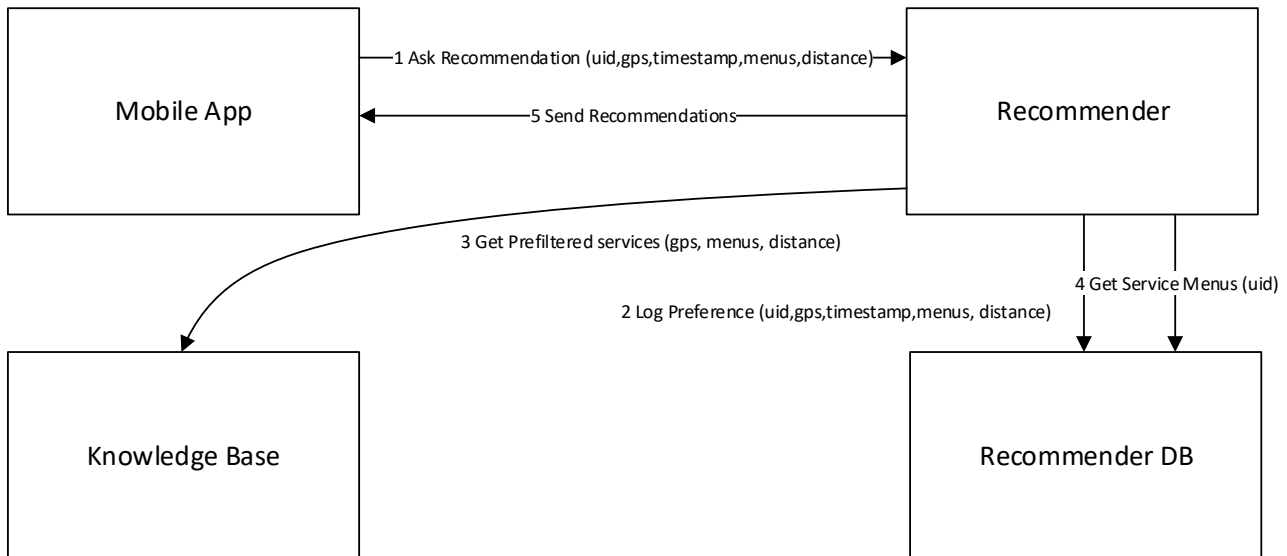
<p><b>http://disit.org/SmartCityRecommender/?action=logViewedTweet&amp;user=3043b85d23d6f4879e1765c2c2e431cbc71d393065af06b03486ba4a04642b5b&amp;twitterId=694606765888016385&amp;group=Twitter1</b></p> <p><b>Result:</b> true, false</p>
--

<b>Nome:</b> Assess	
<b>URL:</b> <a href="http://disit.org/SmartCityRecommender/">http://disit.org/SmartCityRecommender/</a>	
<b>Descrizione</b>	
API per fare l'assessment di un servizio	
Modalità: GET, POST	
<b>Parametri Obbligatoriosi</b>	
action	l'azione richiesta (in questo caso <i>assess</i> )
uid	l'hash dell'id utente calcolato dalla mobile app
serviceUri o genID	l'id del servizio da valutare
vote	il voto (1-5)
suggType	il tipo di assess (assis, recom, engage)
<b>Esempio</b>	
<p><a href="http://disit.org/SmartCityRecommender/?action=assess&amp;uid=3043b85d23d6f4879e1765c2c2e431cbc71d393065af06b03486ba4a04642b5b&amp;serviceUri=prova&amp;vote=1&amp;suggType=recom">http://disit.org/SmartCityRecommender/?action=assess&amp;uid=3043b85d23d6f4879e1765c2c2e431cbc71d393065af06b03486ba4a04642b5b&amp;serviceUri=prova&amp;vote=1&amp;suggType=recom</a></p>	
<b>Result:</b> true	

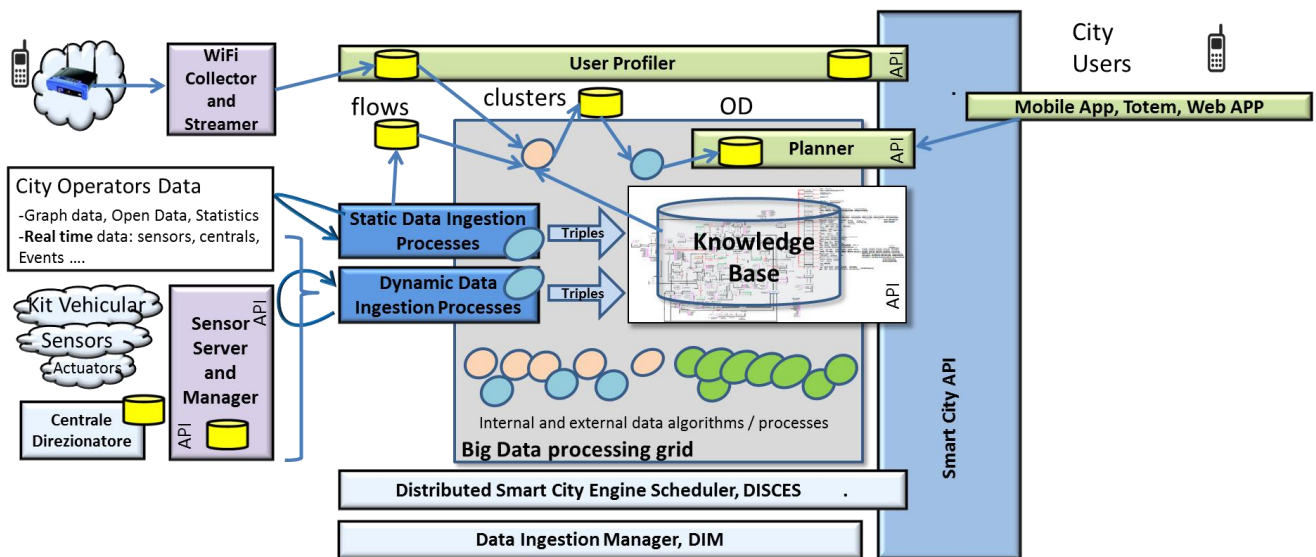
- 1) Il dispositivo mobile invia al recommender una notifica di presenza, nel caso in cui l'utente si sia spostato almeno di N/2 metri (con N pari alla distanza impostata sul dispositivo), oppure quando viene cliccata una raccomandazione, con i seguenti parametri:
  - a. userID;
  - b. gps location: latitude, longitude;
  - c. timestamp;
  - d. menus, le categorie impostate sul dispositivo mobile, se le utente le ha personalizzate, oppure il profilo al quale l'utente appartiene (e.g., studente);
- 2) Il recommender scrive sul database MySQL l'evento registrato inviatogli dal dispositivo mobile come al punto 1;
- 3) Il recommender fa una query sull'RDF store per chiedere la lista dei servizi, considerando l'orario di apertura dei servizi e con i vincoli definiti dall'utente:
  - a. gps location: latitude, longitude;
  - b. timestamp;
  - c. menus, le categorie impostate sul dispositivo mobile, se l'utente le ha personalizzate, oppure quelle associate al profilo al quale l'utente appartiene (e.g., studente);
  - d. categorie dei servizi (Accommodation, CulturalActivity, Education, Emergency, Entertainment, EnvironmentAndAgriculture, FinancialService, GovernmentOffice, HealthCare, Shopping, TourismService, TransferService, WineAndFood ecc.), servizi trasversali (DL, LF, BUS, Meteo, Eventi in città);

4) Chiede con una query su database la lista dei menu associati al profilo utente;

Invia le raccomandazioni al dispositivo mobile secondo le regole definite nelle impostazioni del recommender e i vincoli trovati con i punti 3 e 4.



### 5.8.10 API request planning (TPL e Logistic) (API16) (MIZAR)



<b>Profilo Tool/Algoritmo</b>	
API request planning (API16)	
Nome responsabile	
Partner responsabile	MIZAR
Descrizione Tool/algoritmo	API per richiedere la pianificazione del percorso fra due punti (o sequenza di punti) tramite: mezzi pubblici, auto, e a piedi. API con accesso condizionato in base all'utente, al profilo utente, ai dati richiesti.
Dati primari in ingresso	Punto di inizio e punto di fine del percorso

Dati prodotti in Uscita	Lista di archi del percorso	
Principi (metodo, basi teoriche, etc.)	Dopo aver ricevuto i dati in ingresso, il sistema calcola il percorso migliore e restituisce la lista degli archi da percorrere.	
Casi di test (presenti/assenti)	nessuno	
Posizione casi di test	n.a	
Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Web API	
Single thread/Multithread	Single thread	
Linguaggio di sviluppo (java, Php, ETL)	PHP	
Piattaforme supportate	Linux	
Posizione del codice sorgente	Repository di progetto	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n.a.	n.a.	SAOP
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
Data Objects	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
SOAP	n.a.	n.a.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
n.a.	n.a.	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
nessuna	n.a.	n.a.
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

n.a.	n.a.	

### 5.8.10.1 *Procoltolo SII-PathPlanner, API: PathPlannerRequest*

Per inviare una richiesta al servizio di calcolo percorso, è necessario inviare una richiesta autenticata da un token con la seguente chiamata:

```
http://domain/rest/v1/objects/supportedtypes?token={token};startPoint={WGS84};stopPoint={WGS84};transportationMode=PrivateVehicle
```

Il parametro transportationMode è di default PublicTransport; sono ammessi PrivateVehicle e Pedestrian.

La risposta ottenuta sarà una lista di archi di OpenStreetMap, in formato json, del tipo:

```
[
  {
    "ArcCode": "string",
    "Description": "string",
    "StartGeoInfo": {
      "Latitude": 0,
      "Longitude": 0
    },
    "StopGeoInfo": {
      "Latitude": 0,
      "Longitude": 0
    }
  },
  {
    "ArcCode": "string",
    "Description": "string",
    "StartGeoInfo": {
      "Latitude": 0,
      "Longitude": 0
    },
    "StopGeoInfo": {
      "Latitude": 0,
      "Longitude": 0
    }
  }
]
```

## 5.8.11 API Pubblicazione Dati Supervisore (API17) (MIZAR)

### Smart City API

#### Profilo Tool/Algoritmo

API request planning

Nome responsabile	
Partner responsabile	MIZAR
Descrizione Tool/algoritmo	API per accedere a dati storici per PA e SME. API con accesso condizionato in base all'utente, al

	profilo utente, ai dati richiesti per implementazione di nuovi scenari B2B o B2C	
Dati primari in ingresso	Selezione di dati dalla piattaforma SII	
Dati prodotti in Uscita	Dati in formato OpenData	
Principi (metodo, basi teoriche, etc.)	L'API espone i dati del supervisore mettendoli a disposizione della piattaforma SII. In particolare i dati esposti includono lo storico di Livelli di Servizio, O/D, volumi e conteggi dei sensori, trasporto pubblico, eventi (esposti in formato DENM), previsioni semaforiche (esposte in formato SPaT)	
Casi di test (presenti/assenti)	Nessuno	
Posizione casi di test	n.a.	
Principali problemi non risolti	n.a.	
Principali requisiti pendenti	n.a.	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non Implementato	
Stato implementazione, percentuale	50%	
Eseguitabile/libreria/web app	Web app	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	C#	
Piattaforme supportate	Windows	
Posizione del codice sorgente	Repository di progetto	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	<a href="http://IPserver/rest/api/v1">http://IPserver/rest/api/v1</a> (per misure e dati) <a href="http://IPserver/api/vehicleservice/denm">http://IPserver/api/vehicleservice/denm</a> (per DENM) <a href="http://IPserver/api/vehicleservice/spat">http://IPserver/api/vehicleservice/spat</a> (per spat)	
Indirizzo/i accesso via web (se presenti)	Non applicabile	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n.a.	REST API	HTTP
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
REST API	n.a.	n.a.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
n.a.	n.a.	

<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
n.a.	n.a.	n.a.
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
REST API	n.a.	Proprietaria, ad uso gratuito per il progetto SII

### 5.8.11.1 Protocollo Supervisore-SII, API17 - (MIZAR)

Per ottenere le misure dal supervisore, è prima necessario ottenere l'elenco degli oggetti presenti con la seguente chiamata:

```
http://domain/rest/v1/objects/supportedtypes?token={token}
```

```
[
  {
    "Type": "string",
    "Uri": "string"
  }
]
```

In seguito si deve effettuare una chiamata per ottenere la lista degli oggetti di una determinata tipologia

```
http://domain/rest/v1/objecttypes?token={token}
```

```
[
  {
    "GUID": "string",
    "Code": "string",
    "Description": "string",
    "Type": "string",
    "Uri": "string",
    "Status": "string",
    "GeoInfo": {
      "Projection": "string",
      "Latitude": 0,
      "Longitude": 0,
      "Direction": 0
    },
    "LastUpdateTimeUTC": "string"
  }
]
```

Per ottenere le misure di un determinato oggetto è infine necessario invocare il seguente URL:

```
http://domain/rest/v1/objecttypes/{GUID}/measures?token={token}&fromDate={datefrom}&toDate={dateto}
```

```
{
  "Measures": [
    {
      "MeasureTimeUTC": "2016-06-08T15:45:00Z",
      "Sensors": [
        {
          "Data": [
            {
              "Group": "string",
              "Key": "string",
              "Value": "string"
            }
          ]
        }
      ]
    }
  ]
}
```

Per ottenere invece gli eventi in formato DENM è necessario invocare il seguente URL:

```
http://serveraddress/api/vehicleservice/denms/?applicationkey=21372328277482247&latitude=20.233333&longitude=20.4444555&altitude=240&heading=3.3&speed=50.4
```

ottenendo la seguente risposta

```
<ItsPduMessageList xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ItsPduMessage>
    <Base64Binary>
      AQEAAAABgAAAAACAAInZ4k10AnagOFY7Wk6ShrSdIWAAAAAAGGoMAAAEBA==
    </Base64Binary>
  </ItsPduMessage>
  <ItsPduMessage>
    <Base64Binary>
      AQEAAAABgAAAAACAAQnafzN4AnagOFY7Wk6SlrSdIXAAAAAAGGoMAAAEBA==
    </Base64Binary>
  </ItsPduMessage>
  <ItsPduMessage>
    <Base64Binary>
      AQEAAAABgAAAAACAAynafzN4AnagOFY7Wk6SlrSdIKAAAAAAGGoMAAAEBA==
    </Base64Binary>
  </ItsPduMessage>
  <ItsPduMessage>
    <Base64Binary>
```

```
AQEAAAABgAAAAACAagnafzN4AnagOFY7Wk6SxrSdILAAAAAGGoMAAAEBA==  
</Base64Binary>  
</ItsPduMessage>  
</ItsPduMessageList>
```

Il messaggio DENM è codificato Base64Binary con codifica PER unaligned.

Per ottenere invece gli le previsioni semaforiche in formato SPaT:

```
http://serveraddress/api/vehicleservice/spats?intersectionid=10001;10002?token=25dfweaf4566
```

ottenendo in risposta:

```
<PredictionList xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSc  
hema-instance">  
<PredictionCollection>  
  <Prediction>  
    <IntersectionId>10001</IntersectionId>  
    <TimeStamp>2014-05-20T11:09:14.6+02:00</TimeStamp>  
    <Value>  
      <States>  
        <MovementState>  
          <LaneNumbers>  
            <LaneNumber>1</LaneNumber>  
          </LaneNumbers>  
        <CurrentState>  
          <NumberOfBits>4</NumberOfBits>  
          <TrimZeroBits>>false</TrimZeroBits>  
          <State>0010xxxx</State>  
        </CurrentState>  
        <NextChanges>  
          <Change>  
            <MinTimeToChange>0</MinTimeToChange>  
            <MaxTimeToChange>0</MaxTimeToChange>  
            <LikelyTimeToChange>36</LikelyTimeToChange>  
            <Confidence>0</Confidence>  
            <PassState>>false</PassState>  
            <PredCnt>0</PredCnt>  
          </Change>  
        </States>  
      </Value>
```



```
</Prediction>
<Prediction>
  <IntersectionId>10002</IntersectionId>
  <TimeStamp>2014-05-20T11:09:14.6+02:00</TimeStamp>
  <Value>
    <States>
      <MovementState>
        <LaneNumbers>
          <LaneNumber>1</LaneNumber>
        </LaneNumbers>
      <CurrentState>
        <NumberOfBits>4</NumberOfBits>
        <TrimZeroBits>>false</TrimZeroBits>
        <State>0010xxxx</State>
      </CurrentState>
      <NextChanges>
        <Change>
          <MinTimeToChange>0</MinTimeToChange>
          <MaxTimeToChange>0</MaxTimeToChange>
          <LikelyTimeToChange>36</LikelyTimeToChange>
          <Confidence>0</Confidence>
          <PassState>>false</PassState>
          <PredCnt>0</PredCnt>
        </Change>
      </NextChanges>
    </States>
  </Value>
</Prediction>
</PredictionCollection>
```

## 5.8.12 API verso altre smart city (API18) (UNIFI)

**Smart City API**

**Other Smart Cities.....**

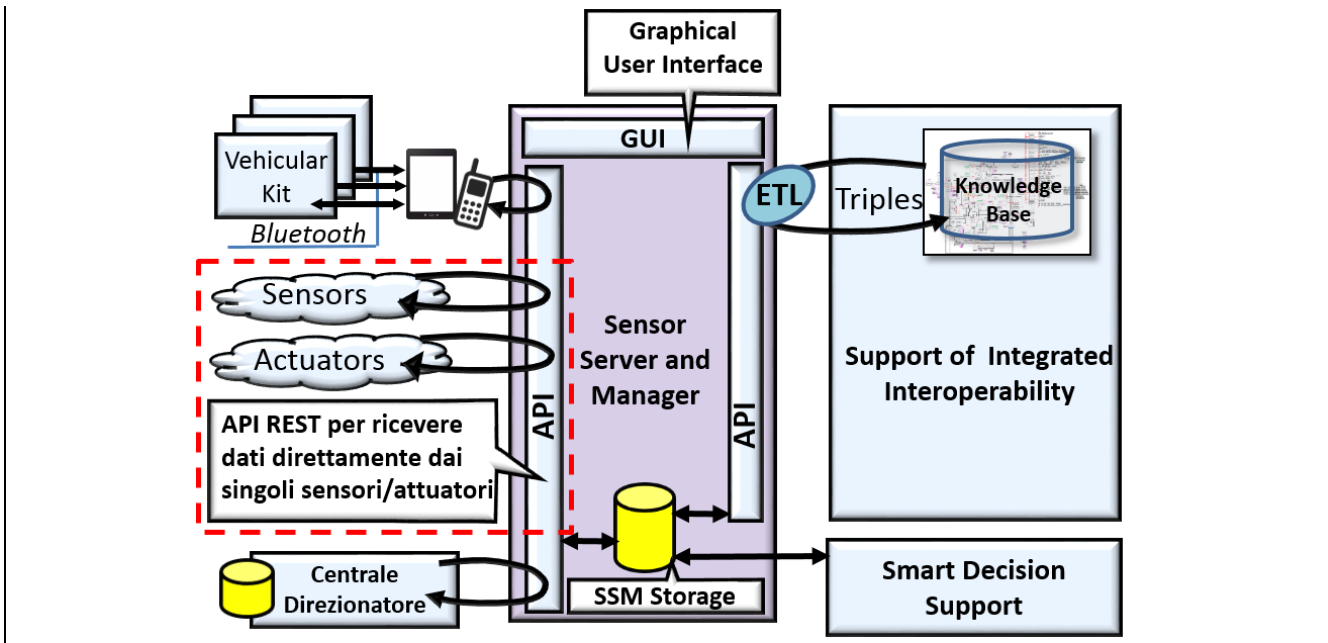
Le API esportate verso altre Smart City sono una visione ridotta rispetto alle Smart City API complete del sistema.

## 5.9 Specifica: Acquisizione dati da sensori

		ECM	Mizar	UNIFI	IN20	TIME	Negentis	Fffknow	liberologico	ataf	busitalia	cttnord	tiemme	arros	elfi	calamai	Midra	Proiert	GEOIN	OUESTIT	SOFTEC	FWINGS
4.2.4	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: Sensori, diretti e centrali		X	X													X					
4.2.8	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: kit privati						X										X					

Principali Sottosistemi interessati dai requisiti da descrivere in questa Subsubsection, saranno i sottosistemi primari della specifica					
	Sotto Attività	Partner coord	Sottosistema	Tool/Level	Descrizione sommaria
4.2.4	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: Sensori, diretti e centrali	UNIFI.D ISIT	Sensori diretti	Sensori, diretti e centrali, modulo salvataggio dati, rx comandi (MAPP02)	Statici e dinamici, sono i sensori che inviano dati verso il Sensor Server and Manager. Gli stessi sensori devono poter ricevere comandi in certi casi.
		UNIFI	Sensori da mobile app	Mobile APP con Wi-Fi e BT, ricezione sensors (API19)	Acquisizione Wi-Fi e BT via mobile app. Sono le Mobile App che fanno chiamate REST per comunicare lo stato dei loro sensori
4.2.8	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: kit privati	NEGEN TIS	Sensori kit	kit privati, modulo salvataggio dati, rx comandi (MAPP03)	Statici e dinamici, sono i Kit Veicolari che inviano dati verso il Sensor Server and Manager. Gli stessi sensori devono poter ricevere comandi, in certi casi

### 5.9.1 Acquisizione dati da Sensori ed invio a Sensor Server e Manager con call REST, Rx Comandi (MAPP02) (UNIFI.DISIT)



## Profilo Tool/Algoritmo

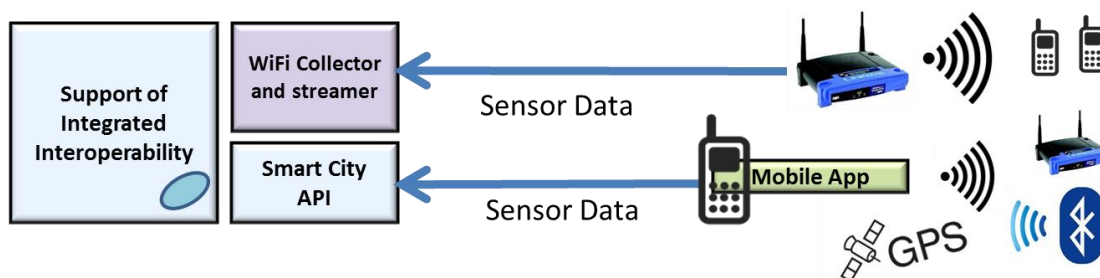
Acquisizione dati da Sensori, invio diretto a centrali Sensor Server and Manager (MAPP02)

Nome responsabile	
Partner responsabile	UNIFI.DISIT
Descrizione Tool/algoritmo	<p>Acquisizione di dati statici e dinamici provenienti direttamente da sensori e attuatori attraverso API REST o tramite Comandi Rx. Queste API coincidono con le API che il Sensor Server and Manager espone nelle smart city API per permettere ai singoli sensori e attuatori di registrare eventi e stato dei sensori/attuatori tramite chiamate REST. Sono inviati dai sensosiri/attuatori verso il Sensor Server and Manager eventi contenuti le seguenti informazioni:</p> <ul style="list-style-type: none"> <li>• ID del sensore/attuatore</li> <li>• Date and time</li> <li>• Tipo di sensore/attuatore</li> <li>• Coordinate GPS e altre informazioni</li> <li>• Valori dei sensori riscontrati che dipendono dal tipo di sensore (temperatura, CANBUS, umidità, numero passeggeri, stato del mezzo, etc.)</li> <li>• Etc.</li> </ul> <p>Le API corrispondono a quelle usate anche dal blocco 'Centrale Direzionatore' (descritte in dettaglio nel paragrafo 5.11.2)</p>
Dati primari in ingresso	Acquisizione di dati provenienti da sensori e/o attuatori tramite API REST.

Dati prodotti in Uscita	Il modulo scrive i dati su Database, inoltre fornisce delle API REST che consentono di esporre i dati dei kit JSON per la conseguente acquisizione verso la base di conoscenza Sii-Mobility, tramite un processo ETL (uno per ciascun tipo di sensore e/o mobile App).	
Principi (metodo, basi teoriche, etc.)	Realizzazione di: API REST, ETL	
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	n/d	
Principali problemi non risolti	Assenti	
Principali requisiti pendenti	Nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0.0%	
Eseguibile/libreria/web app	API REST (metodi http: POST, formato di comunicazione dati: JSON)	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Php	
Piattaforme supportate	Non dipende dal tipo di piattaforma	
Posizione del codice sorgente	UNIFI, Laboratorio DISIT	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	Da stabilire	
Indirizzo/i accesso via web (se presenti)	Da stabilire	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Sensors		
Actuators		
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
MySQL HBase	Database necessario per la memorizzazione di informazioni realtive ai sensori e/o mobile App.	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>

API REST	Php	Nessuna
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

### 5.9.2 Acquisizione dati da Mobile come Sensori: Mobile APP con Wi-Fi e BT (API19) (UNIFI, )



Profilo Tool/Algoritmo	
Acquisizione dati da Mobile come Sensori: Mobile APP con Wi-Fi e BT	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	<p>Acquisizione Wi-Fi e BT via mobile app. Queste API coincidono con le API che il Sensor Server and Manager espone nelle smart city API per permettere alle applicazioni mobili di registrare eventi e stato sensori tramite chiamate REST.</p> <p>Sono inviati dal mobile verso il Sensor Server and Manager:</p> <ul style="list-style-type: none"> <li>• ID del mobile</li> <li>• Date and time</li> <li>• Tipo di device e di APP</li> <li>• Coordinate GPS e altre informazioni</li> <li>• Valori dei sensori riscontrati: Wi-Fi, BT, Beacon,</li> <li>• Ma anche in certi casi: temperatura, CANBUS, umidità, numero passeggeri, stato del mezzo, etc. etc.</li> <li>• Etc. etc.</li> </ul>
Dati primari in ingresso	Acquisizione di dati provenienti da sensori e/o sensori e/o mobile App, tramite API REST.
Dati prodotti in Uscita	Il modulo scrive i dati su Database, inoltre fornisce delle API REST che consentono di esporre i dati dei kit JSON per la conseguente acquisizione

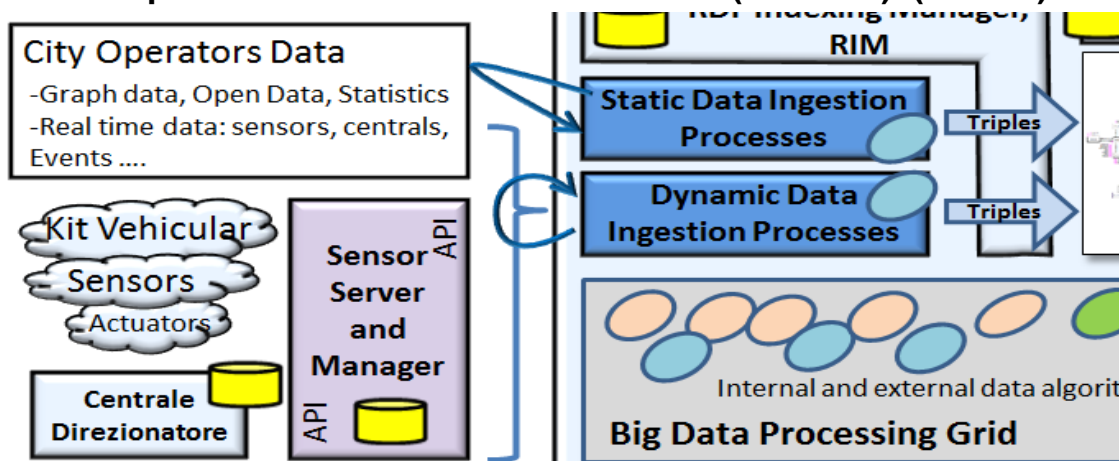
	verso la base di conoscenza Sii-Mobility, tramite un processo ETL (uno per ciascun tipo di sensore e/o mobile App).	
Principi (metodo, basi teoriche, etc.)	Realizzazione di: API REST, ETL	
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	n/d	
Principali problemi non risolti	Assenti	
Principali requisiti pendenti	Nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0.0%	
Eseguibile/libreria/web app	API REST (metodi http: POST, formato di comunicazione dati: JSON)	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Php	
Piattaforme supportate	Non dipende dal tipo di piattaforma	
Posizione del codice sorgente	UNIFI, Laboratorio DISIT	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	Da stabilire	
Indirizzo/i accesso via web (se presenti)	Da stabilire	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Sensors		
Mobile App		
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
MySQL HBase	Database necessario per la memorizzazione di informazioni reattive ai sensori e/o mobile App.	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
API REST	Php	Nessuna
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione</b>	<b>Licenza: GPL, LGPL, PEK,</b>

	usata	proprietaria, commerciale, etc.

**5.9.2.1** Procotollo, API: mobile\_save\_status

API REST per l’inserimento delle informazioni dei sensori e/o dei dati provenienti da mobile (visti sempre come sensori). Si veda la descrizione in dettaglio nel paragrafo 5.11.2

**5.9.3 ETL Acquisizione dati da sensori ambiente (MAPP03) (MIDRA)**



**Profilo Tool/Algoritmo**

Acquisizione dati da sensori ambiente

Nome responsabile	
Partner responsabile	MIDRA
Descrizione Tool/algoritmo	Interfaccia tra sensori e database SII mobility
Dati primari in ingresso	Misure fornite dai sensori
Dati prodotti in Uscita	Fusione dati ed elaborazione statistiche per il Sistema SII Mobility.
Principi (metodo, basi teoriche, etc.)	Non Applicabile
Casi di test (presenti/assenti)	Assenti
Posizione casi di test	n/d
Principali problemi non risolti	Assenti
Principali requisiti pendenti	Interfaccia di scambio con il sistema SII Mobility
<b>Aspetti Tecnologici</b>	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0
Eseguibile/libreria/web app	Eseguibile/web app
Single thread/Multithread	Multithread
Linguaggio di sviluppo (java, Php, ETL)	C, HTML, PHP, SQL,
Piattaforme supportate	Windows Server, Linux
Posizione del codice sorgente	Git
Indirizzo/i web services (se presenti) con	n/d

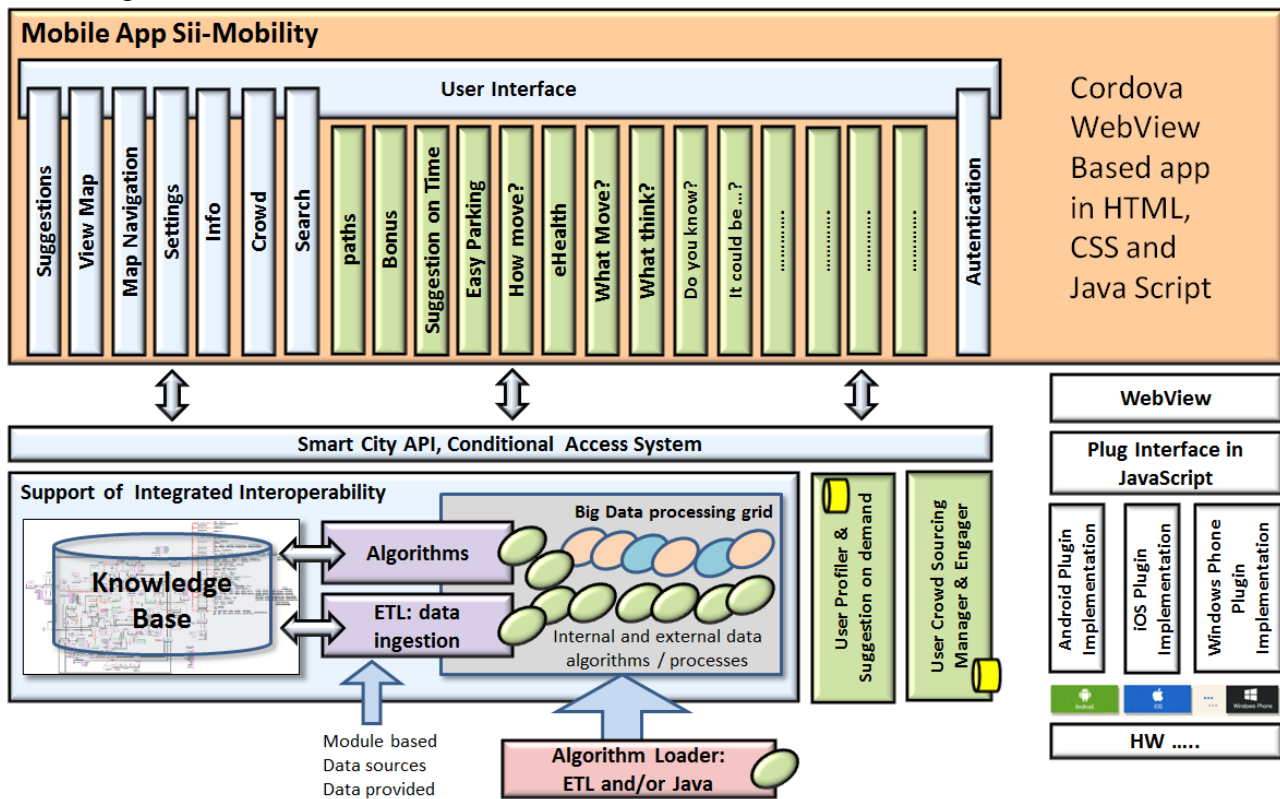
indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)	n/d	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Apache Web Server	APACHE	HTTP/CGI
Pentaho kettle per lo sviluppo di ETL che sono utilizzati per leggere il server di accumulo dati e caricare i dati stessi dentro la knowledge base		
MySQL connector for PHP	MYSQL	SQL
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
Da definire		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
http anche per chiamate REST		HyperText Transfer Protocol
TCP/IP		
TCP/IP over GPRS		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
MySQL	Database relazionale	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
Web	HTML, JavaScript, CSS, PHP	
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
MySQL		GPL



## 5.10 Specifica: applicazioni fisse e mobili campione



Con al seguente architettura.



Questo implica che lo sviluppo di un modulo (di quelli verdi funzionali specificati in seguito) per l'applicazione mobile puo' implicare:

- Lo sviluppo ed il caricamento di dati tramite la creazione di un processo ETL che legge a sua volta un file dati Open via http/FTP
- Lo sviluppo ed il caricamento di un algoritmo, per esempio di routing, intermodale
- Lo sviluppo di un modulo Cordova da integrare nell'applicazione generale secondo direttive specifiche e da concordare con UNIFI che opera l'integrazione e la pubblicazione della APP sui vari portali di market e anche verso hardware specifico purché compatibile Cordova. In caso contrario vanno sviluppati dei plugin Cordova uno per ogni piattaforma hardware/sistema operativo. Se il Sistema operativo o la piattaforma non sono supportati da Cordova va fatto tutto il portini di Cordova che potrebbe essere un costo non gestibile.

Ai moduli base si aggiungono quelli per Promozione eventi, View informazioni per percorsi bus, Bike, etc.; View informazioni su aree: ztl, giardini, etc.; Soluzione integrata di bigliettazione: corsa singola SMS, corse multimodali direttamente nel modulo multimodale; Eventi in Push protezione civile; Previsioni meteo per area...; Sistema di raccolta dati da sensori WiFi, Beacon. Questi moduli sono integrati nel support sviluppato da UNIFI.

Per Sii-mobility le app che su tale base potranno essere messe sui market potranno essere principalmente di due tipi:

- App per utenti a piedi, bike, e car, App per utenti privati che possono o meno avere un kit veicolare per bike e/o CAR.
- App per utenti evoluti come autisti, che si connette anche al kit veicolare evoluto e per bus, solo per Android.

		FCM	Mizar	UNIFI	IN20	TIME	Nesentis	Effknow	liberologico	ataf	busitalia	cttnord	tiemme	argos	elfi	calamai	Midra	Proiect	GEOIN	OUJESTIT	SOETEC	EWINGS
3.5.1	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: general frame per applicazioni as widget per mobile e web, gestione profilo lato server		X																			
3.5.2	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo itinerario, percorso				X														X			
3.5.3	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo profiling, comportamento virtuoso							X													X	
3.5.4	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo suggestion del tempo e nello spazio																		X			X
3.5.5	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo parcheggio easy			X					X											X		
3.5.6	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo come ti sposti, "ma ti sposti ?" → focalizzato come modulo che gestisce l'ecoguida anche in relazione ai dati che possono provenire da un dongle connesso a CANBUS in standard OBDII.																		X			
3.5.7	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo tieniti in forma		X																X			
3.5.8	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo lascia un commento, "che ne pensi....."		X																			X
3.5.9	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo servizi commerciali, "ma lo sai che ?"		X					X														
3.5.10	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo partecipativo, "sarebbe opportuno che ..."																				X	
3.5.11	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo "sapevi che nel...."								X												X	X
3.5.12	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo merci logistica				X														X			
3.5.13	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: altro			X					X											X		

Si ricorda che nel contesto della APP il modulo di **data harvesting** (descritto meglio in deliverable di OR3 sotto la responsabilità di NEAGENTIS) si interfaccia da una parte con i kit veicolari tramite Bluetooth e dall'altra chiama le API dello Sensor Server Manager per inviare i dati e per prendere eventuali direttive di controllo e comando (vedasi attuatori). Lo sviluppo di tale modulo lo si prevede in modo diverso in sistemi operativi diversi: java per Android, etc. la piattaforma primaria e' Android.

Principali Sottosistemi interessati dai requisiti da descrivere in questa Subsubsection, saranno i sottosistemi primari della specifica				
Sotto Attività	Partner coordinato	Sottosistema	Tool/Level	Descrizione sommaria

3.5.1	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: general frame per applicazioni as widget per mobile e web, gestione profilo lato server	UNI FI	City User Tools, CUT	Mobile App Sii-Mobility (APP03)	<p>Mobile App che permetta l'accesso ai dati di Sii-Mobility e che sfrutti i servizi di Sii-Mobility. In questi casi il fuoco è sui servizi personalizzati, sul sistema di guida connessa, sul profiling, sul creare un assistente personale per gli utenti nelle loro categorie. E' di assoluta importanza raccogliere commenti e tracciare il comportamento dell'utente. Permette anche di aiutare gli utenti in mobilità, etc.</p> <p>Questo stesso modulo CORE deve integrare funzioni per:</p> <ul style="list-style-type: none"> <li>- Connessione wifi con ZTL per accesso alle alle aree via autenticazione</li> <li>- Connessione con gestore del traffico tramite IP per acquisire eventi di semaforo, velocita. Etc. per la guida connessa.</li> <li>- Modulo generico per acquisizione dati da CANBUS tramite Bluetooth. (questa parte viene sviluppata nel modulo di aiuto alla guida)</li> </ul>
		UNI FI	City User Tools, CUT	Web Server User Profiling (T15)	Applicazione Web che permetta l'accesso ai dati di profilazione utente di Sii-Mobility. Vengono mostrate, tramite l'ausilio di grafici, le principali statistiche sull'utilizzo del servizio di raccomandazione aggregate o suddivise per profilo utente
		UNI FI	City User Tools, CUT	Registrazione e Autenticazione per App, tool e API (T16)	<p>Soluzione di registrazione ed autenticazione di utenti di App Mobile e WEB di Sii-Mobility. La registrazione permette agli utenti di accumulare punteggi anche tramite i sistemi di partecipazione (con contributi, come foto, commenti, voti, correzione di dati, etc. etc. ).</p> <p>Sviluppo in CORDOVA</p> <p>Moduli aggiungibili: da verificare</p> <p>Core distribuito ai partner</p> <p>Integrazione a cura di UNIFI</p> <p>Modulo per la gestione dei bonus, gigli</p>
3.5.2	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo itinerario, percorso	GEON	City User Tools, CUT	modulo itinerario, percorso (MAPP04)	<p>Modulo per web app e mobile per la richiesta di informazioni riguardo alla pianificazione di un percorso che preveda eventuali tappe intermedie e la loro visualizzazione interattiva.</p> <p>Dovrà essere possibile imporre vincoli sui mezzi da utilizzare e su come indirizzare il calcolo (percorso più veloce più economico, più sostenibile), etc. si veda requisiti del pianificatore e delle API che permetteranno a questo modulo di funzionare.</p>

3.5.3	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo profiling, comportamento virtuoso	SOF TEC H	City User Tools, CUT	modulo profiling, comportamento virtuoso (MAPP05)	Modulo per web app e mobile per la richiesta di informazioni riguardo alle info personali e preferenze ed al suo comportamento passato (e.g., prevalenza bus, car, moto, bike, etc.). Suggerisce comportamenti virtuosi, come l'uso di mezzi a minor impatto ambientale, l'uso di scambiatori, anche arrivando ad offrire bonus se possibile, prendendoli da un carnet di Bonus della PA o delle TPL. Possibile necessità di caricare un modulo computazionale aggiuntivo in piattaforma Sii-Mobility.
3.5.4	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo suggestion del tempo e nello spazio	GEO IN	City User Tools, CUT	modulo suggestion del tempo e nello spazio (MAPP06)	Modulo per web app e mobile che sfrutta gli algoritmi di pianificazione accessibili via API per fornire suggerimenti nel tempo e nello spazio. Per esempio: hai tempo passa da Mario, visto che sei qui visita XX, perché non parcheggi qui visto che hai ancora 20 minuti, oppure anche semplicemente dato un certo punto GPS impostando il tempo a disposizione, viene suggerito un percorso che suggerisce di visitare punti di interesse in base alle preferenze dell'utente. Possibile necessità di caricare un modulo computazionale aggiuntivo in piattaforma Sii-Mobility.
3.5.5	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo parcheggio easy	IN20	City User Tools, CUT	modulo parcheggio easy (MAPP07)	Modulo per web app e mobile che permette di monitorare lo stato dei parcheggi o di un parcheggio in particolare per evitare di arrivare al parcheggio senza sapere se vi sarà posto. Questo, avendo i dati, potrà essere usato in riferimento ai parcheggi coperti o lungo strada. Utilizza le smart City API per accedere ai dati dello stato dei parcheggi. Possibile necessità di caricare un modulo computazionale aggiuntivo in piattaforma Sii-Mobility
3.5.6	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo come ti sposti, "ma ti sposti?"	GEO IN	City User Tools, CUT	modulo come ti sposti, "ma ti sposti?" (MAPP08)	Modulo per web app e mobile che sfrutta gli algoritmi di profilazione e la valutazione del comportamento dell'utente per suggerire soluzioni di mobilità alternative. Utilizza le smart City API per accedere ai dati delle abitudini della persone, delle alternative, dei servizi presenti sul territorio, etc. Possibile necessità di caricare un modulo computazionale aggiuntivo in piattaforma Sii-Mobility.
3.5.7	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo tieniti in forma	UNI FI	City User Tools, CUT	modulo tieniti in forma (MAPP09)	Modulo per web app e mobile che sfrutta gli algoritmi di profilazione e la valutazione del comportamento dell'utente per suggerire soluzioni di mobilità che possono tenerlo in forma. Utilizza le smart City API per accedere ai dati delle abitudini della persone, chiede anche informazioni sull'età della persona, su cosa ha mangiato, etc. Possibile necessità di caricare un modulo computazionale aggiuntivo in piattaforma Sii-Mobility.

3.5.8	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo lascia un commento, "che ne pensi....."	UNI FI	City User Tools, CUT	modulo lascia un commento , "che ne pensi....." (MAPP10)	Modulo per web app e mobile che sfrutta gli strumenti di geolocalizzazione dell'App per capire dove si trova l'utente e se è rimasto fermo per un certo periodo in qualche punto di interesse o che ci è stato, gli può chiedere conferma, ma anche richiedere un contributo: un commento, un score, una foto. Utilizza le smart City API per accedere ai dati delle abitudini della persone e capire i servizi geolocalizzati, etc. Possibile necessità di caricare un modulo computazionale aggiuntivo in piattaforma Sii-Mobility. Per esempio per capire quanto una persona permane, se è possibile dare un etichetta a quel posto come uno dei suoi preferiti, etc.
3.5.9	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo servizi commerciali, "ma lo sai che ?"	UNI FI	City User Tools, CUT	modulo servizi commercia li, "ma lo sai che ?" (MAPP11)	Modulo per web app e mobile che sfrutta gli strumenti di geolocalizzazione dell'App per capire dove si trova l'utente e se è rimasto o sta passando a piedi o in auto per un certo punto di interesse. Che magari gli può interessare sulla base del suo profilo. Il sistema gli invia una notifica in push (il telefono chiede ed il server invia) dove si informa che riguardo a quel punto si possono promuovere commercialmente cose interessanti. <ul style="list-style-type: none"> <li>• Possibile necessità di caricare un modulo computazionale aggiuntivo in piattaforma Sii-Mobility (per elaborare e fare cache delle info estratte da dbpedia o da altri) oppure una base di dati.</li> <li>• Web Tool a supporto della soluzione per accettare la sottomissione di annunci pubblicitari.</li> </ul>
3.5.10	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo partecipativo, "sarebbe opportuno che ..."	SOF TEC H	City User Tools, CUT	modulo partecipati vo, "sarebbe opportuno che ..." (MAPP12)	Se si trova in modalità navigazione la destinazione è nota e lo fa il navigatore. Ma se la sua destinazione non è nota si può dargli una mano con questo modulo. Modulo per web app e mobile che sfrutta gli strumenti di geolocalizzazione dell'App per capire dove si trova l'utente e fornirgli delle indicazioni in real time sulle strade che potrebbe prendere per raggiungere le sue destinazioni abituali tenendo conto anche di eventuali incidenti e traffico, e per fargli ridurre il carico su certe direttrici.
3.5.11	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo "sapevi che nel...."	SOF TEC H	City User Tools, CUT	modulo "sapevi che nel...." (MAPP13)	Modulo per web app e mobile che sfrutta gli strumenti di geolocalizzazione dell'App per capire dove si trova l'utente e se è rimasto o sta passando a piedi o in auto per un certo punto di interesse. Che magari gli può interessare sulla base del suo profilo. Il sistema gli invia una notifica in push (il telefono chiede ed il server invia) dove si informa che riguardo a quel punto si possono raccontare aspetti presi da Wikipedia o da altra sorgente. Possibile necessità di caricare un modulo computazionale aggiuntivo in piattaforma Sii-Mobility (per elaborare e fare cache delle info estratte da dbpedia o da altri) oppure una base di dati.

3.5.1 2	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: modulo merci logistica	TIM E	City User Tools, CUT	modulo merci logistica (MAPP14)	Modulo per web app e mobile per la richiesta di informazioni riguardo alla pianificazione di un percorso di distribuzione merci che preveda eventuali tappe intermedie e la loro visualizzazione interattiva. Dovrà essere possibile imporre vincoli sui mezzi da utilizzare e su come indirizzare il calcolo (percorso più veloce più economico, più sostenibile), etc. si veda requisiti del pianificatore e delle API che permetteranno a questo modulo di funzionare. Possibile necessità di caricare un modulo computazionale aggiuntivo in piattaforma Sii-Mobility.
3.5.1 3	Studio, definizione e Sviluppo di applicazioni fisse e mobili campione: altro	LIBE ROL OGI CO	City User Tools, CUT	Altro (MAPP15)	1) Soluzione per il tracking di flotte di veicoli di operatori ecologici e infomobilità al guidatore, basato su mobile app e applicazione web di backoffice interoperabile con la piattaforma Sii-Mobility. 2) ....

Tutti i moduli interagiscono la mappa ed effettuano query su Smart City API.

	ETL data o processo di analytic	DB storico	attivazione	View only	Immissione dati
<b>modulo itinerario, percorso</b>	Si per computo routing	Centrale, user profile, storico bonus e percorsi	Da-a, da-a-a-a, GPS	no	Si
<b>modulo profiling, comportamento virtuoso</b>	<u>Si per computo bonus</u>	Centrale, user profile, storico bonus	Stato del comportamento o posizione	no	Si
<b>modulo suggestion del tempo e nello spazio</b>	Si per calcolo suggerimenti	Centrale, user profile, storico bonus	Stato del comportamento o posizione	no	No
<b>modulo parcheggio easy</b>	Dati parcheggi RT	Centrale, user profile, storico bonus	Stato del comportamento o posizione	no	No
<b>modulo come ti sposti, "ma ti sposti?"</b>	<u>Si per computo comportamento e suggestion</u>	Centrale, user profile, storico bonus	Stato del comportamento o posizione	Si	Si
<b>modulo tieniti in forma</b>	<u>per computo comportamento e suggestion</u>	Centrale, user profile, storico bonus	Stato del comportamento o posizione	Si	Si
<b>modulo lascia un commento, "che ne pensi....."</b>	<u>per computo comportamento e suggestion</u>	Centrale, user profile, storico bonus	Stato del comportamento o posizione	Si	Si
<b>modulo servizi commerciali, "ma lo sai che?"</b>	<u>per computo comportamento e suggestion</u>	Centrale, user profile, storico bonus	Stato del comportamento o posizione	Si	Si
<b>modulo partecipativo, "sarebbe opportuno che ..."</b>	<u>per computo comportamento e suggestion</u>	Centrale, user profile, storico bonus	Stato del comportamento o posizione	Si	Si

<b>modulo “sapevi che nel....”</b>	<u>per computo comportamento e suggestion</u>	Centrale, user profile, storico bonus	Stato del comportamento o posizione	Si	Si
<b>modulo merci logistica</b>	Si per computo routing	Centrale, user profile, storico bonus e percorsi	Da-a, da-a-a-a, GPS	Si	Si
<b>Modulo guida connessa</b>	Si per computo routing	Centrale, user profile, storico bonus	Stato del comportamento o posizione	Si	Si
<b>Modulo per connessione con ZTL via WIFI</b>	<u>per computo comportamento e suggestion</u>	Centrale, user profile, storico bonus	Stato del comportamento o posizione	Si	Si

Tutti i moduli sono sviluppati in accordo al modello descritto nella sezione 5.10.1. I moduli vengono sviluppati integrando il loro sviluppo nel frame che sarà sviluppato e distribuito da UNIFI. L'integrazione dei moduli sarà effettuata dal personale di UNIFI DISIT Lab. I moduli in consegna a UNIFI DISIT Lab devono arrivare già testati e collaudati, con il frame di test, con allegata una documentazione di massima che descrive le principali funzionalità ed il codice sviluppato.

## 5.10.1 Mobile App Sii-Mobility (APP03) (UNIFI)

Smart City API

Mobile App, Totem, Web APP

## Profilo Tool/Algoritmo

## Mobile App Sii-Mobility (APP03)

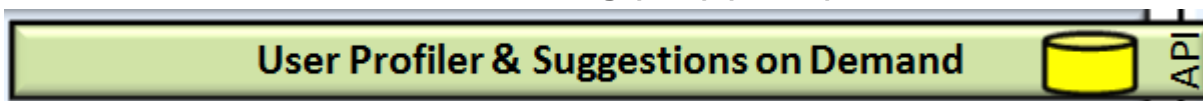
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	<p>Mobile App che permetta l'accesso ai dati di Sii-Mobility e che sfrutti i servizi di Sii-Mobility. In questi casi il fuoco è sui servizi personalizzati, sul sistema di guida connessa, sul profiling, sul creare un assistente personale per gli utenti nelle loro categorie. E' di assoluta importanza raccogliere commenti e tracciare il comportamento dell'utente. Permette anche di mostrare la propria reputazione ad altri utenti, aiutare altri utenti in mobilità, etc.</p> <p>Questo stesso modulo CORE deve integrare funzioni per:</p> <ul style="list-style-type: none"> <li>- Connessione wifi con ZTL per accesso alle aree via autenticazione</li> <li>- Connessione con gestore del traffico tramite IP per acquisire eventi di semaforo, velocità. Etc. per la guida connessa.</li> <li>- Modulo generico per acquisizione dati da CANBUS tramite Bluetooth. (questa parte viene sviluppata nel modulo di aiuto alla guida)</li> </ul>
Dati primari in ingresso	Dati provenienti da API REST, dati provenienti dai sensori presenti nel dispositivo dell'utente (bussola, GPS, Wi-Fi, Bluetooth)
Dati prodotti in Uscita	Visualizzazione grafica dei dati provenienti dalle API Rest
Principi (metodo, basi teoriche, etc.)	
Casi di test (presenti/assenti)	Assenti
Posizione casi di test	n/a
Principali problemi non risolti	n/a
Principali requisiti pendenti	n/a
<b>Aspetti Tecnologici</b>	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0.0%
Eseguibile/libreria/web app	Mobile App e Web App
Single thread/Multithread	Multithread
Linguaggio di sviluppo (java, Php, ETL)	Javascript, HTML, CSS, C#, Objective C, Java
Piattaforme supportate	iOS, Windows Phone, Android
Posizione del codice sorgente	SVN
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se	n/a



necessarie)		
Indirizzo/i accesso via web (se presenti)	www.km4city.org	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Cordova 5.4.0	ServiceMap API	REST API JSON
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTP		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
n/a		
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
Applicazione	Javascript, HTML, CSS	
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
Bootstrap	3.3.6	MIT
Date.Format	1.2.3	MIT
Forge-sha256		
Easy-button		
jquery	1.11.0	MIT
Jquery.fancytree.glyph		MIT
Jquery.fancytree.persist		MIT
Jquery.fancytree		MIT
Jquery.mobile	1.4.5	MIT
Jquery.ui	1.11.4	MIT
L.Terminator		
Leaflet.awesome-markers	2.0.1	
Leaflet	0.7.3	
OpenLayer	3.13.0	
Star-rating	3.5.6	
Twitter-text		
com.pylonproducts.wifiwizard	0.2.9	Apache 2.0

com.red_folder.phonegap.plugin.backgro undservice	2.0.0	Apache 2.0
cordova-plugin-app-event	1.2.0	Apache 2.0
cordova-plugin-appinfo	2.0.3	MIT
cordova-plugin-backbutton	0.3.0	MIT
cordova-plugin-ble-central	1.0.4	Apache 2.0
cordova-plugin-calendar	4.4.7	MIT
cordova-plugin-camera	2.1.0	Apache 2.0
cordova-plugin-device	1.1.1	Apache 2.0
cordova-plugin-device-orientation	1.0.2	Apache 2.0
cordova-plugin-dialogs	1.2.0	Apache 2.0
cordova-plugin-file	4.1.1	Apache 2.0
cordova-plugin-file-transfer	1.5.0	Apache 2.0
cordova-plugin-geolocation	2.1.0	Apache 2.0
cordova-plugin-imagepicker	1.1.0	Apache 2.0
cordova-plugin-inappbrowser	1.2.0	Apache 2.0
cordova-plugin-insomnia	4.2.0	MIT
cordova-plugin-legacy-whitelist	1.1.1	Apache 2.0
cordova-plugin-network-information	1.2.0	Apache 2.0
cordova-plugin-screen-orientation	1.4.0	MIT
cordova-plugin-splashscreen	3.2.0	Apache 2.0
cordova-plugin-statusbar	2.1.0	Apache 2.0
cordova-plugin-vibration	2.1.0	Apache 2.0
cordova-plugin-x-socialsharing	5.0.10	MIT
de.appplant.cordova.plugin.local- notification	0.8.4	Apache 2.0
de.fastr.phonegap.plugins.CheckGPS	0.9.8	MIT
org.disit.siiMobile.backgroundService	0.1	

### 5.10.2 Web Server User Profiling (T15) (UNIFI)



Il profilo utente deve includere:

- Tipologia: cittadino, pendolare, turista, studente, operatore, disabile, etc..
- User id collegato al telefono
- User id collegato al MAC
- User id collegato ad info personali, come: comune, facebook, twitter, linkedin, etc.
- Log delle sue azioni
- Log delle raccomandazioni ricevute, accettate
- Contributi prodotti in crowdsourcing: audio, video, etc...
- Scelte fatte sui menu dell'app o delle App
- Percorsi fatti
- Point di maggiore interesse: casa, scuola, lavoro, palestra, shopping, etc.
- Mezzi maggiormente usati
- Velocità tipica di movimento in città

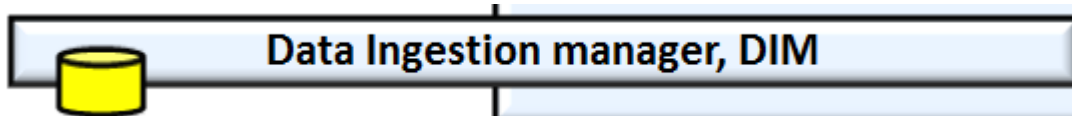
- Abitudini
- Bonus accumulati
- Area di residenza
- Etc..

<b>Profilo Tool/Algoritmo</b>		
Web Server User Profiling (T15)		
Nome responsabile		
Partner responsabile	DISIT-UNIFI	
Descrizione Tool/algoritmo	Applicazione Web che permetta l'accesso ai dati di profilazione utente di Sii-Mobility. Vengono mostrate, tramite l'ausilio di grafici, le principali statistiche sull'utilizzo del servizio di raccomandazione aggregate o suddivise per profilo utente	
Dati primari in ingresso	Connessioni degli utenti agli Access Point FirenzeWifi e Beacon	
Dati prodotti in Uscita	Flussi degli utenti	
Principi (metodo, basi teoriche, etc.)	Lo strumento consente la visualizzazione su mappa dei flussi utenti nella città di Firenze in base alle rilevazioni fornite dagli access point e ai dati di posizione forniti dai dispositivi mobile	
Casi di test (presenti/assenti)	assenti	
Posizione casi di test		
Principali problemi non risolti		
Principali requisiti pendenti		
Aspetti Tecnologici		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Web app	
Single thread/Multithread	Single thread	
Linguaggio di sviluppo (java, Php, ETL)	PHP	
Piattaforme supportate	Linux, Mac, Windows	
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)	<a href="http://disit.org/wifi-firenze">http://disit.org/wifi-firenze</a>	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione

JSON		
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
HTTP REST		
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
wifi	Database degli access point	
Solr	Indice dei dati di accesso degli utenti agli access point	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
web	PHP	
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
Highcharts		

### 5.10.3 Registrazione e Autenticazione e licensing per App (T16) (UNIFI)

#### Smart City API



<h2 style="margin: 0;">Profilo Tool/Algoritmo</h2>	
Registrazione e Autenticazione e licensing per App (T16)	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Soluzione di registrazione ed autenticazione di utenti di App Mobile e WEB di Sii-Mobility. La registrazione permette agli utenti di accumulare punteggi anche tramite i sistemi di partecipazione (con contributi, come foto, commenti, voti, correzione di dati, etc. etc. ).
Dati primari in ingresso	dati utenti e registrazione applicazioni
Dati prodotti in Uscita	Profili utenti e applicazioni
Principi (metodo, basi teoriche, etc.)	La soluzione permetterà di registrare utenti, associando il profilo anche ad social network come facebook o linkedin o google+ usando protocolli come OpenAuth 2.0.

Casi di test (presenti/assenti)	assenti	
Posizione casi di test	n/a	
Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	web app	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	PHP, Java	
Piattaforme supportate	Quelle supportate da PHP e Java	
Posizione del codice sorgente	n/a	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/a	
Indirizzo/i accesso via web (se presenti)	n/a	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con Dtool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>OpenAuth 2.0</b>		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>UtentiApplicazioni</b>	Database con dati degli utenti e delle applicazioni che usano i servizi	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>web</b>	PHP, HTML, Javascript	jQuery, jQuery UI, Bootstrap
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
<b>JQuery</b>		
<b>JQuery UI</b>		

Bootstrap		
-----------	--	--

#### 5.10.4 Mobile APP: Modulo itinerario, percorso (MAPP04) (GEOIN)

Modulo per web app e mobile per la richiesta di informazioni riguardo alla pianificazione di un percorso che preveda eventuali tappe intermedie e la loro visualizzazione interattiva. Dovrà essere possibile imporre vincoli sui mezzi da utilizzare e su come indirizzare il calcolo (percorso più veloce più economico, più sostenibile), etc. si veda requisiti del pianificatore e delle API che permetteranno a questo modulo di funzionare.

Modulo che interagisce con l'utente per la definizione del percorso DA ... A...

- Identificazione DA: posizione corrente, una posizione cercata, una posizione via GPS punto sulla mappa, un servizio, etc.
- Identificazione A: idem
- Tipo di navigazione: a piedi, mezzo privato (car), in bici, in TPL (multimodale), TPL+BICI
- Richiesta di pagamento pedaggio (se CAR)
- Richiesta di emissione biglietto se TPL
- Richiesta di eventuale giorno e ora di arrive se TPL per gli orari
- Descrizione del percorso per punti
- Descrizione dei costi per tratte
- Riproduzione del viaggio con segmenti su mappa
- Lista ultimi viaggi fatti
- Lista ultime destinazioni scelte
- Etc.

Il modulo viene sviluppato in accordo al modello descritto nella sezione 5.10.1, dai partner identificati nella tabella riportata in Sezione 5.10.

GEOIN e' il referente responsabile per lo sviluppo di questo modulo.

Modulo web app per apparato mobile finalizzato alla richiesta di informazioni riguardo alla pianificazione di un percorso. Potrà esse mono o multi modale e prevedere eventuali tappe intermedie. Il risultato avrà sia una visualizzazione grafica su mappa che una descrizione a lista.

Flusso di utilizzo:

1. Si attiva la funzione *calcolo percorso*
2. La app propone la scelta dei mezzi da utilizzare (piedi, bicicletta, auto, mezzo pubblico) costituiti da check box rappresentati tramite icone. Sarà possibile scegliere una o più categorie.
3. Si entra nella sotto procedura per la selezione del punto intermedio o finale
4. *Opzionale: nel caso che il percorso preveda l'uso del mezzo privato si può proporre come destinazione quella del parcheggio più vicino, trasformando il percorso in multimodale ed indicando l'ultima tratta come "a piedi".*
5. Si invia la richiesta
6. Si riceve la risposta in formato Json e la si renderizza sulla mappa
7. Un bottone, aggiunto in un angolo della mappa, permetterà di aprire la finestra per la visualizzazione del percorso in forma testuale.
8. *Opzionale: un link tra la parte grafica e la parte testuale per cui, cliccando un tratto del percorso (grafico), si evidenzia il corrispondente elemento della lista testuale ed in modo*

*analogo cliccando su un elemento della lista, si evidenzia il corrispondente elemento sulla mappa.*

9. *Opzionale: qualora il percorso preveda l'uso di parcheggi o di mezzi del trasporto pubblico la possibilità di acquistare il corrispondente biglietto.*

Procedura di selezione del punto di partenza, tappa intermedia o destinazione

1. Tramite un menu a tendina si potrà scegliere tra:
  - a. Posizione attuale GPS che viene indicata sulla mappa, con associato un pop up che indica il toponimo associato a quella posizione.
  - b. Punto indicato sulla mappa. Il menù si chiude ed un cursore di forma particolare suggerisce all'utente di indicare il punto sulla mappa. Durante questa operazione dovrà essere possibile effettuare pan, zoom ed eventualmente tornare al passo precedente di selezione della modalità. Individuato il punto un pop up indicherà il toponimo associato a quella posizione.
  - c. Ricerca full text. Il menù si chiude ed appare una box contenente una casella di testo in cui digitare la stringa corrispondente a quanto cercato. Nel caso in cui questa produca ambiguità (più punti geografici corrispondenti allo stesso testo), verrà proposta una lista con il testo dei vari risultati permettendo di selezionare quello desiderato se presente, o tornare indietro e perfezionare la digitazione.
2. *Opzionale:* si potrà impostare data ed ora in cui si desidera partire o arrivare al punto di tappa.

Questa procedura Utilizza le SmartCity Api per la funzionalità di Geocoding sia diretta, cioè data una stringa individuare il punto geografico (latitudine, longitudine) corrispondente, sia inversa, dato un punto geografico individuare il toponimo più vicino alla posizione.

Nel caso di geocoding diretto il jJon ritornato dovrà contenere la stringa completa (città, via , cap, civico) oltre alle relative coordinate geografiche. Nel caso di ambiguità il Json sarà costituito da una collection di località.

Calcolo del percorso

É richiesta una API che possa attivare il calcolo in modo mono o multimodale. Per questa app le tappe dovrebbero essere intese come sequenziali per cui non è richiesta alcuna ottimizzazione ed il calcolo potrà essere eseguito come sequenza di percorsi semplici costituiti solo da partenza ed arrivo. L'indicazione dell'ora di partenza e / o di arrivo sarà utilizzato solo nel caso di percorso multimodale dove sono coinvolti mezzi pubblici per poter indicare quando passa il primo mezzo disponibile.

Il Json di ritorno dovrebbe essere costituito da una collection di polilinee georeferenziate ciascuna corredata con le indicazioni di toponomastica e le eventuali indicazioni di manovra.

Questo dato potrà essere utilizzato su richiesta tanto per una visualizzazione grafica che per una testuale del tipo percorri via Roma per 100 e poi svolta a destra in corso Garibaldi.

Api Utilizzate

- Geocodifica diretta ed inversa
- Calcolo del percorso mono multimodale
- Renderizzazione di una polilinea su mappa

### 5.10.5 Mobile APP: modulo profiling, comportamento virtuoso (MAPP05) (SOFTEC)

Modulo per web app e mobile per la richiesta di informazioni riguardo alle info personali e preferenze ed al suo comportamento passato (e.g., prevalenza bus, car, moto, bike, etc.). Suggerisce comportamenti virtuosi, come l'uso di mezzi a minor impatto ambientale, l'uso di scambiatori, anche arrivando ad offrire bonus se possibile, prendendoli da un carnet di Bonus della PA o delle TPL. Possibile necessità di caricare un modulo computazionale aggiuntivo in piattaforma Sii-Mobility. Server side viene sviluppato un modulo che produce informazioni sul comportamento dell'utente: come di muove, dove va di solito, etc. Il modulo dovrebbe informare l'utente sui suoi percorsi tipici, suggerirgli di fare percorsi alternativi, di usare i mezzi pubblici, di usare parcheggi esterni, di andare in bike se possibile, etc.

<b>Scenario</b>	L'utente avvia la app per percorrere un tragitto
<b>Descrizione</b>	Mentre l'utente si muove con app attiva, il server pusha delle "sintesi" di notifiche al fine di suggerirgli scelte di trasferimento migliori rispetto a quelle abitualmente seguite e/o alla condizioni attuali della viabilità. L'utente, tramite feedback ai suggerimenti ricevuti, avvia una scelta interattiva con le possibili soluzioni prospettate dal servizio.
<b>Attori</b>	Il comune cittadino che avvia l'app e ha il GPS attivo
<b>Passi</b>	<ol style="list-style-type: none"> <li>1. L'utente avvia l'app.</li> <li>2. Il server pusha una notifica all'utente per suggerirgli di deviare il suo percorso ideale, ad esempio parcheggiando la macchina in un punto e prendendo l'autobus alla fermata adiacente.</li> <li>3. La notifica permette di scegliere fra un "Ok, accetta suggerimento" oppure un "No, preferisco continuare sul mio percorso abituale". <ol style="list-style-type: none"> <li>a. Cliccando su "No", l'utente sceglie di proseguire sul suo percorso, o comunque decide di ignorare il suggerimento ricevuto.</li> <li>b. Se invece l'utente sceglie "Sì", viene effettuata una nuova richiesta al server (modulo Calcolo percorso) per ottenere il tragitto modificato verso il punto prescelto (parcheggio auto e successivamente fermata autobus)</li> </ol> </li> <li>4. Ogni volta che l'utente decide di accettare un suggerimento ricevuto, guadagna un +1 di bonus, che deve essere associato alla sua utenza mediante una chiamata per ID ad un nuovo servizio server side. Il servizio notifica l'accredito all'utente.</li> </ol>

Il modulo viene sviluppato in accordo al modello descritto nella sezione 5.10.1, dai partner identificati nella tabella riportata in Sezione 5.10.

SOFTECH e' il referente responsabile per lo sviluppo di questo modulo.

### 5.10.6 Mobile APP: modulo suggestion del tempo e nello spazio (MAPP06) (GEOIN)

Modulo per web app e mobile che sfrutta gli algoritmi di pianificazione accessibili via API per fornire suggerimenti nel tempo e nello spazio. Per esempio: hai tempo passa da Mario, visto che



sei qui visita XX, perché non parcheggi qui visto che hai ancora 20 minuti, oppure anche semplicemente dato un certo punto GPS impostando il tempo a disposizione, viene suggerito un percorso che suggerisce di visitare punti di interesse in base alle preferenze dell'utente. Possibile necessità di caricare un modulo computazionale aggiuntivo in piattaforma Sii-Mobility.

Server side viene sviluppato un modulo che produce informazioni sul comportamento dell'utente: come di muove, dove va di solito, etc. Lato server esiste una funzione che permette di sapere cosa vi è intorno ad un punto GPS o nell'intorno di un servizio.

Il modulo potrebbe accedere all'agenda lato mobile tramite Google agenda e suggerire se l'utente ha tempo di fare altre cose, fermarsi prima e farlo a piedi, usare un mezzo diverso, fare visita ad un artigiano, etc.

Anche l'utente potrebbe chiedere "cosa posso fare nei prossimi 30 minuti liberi".....

Il modulo viene sviluppato in accordo al modello descritto nella sezione 5.10.1, dai partner identificati nella tabella riportata in Sezione 5.10.

GEOIN è il referente responsabile per lo sviluppo di questo modulo.

Si presume che lato server il sistema abbia acquisito il comportamento abituale dell'utente, ad esempio tutte le mattine prende l'auto, compie questo itinerario e parcheggia nell'intorno di...

Sulla base del comportamento acquisito, il server deve poter elaborare dei percorsi alternativi che prevedano un impiego delle tratte in bus e delle tratte a piedi oppure l'uso parcheggi scambiatori più convenienti rispetto a quello scelto abitualmente. Queste alternative, organizzate come i "tip" di alcune applicazioni desktop, potrebbero venire proposte all'avvio dell'app, lasciando all'utente la possibilità di selezionare il prossimo suggerimento. Ad esempio:

Stamani potresti non usare l'auto, prendere il bus alla tale fermata, scendere a quest'altra e procedere a piedi fino a destinazione, impiegheresti tanti minuti, ma risparmiaresti x € di carburante riducendo di y le emissioni di CO2 ti interessa?

In caso di risposta affermativa l'app potrebbe provvedere a mostrare su mappa l'itinerario e proporre l'acquisto del biglietto del bus.

La stessa box che propone i suggerimenti potrebbe presentare un bottone per l'invio di richieste. Queste dovrebbero essere selezionabili da una lista contenente un numero limitato di opzioni aggiungendo, a seconda dei casi, uno o più parametri. Le opzioni tra cui scegliere potrebbero essere:

- Ho a disposizione x minuti cosa potrei fare nell'intorno del mio itinerario abituale dato che i miei interessi sono (monumenti, librerie...)
- Che eventi ci sono vicino alla mia destinazione
- Cerco un negozio di... vicino al mio percorso abituale ecc.

#### Api utilizzate

- Acquisizione dati utente per elaborazione comportamento abituale (?)
- Calcolo percorso mono – multimodale sulla base di partenza arrivo abituali
- Acquisto di biglietti di mezzi pubblici
- Elaborazione ed invio suggerimenti in termini di percorso partenza - destinazione che preveda e o includa alternative al mezzo privato
- Indicazioni di eventi o luoghi di interesse che si trovano in una fascia intorno al percorso abituale

### **5.10.7 Mobile APP: modulo parcheggio easy (MAPP07) (LIBEROLOGICO)**

Server side viene sviluppato un modulo che produce informazioni sul comportamento dell'utente: come di muove, dove va di solito, etc. Sono inoltre inviati I dati che riguardano lo stato dei parcheggi di superficie ed interrati. Il modulo può informare l'utente sui parcheggi vicini, mostrando I numero dei posti vuoti, anche con capacità predittiva se possibile, .... dare informazioni specifiche sui parcheggi per disabili, parcheggi blu e bianchi, parcheggi moto, coperti e non coperti, presenza di colonnine di ricarica nel parcheggio (posti liberi), posti moto/auto, posti per scarico merci, etc.; dare informazioni specifiche parcheggi e la loro tariffa, etc... franchigia a tempo; eventualmente accettare prenotazioni. Una volta scelto il parcheggio l'APP può dare il routing, e portare l'utente al pagamento, o connetterlo ai sistemi di pagamento integrati, anche consumando o guadagnando Bonus.

Nella soluzione integrata, tramite i servizi presenti sul territorio potrebbe: (i) tracciare il percorso in ambito urbano, inviando i dati alla centrale operativa per monitoraggio e verifica, (ii) notificare il driver dell'accesso in aree a limitazione di accesso e transito: ZTL, APU; (iii) avvisare l'utente di eventuali difformità rispetto al percorso previsto.

Lato backoffice, il gestore pubblico: configura il tracciato urbano obbligatorio; verifica, per ogni veicolo merci monitorato, il rispetto della sua corrispondenza; allerta il guidatore in caso di difformità.

Questo soddisfa l'esigenza per il gestore pubblico di realizzare una soluzione innovativa per i veicoli merci, a basso TCO (Total Cost of Ownership), per: monitorare lo spostamento di ciascun veicolo nel loro percorso urbano, verificare il rispetto delle ordinanze cittadine di accesso e transito per percorsi urbani obbligatori.

Il modulo viene sviluppato in accordo al modello descritto nella sezione 5.10.1, dai partner identificati nella tabella riportata in Sezione 5.10.

LIBEROLOGICO e' il referente responsabile per lo sviluppo di questo modulo.

### **5.10.8 Mobile APP: modulo come ti sposti, "ma ti sposti ?" (MAPP08) (GEOIN)**

Modulo per web app e mobile che sfrutta gli algoritmi di profilazione e la valutazione del comportamento dell'utente per suggerire soluzioni di mobilità alternative. Utilizza le smart City API per accedere ai dati delle abitudini della persone, delle alternative, dei servizi presenti sul territorio, etc. Possibile necessità di caricare un modulo computazionale aggiuntivo in piattaforma Sii-Mobility.

Questo modulo viene convertito a modulo che permette di fare monitoraggio del tipo di guida in modalità ECOGUIDA.

L'app viene connessa tramite Bluetooth con un dongle OBDII e fornisce informazioni sul tipo di guida, per aiutare l'utente ad avere un comportamento più ecosostenibile.

Le informazioni vengono inviate anche al server per collezionarle sul profilo utente, poter produrre uno storico del comportamento, ma anche statistiche del comportamento.

Il modulo viene sviluppato in accordo al modello descritto nella sezione 5.10.1, dai partner identificati nella tabella riportata in Sezione 5.10.

GEOIN e' il referente responsabile per lo sviluppo di questo modulo.

Soluzione in grado di dare indicazioni e avvisi relativi allo stile di guida. Lo stile di guida può essere dedotto in base alle accelerazioni rilevate su un tragitto di adeguata lunghezza. La presenza di forti accelerazioni sia in direzione longitudinale (accelerazioni e frenate) che trasversale (brusche

sterzate) indicano un comportamento costoso in termini di consumo carburante ed usura del veicolo oltre che pericoloso.

Nel caso in cui l'app sia installata su un mezzo pesante (camion o autobus) il kit veicolare, se collegato alla porta CANBUS, può avere accesso istantaneamente alla posizione del pedale del freno, dell'acceleratore e del numero di giri del motore da cui estrapolare sufficienti indicazioni per elaborare indicazioni sullo stile di guida (forse un po' più complessa è la valutazione dei bruschi cambiamenti di direzione usando solo i dati del can).

Nel caso invece di autoveicoli leggeri l'acquisizione di queste informazioni potrebbe basarsi tanto sul kit di bordo (in grado di rilevare le accelerazioni tanto utilizzando i sensori inerziali (accelerometri e giroscopi), quanto lo stesso GPS; tali informazioni sono in genere abbastanza "rumorose" e se non adeguatamente filtrate possono fornire indicazioni poco attendibili.

Una soluzione alternativa per i veicoli privati è quella di utilizzare un lettore OBDII collegato alla corrispondente porta presente su tutti gli autoveicoli da oltre 10 anni. Tali lettori generalmente economici (tra i 10 ed i 20€) possono essere collegati tramite Bluetooth al dispositivo su cui gira l'app. Si possono riscontrare dei problemi di compatibilità in quanto i protocolli utilizzati dalle varie marche ma anche da serie diverse dello stesso modello sono numerosi (si veda ad esempio <https://www.elmelectronics.com/products/ics/obd/>) e non tutti forniscono le stesse informazioni. Dall'analisi storica delle accelerazioni è possibile dedurre lo stile di guida classificandolo in 4 o 5 categorie e fornendo eventuali suggerimenti per migliorarlo (non aspettare l'ultimo momento per frenare, cambia marcia più spesso ecc.)

L'elaborazione di queste informazioni richiede algoritmi abbastanza complessi che devono essere implementati lato server.

L'app si dovrebbe quindi preoccupare di acquisire i dati ed inviarli al server che li conserverà per un periodo pre impostato (ad esempio una settimanale o un mese) al termine del quale elaborerà e classificherà lo stile di guida relativo al periodo. Per ogni utente si avranno quindi dei dati storici (in numero comunque limitato).

Quando si avvia l'app questa dovrebbe iniziare ad inviare al server, in background, i dati inerenti la valutazione dello stile di guida solo in occasione del superamento di certe soglie (accelerazioni e giri motore). L'utente potrebbe interagire richiedendo quale è il mio punteggio attuale, come mi sono comportato la scorsa settimana ecc.

Api utilizzate

- Acquisizione dati dalla app relativamente ad accelerazioni (se disponibili) o brusche variazioni di velocità, dati ricevuti dal dispositivo di cattura ODBII.
- Elaborazione dello stile guida e classificazione in categorie

### **5.10.9 Mobile APP: modulo tieniti in forma (MAPP09) (UNIFI)**

Modulo per web app e mobile che sfrutta gli algoritmi di profilazione e la valutazione del comportamento dell'utente per suggerire soluzioni di mobilità che possono tenerlo in forma. Utilizza le smart City API per accedere ai dati delle abitudini della persona, chiede anche informazioni sull'età della persona, su cosa ha mangiato, etc. Possibile necessità di caricare un modulo computazionale aggiuntivo in piattaforma Sii-Mobility.

Server side viene sviluppato un modulo che produce informazioni sul comportamento dell'utente: come di muove, dove va di solito, etc.

Si va ad incentivare il comportamento virtuoso, lasciando l'auto per BIKE o BUS, lasciando BUS per BIKE e a PIEDI, etc. Suggerimenti ed incentivi adeguati.

Il modulo viene sviluppato in accordo al modello descritto nella sezione 5.10.1, dai partner identificati nella tabella riportata in Sezione 5.10.

UNIFI è il referente responsabile per lo sviluppo di questo modulo.

Questo modulo permette, a chi ne fa uso, di ricevere suggerimenti su come abbinare un approccio sostenibile alla mobilità con una salutare attività fisica. Può avere una doppia modalità: suggerimento (visto che tutti i giorni...) oppure informativa sulla base di una precisa richiesta (come potrei fare per...).

In entrambi i casi è necessario conoscere alcuni dati atti ad individuare la tipologia di persona (età, sesso, peso, altezza). Questi dati potranno essere inseriti mediante un'apposita finestra e dovranno essere memorizzati sul server insieme agli altri dati dell'utente per essere utilizzati dagli algoritmi di calcolo.

Si assume che il sistema abbia accesso alle abitudini dell'utente ad esempio l'itinerario consueto e la tipologia di mezzi impiegati.

Selezionando *suggerimenti* all'avvio della app, verrà inoltrata una richiesta al server che risponderà con un suggerimento elaborato incrociando dati relativi alla corporatura ed alle abitudini (usi sempre l'auto e fai poco moto lo sai che camminando 10 minuti ogni giorno potresti...).

Selezionando *Informazioni su richiesta* l'utente potrà scegliere il tipo di suggerimento selezionandolo tra alcune categorie (vorrei perdere 5 kg, vorrei camminare ma per non più di 30 minuti), il sistema elaborerà una modalità di spostamento per ottimizzare le richieste provvedendo ad inviarne la descrizione.

Flusso operativo

1. Si attiva la app
2. Si seleziona la modalità operativa:
  - a. Suggerimenti
  - b. Informazioni su richiesta
  - c. Inserimento parametri personali
    - i. Età
    - ii. Sesso
    - iii. Peso
    - iv. Altezza
3. Nel caso si sia selezionato *informazioni su richiesta* si avrà un menu da cui scegliere il tipo di richiesta tra quelli previsti:
  - a. Per una vita più sana (dovresti camminare almeno xx minuti al giorno, prova a cambiare abitudini: elaborazione di un percorso che preveda un tratto a piedi
  - b. Vorrei perdere xx chili. (si elabora un consumo calorico adeguato e si suggeriscono degli spostamenti che prevedano adeguate tratte da farsi a piedi

Algoritmi lato server

- Calcolo percorso
- Valutazione del consumo calorico in base alle distanze percorse a piedi (basati su peso, statura ed età)
- Valutazione della forma fisica in funzione di età e dati corporei si determina lo stato (magro, normale, grasso, obeso)

- Valutazione del fabbisogno giornaliero di moto per mantenersi in salute in funzione dell'età e del rapporto peso / statura.

Api utilizzate:

- Geo codifica diretta ed inversa
- Calcolo del percorso ottimo multi tappa con vincoli relativi alla tipologia del veicolo.
- Richieste al sotto-modulo "forma fisica"

#### **5.10.10 Mobile APP: modulo lascia un commento, "che ne pensi....." (MAPP10) (UNIFI)**

Modulo per web app e mobile che sfrutta gli strumenti di geolocalizzazione dell'App per capire dove si trova l'utente e se è rimasto fermo per un certo periodo in qualche punto di interesse o che ci è stato, gli può chiedere conferma, ma anche richiedere un contributo: un commento, un score, una foto.

Utilizza le smart City API per accedere ai dati delle abitudini della persone e capire i servizi geolocalizzati, etc. Possibile necessità di caricare un modulo computazionale aggiuntivo in piattaforma Sii-Mobility. Per esempio per capire quanto una persona permane, se è possibile dare un'etichetta a quel posto come uno dei suoi preferiti, etc.

Modulo che raccoglie informazioni dall'utente: immagini, commenti, voti, anche sui servizi per esempio TPL, etc, Servizio moderato che porta alla visualizzazione delle immagini o dei commenti solo tramite una validazione da amministratore e gestore.

L'utente si muove con app attiva: può essere sia pendolare che turista ma viene interrogato con lo stesso form dato che non sembra rilevante distinguere i due casi.

Il server pusha una richiesta di commento o valutazione sapendo se l'utente ha appena effettuato un viaggio in bus o visitato una attrazione etc. L'utente può o meno lasciare un commento con il tipo di valutazione che preferisce.

1. L'utente avvia l'app.
2. Il server pusha una notifica all'utente per chiedere un commento o valutazione. L'utente può rifiutare ed in questo caso si chiude l'algoritmo. Il server non chiederà più info all'utente per un periodo di tempo definito.
3. Se l'utente accetta si apre una finestra che contiene :
  - Questionario di tipo chiuso con valutazione a stelletta.
  - Questionario di tipo aperto con campo testo di max tot caratteri
  - Campo inserimento immagini relativo all'attività appena svolta (da verificare che Cordova dia accesso alla fotocamera/Galleria e da verificare lato server la gestione di file di grandi dimensioni e loro visualizzazione su mappa).
4. L'utente sceglie che tipo di questionario compilare (solo questionario chiuso, solo testo, solo foto o più di uno).
5. La App invia le informazioni ricevute al server che le dovrà gestire e fare comparire le statistiche (es stelletta di valutazione di un luogo turistico): nel caso di foto geolocalizzarle sulla mappa o associarle a punti di interesse relativi.

EWINGS: Generazione di questionari (forma chiusa o aperta) in JSON o che altro che poi possono essere presentati all'utente su mobile raccogliendo tali dati e info sul server, sempre nel profilo dell'utente.

### 5.10.11 Mobile APP: modulo servizi commerciali, “ma lo sai che ?” (MAPP11) (UNIFI)

Modulo che suggerisce fatti storici che sono accaduti in città o che ricorda la storia dei personaggi famosi e dei loro legami con la città etc. vie etc. Vedi collegamento con dbPedia, etc. Può essere una sorta di Game per collezionare le informazioni dei personaggi famosi e dei loro legami con il territorio.

Implica:

- Acquisizione di informazioni da distribuire
- Geolocalizzarle
- Definire le regole del gioco e di emissione dei messaggi verso l’utente e regole di ingaggio

Descrizione Tool/algoritmo	<p>Modulo per web app e mobile che sfrutta gli strumenti di geolocalizzazione dell’App per capire dove si trova l’utente e se è rimasto o sta passando a piedi o in auto per un certo punto di interesse (che magari gli può interessare sulla base del suo profilo). Il sistema gli invia una notifica in push (il telefono chiede ed il server invia) dove si informa che riguardo a quel punto si possono promuovere commercialmente cose interessanti.</p> <ul style="list-style-type: none"> <li>• Possibile necessità di caricare un modulo computazionale aggiuntivo in piattaforma Sii-Mobility (per elaborare e fare cache delle info estratte da dbpedia o da altri) oppure una base di dati.</li> <li>• Web Tool a supporto della soluzione per accettare la sottomissione di annunci pubblicitari.</li> </ul> <p>Per il presente modulo sarà inoltre progettata e integrata una parte di back end per il crawling e l’ingestion di informazioni geolocalizzate di servizi commerciali, aziende ecc.</p>	
Dati primari in ingresso	ID utente, dati di geolocalizzazione da GPS di device utenti	
Dati prodotti in Uscita	Notifiche PUSH: raccomandazioni e suggerimenti per l’utente in base al suo profilo, al contesto e alla posizione geografica.	
Principi (metodo, basi teoriche, etc.)	Algoritmi e processi per il crawling e l’ingestion di dati e informazioni di servizi commerciali e aziende, e algoritmi di Natural Language Processing (NLP) per l’estrazione di informazioni di geolocalizzazione dei suddetti servizi.	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
User Profiler (5.13.5)	MySQL Connection Driver	Accesso e condivisione dati su database relazionale MySQL.
Suggestion on Demand (5.13.6)	MySQL Connection Driver	Accesso e condivisione dati su database relazionale MySQL.
Knowledge Base RDF (5.11.4)	SPARQL endpoint, API SPARQL dedicate	SPARQL result object

Il modulo viene sviluppato in accordo al modello descritto nella sezione 5.10.1, dai partner identificati nella tabella riportata in Sezione 5.10.

UNIFI è il referente responsabile per lo sviluppo di questo modulo.

### 5.10.12 Mobile APP: modulo partecipativo, “sarebbe opportuno che ...” (MAPP12) “sapevi che nel....” (MAPP13) (SOFTECH)

Se un utente trova con la APP in modalità navigazione la destinazione è nota e lo segue il navigatore. Ma se la sua destinazione non è nota si può dare un supporto all’utente con questo modulo. Modulo per web app e mobile che sfrutta gli strumenti di geolocalizzazione dell’App per capire dove si trova l’utente e fornirgli delle indicazioni in real time sulle strade che potrebbe prendere per raggiungere le sue destinazioni abituali tenendo conto anche di eventuali incidenti e traffico, e per fargli ridurre il carico su certe direttrici.

Modulo che suggerisce fatti storici che sono accaduti in città o che ricorda la storia dei personaggi famosi e dei loro legami con la città etc. vie etc. Vedi collegamento con dbPedia, etc. Può essere una sorta di Game per collezionare le informazioni dei personaggi famosi e dei loro legami con il territorio.

Implica:

- Acquisizione di informazioni da distribuire
- Geolocalizzarle
- Definire le regole del gioco e di emissione dei messaggi verso l’utente e regole di ingaggio

<b>Scenario</b>	L’utente si sta spostando con app e GPS attivi
<b>Descrizione</b>	In relazione alla posizione dell’utente, il server pusha delle notifiche su Points Of Interests (POI) legati a luoghi e personaggi. L’utente visualizza i POI suggeriti su mappa
<b>Attori</b>	Il comune cittadino che avvia l’app e ha il GPS attivo
<b>Passi</b>	<ol style="list-style-type: none"> <li>1. Il sistema di Engagement decide di fare firing di un messaggio (ad es., informazioni storiche sui punti d’interesse nelle vicinanze)</li> <li>2. L’utente visualizza sulla mappa i POI, in base a coordinate e tipologia ricevute dal servizio.</li> <li>3. Cliccando su una icona, una finestra mostra il dettaglio descrittivo su quel punto d’interesse. <ol style="list-style-type: none"> <li>a. L’utente può scegliere di chiudere il dettaglio descrittivo.</li> <li>b. Oppure può decidere di impostare quel punto come nuova destinazione del proprio tragitto.</li> </ol> </li> </ol>

Il modulo viene sviluppato in accordo al modello descritto nella sezione 5.10.1, dai partner identificati nella tabella riportata in Sezione 5.10.

SOFTECH e’ il referente responsabile per lo sviluppo di questo modulo.

### 5.10.13 Mobile APP: modulo merci logistica (MAPP14) (TIME, GEOIN)

Modulo web APP e mobile APP per la ricezione e consultazione delle liste di consegna prodotte dal modulo A06 di cui a 5.1.7. e dei percorsi ottimizzati prodotti dal modulo T05 di cui a 5.2.4. e relativa visualizzazione interattiva.

I vincoli relativi alla tipologia dei mezzi o eventuali altri vincoli sono già inclusi nei tool prima detti e quindi l’utente dispone di informazioni predefinite.

L’APP consente inoltre un aggiornamento del percorso ottimizzato in real time, per consentire all’utente di visualizzare variazioni di percorso legate alle condizioni dinamiche del traffico o ad altri dati dinamici disponibili sulla piattaforma Sii Mobility, inviando una richiesta al modulo T05 di cui a 5.2.4. ed ottenendo un ricalcolo del percorso.

La visualizzazione sarà sia testuale che tramite mappa con evidenziazione dei punti di consegna/ritiro.

L'APP mobile dovrà inoltre consentire lo scarico dei dati (tramite lettore bar code) relativi alle consegne/ritiri effettuati ed il loro invio al modulo PETL19i cui a 5.19.5.

Modulo per web app e mobile per la richiesta di informazioni riguardo alla pianificazione di un percorso di distribuzione merci che preveda eventuali tappe intermedie e la loro visualizzazione interattiva.

Dovrà essere possibile imporre vincoli sui mezzi da utilizzare e su come indirizzare il calcolo (percorso più veloce più economico, più sostenibile), etc. si veda requisiti del pianificatore e delle API che permetteranno a questo modulo di funzionare. Possibile necessità di caricare un modulo computazionale aggiuntivo in piattaforma Sii-Mobility.

Modulo che interagisce con l'utente per la definizione del percorso DA ... A... come sopra.

In aggiunta si assume trasporto con mezzo pesante

Fornitura delle dimensioni e peso del mezzo

Lista dei punti da toccare lungo il tragitto ... Da A1, A2, A3, A4, etc. con eventuali vincoli di orari.

Il modulo viene sviluppato in accordo al modello descritto nella sezione 5.10.1, dai partner identificati nella tabella riportata in Sezione 5.10.

TIME è il referente responsabile per lo sviluppo di questo modulo.

Questa app dovrebbe ricalcare in buona parte quella del 3.5.2 da cui si differenzia per due aspetti. Il percorso sarà sempre di tipo mono modale. È richiesta una ottimizzazione volta a minimizzare la percorrenza pur soddisfacendo agli eventuali vincoli temporali imposti sulle tappe.

Il flusso operativo potrebbe essere:

1. Si attiva la funzione *logistica*
2. Si inseriscono le tappe, ciascuna rappresentata da una posizione ed opzionalmente da un orario e da un tempo di sosta.
3. Si avvia la procedura
4. Si visualizzano in forma grafica e / o testuale i risultati

La parte interattiva dell'app è costituita dal sotto-modulo per l'inserimento delle tappe, gestita con un meccanismo del tipo aggiungi tappa - rimuovi tappa - fine. Ovvero una finestra che permetta di aggiungere una tappa, modificare o cancellare una tappa già inserita, terminare la procedura e quindi attivare automaticamente la parte di elaborazione.

La posizione relativa ad ogni tappa potrebbe essere inserita con un meccanismo simile a quello del 3.5.2, ma in questo contesto raramente sarà utilizzata la posizione attuale o il punto sulla mappa, si avranno degli indirizzi ed eventualmente una rubrica personale da cui prelevare quelli più utilizzati (ditta Rossi piuttosto che l'indirizzo della sede della ditta Rossi). Questo richiede che l'app possa avere accesso alla rubrica dell'utente memorizzata sul dispositivo. Il mezzo utilizzato sarà generalmente pesante per cui nel calcolo del percorso occorrerà tenere conto dei vincoli imposti da questo (su alcune strade urbane non è consentito la circolazione di mezzi pesanti, o è consentita solo in certe fasce orarie).

Il calcolo del percorso ottimo con in più il vincolo di orari di visita potrebbe non avere una soluzione o fornire una soluzione non ottimale. Un uso alternativo dell'app potrebbe essere quello di fornire in ingresso le tappe con il relativo tempo di sosta ed il sistema provvederà a calcolare il



percorso fornendo per ciascuna tappa la fascia oraria di visita così che il trasportatore possa informare il proprio cliente dell'orario previsto di arrivo per il ritiro o la consegna.

Api utilizzate:

- Geo codifica diretta ed inversa
- Calcolo del percorso ottimo multi tappa con vincoli relativi alla tipologia del veicolo.

<b>Profilo Tool/Algoritmo</b>	
<b>Mobile APP : modulo merci e logistica (MAPP14)</b>	
Nome responsabile	
Partner responsabile	TIME
Descrizione Tool/algoritmo	APP Mobile per la acquisizione delle lise di consegna e dei percorsi ottimizzati generati dai moduli A06 di cui a 5.1.7. e T05 di cui a 5.2.4. L'APP consente inoltre l'interrogazione del modulo T05. per rilevare eventuali variazioni nei percorsi dovuti a condizioni dinamiche, come ad esempio le condizioni di traffico e restituisce il nuovo percorso ottimizzato
Dati primari in ingresso	Dati da Moduli A06-T05
Dati prodotti in Uscita	Liste consegne/ritiri e percorsi ottimizzati in formato text e maps con evidenziazione delle fermate
Principi (metodo, basi teoriche, etc.)	
Casi di test (presenti/assenti)	Da definire
Posizione casi di test	Da definire
Principali problemi non risolti	Nessuno
Principali requisiti pendenti	Nessuno
Aspetti Tecnologici	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0%
Eseguibile/libreria/web app	Mobile APP
Single thread/Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java
Piattaforme supportate	Android, iOS, Windows 8
Posizione del codice sorgente	n.d.
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.d.
Indirizzo/i accesso via web (se presenti)	n.d.

#### **5.10.14 Mobile APP: Altri moduli per app mobile (MAPP15) (LIBEROLOGICO)**

Moduli aggiuntivi saranno identificati in corso d'opera.

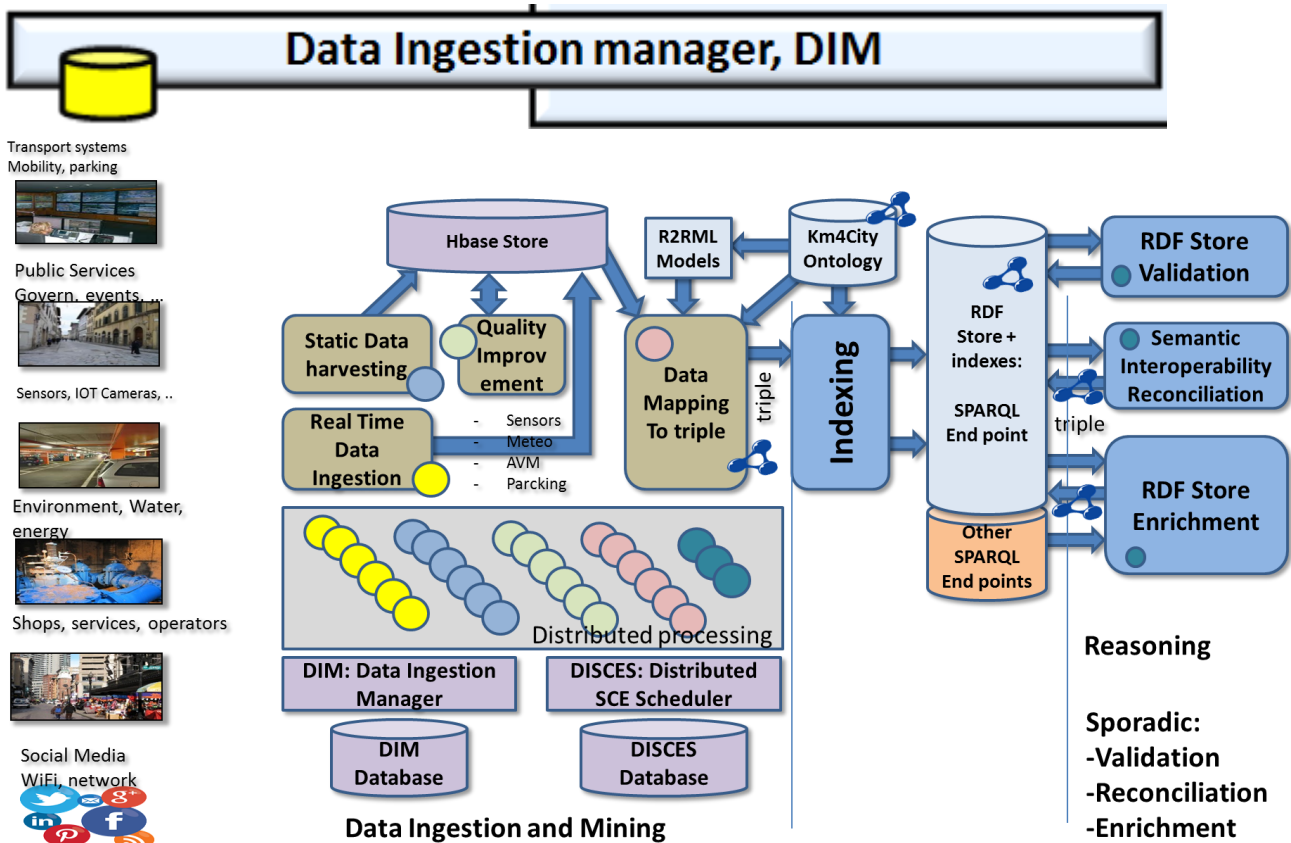
### 5.11 Specifica: moduli acquisizione dati e gestione ingestion process

		FCM	Mizar	UNIFI	IN20	TIME	Negeotis	EffKnow	liberologico	ataf	bustalia	cttnord	tiemme	argos	elfi	calamai	Midra	Proiect	GEOIN	OUJESTIT	SOFTEC	FWINGS	
4.1.1	Studio e sviluppo di moduli acquisizione dati e gestione ingestion process: struttura di ingestion		X	X																			
4.1.2	Studio e sviluppo di moduli acquisizione dati e gestione ingestion process: modello dati ingestion			X																			
4.1.4	Studio e sviluppo di moduli acquisizione dati e gestione ingestion process: scheduler per ingestion		X	X				X															
4.1.6	Studio e sviluppo di moduli acquisizione dati e gestione ingestion process: modello dati big data			X																			
4.1.7	Studio e sviluppo di moduli acquisizione dati e gestione ingestion process: aggregazione / riconciliazione		X	X				X															

Principali Sottosistemi interessati dai requisiti da descrivere in questa Subsubsection, saranno i sottosistemi primari della specifica					
	Sotto Attività	Partner coord	Sottosistema	Tool/Level	Descrizione sommaria
4.1.1	Studio e sviluppo di moduli acquisizione dati e gestione ingestion process: struttura di ingestion	UNIFI	Support of Integrated Interoperability	Data Ingestion manager, DIM (T17)	Strumento che censisce e gestisce il set di data stream (statici o dinamici) da ingerire, con relativi parametri di licensing, scheduling, status, etc. Lo Strumento pianifica e programma DISCES in modo da asincrono e periodico con offset di prima esecuzione.
4.1.2	Studio e sviluppo di moduli acquisizione dati e gestione ingestion process: modello dati ingestion	UNIFI	Data Ingestion	Sensor Server and Manager (protocol e storage temporaneo) (T18)	Collezionatore (crawler) di dati provenienti da sensori speciali, loro storage e navigazione con interfaccia utente. Per esempio i nuovi AVM kit, Bike Kit, etc.
4.1.4	Studio e sviluppo di moduli acquisizione dati e gestione ingestion process: scheduler per ingestion	UNIFI	Support of Integrated Interoperability	Distributed Smart City Engine Scheduler, DISCES (T19)	Strumento per lo scheduling dei processi di ingestione, aggregazione, integrazione, verifica, validazione, quality improvement, arricchimento, riconciliazione, data analytic, big data, processi esterni, etc.
4.1.6	Studio e sviluppo di moduli acquisizione dati e gestione ingestion process: modello dati big data	UNIFI	Knowledge Base	Knowledge Base RDFStore (dati statici e real time) (T20)	The Km4City model based of the city for static and dynamic data. RDF store including ontology and data.
		UNIFI	Knowledge Base	Knowledge Base Statistic RDFStore (dati statistici) (T21)	The Km4City model based of the city for statistic data. RDF store including ontology and data.

		UNIFI	Knowledge Base	RDF Indexer Manager, RIM (T22)	Strumento per l'indicizzazione delle triple negli RDF store e loro versioning.
4.1.7	Studio e sviluppo di moduli acquisizione dati e gestione ingestion process: aggregazione / riconciliazione	UNIFI	ETL e mapping	aggregazione / riconciliazione (A08)	Studio e sviluppo di algoritmi di aggregazione / riconciliazione verso il noSQL database e la base di conoscenza big data di Sii-Mobility, tecniche di link discovering, etc.

### 5.11.1 Data Ingestion manager, DIM (T17) (UNIFI)

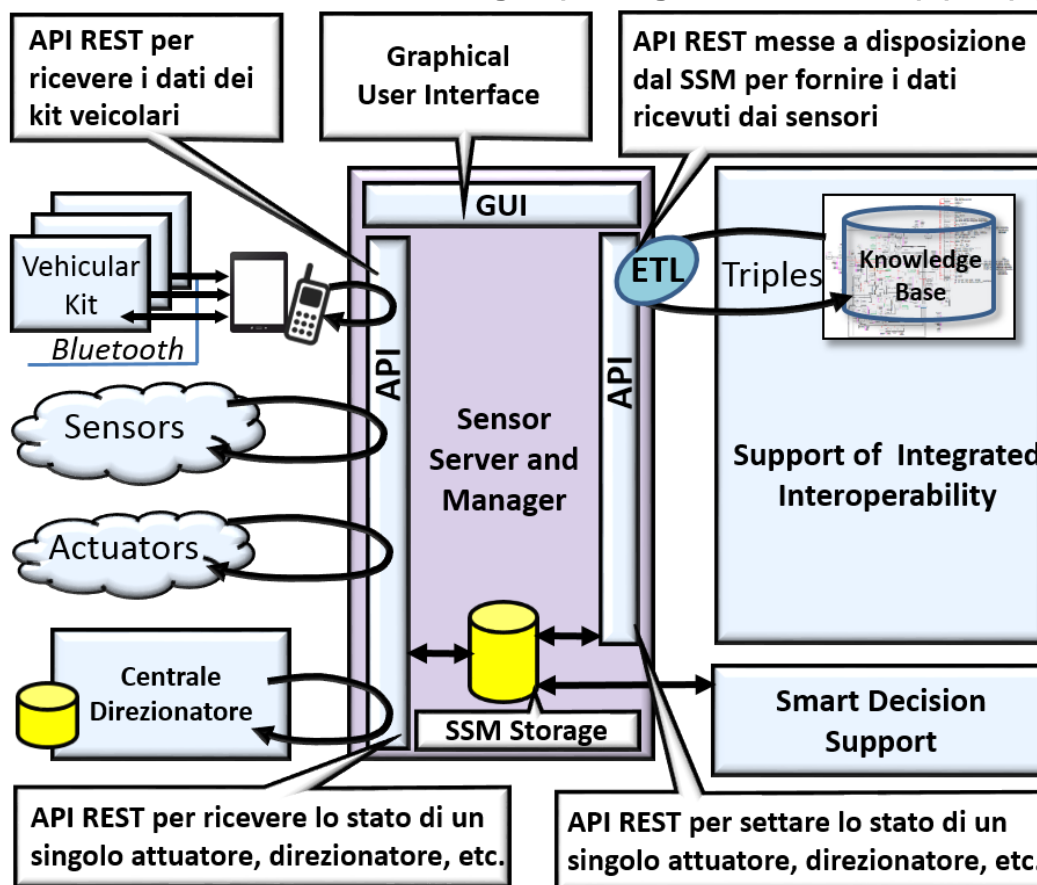


<h2 style="margin: 0;">Profilo Tool/Algoritmo</h2> <h3 style="margin: 0;">Data Ingestion manager, DIM (T17)</h3>	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Strumento che censisce e gestisce il set di data stream (statici o dinamici) da ingerire, con relativi parametri di licensing, scheduling, status, etc. . Lo Strumento pianifica e programma DISCES in modo asincrono e periodico con offset di prima esecuzione.
Dati primari in ingresso	Set di data stream (statici o dinamici) da ingerire

Dati prodotti in Uscita	Comandi di pianificazione e di programmazione del DISCES	
Principi (metodo, basi teoriche, etc.)	La pianificazione e programmazione del DISCES può essere in modo da asincrono e periodico con offset di prima esecuzione.	
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	n/d	
Principali problemi non risolti	Assenti	
Principali requisiti pendenti	Nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0.0%	
Eseguibile/libreria/web app	Web app	
Single thread/Multithread	Single thread	
Linguaggio di sviluppo (java, Php, ETL)	PHP, Javascript, HTML, CSS	
Piattaforme supportate	Unix, Windows	
Posizione del codice sorgente	svn	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)		
	Interfacce API usate	Modello di comunicazione e formato
DataIngestionManager	n/a	n/d
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)		
	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
JSON		JavaScript Object Notation
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)		
	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
HTTP		HyperText Transfer Protocol
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)		
	Descrizione	
MySql	Database relazionale	
<b>Tipo interfaccia utente</b> (web/applicazione)		
	Modello sviluppo, linguaggio	Libreria usata per UI
Web	HTML, JavaScript, CSS	Bootstrap
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)		
	Nome e versione	Licenza: GPL, LGPL, PEK,

	usata	proprietaria, commerciale, etc.
JQuery	1.10.1	Licenza MIT
Bootstrap	3.1.1	Licenza MIT

### 5.11.2 Sensor Server and Manager (storage and interface) (T18) (UNIFI)



Profilo Tool/Algoritmo	
Sensor Server and Manager (T18)	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	<p>Questo tool avrà lo scopo di:</p> <ul style="list-style-type: none"> <li>• Collezionare dati tramite chiamate REST provenienti da: Kit veicolari (e.g. AVM kit, Bike Kit, etc.), da applicazioni mobili, oppure da sensori indipendenti</li> <li>• Mostrare tramite interfaccia utente (GUI), i dati ricevuti e lo stato dei processi di acquisizione. Permettendo inoltre di effettuare la navigazione tra i dati/processi stessi (tramite filtri e ricerche, statistiche, etc.)</li> <li>• Ricevere richieste per fornire lo stato ad alcuni attuatori o sensori che cambiano configurazione, per esempio il direzionatore,</li> </ul>

	semafori, etc. <ul style="list-style-type: none"> <li>Ricevere lo stato per alcuni attuatori o sensori tramite chiamate REST per esempio dallo <i>Smart Decision Support</i> o da <i>Dashboard</i>.</li> </ul>	
Dati primari in ingresso	Dati provenienti da: <ul style="list-style-type: none"> <li>Kit veicolari (e.g. AVM kit, Bike Kit, etc.), sensori</li> <li>Dati da Smart DS e/o da Dashboard</li> </ul>	
Dati prodotti in Uscita	<ul style="list-style-type: none"> <li>Visualizzazione stato processi/dati tramite GUI</li> <li>Sistema di Log, relativo alla ingestione delle informazioni</li> <li>API REST, per l'invio dello stato dei mobile connessi a: (i) Kit veicolari (e.g. AVM kit, Bike Kit, etc.), (ii) sensori. Tali dati, verranno richiesti tramite opportuni processi ETL al blocco Sensor Server and Manager, ed arricchiranno la Knowledge Base.</li> </ul>	
Principi (metodo, basi teoriche, etc.)	Realizzazione di un sistema di API REST, Realizzazione di una interfaccia web, realizzazione della specifica dei messaggi da inviare/riceve tramite le API	
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	UNIFI, Laboratorio DISIT	
Principali problemi non risolti	Nessuno	
Principali requisiti pendenti	Nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Web App	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Php	
Piattaforme supportate	tutti i tipi di Browser	
Posizione del codice sorgente	UNIFI, Laboratorio DISIT	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	Da Stabilire	
Indirizzo/i accesso via web (se presenti)	Da Stabilire	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON	Mobile save status	Rest call
JSON	Sensor save status	Rest call
JSON	actuator get status	Rest call

JSON	actuator save status	Rest call
JSON	get Status	ETL rest call
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
Protocollo HTTP, paradigma REST	Data harvester	Mobile save status
Protocollo HTTP, paradigma REST	Sensors, Data Harvester	Sensor save status
Protocollo HTTP, paradigma REST	Data harvester	actuator get status
Protocollo HTTP, paradigma REST	Smart DS, Dashboard	actuator save status
Protocollo HTTP, paradigma REST	ETL process	Mobile get Status
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
MySQL/HBase	Database necessario per l'ingestion dei dati dei sensori e per il relativo stato	
MySQL	Database necessario per la gestione della GUI (log, dati utenti, etc.)	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
Applicazione web	Php	Da stabilire
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
Da stabilire	Da stabilire	Da stabilire

## API, azioni possibili:

- sensors\_insert\_info
  - Mobile save status: data harvester su mobile e kit veicolare (dati che possono essere relativi a stato sensori mobili, sensori bike sensori can bus, sensori autobus, etc. questi vanno memorizzati su database;
  - Sensor save status: sensori diretti, per esempio un embedded che chiama direttamente il SSM per salvare certi dati;
- Sensor\_get\_info: per acquisire lo stato di mobile, sensori, e attuatori salvato su database dentro il SSM,
- actuator get status: attuatori (per esempio anche il direzionatore), fanno una chiamata rest allo SSM per sapere se vi sono dei valori da aggiornare, delle azioni da fare, per esempio cambiare luce ad un semaforo, inviare un messaggio on scree, etc.
- actuator save status: scrittura da parte di processi che producono decisioni, per esempio lo Smart DS per decidere se il semaforo deve cambiare colore o il direzionatore cambiare senso, etc. L'azione di cambiare la direzione potrebbe anche provenire dalla Dashboard nella Sala di controllo. Anche questa parte di API devono essere autenticate.

### 5.11.2.1 Protocollo API Mobile\_save\_status

<b>Nome:</b> sensors_insert_info	
<b>URL:</b> Da stabilire	
<b>Descrizione</b>	
<p>Questa API viene usata da: Kit veicolari (e.g. AVM kit, Bike Kit, etc.), sensori, attuatori e/o da applicazioni mobile per l'invio dei dati al Sensor Server and Manager.</p> <ul style="list-style-type: none"> <li>○ Mobile save status: data harvester su mobile e kit veicolare (dati che possono essere relativi a stato sensori mobili, sensori bike sensori can bus, sensori autobus, etc. questi vanno memorizzati su database;</li> <li>○ Sensor save status: sensori diretti, per esempio un embedded che chiama direttamente il SSM per salvare certi dati;</li> </ul>	
Protocollo: HTTP, Modalità: POST	
<b>Parametri</b>	
action*	sensors_insert_info
type*	json
(*) = required	
<b>Esempi</b>	
<p>http://....?action=sensors_insert_info&amp;type=json</p> <p><b>JSON:</b></p> <pre>[{   "sensor_id": "2222",   "sensor_type": "vehicular_kit",   "vehicle_type": "Bike",   "info_type": "velocity",   "description": "Segnalazione del sensore 2222",   "submissionTime": "2017-05-01 10:15:34",   "currentState": "RUNNING",   ... }, {   "sensor_id": "3324",   "sensor_type": " vehicular_kit",   "vehicle_type": "Bike",   "info_type": "velocity",   "description": "Segnalazione del sensore 3324",   "submissionTime": "2017-04-01 12:16:04",   "currentState": "RUNNING",   ... }, .... ]</pre>	

### 5.11.2.2 Protocollo API sensors\_get\_info



<b>Nome:</b> sensors_get_info	
<b>URL:</b> Da stabilire	
<b>Descrizione</b>	
Questa API viene usata dai processi ETL per ricevere informazioni inviate dai sensori al Sensor Server and Manager (informazioni che saranno integrate nella Knowledge Base di SiiMobility). L'API sarà autenticata.	
Protocollo: HTTP, Modalità: GET	
<b>Parametri</b>	
action*	sensors_get_info
type*	json
limit	INT
offset	INT
(*) = required	
<b>Esempi</b>	
http://....?action=sensors_get_info&type=json	
<b>JSON sent:</b>	
<pre>[{   "sensor_id": "2222",   "sensor_type": "vehicular_kit",   "vehicle_type": "Bike",   "info_type": "velocity",   "description": "Segnalazione del sensore 2222",   "submissionTime": "2017-05-01 10:15:34",   "currentState": "RUNNING",   ... }, {   "sensor_id": "3324",   "sensor_type": " vehicular_kit",   "vehicle_type": "Bike",   "info_type": "velocity",   "description": "Segnalazione del sensore 3324",   "submissionTime": "2017-04-01 12:16:04",   "currentState": "RUNNING",   ... }, .... ]</pre>	

### 5.11.2.3 Protocollo API actuator\_get\_status

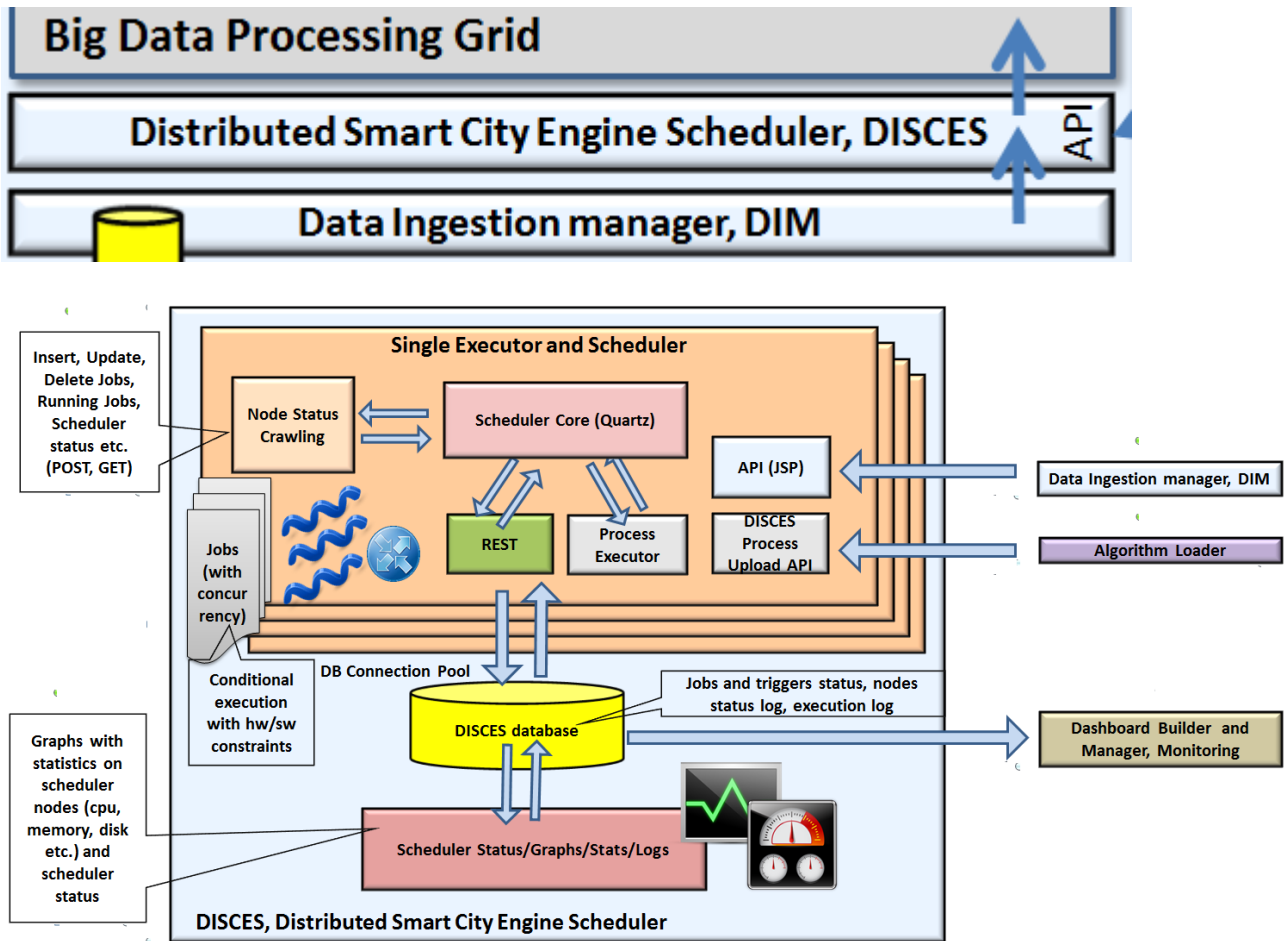
<b>Nome:</b> actuator_get_status	
<b>URL:</b> Da stabilire	
<b>Descrizione</b>	
Questa API viene usata da: sensori/attuatori/kit veicolari/Centrale Direzioneatore/etc. per ricevere informazioni sul proprio stato e/o dal blocco Smart Decision Support. L'API sarà autenticata.	
Protocollo: HTTP, Modalità: GET	

Parametri	
action*	sensor_get_status
type*	json
sensor_id*	Alphanumeric value
(*) = required	
Esempi	
<p>http://....?action=sensor_get_status&amp;sensor_id=2222&amp;type=json</p> <p><b>JSON:</b></p> <pre>{   "sensor_id": "2222",   "submissionTime": "2017-04-01 12:16:04",   "currentState": "RUNNING",   ... }</pre>	

#### 5.11.2.4 Protocollo API actuator\_set\_status

Nome: sensor_set_status	
<b>URL:</b> Da stabilire	
Descrizione	
Questa API viene usata dal blocco Smart Decision Support e/o dashboard per impostare lo stato dei sensori (e.g. aggiornare le direttive). L'API sarà autenticata.	
Protocollo: HTTP, Modalità: POST	
Parametri	
action*	sensor_set_status
type*	json
sensor_id*	Alphanumeric value
(*) = required	
Esempi	
Ricezione direttive relative allo stato del sensore.	
<p>http://....?action=sensor_set_status&amp;sensor_id=2222&amp;type=json</p> <p><b>JSON:</b></p> <pre>{   "sensor_id": "2222",   "submissionTime": "2017-04-01 12:16:04",   "currentState": "OFF",   ... }</pre>	

### 5.11.3 Distributed Smart City Engine Scheduler, DISCES (T19) (UNIFI)



<h2 style="margin: 0;">Profilo Tool/Algoritmo</h2>	
Distributed Smart City Engine Scheduler, DISCES (T19)	
Nome responsabile	
Partner responsabile	DISIT-UNIFI
Descrizione Tool/algoritmo	Scheduler distribuito di processi e chiamate REST con supporto a vincoli di esecuzione hardware/software Strumento per lo scheduling dei processi di ingestione, aggregazione, integrazione, verifica, validazione, quality improvement, arricchimento, riconciliazione, data analytic, big data, processi esterni, etc.
Dati primari in ingresso	Job data
Dati prodotti in Uscita	Esecuzione del Job
Principi (metodo, basi teoriche, etc.)	I Job vengono eseguiti secondo la temporizzazione prescelta dal primo nodo libero del cluster, con supporto alla non-concorrenza
Casi di test (presenti/assenti)	assenti
Posizione casi di test	
Principali problemi non risolti	

Principali requisiti pendenti		
Aspetti Tecnologici		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Web app	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java	
Piattaforme supportate	Linux, Mac, Windows, Unix	
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	http://[hostname]:8080/SmartCloudEngine	
Indirizzo/i accesso via web (se presenti)	http://[hostname]:8080/SmartCloudEngine	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
RDF Store	SPARQL Query	HTTP REST
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
JSON	RDF Store	
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
HTTP REST	RDF Store	
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
quartz	Database per la persistenza dei dati dello scheduler	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
Web	PHP	
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
Quartz	2.2.1	Apache 2.0

### 5.11.3.1 Protocollo esposto

Il modulo espone una interfaccia REST su protocollo HTTP per l'accesso alle funzioni di scheduling.

**Nome:** checkExistJob

<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per controllare l'esistenza di un Job sullo scheduler.	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22checkExistJob%22,%22jobName%22:%22job1%22,%22jobGroup%22:%22group1%22}	
<b>JSON Result:</b>	
{ "0":["response"], "1":"false" }	

<b>Nome:</b> checkExistTrigger	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per controllare l'esistenza di un Trigger sullo scheduler.	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
triggerName	Il nome del trigger
triggerGroup	Il gruppo del trigger
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22checkExistTrigger%22,%22jobName%22:%22trigger1%22,%22triggerGroup%22:%22group1%22}	
<b>JSON Result:</b>	
{ "0":["response"], "1":"false" }	

<b>Nome:</b> clear, clearScheduler	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per cancellare tutti i job e i trigger presenti sullo scheduler	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
<b>Esempio</b>	
<a href="http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22clear%22}">http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22clear%22}</a>	
<b>Result:</b>	
<p>true</p>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22	
<b>JSON Result:</b>	

```
{ "0":["response"],"1":"true" }
```

<b>Nome:</b> deleteCalendar	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per cancellare un calendario presente sullo scheduler	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
calendarName	Il nome del calendario
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22, %22 calendarName %22:%22 %22}	
<b>Result:</b>	
<p>true</p>	

<b>Nome:</b> deleteJob	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per cancellare un job dallo scheduler, e tutti i trigger associati	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22,%22 jobName%22:%22job1%22, %22jobGroup%22:%22 %22}	
<b>JSON Result:</b>	
{ "0":["response"],"1":"true" }	

<b>Nome:</b> getCurrentlyExecutingJobs	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere la lista dei job in esecuzione	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22}	

**JSON Result:**  
 {"0":["response"],"1":{"0":{"job1\\","group1\\","Sat Mar 12 14:07:44 CET 2016\\}}}}

<b>Nome:</b> getTriggersOfJob	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere la lista dei trigger di un job	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22 } <b>JSON Result:</b> {"0":["response"],"1":{"0":{"0":{"Ljava.lang.String;@2257088e}}}}	

<b>Nome:</b> getJobDetail	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere i dettagli di un job	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22, %22jobName%22:%22 %22, %22jobGroup%22:%22 %22} <b>JSON Result:</b> {"#isNonConcurrent":["{"processPath\\":\\"\\opt\\jdk1.8.0_51\\bin\\java\\"}, {"cp\\":\\"- classpath\\"}, {"lib\\":\\"\\var\\www\\html\\sce\\test\\lib\\*\\"}, {"InsertTweets\\":\\"inserttwee ts.InsertTweets\\"}, {"insertTweets.properties\\":\\"\\var\\www\\html\\sce\\test\\lib\\insertTw eets.properties\\"}]}]}	

<b>Nome:</b> getJobGroupNames	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere la lista dei gruppi	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22}	

**JSON Result:**  
 {"0":"InsertTweetsRecommender","1":"TwitterVigilanceIndexing"}

<b>Nome:</b> getPausedTriggerGroups	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere la lista dei trigger in pausa	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22getPausedTriggerGroups%22}	
<b>JSON Result:</b>	
{"0":"InsertTweetsRecommender","1":"TwitterVigilanceIndexing"}	

<b>Nome:</b> getSchedulerInstanceId	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere l'id dell'istanza dello scheduler	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22%22}	
<b>Result:</b>	
debian1456734408633	

<b>Nome:</b> getSchedulerName	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere il nome dell'istanza dello scheduler	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22getSchedulerName%22}	
<b>Result:</b>	
SCE	

<b>Nome:</b> getTriggerGroupNames	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	



API per chiedere la lista di tutti i trigger	
Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22getTriggerGroupNames%22}	
<b>JSON Result:</b> { "0": "9cc52dfb-8360-4f75-9ed2-1c0c61530f46", "1": "4b2cf714-cd6c-4265-8b2e-702a80150736" }	

<b>Nome:</b> getTriggerKeys	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere la lista di tutti i trigger di un gruppo	
Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:
triggerGroup	Il nome del gruppo
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22getTriggerKeys%22,%22triggerGroup%22:%22group1%22}	
<b>JSON Result:</b> { "0": "9cc52dfb-8360-4f75-9ed2-1c0c61530f46", "1": "4b2cf714-cd6c-4265-8b2e-702a80150736" }	

<b>Nome:</b> getTriggersOfJob	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere la lista di tutti i trigger di un job	
Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22getTriggersOfJob%22,%22jobName%22:%22job1%22,%22jobGroup%22:%22group1%22}	
<b>JSON Result:</b> { "0": ["response"], "1": { "0": [Ljava.lang.String;@4710c620] } }	

<b>Nome:</b> getTriggerState	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere lo stato di un trigger	

Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:
triggerName	Il nome del trigger
triggerGroup	Il gruppo del trigger
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22getTriggerState%22,%22triggerName%22:%22trigger1%22,%22triggerGroup%22:%22group1%22}	
<b>Result:</b>	
NONE	

<b>Nome:</b> interruptJob	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere l'interruzione di tutte le istanze di un job	
Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22,%22jobName%22:%22job1%22,%22jobGroup%22:%22group1%22}	
<b>JSON Result:</b>	
{ "0":["response"],"1":"true" }	

<b>Nome:</b> interruptJobInstance	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere l'interruzione di un'istanza di un job	
Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
fireInstanceid	L'id dell'istanza del job
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22,%22jobName%22:%22job1%22,%22jobGroup%22:%22group1%22,%22fireInstanceid%22:%22debian14567344086331456734410234%22}	
<b>JSON Result:</b>	
{ "0":["response"],"1":"true" }	

<b>Nome:</b> interruptJobs	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere l'interruzione di tutti i job in esecuzione	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22interruptJobs%22}	
<b>JSON Result:</b>	
{ "0":["response"],"1":"true" }	

<b>Nome:</b> interruptFireInstanceId	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere l'interruzione di un'istanza di un job	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
fireInstanceId	L'id dell'istanza del job
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22interruptFireInstanceId%22,%22fireInstanceId%22:%22debian14567344086331456734410234%22}	
<b>Result:</b>	
<p>true</p>	

<b>Nome:</b> isInStandbyMode	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere se lo scheduler è in standby	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22isInStandbyMode%22}	
<b>JSON Result:</b>	
{ "0":["response"],"1":"false" }	

<b>Nome:</b> isShutdown	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	

API per chiedere se lo scheduler è shutdown	
Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22isShutdown%22}	
<b>JSON Result:</b> { "0":["response"], "1":"false" }	

<b>Nome:</b> isStarted	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere se lo scheduler è in esecuzione	
Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22isStarted%22}	
<b>JSON Result:</b> { "0":["response"], "1":"true" }	

<b>Nome:</b> pauseAll	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere di mettere in pausa tutti i trigger (compresi quelli futuri)	
Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22pauseAll%22}	
<b>JSON Result:</b> { "0":["response"], "1":"true" }	

<b>Nome:</b> pauseJob	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere di mettere in pausa un job	
Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:

jobName	Il nome del job
jobGroup	Il gruppo del job
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22pauseJob%22,%22jobName%22:%22job1%22,%22jobGroup%22:%22group1%22}	
<b>JSON Result:</b>	
{"0":["response"],"1":"true"}	

<b>Nome:</b> pauseJobs	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere di mettere in pausa i job di un gruppo	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
jobGroup	Il gruppo del job
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22}	
<b>Result:</b>	
<p>true</p>	

<b>Nome:</b> pauseTrigger	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere di mettere in pausa un trigger	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
triggerName	Il nome del trigger
triggerGroup	Il gruppo del trigger
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22,%22triggerName%22:%22 %22,%22triggerGroup%22:%22 %22}	
<b>Result:</b>	
{"0":["response"],"1":"true"}	

<b>Nome:</b> pauseTriggers	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere di mettere in pausa i trigger di un gruppo	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	

id	L'id dell'azione richiesta:
groupName	Il nome del gruppo
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22,%22groupName%22:%22 %22}	
<b>Result:</b>	
<p>true</p>	

<b>Nome:</b> rescheduleJob	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere il rescheduling di un job	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
withIdentityNameGroup	Il nomi del job e del gruppo separati da un punto (.)
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22,%22withIdentityNameGroup%22:%22job1. %22}	
<b>JSON Result:</b>	
{"0":["response"],"1":"true"}	

<b>Nome:</b> resumeAll	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere il resume di tutti i job e i trigger	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22}	
<b>JSON Result:</b>	
{"0":["response"],"1":"true"}	

<b>Nome:</b> resumeJob	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere il resume di un job	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	

id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22, %22jobName%22:%22 %22, %22jobGroup%22:%22 %22}	
<b>JSON Result:</b> { "0":["response"], "1":"true" }	

<b>Nome:</b> resumeJobs	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere il resume dei job di un gruppo	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
groupName	Il nome del gruppo
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22,%22 groupName%22:%22 %22}	
<b>JSON Result:</b> { "0":["response"], "1":"true" }	

<b>Nome:</b> resumeTrigger	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere il resume di un trigger	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
triggerName	Il nome del trigger
triggerGroup	Il gruppo del trigger
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22,%22 triggerName%22:%22 %22,%22 triggerGroup%22:%22 %22}	
<b>JSON Result:</b> { "0":["response"], "1":"true" }	

<b>Nome:</b> resumeTriggers	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere il resume dei trigger di un gruppo	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:

groupName	Il nome del gruppo
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22%22,%22groupName%22:%22%22}	
<b>Result:</b> <p>true</p>	

<b>Nome:</b> shutdownScheduler	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere lo shutdown dello scheduler	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
waitForJobsToComplete	Se è true lo scheduler è messo in shutdown al completamento dei job, in caso contrario immediatamente.
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22%22,%22waitForJobsToComplete%22:%22%22}	
<b>JSON Result:</b> {"0":["response"],"1":"true"}	

<b>Nome:</b> standbyScheduler	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere lo standby dello scheduler	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22standbyScheduler%22}	
<b>JSON Result:</b> {"0":["response"],"1":"true"}	

<b>Nome:</b> startScheduler	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere lo start dello scheduler	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	



id	L'id dell'azione richiesta:
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22startScheduler%22}	
<b>JSON Result:</b>	
{"0":["response"],"1":"true"}	

<b>Nome:</b> startDelayed	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere lo start dello scheduler ritardato	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
seconds	I secondi di ritardo
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22,%22seconds%22:%22 %22}	
<b>Result:</b>	
<p>true</p>	

<b>Nome:</b> truncateCatalinaLog	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere il truncate del log di Catalina (Tomcat)	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22}	
<b>JSON Result:</b>	
{"0":["response"],"1":"true"}	

<b>Nome:</b> triggerJob	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere l'esecuzione di un job	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
jobName	Il nome del job

jobGroup	Il gruppo del job
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22, %22jobName%22:%22 %22, %22jobGroup%22:%22 %22}	
<b>JSON Result:</b>	
{"0":["response"],"1":"true"}	

<b>Nome:</b> unscheduleJob	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere la rimozione di un job dallo scheduler	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22, %22jobName%22:%22 %22, %22jobGroup%22:%22 %22}	
<b>JSON Result:</b>	
{"0":["response"],"1":"true"}	

<b>Nome:</b> listJobTriggers	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere la lista dei trigger di un job	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22, %22jobName%22:%22 %22, %22jobGroup%22:%22 %22}	
<b>JSON Result:</b>	
{"0":["Trigger Name","Trigger Group","Calendar Name","Description","End Time","Final Fire Time","Misfire Instruction","Next Fire Time","Previous Fire Time","Priority","Start Time","May Fire Again"],"1":["7e369086-851f-4191-927f-229327d0f44c","4b2cf714-cd6c-4265-8b2e-702a80150736","","","","0","Thu Mar 17 12:00:00 CET 2016","Thu Mar 17 06:00:00 CET 2016","5","Thu Dec 10 12:00:00 CET 2015","true"]}	

<b>Nome:</b> updateJobDataMap	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere la lista dei trigger di un job	
Modalità: GET, POST	

Parametri Obbligatori	
id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
jobDataMap	Il job data map in formato JSON
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22, %22jobName%22:%22 %22, %22jobGroup%22:%22 %22, %22jobDataMap%22:%22 %22}	
<b>JSON Result:</b> {"0":["response"],"1":"true"}	

Nome: addJob	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
Descrizione	
API per chiedere di aggiungere un job allo scheduler	
Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
jobClass	Il job in formato JSON
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22, %22jobName%22:%22 %22, %22jobGroup%22:%22 %22, %22jobClass%22:%22 %22}	
<b>JSON Result:</b> {"0":["response"],"1":"true"}	

Nome: getNotificationEmail	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
Descrizione	
API per chiedere l'email di notifica associata al job	
Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22 %22, %22jobName%22:%22 %22, %22jobGroup%22:%22 %22}	
<b>JSON Result:</b> {"0":["response"],"1":"true"}	

Nome: getJobDataMap	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	

Descrizione	
API per chiedere il data map di un job in formato JSON	
Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
Esempio	
<pre>http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22getJobDataMap%22,%22jobName%22:%22    %22,%22jobGroup%22:%22    %22}</pre>	
<b>JSON Result:</b> <pre>{ "#processParameters": [{"processPath": "\\opt\\jdk1.8.0_51\\bin\\java"}, {"cp": "-classpath"}, {"lib": "\\var\\www\\html\\sce\\test\\lib\\*"}, {"TwitterIndexing": "twitterindexing.TwitterIndexing"}, {"twitter.properties": "\\var\\www\\html\\sce\\test\\lib\\twitter.properties"}], "#isNonConcurrent": "true" }</pre>	

Nome: getJobFireTimes	
URL: http://hostname:8080/SmartCloudEngine	
Descrizione	
API per chiedere i tempi di firing di un job	
Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
Esempio	
<pre>http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22getJobFireTimes%22,%22jobName%22:%22    %22,%22jobGroup%22:%22    %22}</pre>	
<b>JSON Result:</b> <pre>{"startTime": "Thu Dec 10 12:00:00 CET 2015", "state": "BLOCKED", "previousFireTime": "Thu Dec 10 12:00:00 CET 2015" }</pre>	

Nome: getJobTriggers	
URL: http://hostname:8080/SmartCloudEngine	
Descrizione	
API per chiedere i trigger di un job	
Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
Esempio	
<pre>http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22getJobTriggers%22,</pre>	

%22jobName%22:%22 %22, %22jobGroup%22:%22 %22}
<b>JSON Result:</b> { "0":["7e369086-851f-4191-927f-229327d0f44c", "4b2cf714-cd6c-4265-8b2e-702a80150736"]}

<b>Nome:</b> getSchedulerMetadata	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere i metadati dello scheduler	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22getSchedulerMetadata%22}	
<b>JSON Result:</b> { "Scheduler instance id":["debian1456734408633", "Reports the instance id of the scheduler"], "Running since":["Mon Feb 29 09:26:48 CET 2016", "Reports the date at which the scheduler started running"], "Number of jobs executed":[1826, "Reports the number of jobs executed since the scheduler started"], "Scheduler started":["yes", "Reports whether the scheduler has been started"], "JobStore supports persistence":["yes", "Reports whether or not the scheduler's JobStore instance supports persistence"], "Remote Scheduler":["no", "Reports whether the scheduler is being used remotely (via RMI)"], "Scheduler shutdown":["no", "Reports whether the scheduler has been shutdown"], "Standby mode":["no", "Reports whether the scheduler is in standby mode"], "JobStore Clustered":["yes", "Reports whether or not the scheduler's JobStore is clustered"], "Scheduler name":["SCE", "Reports the name of the scheduler"]}	

<b>Nome:</b> getSystemStatus	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere lo stato del nodo dello scheduler	
Modalità: GET, POST	
<b>Parametri Obbligatori</b>	
id	L'id dell'azione richiesta:
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22getSystemStatus%22}	
<b>JSON Result:</b> { "Operating System version":["3.16.0-4-amd64", "Reports the operating system version"], "Operating System architecture":["amd64", "Returns the operating system architecture"], "CPU load (JVM)":["6.318334157463876E-4", "Returns the recent cpu usage for the Java Virtual Machine process. This value is a double in the [0.0, 1.0] interval. A value of 0.0 means that none of the CPUs were running threads from the JVM process during the recent period of time observed, while a value of 1.0 means that all CPUs were actively running threads from the JVM 100% of the time during the recent period being observed. Threads from the JVM include the application threads as well as the JVM internal threads. All values between 0.0 and 1.0 are possible depending of the activities going on in the JVM process and the whole system. If the Java Virtual	

Machine recent CPU usage is not available, the value reports a negative value"], "Number of processors": ["8", "Reports the number of processors available to the Java virtual machine"], "Process CPU time": ["5379360000000", "Returns the cpu time used by the process on which the Java virtual machine is running in nanoseconds. The returned value is of nanoseconds precision but not necessarily nanoseconds accuracy. This value reports -1 if the platform does not support this operation"], "Total physical memory": ["5.0756255744E10", "Returns the total amount of physical memory in bytes"], "CPU load": ["0.129306082083402", "Returns the recent cpu usage for the whole system. This value is a double in the [0.0, 1.0] interval. A value of 0.0 means that all CPUs were idle during the recent period of time observed, while a value of 1.0 means that all CPUs were actively running 100% of the time during the recent period being observed. All values between 0.0 and 1.0 are possible depending of the activities going on in the system. If the system recent cpu usage is not available, the value reports a negative value"], "Free physical memory": ["32348704768", "Reports the amount of free physical memory in bytes"], "Committed virtual memory": ["6919090176", "Reports the amount of virtual memory that is guaranteed to be available to the running process in bytes, or -1 if this operation is not supported"], "Free swap space": ["7839490048", "Reports the amount of free swap space in bytes"], "Total swap space": ["8.585736192E9", "Returns the total amount of swap space in bytes"], "System Load average": ["1.31", "Reports the system load average for the last minute. The system load average is the sum of the number of runnable entities queued to the available processors and the number of runnable entities running on the available processors averaged over a period of time. The way in which the load average is calculated is operating system specific but is typically a damped time-dependent average.\n\nIf the load average is not available, a negative value is returned.\n\nThis value is designed to provide a hint about the system load and may be queried frequently. The load average may be unavailable on some platform where it is expensive to implement this method"], "Operating System name": ["Linux", "Reports the operating system name"]}

<b>Nome:</b> getConnectionPoolInfo	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere le informazioni del connection pool per il database	
Modalità: GET, POST	
<b>Parametri Obbligatorii</b>	
id	L'id dell'azione richiesta:
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22getConnectionPoolInfo%22}	
<b>JSON Result:</b>	
{ "MaxIdle":8,"MinIdle":0,"MaxActive":10,"NumActive":0,"TestOnReturn":false,"TestWhileIdle":false,"SoftMinEvictableIdleTimeMillis":-1,"TestOnBorrow":false,"TimeBetweenEvictionRunsMillis":-1,"MaxWait":-1,"WhenExhaustedAction":2,"NumIdle":1,"Lifo":true,"MinEvictableIdleTimeMillis":1800000,"NumTestsPerEvictionRun":3 }	

<b>Nome:</b> buildTriggerForJob	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere la creazione di un trigger per un job	

Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:
jobName	Il nome del job
jobGroup	Il gruppo del job
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%buildTriggerForJob%22,%22jobName%22:%job1%22,%22jobGroup%22:%group1%22}	
<b>Result:</b>	
<p>>true</p>	

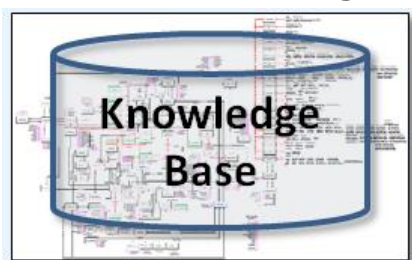
<b>Nome:</b> setJobProgress	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere di impostare la percentuale di avanzamento di un job	
Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:
fire_instance_id	Il fire instance id del job
progress	La percentuale di avanzamento del job
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22,%22 fire_instance_id %22:%job1%22,%22progress%22:%30%22}	
<b>Result:</b>	
<p>>true</p>	

<b>Nome:</b> getJobProgress	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere la percentuale di avanzamento di un job	
Modalità: GET, POST	
Parametri Obbligatori	
id	L'id dell'azione richiesta:
fire_instance_id	Il fire instance id del job
Esempio	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22,%22 fire_instance_id %22:%job1%22}	
<b>Result:</b>	
<p>30</p>	

<b>Nome:</b> getJobProgress	
<b>URL:</b> http://hostname:8080/SmartCloudEngine	
<b>Descrizione</b>	
API per chiedere la percentuale di avanzamento di un job	
Modalità: GET, POST	
Parametri Obbligatori	

id	L'id dell'azione richiesta:
fire_instance_id	Il fire instance id del job
<b>Esempio</b>	
http://192.168.0.23:8080/SmartCloudEngine/index.jsp?json={%22id%22:%22,%22 fire_instance_id %22:%22job1%22}	
<b>Result:</b>	
<p>10</p>	

#### 5.11.4 Knowledge Base RDFStore (dati statici e real time) (T20) (UNIFI)

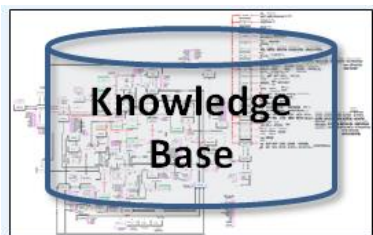


<b>Profilo Tool/Algoritmo</b>	
Knowledge Base RDFStore (dati statici e real time)	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	The Km4City model based of the city for static and dynamic data. RDF store including ontology and data.
Dati primari in ingresso	triple in vario formato e query
Dati prodotti in Uscita	risultati delle query SPARQL
Principi (metodo, basi teoriche, etc.)	Questo RDF store dovrà supportare query geografiche, query full-text, inferenza RDFS, essere un quad store e dovrà supportare protocolli standard definiti nella specifica W3C
Casi di test (presenti/assenti)	assenti
Posizione casi di test	n/a
Principali problemi non risolti	nessuno
Principali requisiti pendenti	nessuno
<b>Aspetti Tecnologici</b>	
Stato (proposto/approvato)	proposto
Implementato/non implementato	non implementato
Stato implementazione, percentuale	0%
Eseguibile/libreria/web app	Eseguibile
Single thread/Multithread	Multithread
Linguaggio di sviluppo (java, Php, ETL)	C
Piattaforme supportate	Linux, Windows
Posizione del codice sorgente	n/a
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se	



necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
SPARQL 1.1 HTTP Protocol,		
SPARQL 1.1 Graph Protocol		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

### 5.11.5 Knowledge Base Statistic RDFStore (dati statistici) (T21) (UNIFI)

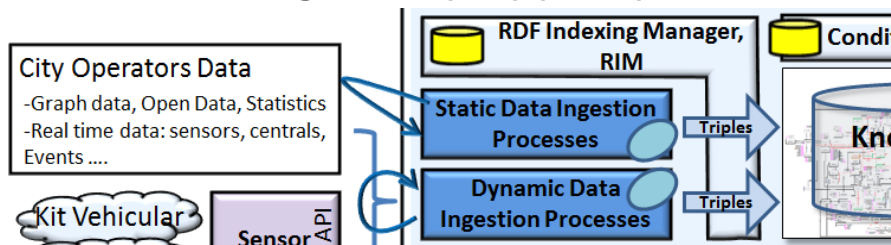


<b>Profilo Tool/Algoritmo</b>	
Knowledge Base Statistic RDFStore (dati statistici)	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	The Km4City model based of the city for statistic data. RDF store including ontology and data.
Dati primari in ingresso	triple in vario formato, query SPARQL

Dati prodotti in Uscita	risposte a query SPARQL	
Principi (metodo, basi teoriche, etc.)	RDF store con stesse caratteristiche di Knowledge Base RDFStore	
Casi di test (presenti/assenti)	assenti	
Posizione casi di test	n/a	
Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	C	
Piattaforme supportate	Linux, Windows	
Posizione del codice sorgente	n/a	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
N3, RDF/XML, Turtle, NTriple		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
SPARQL 1.1 HTTP Protocol		
SPARQL 1.1 Graph Protocol		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

Virtuoso	7.2.1	GPL

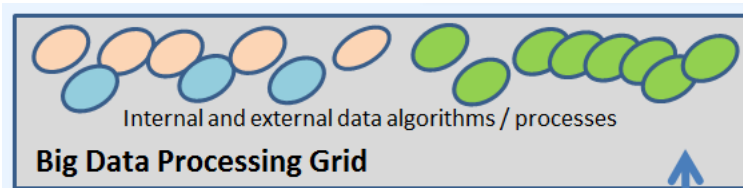
### 5.11.6 RDF Indexer Manager, RIM (T22) (UNIFI)



<h2 style="margin: 0;">Profilo Tool/Algoritmo</h2> <h3 style="margin: 0;">RDF Indexer Manager, RIM</h3>		
Nome responsabile		
Partner responsabile	UNIFI	
Descrizione Tool/algoritmo	Strumento per l'indicizzazione delle triple negli RDF store e loro versioning	
Dati primari in ingresso	Triple RDF da indicizzare	
Dati prodotti in Uscita	Bash script che verrà utilizzato per generare e popolare l'RDF Store.	
Principi (metodo, basi teoriche, etc.)	Preparazione di un indice di triple RDF chiamato indice descrittore mediante una interfaccia web di tipo "passo passo" dove l'utente sarà in grado di selezionare diversi set di triple ("set di dati") per la costruzione ed il versionamento di RDF Store	
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	n/d	
Principali problemi non risolti	Assenti	
Principali requisiti pendenti	Nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0.0%	
Eseguibile/libreria/web app	Web app	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	PHP, Javascript, HTML, CSS, Bash	
Piattaforme supportate	Linux, Windows	
Posizione del codice sorgente	Svn	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
Nomi tool/moduli usati <small>(aggiungere una riga per ogni tool/modulo usato)</small>	Interfacce API usate	Modello di comunicazione e

		formato
isql-vt	n/a	CLI - SPARQL
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
JSON		JavaScript Object Notation
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
HTTP		HyperText Transfer Protocol
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	Descrizione	
MySQL	Database relazionale	
Virtuoso	RDF Store	
<b>Tipo interfaccia utente</b> (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
Web	HTML, JavaScript, CSS	Bootstrap
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
JQuery	1.10.1	Licenza MIT
Bootstrap	3.1.1	Licenza MIT

### 5.11.7 Moduli e processo di aggregazione / riconciliazione (A08) (UNIFI)



<b>Profilo Tool/Algoritmo</b>	
Moduli e processo di aggregazione / riconciliazione	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Studio e sviluppo di algoritmi di aggregazione / riconciliazione verso il noSQL database e la base di conoscenza big data di Sii-Mobility, tecniche di link discovering, etc.
Dati primari in ingresso	Dataset provenienti dai blocchi: Static data ingestion, Dynamic data ingestion, Sensor Server and Manager.
Dati prodotti in Uscita	Serie di Triple, come risultato del mapping tra i

	dataset (riconciliati, arricchiti) e l'ontologia Km4city.	
Principi (metodo, basi teoriche, etc.)	NoSql Database, conoscenza della Km4city Ontology, realizzazione di processi ETL per effettuare le operazioni di: aggregazione, riconciliazione e mapping con l'ontologia km4city.	
Casi di test (presenti/assenti)	Nessuno	
Posizione casi di test	UNIFI, Laboratorio DISIT	
Principali problemi non risolti	Nessuno	
Principali requisiti pendenti	Nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Eseguibili	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	ETL	
Piattaforme supportate	Windows, Unix	
Posizione del codice sorgente	UNIFI, Laboratorio DISIT	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	Da stabilire	
Indirizzo/i accesso via web (se presenti)	Da stabilire	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Pentaho Kettle tool		
Karma Data integration tool		
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
Mysql	Database necessario per il mapping dei dataset nella ontologia Km4city	
Hbase, hadoop	Database necessario per la gestione delle triple (Base di Conoscenza)	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>

Pentaho Kettle tool: interfaccia grafica	Java, js, funzioni pentaho (disponibili nella interfaccia grafica)	
Karma: interfaccia web		
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

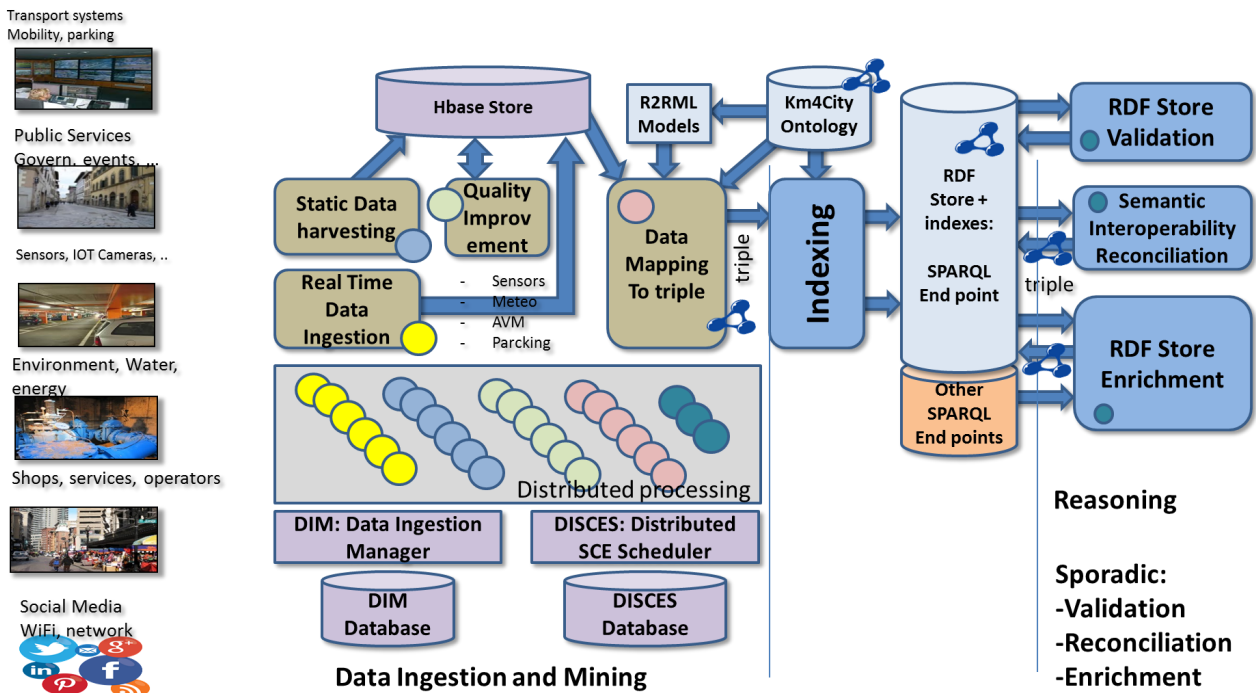
### 5.12 Specifica: Acquisizione dati da enti gestori, LOD, sorgenti esterne

		FCM	Mizar	UNIFI	IN20	TIME	Negetis	EffKnow	liberologico	ataf	busitalia	cttnord	tiemme	argos	elfi	calamai	Midra	Proiect	GEOIN	OIJESTIT	SOFTEC	FWINGS
4.1.3	Studio e sviluppo di moduli acquisizione dati e gestione ingestion process: moduli acquisizione dati, OD, ETL	X	X			X			X								X		X			
4.1.5	Studio e sviluppo di moduli acquisizione dati e gestione ingestion process: quality improvement		X	X		X		X	X								X		X			
4.2.1	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: gestori TPL vari, web service		X			X																
4.2.2	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: OD comuni vari, ETL	X	X			X																
4.2.3	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: LOD vari, LOG			X																		
4.2.5	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: centrale ambiente e meteo	X	X																	X		
4.2.6	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: autostrade e sensori																X					
4.2.7	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: Ferrovie (static e real time)																X					
4.2.9	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: Connessione con protezione civile, emergenza			X																		

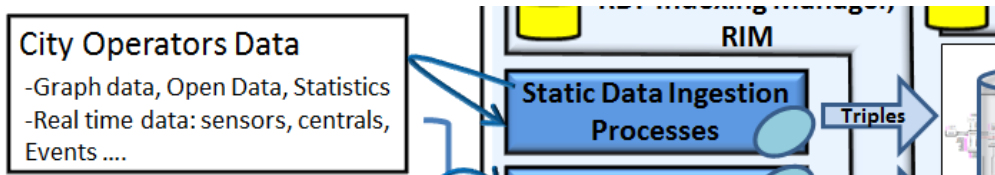
Principali Sottosistemi interessati dai requisiti da descrivere in questa Subsubsection, saranno i sottosistemi primari della specifica					
	Sotto Attività	Partner coord	Sottosistema	Tool/Level	Descrizione sommaria
4.1.3	Studio e sviluppo di moduli acquisizione dati e gestione ingestion process: moduli acquisizione dati, OD, ETL	UNIFI	Data Ingestion	ETL process on Kettle: Static Data Ingestion processi di base (PETL01)	Un set di moduli base per programmi ETL per formalizzare processi di ingestione dei dati, il quality improvement dei dati, la triplication (conversione in triple per RDF store), verifica di consistenza e completezza, etc. Gli ETL vengono messi in esecuzione da DISCES in modo periodico o aperiodico.
		UNIFI	Data Ingestion	Lista Dati da Ingerire ed integrare. (OD01)	Lista degli Open Data e Real Time Data che potrebbero essere necessari nelle varie aree di sperimentazione può essere recuperata dalla tabella in Appendice 1. Possibili protocolli: DATEX II, SIRI, IETF, REST, OD.
4.1.5	Studio e sviluppo di moduli acquisizione dati e gestione ingestion process: quality improvement	MIZAR	Processi ETL su Kettle	Quality improvement (PETL02)	Processi di quality improvement, dati statici nelle varie aree di sperimentazione sviluppati come ETL.

4.2.1	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: gestori TPL vari, web service	MIZAR	Processi ETL su Kettle	Acquisizione dati da gestori TPL vari, web service (PETL03)	Processi di acquisizione dati statici e dinamici dei vari gestori TPL nelle varie aree di sperimentazione, sviluppati come ETL.
4.2.2	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: OD comuni vari, ETL	UNIFI	Processi ETL su Kettle	Acquisizione OD comuni vari, ETL (PETL04)	Processi di acquisizione dati statici e dinamici relativi a Open data di servizi nelle varie aree di sperimentazione, sviluppati come ETL.
4.2.3	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: LOD vari, LOG	UNIFIDISIT	Processi ETL su Kettle	Enrichment via LOD, RDF endpoint (PETL05)	Processi di acquisizione dati statici e dinamici relativi a LOD per enrichment di servizi nelle varie aree di sperimentazione, sviluppati come ETL o Java process.
4.2.5	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: centrale ambiente e meteo	UNIFIDISIT	Processi ETL su Kettle	Acquisizione dati Lamma (PETL06)	Processi di acquisizione dati statici e dinamici relativi a centrale meteo da LAMMA per la regione Toscana nelle varie aree di sperimentazione, sviluppati come ETL.
		UNIFILSTT	Processi ETL su Kettle	Acquisizione dati Ambiente (PETL07)	Processi di acquisizione dati statici e dinamici relativi a ambiente (per esempio da ARPAT) per la regione Toscana nelle varie aree di sperimentazione, sviluppati come ETL. Sviluppo di accordo per l'acquisizione dati o da WEB.
4.2.6	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: autostrade e sensori	MIDRA	Processi ETL su Kettle	Acquisizione dati Autostrade (PETL08)	Processi di acquisizione dati Statici e dinamici relativi a: Tempi di percorrenza, Flussi traffico, Incidenti, Lavori, Interruzioni (per esempio da AUTOSTRADE) per la regione Toscana nelle varie aree di sperimentazione, sviluppati come ETL. Sviluppo di accordo per l'acquisizione dati o da WEB.
4.2.7	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: Ferrovie (static e real time)	MIDRA	Processi ETL su Kettle	Acquisizione Dati Ferrovie (PETL09)	Processi di acquisizione dati Statici e dinamici relativi a: Orari, Ritardi, Anticipi, Soppressioni treni, Interruzioni (per esempio da Ferrovie dello Stato) per la regione Toscana nelle varie aree di sperimentazione, sviluppati come ETL. Sviluppo di accordo per l'acquisizione dati o da WEB.
4.2.9	Acquisizione dati da enti gestori, LOD, sorgenti esterne, da sensori, etc.: Connessione con protezione civile, emergenza	UNIFI	Processi ETL su Kettle	Acquisizione dati per emergenza (PETL10)	Processi di acquisizione dati Statici e dinamici relativi a: Connessione con protezione civile, emergenza, nelle varie aree di sperimentazione, sviluppati come ETL. Sviluppo di accordo per l'acquisizione dati o da WEB.





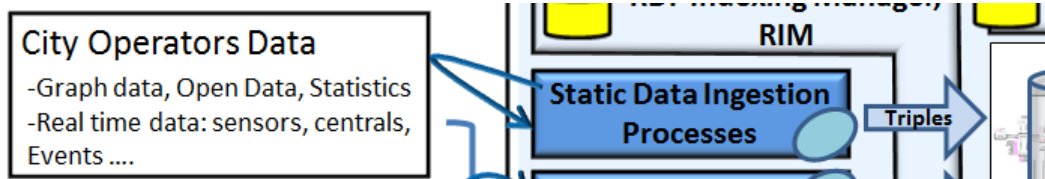
### 5.12.1 Processi ETL Static Data Ingestion, grafo e altri static data, Lista Dati da Ingerire ed integrare (PETL01) (UNIFI)



<h2 style="margin: 0;">Profilo Tool/Algoritmo</h2>	
Processi ETL Static Data Ingestion, grafo e altri static data. Lista Dati da Ingerire ed integrare.	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	<p>Un set di moduli base per programmi ETL per formalizzare processi di ingestione dei dati, il quality improvement dei dati, la triplication (conversione in triple per RDF store), verifica di consistenza e completezza, etc. Gli ETL vengono messi in esecuzione da DISCES in modo periodico o aperiodico.</p> <p>Lista degli Open Data e Real Time Data che potrebbero essere necessari nelle varie aree di sperimentazione può essere recuperata dalla tabella in Appendice 1.</p> <p>Possibili protocolli: DATEX II, SIRI, IETF, REST, OD.</p>
Dati primari in ingresso	Data set da ingerire
Dati prodotti in Uscita	Triple RDF relative ai dataset ingeriti
Principi (metodo, basi teoriche, etc.)	Gli ETL vengono messi in esecuzione da DISCES in modo periodico o aperiodico.

Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	n/d	
Principali problemi non risolti	Assenti	
Principali requisiti pendenti	Nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0.0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	ETL	
Piattaforme supportate	Unix, Windows	
Posizione del codice sorgente	svn	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Ingestion	n/a	REST API, CSV, KMZ, SHP
Quaity Improvement	n/a	n/a
Triplification	n/a	n/a
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
CSV		comma-separated values
KMZ/KML		Keyhole Markup Language
SHP		Shape format file
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTP		HyperText Transfer Protocol
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
MySql	Database relazionale	
HBase	Database NoSql distribuito su filesystem Hadoop	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
n/a		
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
n/a		

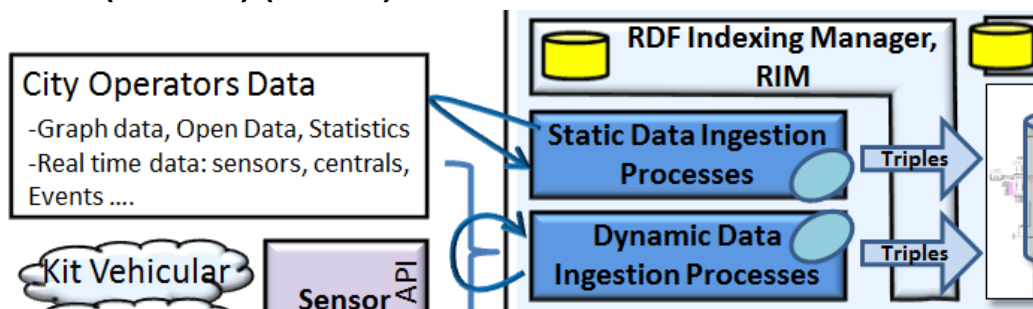
### 5.12.2 Processi ETL per: Quality improvement (PETL02) (MIZAR)



Profilo Tool/Algoritmo		
Processi ETL per: Quality improvement		
Nome responsabile		
Partner responsabile	MIZAR	
Descrizione Tool/algoritmo	Processi di quality improvement, dati statici nelle varie aree di sperimentazione sviluppati come ETL. Gli algoritmi specificati verranno implementati nei singoli ETL collegati ai sottosistemi.	
Dati primari in ingresso	Dati dai sottosistemi	
Dati prodotti in Uscita	Dati per SII	
Principi (metodo, basi teoriche, etc.)	L'algoritmo valida i dati di ingresso applicando delle regole di filtraggio per i campioni "fuori profilo"; inoltre in caso di buchi di dati in ingresso per brevi periodi, l'algoritmo ricostruisce i dati mancanti sulla base delle informazioni storiche e dei dati ricevuti prima e dopo la mancanza dei dati	
Casi di test (presenti/assenti)	Nessuno	
Posizione casi di test	n.a.	
Principali problemi non risolti	n.a.	
Principali requisiti pendenti	n.a.	
Aspetti Tecnologici		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non Implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Solo specifiche	
Single thread/Multithread	n.a.	
Linguaggio di sviluppo (java, Php, ETL)	ETL	
Piattaforme supportate	n.a.	
Posizione del codice sorgente	n.a.	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.a.	
Indirizzo/i accesso via web (se presenti)	n.a.	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
n.a.	n.a.	n.a.

Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
n.a.	n.a.	n.a.
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
n.a.	n.a.	n.a.
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
n.a.	n.a.	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
n.a.	n.a.	n.a.
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
n.a.	n.a.	n.a.

### 5.12.3 Processi ETL per acquisizione dati da: gestori TPL vari, web service (PETL03) (MIZAR)

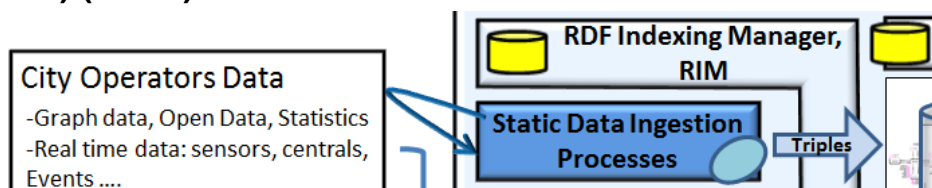


<h2 style="margin: 0;">Profilo Tool/Algoritmo</h2>	
Processi ETL per acquisizione dati da: gestori TPL vari, web service	
Nome responsabile	
Partner responsabile	MIZAR
Descrizione Tool/algoritmo	Processi di acquisizione dati statici e dinamici dei vari gestori TPL nelle varie aree di sperimentazione, sviluppati come ETL. Dati tpl si intende: <ul style="list-style-type: none"> <li>■ Descrizione percorsi e corse, orari, posizione paline,</li> <li>■ RealTime: posizione mezzi</li> </ul> MIZAR ha gestori locati in Firenze, Prato, Pistoia, Pisa (CTTNORD), Siena, Arezzo (TIEMME), per oltre

	700 mezzi. Vano sviluppati moduli per tutte queste centrali.	
Dati primari in ingresso	Dati relativi ad orari ed in tempo reale per i mezzi di trasporto pubblico	
Dati prodotti in Uscita	Dati relativi ad orari ed in tempo reale per i mezzi trasporto pubblico	
Principi (metodo, basi teoriche, etc.)	I dati relativi ad orari ed in tempo reale per i mezzi trasporto pubblico vengono letti nel formato proprietario del sistema di trasporto pubblico e trasformati in modelli di dato normalizzati per la piattaforma SII	
Casi di test (presenti/assenti)	nessuno	
Posizione casi di test	n.a.	
Principali problemi non risolti	n.a.	
Principali requisiti pendenti	nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Specifica generiche, i moduli sono prodotti nei task successivi (PETL15-PETL18)	
Single thread/Multithread	n.a.	
Linguaggio di sviluppo (java, Php, ETL)	ETL	
Piattaforme supportate	n.a.	
Posizione del codice sorgente	n.a.	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.a.	
Indirizzo/i accesso via web (se presenti)	n.a.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n.a.	n.a.	n.a.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
n.a.	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
n.a.	n.a.	n.a.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
n.a.	n.a.	

Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
n.a.	n.a.	n.a.
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
n.a.	n.a.	n.a.

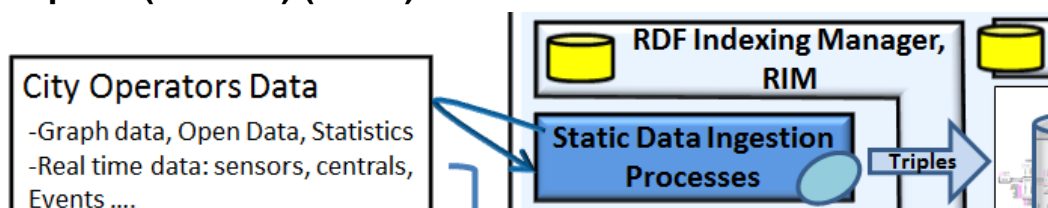
### 5.12.4 Processi ETL per acquisizione dati da: OD comuni vari, ETL (PETL04) (UNIFI)



<h2 style="text-align: center;">Profilo Tool/Algoritmo</h2>	
Processi ETL per acquisizione dati da: OD comuni vari, ETL	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Processi di acquisizione dati statici e dinamici relativi a Open data di servizi nelle varie aree di sperimentazione, sviluppati come ETL (Extraction transforming Loading).
Dati primari in ingresso	Open Data messi a disposizione dai comuni.
Dati prodotti in Uscita	Dati rielaborati tramite processi di: ingestion, Quality improvement, triplication (conversione in triple per RDF store), verifica di consistenza e completezza, etc. I dati così trattati risulteranno arricchiti dal punto di vista semantico e integrati in un'unica Knowledge Base, in accordo con la multi-ontologia km4city.
Principi (metodo, basi teoriche, etc.)	Uso di strumenti per la realizzazione dei processi ETL, uso di strumenti per la modellazione semantica dei dataset
Casi di test (presenti/assenti)	Assenti
Posizione casi di test	n/d
Principali problemi non risolti	Assenti
Principali requisiti pendenti	Nessuno
Aspetti Tecnologici	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0.0%
Eseguibile/libreria/web app	Eseguibile

Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Tool grafico per la realizzazione di ETL, uso di javascript, java, SPARQL, SQL	
Piattaforme supportate	Unix, Windows	
Posizione del codice sorgente	svn	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>SPARQL 1.1 HTTP Protocol</b>		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

### 5.12.5 Processi ETL per acquisizione dati da: Enrichment via LOD, RDF end point (PETL05) (UNIFI)



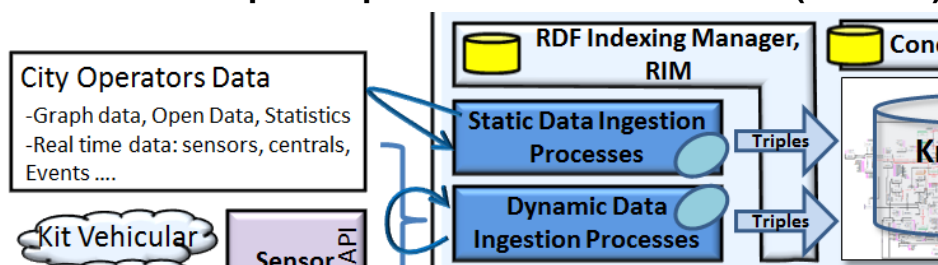
## Profilo Tool/Algoritmo

Processi per acquisizione dati da: LOD vari, LOG		
Nome responsabile		
Partner responsabile	UNIFI	
Descrizione Tool/algorithmo	Processi di acquisizione dati statici e dinamici relativi a LOD per enrichment di servizi nelle varie aree di sperimentazione, sviluppati come ETL o Java process.	
Dati primari in ingresso	dati presenti su rdf store	
Dati prodotti in Uscita	triple di collegamento con dbpedia	
Principi (metodo, basi teoriche, etc.)	Un sistema di analisi del testo per la identificazione dei nomi applicato ad un insieme di attributi per individuare nomi in alcune data properties per effettuare un collegamento con le entità di dbpedia	
Casi di test (presenti/assenti)	assenti	
Posizione casi di test	n/a	
Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Eseguibile Java + web app	
Single thread/Multithread	Mono thread	
Linguaggio di sviluppo (java, Php, ETL)	Java	
Piattaforme supportate	Quelle supportate da Java	
Posizione del codice sorgente	n/a	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/a	
Indirizzo/i accesso via web (se presenti)	n/a	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
RDF Store		SPARQL 1.1 HTTP Protocol
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
SPARQL 1.1 HTTP Protocol		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	



nomi	Contiene i nomi identificati nelle varie proprietà	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
Web	JSP, HTML, Javascript	Jquery
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
GATE	7.0	LGPL

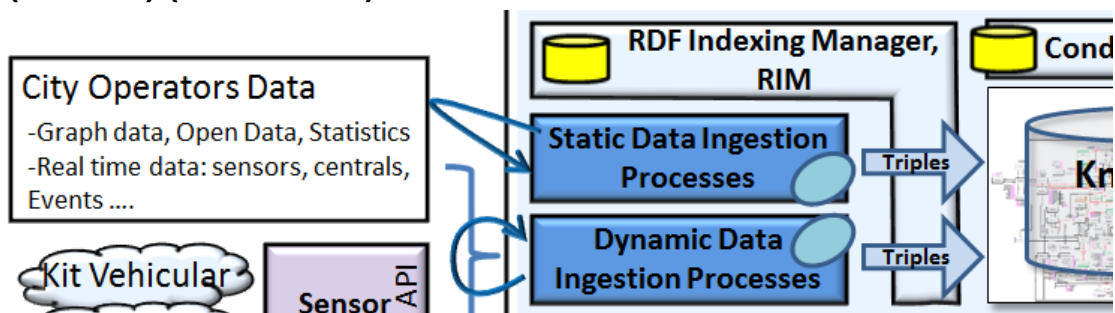
### 5.12.6 Processi ETL per acquisizione dati da: Lamma (PETL06) (UNIFI)



<h2 style="text-align: center;">Profilo Tool/Algoritmo</h2> <h3 style="text-align: center;">Processi ETL per acquisizione dati da: Lamma</h3>	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Processi di acquisizione dati statici e dinamici relativi a centrale meteo da LAMMA per la regione Toscana nelle varie aree di sperimentazione, sviluppati come ETL.
Dati primari in ingresso	Dati messi a disposizione dal Lamma relativi alle previsioni del tempo, relativi ai comuni della Toscana. I dati saranno aggiornati con cadenza di 12 ore.
Dati prodotti in Uscita	Dati rielaborati tramite processi di: ingestion, Quality improvement, triplication (conversione in triple per RDF store), verifica di consistenza e completezza, etc. I dati così trattati risulteranno arricchiti dal punto di vista semantico e integrati in un'unica Knowledge Base, in accordo con la multi-ontologia km4city.
Principi (metodo, basi teoriche, etc.)	Uso di strumenti per la realizzazione dei processi ETL, uso di strumenti per la modellazione semantica dei dataset
Casi di test (presenti/assenti)	Assenti

Posizione casi di test	n/d	
Principali problemi non risolti	Assenti	
Principali requisiti pendenti	Nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0.0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Tool grafico per la realizzazione di ETL, uso di javascript, java, SPARQL, SQL	
Piattaforme supportate	Unix, Windows	
Posizione del codice sorgente	svn	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
SPARQL 1.1 HTTP Protocol		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
HBase		
MySQL		
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

### 5.12.7 Processi ETL per acquisizione dati da: Ambiente, ARPAT (PETL07) (UNIFI.LSTT)



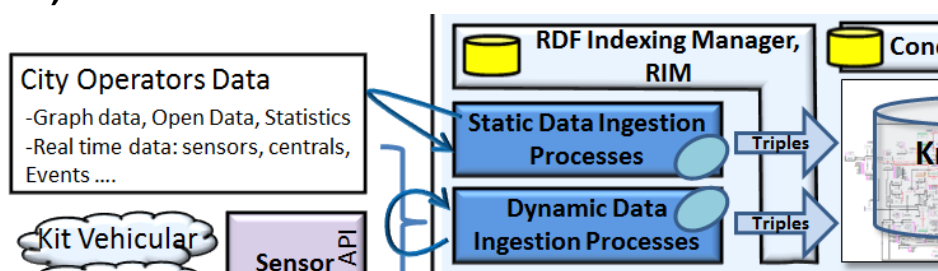
## Profilo Tool/Algoritmo

### Processi ETL per acquisizione dati da: Ambiente, ARPAT

Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Un set di programmi ETL per formalizzare processi di ingestione dei dati, il quality improvement dei dati, la triplificazione (conversione in triple per RDF store), verifica di consistenza e completezza, etc
Dati primari in ingresso	Data Set reso disponibile da ARPAT. In particolare, acquisizione di (i) dati statici sul posizionamento delle stazioni di monitoraggio e di (ii) dati dinamici sulla qualità dell'aria (es. concentrazione PM10, PM2.5, NO2, SO2, CO, Benzene e H2S).
Dati prodotti in Uscita	Triple RDF che rappresentano i dati del dataset in input
Principi (metodo, basi teoriche, etc.)	Gli ETL vengono messi in esecuzione da DISCES in modo periodico o aperiodico, in base alla tipologia dei dati
Casi di test (presenti/assenti)	Assenti
Posizione casi di test	n/d
Principali problemi non risolti	Assenti
Principali requisiti pendenti	Nessuno
Aspetti Tecnologici	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0.0%
Eseguibile/libreria/web app	Eseguibile
Single thread/Multithread	Multithread
Linguaggio di sviluppo (java, Php, ETL)	ETL
Piattaforme supportate	Unix, Windows
Posizione del codice sorgente	Svn
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d
Indirizzo/i accesso via web (se presenti)	n/d

Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
Ingestion	n/a	REST API, CSV
Quaity Improvement	n/a	n/a
Triplification	n/a	n/a
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
CSV		comma-separated values
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
HTTP		HyperText Transfer Protocol
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
MySql	Database relazionale	
HBase	Database NoSql distribuito su filesystem Hadoop	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

### 5.12.8 Processi ETL per acquisizione dati da: Autostrade (PETL08) (MIDRA)



## Profilo Tool/Algoritmo

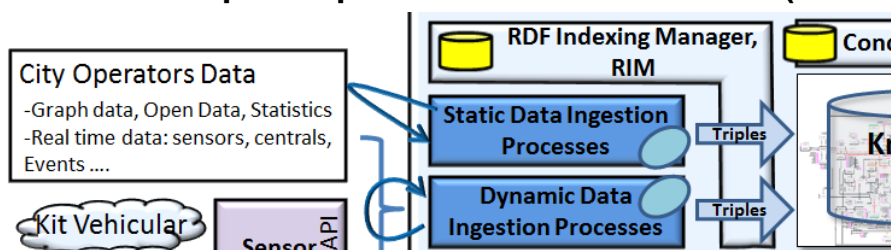
Processi ETL per acquisizione dati da: Autostrade

Nome responsabile	
Partner responsabile	MIDRA
Descrizione Tool/algoritmo	Interfaccia tra sistema di controllo accessi Autostrade (TELEPASS) e database SII Mobility Processi di acquisizione dati Statici e dinamici relativi a: Tempi di percorrenza , Flussi traffico, Incidenti, Lavori, Interruzioni (per esempio da AUTOSTRADE) per la regione Toscana nelle varie aree di sperimentazione,

	sviluppati come ETL. Sviluppo di accordo per l'acquisizione dati o da WEB.	
Dati primari in ingresso	Informazioni su transazioni TELEPASS e accesso alla rete autostradale, tempi di percorrenza e velocità di transito rilevate via TUTOR.	
Dati prodotti in Uscita	Statistiche di utilizzo del servizio e dati per reportistica traffico, utilizzabili per generare stime di QoS per navigazione assistita e monitoraggio (con possibilità di integrazione con informazioni di traffico per altre infrastrutture, es. ferrovie).	
Principi (metodo, basi teoriche, etc.)	Non applicabile	
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	n/d	
Principali problemi non risolti	Assenti	
Principali requisiti pendenti	Interfaccia di scambio con il sistema SII Mobility	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0	
Eseguitabile/libreria/web app	Eseguitabile + web app	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	C, HTML, PHP, SQL	
Piattaforme supportate	Windows Server, Linux	
Posizione del codice sorgente	Git	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Apache Web Server	APACHE	HTTP/CGI
MySQL connector for PHP	MYSQL	SQL
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
Da definire		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTP		Hypertext Transfer Protocol
TCP/IP		
ad-hoc AUTOSTRADE		TELEPASS
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
MySQL	Database relazionale	

Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
Web	HTML, PHP, JavaScript, CSS	
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
MySQL		GPL

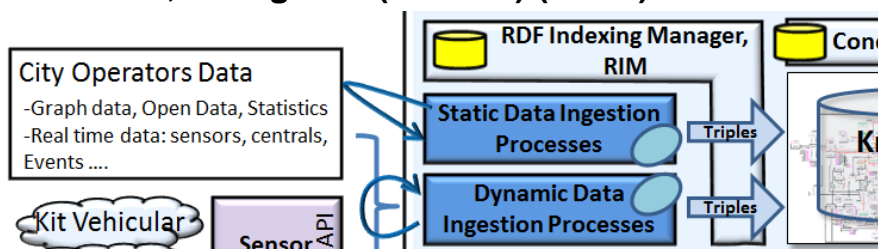
### 5.12.9 Processi ETL per acquisizione dati da: Ferrovie (PETL09) (MIDRA)



Profilo Tool/Algoritmo	
Processi ETL per acquisizione dati da: Ferrovie	
Nome responsabile	
Partner responsabile	MIDRA
Descrizione Tool/algoritmo	Processi di acquisizione dati Statici e dinamici relativi a: Orari, Ritardi, Anticipi, Soppressioni treni, Interruzioni (per esempio da Ferrovie dello Stato) per la regione Toscana nelle varie aree di sperimentazione, sviluppati come ETL. Sviluppo di accordo per l'acquisizione dati o da WEB. Interfaccia tra sistema di monitoraggio traffico e database SII Mobility
Dati primari in ingresso	Informazioni sul traffico ferroviario e flussi di percorrenza. Dati da server rete ferroviaria e da sensori lungo linea per monitoraggio stato infrastrutture.
Dati prodotti in Uscita	Informazioni dettagliate in database riguardanti lo stato del servizio ferroviario a livello nazionale, interfacciabile con sistema Autostrade/TELEPASS per integrazione in sistemi di navigazione assistita
Principi (metodo, basi teoriche, etc.)	Non applicabile
Casi di test (presenti/assenti)	Assenti
Posizione casi di test	n/d
Principali problemi non risolti	Assenti
Principali requisiti pendenti	Interfaccia di scambio con il sistema SII Mobility
Aspetti Tecnologici	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato

Stato implementazione, percentuale	0	
Eseguibile/libreria/web app	Eseguibile/web app	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	C, HTML, PHP, SQL	
Piattaforme supportate	Windows Server, Linux	
Posizione del codice sorgente	Git	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
Apache Web Server	APACHE	HTTP/CGI
MySQL connector for PHP	MYSQL	SQL
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
Da definire		
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
HTTP		HyperText Transfer Protocol
TCP/IP		
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
MySQL	Database relazionale	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
Web	HTML, PHP, JavaScript, CSS	
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
MySQL		GPL

### 5.12.10 Processi ETL per acquisizione dati da: Connessione con protezione civile, emergenza (PETL10) (UNIFI)



<b>Profilo Tool/Algoritmo</b>		
Processi ETL Static Data Ingestion, grafo e altri static data. Lista Dati da Ingerire ed integrare.		
Nome responsabile		
Partner responsabile	UNIFI	
Descrizione Tool/algoritmo	Processi di acquisizione dati Statici e dinamici relativi a: Connessione con protezione civile,	
Dati primari in ingresso	Data set da ingerire	
Dati prodotti in Uscita	Triple RDF relative ai dataset ingeriti	
Principi (metodo, basi teoriche, etc.)	Gli ETL vengono messi in esecuzione da DISCES in modo periodico o aperiodico.	
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	n/d	
Principali problemi non risolti	Assenti	
Principali requisiti pendenti	Nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0.0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	ETL	
Piattaforme supportate	Unix, Windows	
Posizione del codice sorgente	svn	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
Ingestion	n/a	REST API, CSV, KMZ, SHP
Quaity Improvement	n/a	n/a
Triplification	n/a	n/a
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
CSV		comma-separated values
KMZ/KML		Keyhole Markup Language
SHP		Shape format file
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
HTTP		HyperText Transfer Protocol
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
MySql	Database relazionale	



HBase	Database NoSql distribuito su filesystem Hadoop	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
n/a		
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
n/a		

### 5.13 Specifica: CrowdSourcing, Qualificazione delle sorgenti, integrazione multilingua

		FCM	Mizar	UNIFI	IN20	TIME	Neentis	EffKnow	liberologico	ataf	bustalia	cttnord	tiemme	areos	elfi	calamai	Midra	Proiect	GEOIN	OUJESTIT	SOFTEC	FWINGS
4.3.1	CrowdSourcing, Qualificazione delle sorgenti, integrazione multilingua: get/set data twitter			X																		
4.3.2	CrowdSourcing, Qualificazione delle sorgenti, integrazione multilingua: NLP processing: twitter e altri text comments			X																		
4.3.3	CrowdSourcing, Qualificazione delle sorgenti, integrazione multilingua: profile integration SN			X																		
4.3.4	CrowdSourcing, Qualificazione delle sorgenti, integrazione multilingua: deduction system			X																		

Principali Sottosistemi interessati dai requisiti da descrivere in questa Subsubsection, saranno i sottosistemi primari della specifica					
	Sotto Attività	Partner coord	Sottosistema	Tool/Level	Descrizione sommaria
4.3.1	CrowdSourcing, Qualificazione delle sorgenti, integrazione multilingua: get/set data twitter	UNIFI	Social Media Monitoring	Social Media Analysis Interface (T23)	Interfaccia web autenticata per gestire l'accesso ai dati che arrivano da social media crawler.
4.3.2	CrowdSourcing, Qualificazione delle sorgenti, integrazione multilingua: NLP processing: twitter e altri text comments	UNIFI	Social Media Monitoring	Social Media Crawler and manager (T24)	Tool per il crawling di social media con varie configurazioni. Per esempio Twitter. La soluzione si integra con il sistema di Crowd Sourcing della soluzione Sii-Mobility.
4.3.3	CrowdSourcing, Qualificazione delle sorgenti, integrazione multilingua: profile integration SN	UNIFI	Crowd collector	User Crowd Sourcing Manager UCSMANE (T25)	Supporto per la partecipazione, produzione di richieste geolocalizzate e contestualizzate e profilate. L'utente deve poter fornire associati ai servizi: score, commenti e immagini.
		UNIFI	User engager	User Engager, UCSMANE (T26)	Il sistema deve poter definire regole di ingaggio verso l'utente che possono scattare sulla base del suo comportamento, del contesto, del tempo, della posizione, dello stato, etc. Suggestioni per un comportamento sano, gestione bonus, etc.
4.3.4	CrowdSourcing, Qualificazione delle sorgenti, integrazione multilingua: deduction system	UNIFI	User Profiling	User Profiler, UPROSOD (T27)	User profiling. Il Profilo utente deve poter capire le abitudini di vari cluster di utenti: cittadini, turisti, studenti, pendolari. Per abitudini si intende: orari, tipo di veicolo utilizzato, luoghi importanti, percorsi tipici di spostamento in città, etc.

		UNIFI	Suggestion on demand	Suggestion on Demand, UPROSOD (T28)	supporto per la produzione di suggestion on demand. Produzione sulla base del profilo di suggerimenti personalizzati per gli utenti e le categorie, stimoli per la partecipazione alla smart city (contributi, immagini, commenti, advertising, etc.), informazioni di emergenza (protezione civile, note sul traffico, etc.), suggerimenti di guida connessa (segnaletica, semafori, servizi, parcheggi, TPL, colonnine, etc. )
--	--	-------	----------------------	-------------------------------------	--

### 5.13.1 Social Media Analysis Interface (T23) (UNIFI)

## Social Media Analysis Interface

Uso e parziale sviluppo di questo modulo già in possesso di UNIFI

<b>Profilo Tool/Algoritmo</b>	
Social Media Analysis Interface	
Nome responsabile	
Partner responsabile	UNIFI.DISIT
Descrizione Tool/algoritmo	Il tool Social Media Analysis Interface consente agli utenti (prevedendo due tipologie di accesso differenziate: autenticato e non) la realizzazione e gestione di canali per ricerche tematiche mirate su social media (per keyword, hashtags ecc.), nonché analisi statistiche dei dati ingeriti, visualizzazione di trend temporali e valutazione di Sentiment Analysis sui dati di crowdsourcing collezionati dal tool Social Media Crawler and Manager (5.13.2).
Dati primari in ingresso	Flusso di dati generato dal tool Social Media Crawler and Manager (5.13.2); Dati testuali in linguaggio naturale (ad esempio post collezionati da microblog, Twitter e altri social media).
Dati prodotti in Uscita	Dati statistici su utenti e messaggi, post collezionati e analizzati in base ai canali e alle chiavi di ricerca scelte; dati di polarità di Natural Language Processing (orientato allo Sentiment Analysis) visualizzati come grafici customizzabili che ne descrivono i trend nel tempo.
Principi (metodo, basi teoriche, etc.)	Algoritmi statistici e di calcolo, algoritmi di Natural Language Processing (ad esempio POS-Tagging, Word Sense Disambiguation; Named Entity Recognition ecc.).
Casi di test (presenti/assenti)	Sono previsti, in via di definizione, casi di test mirati all'assessment delle principali funzionalità previste dal tool Social Media Analysis Interface: gestione degli utenti, dei canali e dei parametri di

	ricerca, valutazione della produzione e integrazione di informazioni di Sentiment Analysis al fine di individuare e monitorare eventi critici e di interesse.	
Posizione casi di test	.	
Principali problemi non risolti	.	
Principali requisiti pendenti	.	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Web application.	
Single thread/Multithread	Multithread, con processi gestiti tramite uno scheduler engine dedicato.	
Linguaggio di sviluppo (java, Php, ETL)	Php, Java, Javascript	
Piattaforme supportate	Linux, Windows	
Posizione del codice sorgente	.	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	.	
Indirizzo/i accesso via web (se presenti)	.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Social Media Crawler and Manager (5.13.2)	Rest API; Php and Java servlet API per HTTP request; MySQL connector driver.	- JSON. Esempio di formato: "docs": [ { "id": "data": "message": "autore": "ID_chiave_ricerca": "Count_answer_repost": "location": ... } ] - Accesso e condivisione dati su database relazionale MySQL.
User Crowd Sourcing Manager & Engager (5.13.3)	MySQL connector driver.	Accesso e condivisione dati su database relazionale MySQL.
SmartDS (5.1.8)	MySQL connector driver.	Accesso e condivisione dati su database relazionale MySQL.
Dashboard Builder and Manager (5.1.9)	MySQL connector	Accesso e condivisione dati

	driver.	su database relazionale MySQL.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
Dati testuali non strutturati o semi-strutturati (ingeriti da file testo o da web)	Social Media Crawler and Manager (5.13.2)	
JSON		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
TCP		
HTTP		
REST	Social Media Crawler and Manager (5.13.2)	
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	Descrizione	
social_media_content_extraction	Database MySQL per l'archiviazione dei dati processati relativi a Natural Language Processing e Sentiment Analysis	
<b>Tipo interfaccia utente</b> (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
Web application con accesso autenticato differenziato	PHP, Java, Javascript	Highcharts, Bootstrap
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
GATE (Java Suite per Text Engineering e Natural Language Processing)	GATE v7.1	GNU Lesser General Public License
TreeTagger (Tool per annotazioni testuali)	TreeTagger	Open Source per uso non commerciale

### 5.13.2 Social Media Crawler and manager (T24) (UNIFI)



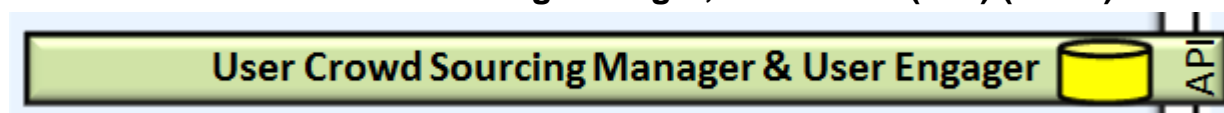
Uso e parziale sviluppo di questo modulo già in possesso di UNIFI

<b>Profilo Tool/Algoritmo</b>	
Social Media Crawler and manager	
Nome responsabile	
Partner responsabile	DISIT.UNIFI
Descrizione Tool/algoritmo	Lo scheduler lancia n thread di processi crawling

	sull'host centrale e sui nodi slave. Ogni thread recupera dal db centrale le chiavi di ricerca che usa per interrogare il social media ad esempio Twitter. I dati ottenuti vengono memorizzati sul cluster hbase.	
Dati primari in ingresso	Chiavi di ricerca	
Dati prodotti in Uscita	Social messages ad esempio Tweets, User profiles	
Principi (metodo, basi teoriche, etc.)	Algoritmi di scheduling, Programmazione concorrente, Paradigma OO, Programmazione Client-Server	
Casi di test (presenti/assenti)	n/a	
Posizione casi di test	n/a	
Principali problemi non risolti	n/a	
Principali requisiti pendenti	n/a	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0.0 %	
Eseguibile/libreria/web app	Codice interpretato	
Single thread/Multithread	Multi-thread	
Linguaggio di sviluppo (java, Php, ETL)	PHP	
Piattaforme supportate	Unix /Windows /OS X	
Posizione del codice sorgente	Svn disit	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/a	
Indirizzo/i accesso via web (se presenti)	n/a	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Scheduler	n/a	n/a
TwitterCrawlerHBase	n/a	n/a
Logger	n/a	n/a
TwitterAPIExchange	Twitter API 1.1	REST
HBaseAPIExchange	HBASE API 2.0	REST
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTP	n/a	Hyper Trasnfert Protocol
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	

Mysql	Database basato su modello relazionale	
HBase	Database distribuito basato su modello google big table	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
applicazione	php	n/a
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
TwitterAPIExchange	n/a	MIT Licence
HBaseAPIExchange	0.1	Proprietaria (DISIT)

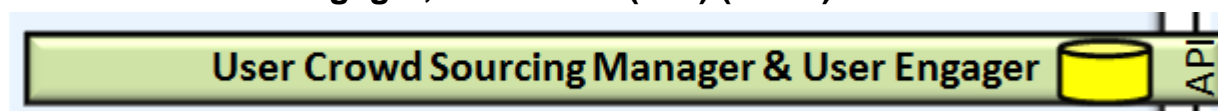
### 5.13.3 User Crowd Sourcing Manager, UCSMANE (T25) (UNIFI)



Profilo Tool/Algoritmo	
User Crowd Sourcing Manager, UCSMANE	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Supporto per la validazione della partecipazione, produzione di richieste geolocalizzate e contestualizzate e profilate. L'utente deve poter fornire associati ai servizi: score, commenti e immagini.
Dati primari in ingresso	dati forniti dagli utenti
Dati prodotti in Uscita	validazione dei dati
Principi (metodo, basi teoriche, etc.)	Interfaccia che permette la validazione dei dati sottomessi dagli utenti: foto, commenti. I dati forniti possono essere accettati o rifiutati inoltre per le foto è possibile ruotarle.
Casi di test (presenti/assenti)	assenti
Posizione casi di test	n/a
Principali problemi non risolti	nessuno
Principali requisiti pendenti	nessuno
Aspetti Tecnologici	
Stato (proposto/approvato)	proposto
Implementato/non implementato	non implementato
Stato implementazione, percentuale	0%
Eseguibile/libreria/web app	web app
Single thread/Multithread	multi thread
Linguaggio di sviluppo (java, Php, ETL)	Java JSP
Piattaforme supportate	Quelle supportate da Java
Posizione del codice sorgente	n/a

Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/a	
Indirizzo/i accesso via web (se presenti)	n/a	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
ServiceMap	informazioni sottomesse dagli utenti attraverso le API (foto, commenti, stelle)	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
Web	Java JSP	HTML
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
jQuery		MIT License

### 5.13.4 User Engager, UCSMANE (T26) (UNIFI)



<b>Profilo Tool/Algoritmo</b>	
User Engager, UCSMANE	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Il sistema deve poter definire regole di ingaggio verso l'utente che possono scattare sulla base del suo comportamento, del contesto, del tempo, della posizione, dello stato, etc. Suggerimenti per un comportamento sano, etc.



Dati primari in ingresso	Identificativo, stato, profilo e lingua del device dell'utente, indicazione geografica	
Dati prodotti in Uscita	Engagement dell'utente in formato JSON	
Principi (metodo, basi teoriche, etc.)	Il sistema applica al contesto degli utenti attivi un set di regole precedentemente inserito	
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test		
Principali problemi non risolti	Nessuno	
Principali requisiti pendenti	Nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	20%	
Eseguibile/libreria/web app	Libreria	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java	
Piattaforme supportate	Piattaforme supportate da Java	
Posizione del codice sorgente	svn	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	Assente	
Indirizzo/i accesso via web (se presenti)	Assente	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
ServiceMap	Services	HTTP REST
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON	ServiceMap	
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTP REST	ServiceMap	HTTP REST
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
Sensor	Dati sullo stato dell'utente	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale,</b>

		etc.
Drools	org.drools 6.3.0.Final	Apache License 2.0

### 5.13.5 User Profiler, UPROSOD (T27) (UNIFI)



<b>Profilo Tool/Algoritmo</b> User Profiler, UPROSOD	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	<p>User profiling. Il Profilo utente deve poter capire le abitudini di vari cluster di utenti: cittadini, turisti, studenti, pendolari. Per abitudini si intende: orari, tipo di veicolo utilizzato, luoghi importanti, percorsi tipici di spostamento in città, etc. Nel profilo utente si devono memorizzare anche informazioni come:</p> <ul style="list-style-type: none"> <li>• Profilo, lingua, età presunta, sesso presunto</li> <li>• Azioni effettuate, di recente e tipiche</li> <li>• Come si muove (con che mezzo), dove abita, etc.</li> <li>• Che cosa cerca, che cosa guarda</li> <li>• Sui contributi come immagini, commenti, voti, etc.</li> <li>• Bonus ricevuti</li> <li>• Comperamenti ottimi che hanno prodotto buoni/bonus</li> <li>• Stato di test per recommender, per engager, per buoni, etc.</li> <li>• Etc.</li> </ul> <p>Il tool dovrebbe mostrare la lista utenti, con dei filtri e ricerche, ed alcune statistiche di base.</p>
Dati primari in ingresso	dati di posizione e preferenze degli utenti
Dati prodotti in Uscita	analisi del profilo utente
Principi (metodo, basi teoriche, etc.)	Il modulo fornisce un'analisi del profilo utente in modo da poter fornire raccomandazioni specifiche
Casi di test (presenti/assenti)	assenti
Posizione casi di test	
Principali problemi non risolti	nessuno
Principali requisiti pendenti	nessuno
Aspetti Tecnologici	
Stato (proposto/approvato)	proposto

Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	web app	
Single thread/Multithread	single thread	
Linguaggio di sviluppo (java, Php, ETL)	Java, PHP	
Piattaforme supportate	Windows, Linux, Unix	
Posizione del codice sorgente	svn	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>User profile</b>		REST HTTP
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>JSON</b>		JavaScript Object Notation
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>HTTP</b>		HyperText Transfer Protocol
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>MySQL</b>	DBMS	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>web</b>		JQuery
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

### 5.13.6 Suggestion on Demand, UPROSOD (T28) (UNIFI)



<h2 style="margin: 0;">Profilo Tool/Algoritmo</h2> <p style="margin: 0;">Suggestion on Demand, UPROSOD</p>	
Nome responsabile	

Partner responsabile	UNIFI	
Descrizione Tool/algoritmo	supporto per la produzione di suggestion on demand. Produzione sulla base del profilo di suggerimenti personalizzati per gli utenti e le categorie, stimoli per la partecipazione alla smart city (contributi, immagini, commenti, advertising, etc.), informazioni di emergenza (protezione civile, note sul traffico, etc.), suggerimenti di guida connessa (segnaletica, semafori, servizi, parcheggi, TPL, colonnine, etc. )	
Dati primari in ingresso	posizione e preference degli utenti	
Dati prodotti in Uscita	produzione di suggerimenti personalizzati	
Principi (metodo, basi teoriche, etc.)	Il modulo produce suggerimenti date le preferenze dell'utente	
Casi di test (presenti/assenti)	assenti	
Posizione casi di test		
Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	web app	
Single thread/Multithread	single thread	
Linguaggio di sviluppo (java, Php, ETL)	Java, PHP	
Piattaforme supportate	Windows, Linux, Unix	
Posizione del codice sorgente	svn	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
suggestion		HTTP REST
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON		JavaScript Object Notation
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
REST HTTP		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	

MySQL	DBMS	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
web		jQuery
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

### 5.14 Specifica: strumenti di pubblicazione dati

		ECM	Mizar	UNIFI	IN20	TIME	Negentis	Effknow	liberologico	ataf	busitalia	cttnord	tiemme	areos	elfi	calamai	Midra	Proiert	GEOIN	OUJSTIT	SOITEC	FWINIGS
4.4.1	Strumenti di Pubblicazione e navigazione nei dati integrati, pubblicazione: RDF store access, API			X																		
4.4.2	Strumenti di Pubblicazione e navigazione nei dati integrati, pubblicazione: rendering on MAPs			X						X	X	X	X									
4.4.3	Strumenti di Pubblicazione e navigazione nei dati integrati, pubblicazione: rendering dati con varie view non solo geo grafiche		X	X		X				X	X	X	X						X		X	
4.4.4	Strumenti di Pubblicazione e navigazione nei dati integrati, pubblicazione: generazione view interattive per totem, web, e mobile		X			X				X	X	X	X						X		X	
4.4.5	Strumenti di Pubblicazione e navigazione nei dati integrati, pubblicazione: monitoraggio azioni utente, integrazione profiling		X	X						X	X	X	X						X		X	
4.4.6	Strumenti di Pubblicazione e navigazione nei dati integrati, pubblicazione: gestione utenza del sistema		X			X				X	X	X	X						X		X	

Principali Sottosistemi da descrivere in questa Subsubsection, sono i sottosistemi primari della specifica					
	Sotto Attività	Partner coord	Sottosistema	Tool/Level	Descrizione sommaria
4.4.1	Strumenti di Pubblicazione e navigazione nei dati integrati, pubblicazione: RDF store access, API	UNIFI	Sii-Mobility Developer Tools	SPARQL RDF store interface (T29)	Strumento per la formalizzazione delle query in SPARQL verso RDF endpoint. Questo strumento integra un supporto per la verifica delle licenze rispetto alle query effettuate ed al profilo utente.
		UNIFI	Sii-Mobility Developer Tools	Linked Open Graph, LOG (T30)	Strumento per la navigazione su RDF store e LD/LOD che permette al programmatore di applicazioni Sii-Mobility di salvare configurazioni e condividerle, ma anche di comprendere il modello e fare delle query navigando su RDF store multipli.
4.4.2	Strumenti di Pubblicazione e navigazione nei dati integrati, pubblicazione: rendering on MAPs	UNIFI	Sii-Mobility Developer Tools	Service Map, ServiceMap (T31)	Strumento per la navigazione (geografica, logica, NLP) sulla base delle informazioni contenute sulla Knowledge Base. Strumento di produzione delle query (Smart City Api) per web app. Mobile e totem, etc., che le invia via email. SM permette anche di produrre IFRAME per l'embedding di segmenti ServiceMap con query dentro a pagine web di terzi.

		UNIFI	Sii-Mobility Developer Tools	Statistic Data Map, StatisticMap (T32)	Strumento per la navigazione (geografica e statistica temporale) sulla base delle informazioni contenute sulla Knowledge Base statistica. Strumento di produzione delle viste statistiche sui dati tramite web. Per esempio distribuzioni di inquinamento, di traffico, di meteo, incidenti, etc.
4.4.3	Strumenti di Pubblicazione e navigazione nei dati integrati, pubblicazione: rendering dati con varie view non solo geo grafiche	GEOIN	Sii-Mobility Developer Tools	rendering dati con varie view non solo geo grafiche (T33)	A seconda della tipologia del dato e dell'utente cui si indirizza si dovranno avere rese diverse, ad esempio lo stesso dato potrà apparire in forma tabellare per l'utente avanzato e sotto forma di grafico o tema su mappa per l'utente comune.
		GEOIN	City User Tools, CUT	Web App Sii-Mobility per internet browser (T34)	Web App per internet browser che permetta l'accesso ai dati di Sii-Mobility e che sfrutta i servizi di Sii-Mobility. In questo caso il fuoco è sulla presentazione dei dati, di visioni statistiche di insieme, di informazioni tipo Dashboard, etc.
4.4.4	Strumenti di Pubblicazione e navigazione nei dati integrati, pubblicazione: generazione view interattive per totem, web, e mobile	SOFTECH	Sii-Mobility Developer Tools	generazione view interattive per totem, web, e mobile (T35)	Sviluppo di moduli specifici per la visualizzazione dati e viste su Totem o mobile.
		SOFTECH	City User Tools, CUT	Web App Sii-Mobility per Totem (T36)	Totem che permetta l'accesso ai dati di Sii-Mobility e che sfrutti i servizi di Sii-Mobility. In questo caso il fuoco è sulla presentazione dei dati, di visioni statistiche di insieme, di informazioni tipo Dashboard, etc. E' di interesse anche raccogliere commenti e tracciare il comportamento dell'utente.
4.4.5	Strumenti di Pubblicazione e navigazione nei dati integrati, pubblicazione: monitoraggio azioni utente, integrazione profiling	UNIFI.D ISIT	City User Tools, CUT	Monitoraggio azioni utente su Web, Totem, User Profiling (T37)	monitoraggio azioni utente, integrazione profiling, sugli strumenti di pubblicazione.... Vedasi profiling in Section 7.6.
4.4.6	Strumenti di Pubblicazione e navigazione nei dati integrati, pubblicazione: gestione utenza del sistema	TIME	Support of Integrated Interoperability	Authentication Support Tool (T38)	Supporto per l'autenticazione centralizzata ai servizi del cittadino e dei tool di Sii-Mobility. Supporto per l'accesso condizionato a tutti gli strumenti Sii-Mobility: DIM, RIM, DBUM, etc. tramite user name e password e social media.

### 5.14.1 SPARQL RDF store interface (T29) (UNIFI)

#### SPARQL RDF Store Interface

Profilo Tool/Algoritmo		
SPARQL RDF store interface		
Nome responsabile		
Partner responsabile	UNIFI	
Descrizione Tool/algoritmo	Strumento per la formalizzazione delle query in SPARQL verso RDF endpoint. Questo strumento integra un supporto per la verifica delle licenze rispetto alle query effettuate ed al profilo utente.	
Dati primari in ingresso	Query SPARQL	
Dati prodotti in Uscita	risultati delle query e valutazioni della query sulla base delle licenze dei dati usati nella query.	
Principi (metodo, basi teoriche, etc.)	La valutazione della licenza viene fatta usando un servizio esposto da Conditional Access Data	
Casi di test (presenti/assenti)	assenti	
Posizione casi di test	n/a	
Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	web app	
Single thread/Multithread	multi thread	
Linguaggio di sviluppo (java, Php, ETL)	PHP, HTML, Javascript	
Piattaforme supportate	Quelle supportate da PHP	
Posizione del codice sorgente	n/a	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/a	
Indirizzo/i accesso via web (se presenti)	n/a	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
RDF Store		SPARQL 1.1 HTTP Protocol
Conditional Access API (sparql checker)		HTTP Protocol
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
SPARQL XML results		
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione



SPARQL 1.1 HTTP Protocol		
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
web	HTML, Javascript	
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
Flint		MIT License
CodeMirror		MIT License

## 5.14.2 Linked Open Graph, LOG (T30) (UNIFI)

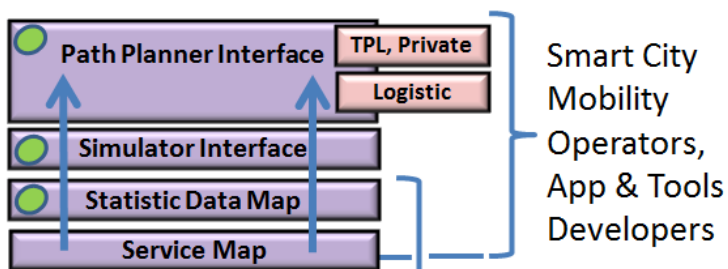
### Linked Open Graph

Uso e parziale sviluppo di questo modulo già in possesso di UNIFI

<b>Profilo Tool/Algoritmo</b>	
Linked Open Graph, LOG	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Strumento per la navigazione su RDF store e LD/LOD che permette al programmatore di applicazioni Sii-Mobility di salvare configurazioni e condividerle, ma anche di comprendere il modello e fare delle query navigando su RDF store multipli.
Dati primari in ingresso	RDFStore su cui navigare
Dati prodotti in Uscita	Viasualizzazione a grafo dei dati presenti
Principi (metodo, basi teoriche, etc.)	Il tool visualizza un grafo delle relazioni tra entità presenti un uno o più RDF Store ma anche dati accessibili in modo Linked Data.
Casi di test (presenti/assenti)	assenti
Posizione casi di test	n/a
Principali problemi non risolti	nessuno
Principali requisiti pendenti	nessuno
Aspetti Tecnologici	
Stato (proposto/approvato)	proposto
Implementato/non implementato	non implementato
Stato implementazione, percentuale	0%
Eseguitabile/libreria/web app	web app
Single thread/Multithread	Multithread
Linguaggio di sviluppo (java, Php, ETL)	PHP

Piattaforme supportate	Quelle supportate da PHP e mysql	
Posizione del codice sorgente	n/a	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/a	
Indirizzo/i accesso via web (se presenti)	log.disit.org	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
SPARQL XML Results		
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
HTTP		
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
LOG	contiene informazioni di configurazione e grafi che possono essere salvati, recuperati e condivisi con altri utenti.	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
Web	PHP, HTML, javascript	javascript D3, JQuery
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
D3		BSD Licence
JQuery		MIT Licence
PHP SPARQL Lib		LGPL

### 5.14.3 Service Map, ServiceMap (T32) (UNIFI)

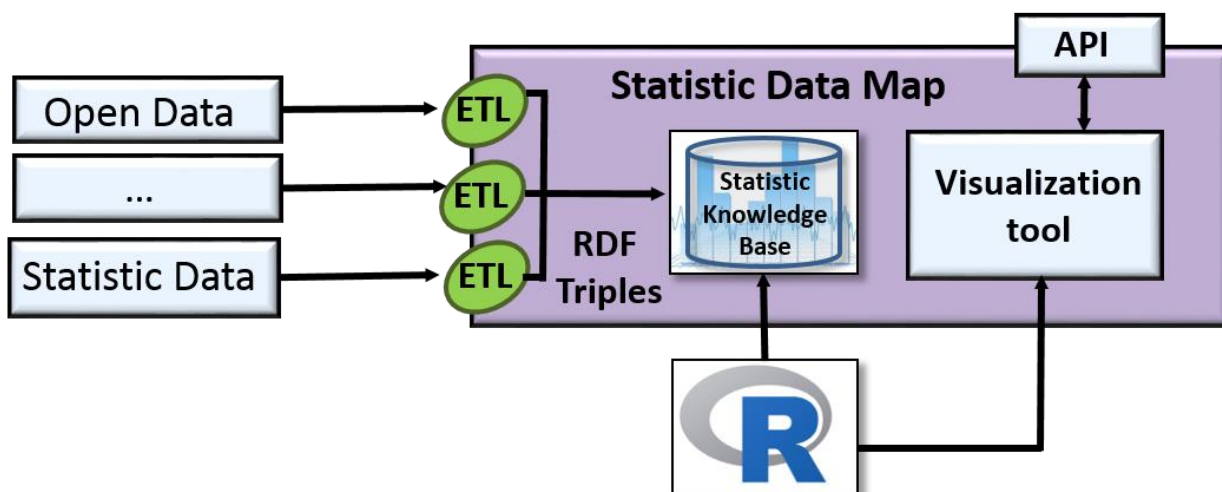


<b>Profilo Tool</b>	
Service Map, ServiceMap	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool	<p>Il Service Map è una interfaccia web realizzata per visualizzare su una mappa geografica i servizi, eventi, flussi di traffico, etc. localizzati nei comuni in cui verranno raccolti i datasets. Inoltre permetterà il salvataggio delle query effettuate sullo store RDF, che potranno essere utilizzate dagli sviluppatori per Applicazioni realizzate a partire dalla Ontologia Sii-Mobility.</p> <ul style="list-style-type: none"> <li>• Attori a cui è destinato il tool: analisti e sviluppatori di Sii-Mobility.</li> <li>• Dati visualizzati: relativi prevalentemente alla Regione Toscana a partire da Firenze, Arezzo, Pisa, Prato, Pistoia (si veda tab. xxxx).</li> </ul>
Dati primari in ingresso	<p>I dati in ingresso potranno provenire da:</p> <ul style="list-style-type: none"> <li>• Interrogazioni dirette sull'RDF store di Sii-Mobility (mediante query SPARQL)</li> <li>• Api REST di Sii-Mobility (che restituiranno dati in formato JSON)</li> <li>• Linked Open Graph, LOG (servizio che restituirà dati in formato JSON)</li> </ul>
Dati prodotti in Uscita	Visualizzazione grafica su una mappa geografica (Applicazione Web): di quanto presente nella ontologia di Sii-Mobility: servizi, eventi, meteo, flussi di traffico, etc.
Principi (metodo, basi teoriche, etc.)	Il Service Map sarà basato sulla Ontologia Sii-Mobility (derivata dal modello Km4City, <a href="http://www.disit.org/km4city">http://www.disit.org/km4city</a> ) e servirà per visualizzare un insieme di servizi presenti sul territorio della Regione Toscana, visualizzando poi i risultati su una mappa geografica costruita sfruttando la libreria javascript LeafLet.
Casi di test (presenti/assenti)	Modalità per testare/eseguire le query SPARQL che saranno poi alla base del Service Map: utilizzo dell'interfaccia web dell'RDF store Virtuoso di Sii-Mobility
Posizione casi di test	
Principali problemi non risolti	Nessuno
Principali requisiti pendenti	Nessuno
Aspetti Tecnologici	
Stato (proposto/approvato)	Approvato
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0%
Eseguibile/libreria/web app	Web App
Single thread/Multithread	Multithread
Linguaggio di sviluppo (java, Php,	java

ETL)		
Piattaforme supportate	tutti i tipi di Browser	
Posizione del codice sorgente	UNIFI, Laboratorio DISIT	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	192.168.0.206:8890/sparql http://log.disit.org/service/	
Indirizzo/i accesso via web (se presenti)	<a href="http://servicemap.disit.org">http://servicemap.disit.org</a> Accesso Open, nessun tipo di credenziale	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
SPARQL RDF store interface, di Sii-Mobility (Virtuoso)	API REST	JSON
Linked Open Graph, LOG (http://log.disit.org/service)	API REST	JSON
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON	SPARQL RDF store interface	
JSON	Linked Open Graph, LOG	
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
Protocollo HTTP, paradigma REST	SPARQL RDF store interface	
Protocollo HTTP, paradigma REST	Linked Open Graph, LOG	
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
RDF store Virtuoso	Database necessario per la memorizzazione della Ontologia e dei dati mappati da Sii-Mobility	
MySQL	Database necessario per la memorizzazione di: impostazioni di base del tool Service Map, query effettuate tramite il Servi, etc.	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
Web App	Java	
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
Leaflet: serve per la visualizzazione di mappe geografiche (http://leafletjs.com)	Leaflet: - Leaflet Gps Control 1.1.0 - Leaflet AwesomeMarkers - Leaflet PolylineDecorator	Licenza: MIT license
Mapbox: serve per la visualizzazione degli standard Layer di Leaflet (satellite, grayscale,	- Mapbox v2.1.4	Licenza: <a href="https://github.com/mapbox/mapbox.js/blob/mb-">https://github.com/mapbox/mapbox.js/blob/mb-</a>

streets, <a href="https://www.mapbox.com">https://www.mapbox.com</a> )		<a href="pages/LICENSE.md">pages/LICENSE.md</a>
Jquery Ajax	jQuery v1.10.1	MIT Licence <a href="https://jquery.org/license">https://jquery.org/license</a>
Highcharts: serve per la visualizzazione di grafici <a href="http://code.highcharts.com">http://code.highcharts.com</a>	Highcharts	CC BY-NC 3.0 <a href="http://creativecommons.org/licenses/by-nc/3.0/">http://creativecommons.org/licenses/by-nc/3.0/</a>

#### 5.14.4 Statistic Data Map, StatisticMap (T32) (UNIFI)



Profilo Tool/Algoritmo	
Statistic Data Map, StatisticMap	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	<p>Strumento per la navigazione (geografica e statistica temporale) sulla base delle informazioni contenute sulla Knowledge Base statistica. Strumento di produzione delle viste statistiche sui dati tramite web. Per esempio distribuzioni di inquinamento, di traffico, di meteo, incidenti, etc. Il tool valuta relazioni tra variabili all'interno di un data set, partendo da una prima analisi descrittiva per poi andare ad elaborare, tramite modelli statistici, eventuali relazioni tra le variabili di interesse.</p> <p>Statistiche descrittive di riferimento:</p> <ul style="list-style-type: none"> <li>• informazioni sulla distribuzione delle variabili presenti (e.g. segnalazione di minimo/massimo, primo, secondo (la mediana) e terzo quartile della distribuzione, etc.;</li> <li>• informazioni sul grado di associazione lineare tra le variabili (matrice di correlazione)</li> </ul>

	Modelli statistici per la valutazione di eventuali relazioni di una o più variabili esplicative e i valori attesi della variabile risposta: <ul style="list-style-type: none"> <li>• Regressione lineare semplice e multipla;</li> <li>• Regressione logistica;</li> </ul> Modelli per l'analisi di serie temporali.	
Dati primari in ingresso	Open Data, Private Data, Statiche, etc.	
Dati prodotti in Uscita	Visualizzazione grafica delle statistiche	
Principi (metodo, basi teoriche, etc.)	Modelli statistici	
Casi di test (presenti/assenti)	Nessuno	
Posizione casi di test	Nessuno	
Principali problemi non risolti	Nessuno	
Principali requisiti pendenti	Nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Serie di ETL / Web App	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	JAVA	
Piattaforme supportate	Unix, Windows / Tutti tipi di Browser	
Posizione del codice sorgente	Laboratorio DISIT, UNIFI	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
R - <a href="https://cran.r-project.org">https://cran.r-project.org</a>		
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>

<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

### 5.14.5 Rendering dati con varie view non solo geo grafiche (T33) (GEOIN)



<b>Profilo Tool/Algoritmo</b>	
rendering dati con varie view non solo geo grafiche	
Nome responsabile	
Partner responsabile	GEOIN
Descrizione Tool/algoritmo	A seconda della tipologia del dato e dell'utente cui si indirizza si dovranno avere rese diverse, ad esempio lo stesso dato potrà apparire in forma tabellare per l'utente avanzato e sotto forma di grafico o tema su mappa per l'utente comune.
Dati primari in ingresso	I dati di Sii-Mobility di cui è richiesta la visualizzazione (ad esempio nel caso di un percorso si potrà avere una lista per una descrizione testuale o la mappa con sovrapposta la linea che lo graficizza)
Dati prodotti in Uscita	Pagina HTML con la resa del dato, o dati formattati JSON per una visualizzazione gestita lato client con procedure Java Script
Principi (metodo, basi teoriche, etc.)	.
Casi di test (presenti/assenti)	
Posizione casi di test	
Principali problemi non risolti	
Principali requisiti pendenti	Definire per ogni tipologia di dato le rese da implementare
<b>Aspetti Tecnologici</b>	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	non implementato
Stato implementazione, percentuale	0%
Eseguitabile/libreria/web app	libreria
Single thread/Multithread	Single thread
Linguaggio di sviluppo (java, Php, ETL)	Java / C++ javaScript
Piattaforme supportate	Linux / Windows
Posizione del codice sorgente	GIT
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se	

necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON		
HTML		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTP		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
WEB		
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

#### 5.14.6 Web App Sii-Mobility per internet browser (T34) (GEOIN)

**Mobile App, Totem, Web APP**

**Participation Platform**

### Profilo Tool/Algoritmo

Web App Sii-Mobility per internet browser

Nome responsabile	
Partner responsabile	GEOIN
Descrizione Tool/algoritmo	Web App per internet browser che permetta l'accesso ai dati di Sii-Mobility e che sfrutta i servizi di Sii-Mobility. In questo caso il fuoco è sulla presentazione dei dati, di visioni statistiche di insieme, di informazioni tipo Dashboard, etc.



Dati primari in ingresso	I dati di Sii-Mobility. Alcuni di questi potranno essere consultati liberamente senza bisogno di autenticazione. Per i dati più sensibili si prevede un accesso con username e password che consentirà la consultazione dei dati in base ai privilegi associati al profilo.	
Dati prodotti in Uscita	Pagine HTML	
Principi (metodo, basi teoriche, etc.)	Metodi per l'analisi statistica	
Casi di test (presenti/assenti)		
Posizione casi di test		
Principali problemi non risolti		
Principali requisiti pendenti	Struttura delle pagine da consultare	
Aspetti Tecnologici		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Web App	
Single thread/Multithread	multithread	
Linguaggio di sviluppo (java, Php, ETL)	java, java script	
Piattaforme supportate	utilizzo dei principali browser web	
Posizione del codice sorgente	git	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)	La homepage di Sii-Mobility ( <a href="http://www.sii-mobility.org">www.sii-mobility.org</a> )	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
HTML		
JSON		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTP		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>

Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

### 5.14.7 Generazione view interattive per totem, web, e mobile (T35) (SOFTECH)

**Participation Platform**

**Mobile App, Totem, Web APP**

<b>Profilo Tool/Algoritmo</b>	
generazione view interattive per totem, web, e mobile	
Nome responsabile	
Partner responsabile	SOFTEC
Descrizione Tool/algoritmo	Sviluppo di moduli specifici per la visualizzazione dati e viste su Totem o mobile.
Dati primari in ingresso	API REST di Sii-Mobility (response JSON)
Dati prodotti in Uscita	Visualizzazione grafica di mappe e tabelle statistiche
Principi (metodo, basi teoriche, etc.)	Interrogazione dei servizi esposti per proporre all'utente web app la consultazione dei dati ritornati dal server e l'eventuale interattività per la richiesta di altre informazione (con conseguenti nuovi chiamate ai servizi)
Casi di test (presenti/assenti)	assenti
Posizione casi di test	n/a
Principali problemi non risolti	assenti
Principali requisiti pendenti	assenti
Aspetti Tecnologici	
Stato (proposto/approvato)	proposto
Implementato/non implementato	non implementato
Stato implementazione, percentuale	0%
Eseguibile/libreria/web app	web app
Single thread/Multithread	single thread
Linguaggio di sviluppo (java, Php, ETL)	Php
Piattaforme supportate	web browser e web view su mobile
Posizione del codice sorgente	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	
Indirizzo/i accesso via web (se presenti)	

Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
WS Sii-Mobility	API REST	JSON
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
Json	API REST	
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
Http/Rest	API Sii-Mobility	
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
Sii-Mobility	Dati raggruppati in risposta aolle query necessarie all'utente della web app	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
Web app	PHP, Javascript, Android	Jquery UI
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
Jquery		

### 5.14.8 Web App Sii-Mobility per Totem (T36) (SOFTECH)

#### Mobile App, Totem, Web APP

Profilo Tool/Algoritmo	
Web App Sii-Mobility per Totem	
Nome responsabile	
Partner responsabile	SOFTEC
Descrizione Tool/algoritmo	Totem che permetta l'accesso ai dati si Sii-Mobility e che sfrutti i servizi di Sii-Mobility. In questo caso il fuoco è sulla presentazione dei dati, di visioni statistiche di insieme, di informazioni tipo Dashboard, etc. E' di interesse anche raccogliere commenti e tracciare il comportamento dell'utente.
Dati primari in ingresso	API REST di Sii-Mobility (response JSON)
Dati prodotti in Uscita	Visualizzazione tabelle per dati statistici
Principi (metodo, basi teoriche, etc.)	Renderizzazione dati aggregati in forma tabellare,

	filtrabili e ordinabili	
Casi di test (presenti/assenti)	assenti	
Posizione casi di test	n/a	
Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	web app	
Single thread/Multithread	single thread	
Linguaggio di sviluppo (java, Php, ETL)	PHP	
Piattaforme supportate	web browser	
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
WS Sii-Mobility	API rest	Json
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
Json	API Rest	
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
Http/rest	WS Sii-Mobility	
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
Sii-Mobility	Dati aggregati per postazione totem di riferimento	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
Web	PHP, Javascript	JQuery UI
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
Jquery		

### 5.14.9 monitoraggio azioni utente, integrazione profiling, lato totem (T37) (UNIFI.DISIT)

#### Mobile App, Totem, Web APP

Profilo Tool/Algoritmo		
monitoraggio azioni utente, integrazione profiling, lato totem		
Nome responsabile		
Partner responsabile	UNIFI.DISIT	
Descrizione Tool/algoritmo	monitoraggio azioni utente, integrazione profiling, sugli strumenti di pubblicazione.... Vedasi profiling in Section 7.6.	
Dati primari in ingresso	posizione GPS dell'utente, timestamp, e tipo di attività svolta tramite mobile app	
Dati prodotti in Uscita	log dei dati di utilizzo su database	
Principi (metodo, basi teoriche, etc.)		
Casi di test (presenti/assenti)	assenti	
Posizione casi di test		
Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	implementato	
Stato implementazione, percentuale	100%	
Eseguibile/libreria/web app	web app	
Single thread/Multithread	single thread	
Linguaggio di sviluppo (java, Php, ETL)	Java	
Piattaforme supportate	Windows, Linux	
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
Nomi tool/moduli usati <small>(aggiungere una riga per ogni tool/modulo usato)</small>	Interfacce API usate	Modello di comunicazione e formato
Web page con statistiche di utilizzo suddivise per profilo utente		JSON
Generatore di dati di utilizzo	MySQL connector	JSON
Web app inserimento dati	REST HTTP	JSON
Formati usati <small>(aggiungere una riga per ogni formato usato)</small>	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
JSON		
Protocolli usati <small>(aggiungere una riga per ogni protocollo usato)</small>	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione

		definizione
REST HTTP		Utilizzato per inviare i dati al database, tramite la web app
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
wifi	Il database contiene i singoli eventi di connessione/disconnessione registrati per ogni utente	
recommender	Il database contiene i singoli eventi di utilizzo registrati per ogni utente	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
web	Java	Highcharts JS
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

#### 5.14.10 Authentication Support Tool (T38) (TIME)

### Authentication Support Tool

Il modulo gestisce la registrazione, profilazione ed autenticazione degli utenti per l'accesso ai servizi SII Mobility, con modalità SSO ed accesso sia in modalità centralizzata che in modalità federativa, tramite Google, Facebook e tramite reti civiche locali.

Una volta effettuata la registrazione e ricevute le credenziali di accesso (username –numero telefonico e password), l'utente accede ai tutti i servizi abilitati per il proprio profilo.

Profilo Tool/Algoritmo	
Authentication Support Tool	
Nome responsabile	
Partner responsabile	TIME
Descrizione Tool/algoritmo	<p>Supporto per l'autenticazione centralizzata degli utenti dei servizi e dei tool di Sii-Mobility. Supporto per l'accesso condizionato a tutti gli strumenti Sii-Mobility: DIM, RIM, DBUM, etc. tramite user name e password e social media, da dispositivi fissi (PC) o mobili (Tablet, Smartphone). Compatibile con SPID.</p> <p>La piattaforma consente la registrazione e profilazione dell'utente, secondo una suddivisione per categorie (cittadino, operatore, amministratore, etc.) e relativa</p>

	<p>definizione dei diritti di accesso ai diversi servizi e categorie di servizi.</p> <p>La piattaforma stessa abilita il SSO per l'accesso unico a tutti i servizi accreditati per il profilo utente e consente l'accesso anche in modo federativo con accesso tramite Google o Facebook, nonché tramite reti civiche esistenti.</p> <p>Dopo la prima registrazione l'accesso è consentito tramite username (nr. Telefonico) e password e limitato ai servizi abilitati per il profilo utente corrispondente (cittadino, operatore, amministratore, etc.). I cookies scaricati sul dispositivo dell'utente consentono l'accesso senza necessità di ridigitare username e password.</p>	
Dati primari in ingresso	Dati registrazione utenti	
Dati prodotti in Uscita		
Principi (metodo, basi teoriche, etc.)	SSO, SOA	
Casi di test (presenti/assenti)	Da definire	
Posizione casi di test	Da definire	
Principali problemi non risolti	Nessuno	
Principali requisiti pendenti	Nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non Implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Web app	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java	
Piattaforme supportate	Windows, Linux	
Posizione del codice sorgente	n.d.	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.d.	
Indirizzo/i accesso via web (se presenti)	n.d.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Liferay Portal	Java API	
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
Html		
Portlet		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
http/https		

SOAP		
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
MySQL	RDBMS	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
Web	java	
Mobile APP	java	
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
Liferay Portal	7.0	GNU
		LGPL

### 5.15 Specifica: Integrazione con sistemi esterni monitoraggio (MIZAR)

		ECM	Mizar	LINIEI	IN20	TIME	Negentis	Effknow	liberalogic	ataf	husitalia	cttnord	tiemme	areos	elfi	calamai	Midra	Proiect	GFOIN	OLJESTIT	SOFTTEC	EMWINGS
4.5.1	Integrazione con sistemi esterni monitoraggio, sessioni specifiche di training, accettazione: integ con altre centrali, modelli	X	X																			
4.5.2	Integrazione con sistemi esterni monitoraggio, sessioni specifiche di training, accettazione: modelli dati: base, pianificazione, sensori, gps-vel	X	X																			

Principali Sottosistemi interessati dai requisiti da descrivere in questa Subsubsection, saranno i sottosistemi primari della specifica

	Sotto Attività	Partner coord	Sottosistema	Tool/Level	Descrizione sommaria
4.5.1	Integrazione con sistemi esterni monitoraggio, sessioni specifiche di training, accettazione: integ con altre centrali, modelli	MIZAR	Support of Integrated Mobility	Support of Integrated Mobility	L'integrazione dei vari sottosistemi, dovrà avvenire mediante un'omogeneizzazione degli standard adoperati. Considerato che i sistemi da integrare sono già esistenti e operativi sul campo, si procederà, al fine di non alterare il funzionamento dei sottosistemi, con lo sviluppo di una serie di adattatori e traduttori in grado di interconnettere gli stessi con la piattaforma.



4.5.2	Integrazione con sistemi esterni monitoraggio, sessioni specifiche di training, accettazione: modelli dati: base, pianificazione, sensori, gps-vel	MIZAR	Support of Integrated Mobility	Support of Integrated Mobility	Viene sviluppato un sistema che permetta il monitoraggio integrato dell'area. connessione effettiva con centrali e gestori in modo da poter stabile un flusso dati costante per le fasi di sperimentazione, e poter fornire dati ed elaborazioni in tempo reale
-------	--	-------	--------------------------------	--------------------------------	---

### 5.16 Specifica: Sistema di partecipazione e sensibilizzazione del cittadino (GEOIN)

Sottoattività e gruppi di lavoro per questi requisiti		FCM	Mizar	UNIEI	IN20	TIME	Negeintis	Effknow	liberologico	ataf	busitalia	cttnord	tiemme	argos	elfi	calamai	Midra	Proiect	GEOIN	OLJESTIT	SOFTEC	FWINGS
4.6.1	Sistema di Partecipazione e Sensibilizzazione del cittadino: Portale web di partecipazione e sensibilizzazione		X		X														X		X	
4.6.2	Sistema di Partecipazione e Sensibilizzazione del cittadino: Piattaforma di application management																		X		X	
4.6.3	Sistema di Partecipazione e Sensibilizzazione del cittadino: Dashboard per la consultazione delle metriche		X																X		X	
4.6.4	Sistema di Partecipazione e Sensibilizzazione del cittadino: Totem da posizionarsi					X																X

**Principali Sottosistemi interessati dai requisiti da descrivere in questa Subsubsection, saranno i sottosistemi primari della specifica**

	Sotto Attività	Partner coord	Sottosistema	Tool/Level	Descrizione sommaria
4.6.1	Sistema di Partecipazione e Sensibilizzazione del cittadino: Portale web di partecipazione e sensibilizzazione	SOFT ECH	Participation Platform	Participation and Sensibilization Platform (T40)	Pagine web per stimolare l'attenzione al City User in base alle sue necessità. Le stesse pagine saranno utilizzate per raccogliere commenti e suggerimenti da parte dei city user.
4.6.2	Sistema di Partecipazione e Sensibilizzazione del cittadino: Piattaforma di application management	GEOIN	Participation Platform	Participation Platform (T39)	La registrazione permette agli utenti di accumulare punteggi (migliorare la propria reputation) anche tramite i sistemi di partecipazione (con contributi, come foto, commenti, voti, correzione di dati, dati di viabilità, etc. etc. via mobile).

		UNIFI DISIT and CSL	Participa tion Platform	User Engager, UCSMANE (T26bis)	Strumento per stimolare la partecipazione dei cittadini ad utilizzare Sii-Mobility rispetto ad altre soluzioni. Lo strumento permette di definire regole che possono essere attivate con varie condizioni di firing, e produrre vari tipi di azioni. Il comportamento collaborativo può aumentare la reputation degli utenti fra quelli che si sono registrati.
4.6.3	Sistema di Partecipazione e Sensibilizzazione del cittadino: Dashboard per la consultazione delle metriche	SOFT ECH	Participa tion Platform	Dashboard of Participation and Sensibilization Platform (D03)	Metriche specifiche sull'andamento del sistema di partecipazione. Le stesse metriche devo poter essere riportate in dashboard.
4.6.4	Sistema di Partecipazione e Sensibilizzazione del cittadino: Totem da posizionarsi	TIME	Totem	Totem (EHW04)	Sviluppo hardware di soluzioni specifiche per l'uso come Totem in are pubbliche con interfaccia touch.

### 5.16.1 Participation and Sensibilization Platform, Parti di base e presentazione informazioni (T40) (SOFTECH)

#### Participation Platform

Profilo Tool/Algoritmo	
Participation and Sensibilization Platform, Parti di base e presentazione informazioni	
Nome responsabile	
Partner responsabile	SOFTEC
Descrizione Tool/algoritmo	Pagine web per stimolare l'attenzione al City User in base alle sue necessità. Le stesse pagine saranno utilizzate per raccogliere commenti e suggerimenti da parte dei city user.
Dati primari in ingresso	Dati statici di informazione circa le maggiori tematiche di viabilità/sostenibilità. Compilazione web form pubbliche, con validazione dati e accettazione captcha, immessi direttamente dagli utenti. Elenco commenti per tematica/pagina.
Dati prodotti in Uscita	Pagine statiche informative. Elenco commenti utenti.
Principi (metodo, basi teoriche, etc.)	L'utente riceve informazioni sulle tematiche rilevanti nella sua esperienza d'uso quotidiano, legge i commenti di altri utenti sull'argomento, interviene commentando a sua volta.
Casi di test (presenti/assenti)	nessuno
Posizione casi di test	nessuno
Principali problemi non risolti	Presenza/assenza di un profilo utente.
Principali requisiti pendenti	Eventuale autenticazione utente.

<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	web app	
Single thread/Multithread	single thread	
Linguaggio di sviluppo (java, Php, ETL)	PHP	
Piattaforme supportate	web browser	
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Sii Mobility	API REST	JSON
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON	API REST	
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
http/Rest	API Sii-Mobility	
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
Sii-Mobility	Base dati per informazioni semi statiche sui contenuti da mostrare	
Sii-Mobility	Commenti utenti	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
Web App	PHP, Javascript	JQuery UI
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
JQuery		

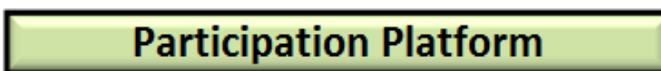
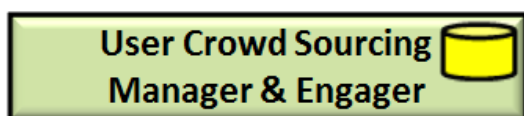
### 5.16.2 Participation and Sensibilization Platform, raccolta suggerimenti e contributi (T39) (GEOIN)

**Participation Platform**

<b>Profilo Tool/Algoritmo</b>		
Participation and Sensibilization Platform, raccolta suggerimenti e contributi		
Nome responsabile		
Partner responsabile	GEOIN	
Descrizione Tool/algoritmo	La registrazione permette agli utenti di accumulare punteggi (migliorare la propria reputation) anche tramite i sistemi di partecipazione (con contributi, come foto, commenti, voti, correzione di dati, dati di viabilità, etc. etc. via mobile).	
Dati primari in ingresso	I dati acquisiti dagli utenti che fanno uso delle app di Sii-Mobility e che ne tracciano le abitudini ed il comportamento.	
Dati prodotti in Uscita	Report	
Principi (metodo, basi teoriche, etc.)	valutazione del comportamento in base alla tipologia degli spostamenti effettuati	
Casi di test (presenti/assenti)		
Posizione casi di test		
Principali problemi non risolti		
Principali requisiti pendenti	Mappa del sito	
Aspetti Tecnologici		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Web app	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java, JavaScript	
Piattaforme supportate	Web browser	
Posizione del codice sorgente	GIT	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)	Accesso tramite la home page di Sii-Mobility	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
HTML		
JSON		
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
HTTP		

Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

### 5.16.3 Participation and Sensibilization Platform, Engager, strumento di coinvolgimento (T26bis)(UNIFI DISIT, UNIFI CSL)



Profilo Tool/Algoritmo	
Participation and Sensibilization Platform, Engager, strumento di coinvolgimento	
Nome responsabile	
Partner responsabile	UNIFI
Descrizione Tool/algoritmo	Strumento per stimolare la partecipazione dei cittadini ad utilizzare Sii-Mobility rispetto ad altre soluzioni. Lo strumento permette di definire regole che possono essere attivate con varie condizioni di firing, e produrre vari tipi di azioni. Il comportamento collaborativo può aumentare la reputation degli utenti fra quelli che si sono registrati.
Dati primari in ingresso	Contesto dell'utente
Dati prodotti in Uscita	Engagement e Assistance dell'utente in formato JSON
Principi (metodo, basi teoriche, etc.)	Event Condition Action
Casi di test (presenti/assenti)	Assenti
Posizione casi di test	
Principali problemi non risolti	Nessuno
Principali requisiti pendenti	Nessuno
Aspetti Tecnologici	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	15%
Eseguibile/libreria/web app	Libreria
Single thread/Multithread	Multithread
Linguaggio di sviluppo (java, Php, ETL)	Java
Piattaforme supportate	Piattaforme supportate da Java

Posizione del codice sorgente	svn	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	<a href="#">nessuno</a>	
Indirizzo/i accesso via web (se presenti)	nessuno	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
ServiceMap	Services	HTTP REST
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTP REST		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
Engager	Dati dell'engager	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
spring	org.springframework 1.3.3.RELEASE	Apache License 2.0

#### 5.16.4 Dashboard of Participation and Sensibilization Platform (D03) (SOFTECH)

**Participation Platform**

<b>Profilo Tool/Algoritmo</b>	
Dashboard of Participation and Sensibilization Platform	
Nome responsabile	
Partner responsabile	SOFTEC
Descrizione Tool/algoritmo	Metriche specifiche sull'andamento del sistema di

	partecipazione. Le stesse metriche devo poter essere riportate in dashboard.	
Dati primari in ingresso	Dati aggregati di origine Sii-Mobility	
Dati prodotti in Uscita	Tabelle riepilogative statistiche	
Principi (metodo, basi teoriche, etc.)	In relazione alle metriche principali individuate, verranno mostrate tabelle riassuntive dei dati ricevuti dal server	
Casi di test (presenti/assenti)	nessuno	
Posizione casi di test	nessuno	
Principali problemi non risolti	nessuno	
Principali requisiti pendenti	nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	proposto	
Implementato/non implementato	non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	web app	
Single thread/Multithread	single thread	
Linguaggio di sviluppo (java, Php, ETL)	PHP	
Piattaforme supportate	web browser	
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Sii-Mobility	API REST	JSON
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON	Sii-Mobility	
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
http/rest	API REST	
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
Sii-Mobility	Dati riepilogativi aggregati per metrica	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
Web app	PHP, Javascript	Jquery UI
<b>Libreria usata</b>	<b>Nome e versione</b>	<b>Licenza: GPL, LGPL, PEK,</b>

(aggiungere una riga per ogni libreria usata)	usata	proprietaria, commerciale, etc.
JQuery		

### 5.16.5 Participation Platform, Totem (EHW04) (TIME)

#### Participation Platform

#### Mobile App, Totem, Web APP

Questo modulo concerne la acquisizione, configurazione ed installazione di totem con touch screen per la consultazione dei servizi pubblici forniti da SII Mobility ed eventualmente per l'accesso come utente accreditato. I servizi sono accessibili tramite interfaccia web.

I totem sono dispositivi da esterno, con caratteristiche di ergonomia e autonomia di alimentazione. I totem saranno dotati di SW per lo screen reader, allo scopo di essere utilizzati anche da parte di non vedenti, e di una tastiera ribassata, per l'usabilità da parte di utenti a mobilità ridotta.

Profilo Tool/Algoritmo		
Participation Platform, Totem		
Nome responsabile		
Partner responsabile	TIME	
Descrizione Tool/algoritmo	Acquisizione, configurazione ed installazione di totem per esterno con touch screen ed interfaccia web.	
Dati primari in ingresso		
Dati prodotti in Uscita		
Principi (metodo, basi teoriche, etc.)		
Casi di test (presenti/assenti)	Non previsti	
Posizione casi di test	Non previsti	
Principali problemi non risolti	Nessuno	
Principali requisiti pendenti	Nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non Implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Web app	
Single thread/Multithread	Single thread	
Linguaggio di sviluppo (java, Php, ETL)	<b>Java</b>	
Piattaforme supportate	Windows	
Posizione del codice sorgente	n.d.	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.d.	
Indirizzo/i accesso via web (se presenti)	n.d.	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
NVDA (Non Visual Desktop Access)		
eSpeak		



<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
http/https		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	Descrizione	
<b>Tipo interfaccia utente</b> (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
web	html	
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
NVDA		GNU ver. 2

**Bacheca Touch 22” – per Esterni (versione per diversamente abili )**

Struttura metallica verniciata a polvere con sportello sotto chiave di sicurezza  
 Predisposizione per ancoraggio a parete / su supporto metallico ancorato a terra  
 Display 22” alta luminosità Full HD risoluzione 1920x1080 Luminosità 1000 cd/m2 Contrasto 3000:1  
 Pannello Touch Screen Capacitivo per esterni MultiTouch  
 Sensore regolazione automatica luminosità  
 Protezione Vetro Temperato Antivandalo 4 mm PC integrato  
 CPU Intel G3260 3,3 GHz, RAM 4 Gb, HDD 500 Gb  
 Impianto Automatico ventilazione interna Temperatura d’esercizio -20°/+50°  
 Protezione differenziale Magnetotermico  
 Alimentazione AC 110-240V  
 Tastiera metallica IP65 con Touch Pad  
 Wi.Fi 802.11 b/g/n con antenna  
 Sistema Operativo Preinstallato Windows 7 Prof 64 bit

**Bacheca Touch 22” – per Esterni (versione per normodotati)**

Struttura metallica verniciata a polvere con sportello sotto chiave di sicurezza  
 Predisposizione per ancoraggio a parete / su supporto metallico ancorato a terra  
 Display 22” alta luminosità Full HD risoluzione 1920x1080 Luminosità 1000 cd/m2 Contrasto 3000:1  
 Pannello Touch Screen Capacitivo per esterni MultiTouch  
 Sensore regolazione automatica luminosità  
 Protezione Vetro Temperato Antivandalo 4 mm PC integrato  
 CPU Intel G3260 3,3 GHz, RAM 4 Gb, HDD 500 Gb  
 Impianto Automatico ventilazione interna Temperatura d’esercizio -20°/+50°

Protezione differenziale Magnetotermico  
 Alimentazione AC 110-240V  
 Wi.Fi 802.11 b/g/n con antenna  
 Sistema Operativo Preinstallato Windows 7 Prof 64 bit

### 5.17 Specifica: sistema di bigliettazione integrato e sua integrazione

Sottoattività e gruppi di lavoro per questi requisiti		FCM	Mizar	UNIFI	IN20	TIME	Nesentis	Fffknow	liberologico	ataf	busitalia	cttnord	fiemme	argos	elfi	calamai	Midra	Proiect	GEOIN	OUJESTIT	SOETEC	FWINGS	
2.1.4	Sviluppo del Sistema Interoperabile integrato, SII: bigliettazione						X																
5.1.1	Integrazione con il sistema di bigliettazione integrato: TPL e sistema toscano						X			X	X	X	X										
5.1.2	Integrazione con il sistema di bigliettazione integrato: parcheggi vari				X		X	X												X			
5.1.3	Integrazione con il sistema di bigliettazione integrato: bike e car sharing						X																
5.1.4	Integrazione con il sistema di bigliettazione integrato: treno, ferrovie						X																
5.1.5	Integrazione con il sistema di bigliettazione integrato: ZTL e bonus				X		X	X												X			
5.1.6	Integrazione con il sistema di bigliettazione integrato: Autostrada																X						
5.1.7	Integrazione con il sistema di bigliettazione integrato: integrazione su profiling				X		X																
5.1.8	Integrazione con il sistema di bigliettazione integrato: tariffazione dinamica				X		X	X												X			

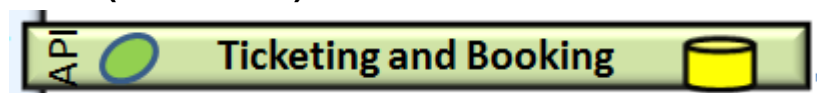
**Principali Sottosistemi interessati dai requisiti da descrivere in questa Subsubsection, saranno i sottosistemi primari della specifica**

	Sotto Attività	Partner coord	Sottosistema	Tool/Level	Descrizione sommaria
2.1.4	Sviluppo del Sistema Interoperabile integrato, SII: bigliettazione	NEGENTIS	Ticketing and Booking	Ticketing interoperabile anche in funzione dei cambiamenti. (T41)	Strumento di integrazione con i vari canali di ticketing e di pagamento delle TPL, parking, altri mezzi di trasporto. Integrazione dei metodi pagamento con transponder, sms, telefono, etc.
		NEGENTIS	Ticketing and Booking	Booking (MBT01)	Accettazione delle prenotazioni dei titoli di viaggio composti, multimodali, ma potrebbe anche fornire prenotazioni sull'accesso a servizi a sportello
5.1.1	Integrazione con il sistema di bigliettazione integrato: TPL e sistema toscano	NEGENTIS	bigliettazione	bigliettazione integrato: TPL e sistema toscano (MBT02)	Strumento di integrazione con i vari canali di ticketing e di pagamento delle TPL

5.1.2	Integrazione con il sistema di bigliettazione integrato: parcheggi vari	NEGENTIS	bigliettazione	bigliettazione integrato: parcheggi vari (MBT03)	Strumento di integrazione con i vari canali di ticketing e di pagamento di parcheggi di vario tipo
5.1.3	Integrazione con il sistema di bigliettazione integrato: bike e car sharing	NEGENTIS	bigliettazione	bigliettazione integrato: bike e car sharing (MBT04)	Strumento di integrazione con i vari canali di ticketing e di pagamento per car sharing e bike sharing
5.1.4	Integrazione con il sistema di bigliettazione integrato: treno, ferrovie	NEGENTIS	bigliettazione	bigliettazione integrato: treno, ferrovie (MBT05)	Strumento di integrazione con i vari canali di ticketing e di pagamento di tratte ferroviarie
5.1.5	Integrazione con il sistema di bigliettazione integrato: ZTL e bonus	NEGENTIS	bigliettazione	bigliettazione integrato: ZTL e bonus (MBT06)	Strumento di accounting dei bonus per usarli per il pagamento di parcheggi, ZTL, e ticket in genere
5.1.6	Integrazione con il sistema di bigliettazione integrato: Autostrada	MIDRA	bigliettazione	bigliettazione integrato: Autostrada (MBT07)	Richiede l'integrazione con il sistema di bigliettazione sviluppato da Autostrade basato sulle diverse soluzioni Telepass. Dovranno essere studiate soluzioni in grado di ricevere e acquisire dati dal sistema centrale Autostrade per l'Italia.
5.1.7	Integrazione con il sistema di bigliettazione integrato: integrazione su profiling	NEGENTIS	bigliettazione	bigliettazione integrato: integrazione su profiling (MBT08)	Condivisione con il sistema di User profiling per l'acquisizione dei bonus e per l'informazione di cosa e quando ogni utente ha comprato come biglietti. Sempre come utente autenticato
5.1.8	Integrazione con il sistema di bigliettazione integrato: tariffazione dinamica	NEGENTIS	bigliettazione	bigliettazione integrato: tariffazione dinamica (MBT09)	Sperimentazione di calcolo per la bigliettazione dinamica

Gli accordi fra Sii-Mobility e gli erogatori del servizio per l'integrazione di pagamento sarà coordinato dallo stesso partner responsabile del task specifico: NEGENTIS, MIDRA, etc., che agirà a nome del consorzio per questi aspetti tenendo sempre informato ed in CC il coordinatore del progetto.

### 5.17.1 Ticketing interoperabile anche in funzione dei cambiamenti (T41) (NEGENTIS)

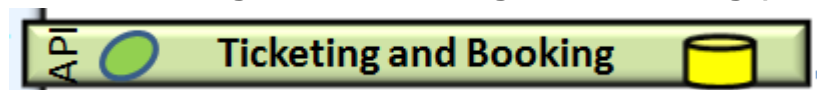


<h2 style="text-align: center;">Profilo Tool/Algoritmo</h2>	
Ticketing interoperabile anche in funzione dei cambiamenti	
Nome responsabile	
Partner responsabile	NEGENTIS
Descrizione Tool/algoritmo	Il sistema di ticketing si integra con i vari canali di ticketing messi a disposizione dai gestori dei servizi presenti sul territorio (e.g., TPL, parking, altri mezzi di

	<p>trasporto). Il biglietto che sarà emesso dalla piattaforma è unico per tutti i servizi previsti nel percorso selezionato. Al fine dell'emissione del biglietto l'utente dovrà pagare la somma di denaro richiesto alla piattaforma Sii-Mobility (transponder, sms, telefono). La piattaforma Sii-Mobility a sua volta si integra con i sistemi di pagamento previsti dai gestori dei singoli servizi.</p> <p>Si prevede che il sistema di bigliettazione abbia un listino dei prezzi per l'accesso ai diversi servizi disponibili che è stato concordato con i diversi gestori sulla base di tale listino è calcolato il prezzo del biglietto integrato. Il pagamento è fatto dall'utente alla piattaforma Sii-Mobility che dovrà essere dotata di un sistema per ricevere le transazioni. Periodicamente, a fine mese/fine settimana, il sistema Sii-Mobility chiuderà le posizioni aperte con i diversi gestori di servizio utilizzando il metodo di pagamento previsto dai singoli gestori.</p>	
Dati primari in ingresso	Elenco dei servizi richiesti dall'utente e date, utente della piattaforma Sii-MOBility	
Dati prodotti in Uscita	Biglietto unico per i servizi richiesti	
Principi (metodo, basi teoriche, etc.)		
Casi di test (presenti/assenti)		
Posizione casi di test		
Principali problemi non risolti		
Principali requisiti pendenti		
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)		
Implementato/non implementato		
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app		
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java, etc	
Piattaforme supportate	Linux	
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>

<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

### 5.17.2 bigliettazione integrato: Booking (MBT01) (NEGENTIS)



<b>Profilo Tool/Algoritmo</b>	
bigliettazione integrato: Booking	
Nome responsabile	
Partner responsabile	NEGENTIS
Descrizione Tool/algoritmo	Il sistema deve poter fornire per un punto di partenza ed un punto di arrivo richiesto dall'utente, la possibile pianificazione del percorso usando i diversi servizi messi a disposizione dai gestori. Se il percorso proposto (verificato a priori come realizzabile dal sistema di planning) è accettato dall'utente allora il sistema deve poter pagare il biglietto "complessivo" sulla base del listino previsto da Sii-Mobility ed emettere il biglietto unico ottenuto componendo i titoli di viaggio
Dati primari in ingresso	Tratta di interesse: punto di partenza punto di arrivo, data del viaggio, servizio richiesto, utente Sii-Mobility.
Dati prodotti in Uscita	Elenco dei servizi (con singoli percorsi) che portano l'utente dal punto di partenza a quello di arrivo
Principi (metodo, basi teoriche, etc.)	
Casi di test (presenti/assenti)	
Posizione casi di test	
Principali problemi non risolti	
Principali requisiti pendenti	
Aspetti Tecnologici	

Stato (proposto/approvato)		
Implementato/non implementato		
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app		
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java, etc	
Piattaforme supportate	Linux	
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

### 5.17.3 Bigliettazione integrato: TPL e sistema toscano (MBT02) (NEGENTIS)



<b>Profilo Tool/Algoritmo</b>		
bigliettazione integrato: TPL e sistema toscano		
Nome responsabile		
Partner responsabile	NEGENTIS	
Descrizione Tool/algoritmo	Il sistema deve poter richiedere ai gestori del traffico pubblico locale (TPL) coinvolti nel progetto l'emissione di un biglietto sulla base della tratta selezionata, e.g., biglietto a tempo o altro. Si noti che l'emissione del biglietto deve essere fatta in accordo al listino che sarà stabilito tra Sii-Mobility ed i gestori di TPL. L'emissione del biglietto non dovrà essere convenzionato alla chiusura di una transazione relativa al pagamento del singolo biglietto. Il pagamento da Sii-Mobility verso il gestore TPL avverrà sulla base di procedure batch di pagamento settimanali e/o mensili.	
Dati primari in ingresso	Tratta di interesse: punto di partenza punto di arrivo, data del viaggio, servizio richiesto, utente Sii-Mobility.	
Dati prodotti in Uscita	Biglietto emesso dalla TPL per la tratta d'interesse	
Principi (metodo, basi teoriche, etc.)		
Casi di test (presenti/assenti)		
Posizione casi di test		
Principali problemi non risolti		
Principali requisiti pendenti		
Aspetti Tecnologici		
Stato (proposto/approvato)		
Implementato/non implementato		
Stato implementazione, percentuale		
Eseguibile/libreria/web app		
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java, etc	
Piattaforme supportate	Linux	
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione</b>

		definizione
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

#### 5.17.4 bigliettazione integrato: parcheggi vari (MBT03) (NEGENTIS)



<b>Profilo Tool/Algoritmo</b>	
bigliettazione integrato: parcheggi vari	
Nome responsabile	
Partner responsabile	NEGENTIS
Descrizione Tool/algoritmo	Il sistema deve poter richiedere ai gestori del parcheggio coinvolti nel progetto l'emissione di un biglietto sulla base del periodo di permanenza. Si noti che l'emissione del biglietto deve essere fatta in accordo al listino che sarà stabilito tra SII-Mobility ed i gestori del parcheggio. L'emissione del biglietto non dovrà essere convenzionato alla chiusura di una transazione relativa al pagamento del singolo biglietto. Il pagamento da SII-Mobility verso il gestore del parcheggio avverrà sulla base di procedure batch di pagamento settimanali e/o mensili.
Dati primari in ingresso	Periodo di interesse: parcheggio, data/ora arrivo, data/ora partenza, servizio richiesto, utente SII-Mobility.
Dati prodotti in Uscita	Biglietto emesso dal parcheggio per il periodo d'interesse
Principi (metodo, basi teoriche, etc.)	
Casi di test (presenti/assenti)	
Posizione casi di test	
Principali problemi non risolti	
Principali requisiti pendenti	
Aspetti Tecnologici	



Stato (proposto/approvato)		
Implementato/non implementato		
Stato implementazione, percentuale		
Eseguibile/libreria/web app		
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java, etc	
Piattaforme supportate	Linux	
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

### 5.17.5 Bigliettazione integrato: bike e car sharing (MBT04) (NEGENTIS)



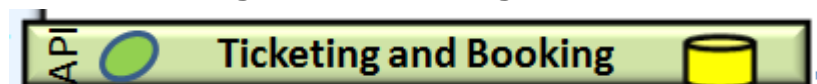
## Profilo Tool/Algoritmo

bigliettazione integrato: bike e car sharing

Nome responsabile		
Partner responsabile	NEGENTIS	
Descrizione Tool/algorithmo	Il sistema deve poter richiedere ai gestori dei servizi di bike/car sharing coinvolti nel progetto l'emissione di un biglietto sulla base del periodo di uso. Si noti che l'emissione del biglietto deve essere fatta in accordo la listino che sarà stabilito tra Sii-Mobility ed i gestori del servizio di bike/car sharing. L'emissione del biglietto non dovrà essere convenzionato alla chiusura di una transazione relativa al pagamento del singolo biglietto. Il pagamento da Sii-Mobility verso il gestore di bike/car sharing avverrà sulla base di procedure batch di pagamento settimanali e/o mensili.	
Dati primari in ingresso	Periodo di interesse: punto prelievo mezzo, data/ora arrivo, data/ora partenza, servizio richiesto, utente Sii-Mobility.	
Dati prodotti in Uscita	Biglietto emesso dal gestore del servizio di car/bike sharing	
Principi (metodo, basi teoriche, etc.)		
Casi di test (presenti/assenti)		
Posizione casi di test		
Principali problemi non risolti		
Principali requisiti pendenti		
Aspetti Tecnologici		
Stato (proposto/approvato)		
Implementato/non implementato		
Stato implementazione, percentuale		
Eseguibile/libreria/web app		
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java, etc	
Piattaforme supportate	Linux	
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>

<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

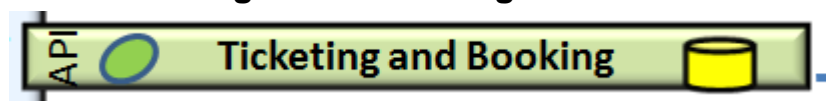
### 5.17.6 Bigliettazione integrato: treno, ferrovie (MBT05) (NEGENTIS)



<b>Profilo Tool/Algoritmo</b>	
bigliettazione integrato: treno, ferrovie	
Nome responsabile	
Partner responsabile	NEGENTIS
Descrizione Tool/algoritmo	Il sistema deve poter richiedere al gestore del servizio ferroviario coinvolto nel progetto l'emissione di un biglietto sulla base della tratta selezionata. Si noti che l'emissione del biglietto deve essere fatta in accordo al listino che sarà stabilito tra Sii-Mobility ed il gestore del servizio ferroviario. L'emissione del biglietto non dovrà essere convenzionata alla chiusura di una transazione relativa al pagamento del singolo biglietto. Il pagamento da Sii-Mobility verso il gestore avverrà sulla base di procedure batch di pagamento settimanali e/o mensili.
Dati primari in ingresso	Tratta di interesse: punto di partenza punto di arrivo, data del viaggio, servizio richiesto, utente Sii-Mobility.
Dati prodotti in Uscita	Biglietto emesso dal gestore per la tratta d'interesse
Principi (metodo, basi teoriche, etc.)	
Casi di test (presenti/assenti)	
Posizione casi di test	
Principali problemi non risolti	
Principali requisiti pendenti	
<b>Aspetti Tecnologici</b>	
Stato (proposto/approvato)	
Implementato/non implementato	
Stato implementazione, percentuale	
Eseguibile/libreria/web app	

Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java, etc	
Piattaforme supportate	Linux	
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

**5.17.7 Bigliettazione integrato: ZTL e bonus (MBT06) (NEGENTIS)**

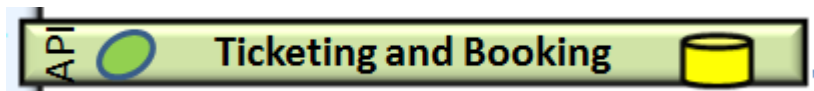


<b>Profilo Tool/Algoritmo</b>	
bigliettazione integrato: ZTL e bonus	
Nome responsabile	
Partner responsabile	NEGENTIS
Descrizione Tool/algoritmo	Il sistema deve poter ricevere dal gestore del traffico del comune d'interesse informazioni relative ai varchi

	<p>per l'accesso alle zone a traffico limitato (ZTL). Se il percorso d'interesse attraversa una o più ZTL allora deve essere segnalato (se consentito) l'attraversamento oppure il divieto. L'accesso alle zone a traffico limitato devono essere disincentivate tramite la definizione di listini al cui interno sono premiati gli spostamenti che hanno minor impatto sull'ambiente ed in ogni caso che non privilegiano l'uso del mezzo di trasporto privato Il sistema di bigliettazione deve poter ricevere dal modulo di profilazione utente informazioni che sulla base del listino possono far accumulare punti/bonus al soggetto. Il sistema di calcolo della tariffa provvederà l'uso dei punti/bonus accumulati dall'utente per ottenere sconti da usare per il pagamento dei biglietti d'interesse. A tal fine dovranno essere concordate tra Sii-Mobility ed i diversi gestori politiche di sconto e/o agevolazione degli utenti virtuosi.</p>	
Dati primari in ingresso	Tratta di interesse: punto di partenza punto di arrivo, data del viaggio.	
Dati prodotti in Uscita	Informazioni sulle ZTL, divieti e/o modalità di accesso.	
Principi (metodo, basi teoriche, etc.)		
Casi di test (presenti/assenti)		
Posizione casi di test		
Principali problemi non risolti		
Principali requisiti pendenti		
Aspetti Tecnologici		
Stato (proposto/approvato)		
Implementato/non implementato		
Stato implementazione, percentuale		
Eseguibile/libreria/web app	Multithread	
Single thread/Multithread	Java, etc	
Linguaggio di sviluppo (java, Php, ETL)	Linux	
Piattaforme supportate		
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>

<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	Descrizione	
<b>Tipo interfaccia utente</b> (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

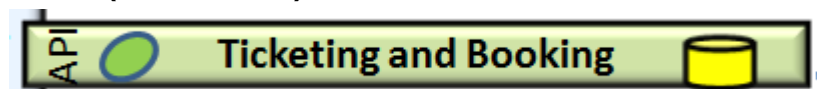
### 5.17.8 Bigliettazione integrata: Autostrada (MBT07) (MIDRA)



<b>Profilo Tool/Algoritmo</b>	
bigliettazione integrata: Autostrada	
Nome responsabile	
Partner responsabile	MIDRA
Descrizione Tool/algoritmo	Sistema automatico di bigliettazione per accesso al servizio Autostrade
Dati primari in ingresso	Informazioni identificative dell'utente per addebito memorizzate all'interno del terminale mobile OBU (On Board Unit)
Dati prodotti in Uscita	Addebito telematico gestito via server Autostrade. Statistiche di utilizzo del servizio e dati per reportistica traffico, utilizzabili per generare stime di QoS per navigazione assistita e monitoraggio (con possibilità di integrazione con informazioni di traffico per altre infrastrutture, es. ferrovie).
Principi (metodo, basi teoriche, etc.)	Non applicabile
Casi di test (presenti/assenti)	Assenti
Posizione casi di test	n/d
Principali problemi non risolti	Assenti
Principali requisiti pendenti	Interfaccia di scambio con il sistema SII Mobility
Aspetti Tecnologici	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0

Eseguibile/libreria/web app	Eseguibile + web app	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	C, HTML, PHP, SQL, ETL	
Piattaforme supportate	Windows Server, Linux	
Posizione del codice sorgente	Git	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
Nomi tool/moduli usati <small>(aggiungere una riga per ogni tool/modulo usato)</small>	Interfacce API usate	Modello di comunicazione e formato
Apache Web Server	APACHE	HTTP/CGI
MySQL connector for PHP	MYSQL	SQL
Formati usati <small>(aggiungere una riga per ogni formato usato)</small>	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
Da definire		
Protocolli usati <small>(aggiungere una riga per ogni protocollo usato)</small>	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
HTTP		Hypertext Transfer Protocol
TCP/IP		
ad-hoc AUTOSTRADE		TELEPASS
Nomi database usati <small>(aggiungere una riga per ogni db usato)</small>	Descrizione	
MySQL	Database relazionale	
Tipo interfaccia utente <small>(web/applicazione)</small>	Modello sviluppo, linguaggio	Libreria usata per UI
Web	HTML, PHP, JavaScript, CSS	
Libreria usata <small>(aggiungere una riga per ogni libreria usata)</small>	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
MySQL		GPL

### 5.17.9 Bigliettazione integrato: integrazione su user profiling (MBT08) (NEGENTIS)



## Profilo Tool/Algoritmo

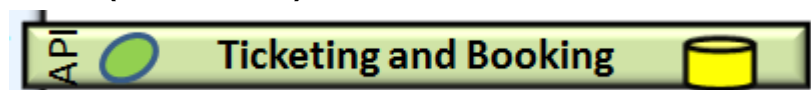
bigliettazione integrato: integrazione su user profiling

Nome responsabile		
Partner responsabile	NEGENTIS	
Descrizione Tool/algoritmo	Il sistema deve poter ricevere deve poter ricevere dal modulo di profilazione di Sii-Mobility informazioni sul comportamento dell'utente (che dovrà essere noto all'interno della piattaforma Sii-Mobility). Le informazioni raccolte dovranno poter essere usate dal modulo di tariffazione per consentire l'accumulo di punti/bonus da usare per il "pagamento" dei titoli di viaggio. A tal fine dovranno essere concordate tra Sii-Mobility ed i diversi gestori politiche di sconto e/o agevolazione degli utenti virtuosi.	
Dati primari in ingresso	Identificativo utente Sii-Mobility, percorso e modalità	
Dati prodotti in Uscita	Informazioni relative al comportamento dell'utente ai fini del calcolo di bonus/punti.	
Principi (metodo, basi teoriche, etc.)		
Casi di test (presenti/assenti)		
Posizione casi di test		
Principali problemi non risolti		
Principali requisiti pendenti		
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)		
Implementato/non implementato		
Stato implementazione, percentuale		
Eseguibile/libreria/web app		
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java, etc	
Piattaforme supportate	Linux	
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>



Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

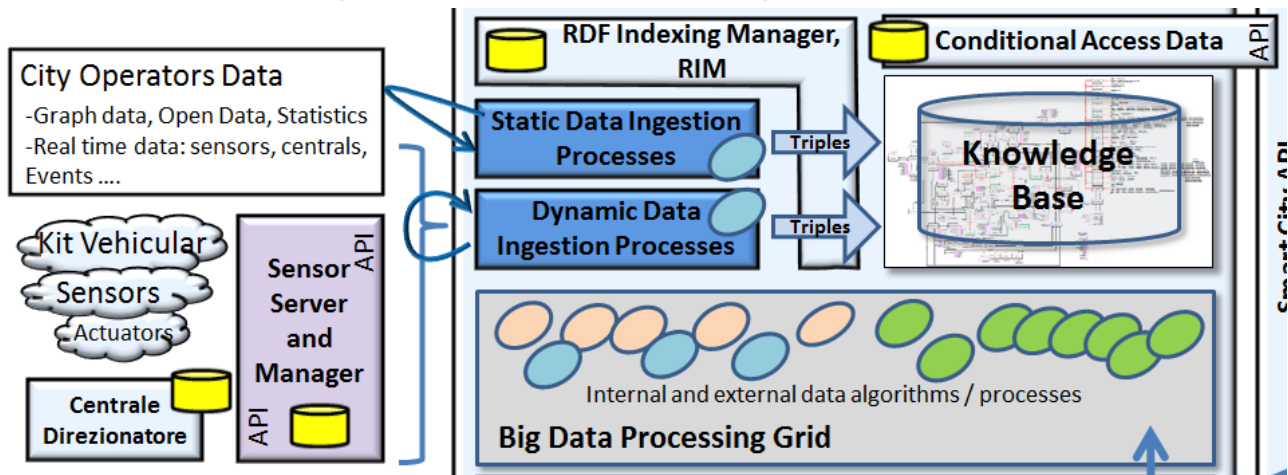
### 5.17.10 Bigliettazione integrato: tariffazione dinamica (MBT09) (NEGENTIS)



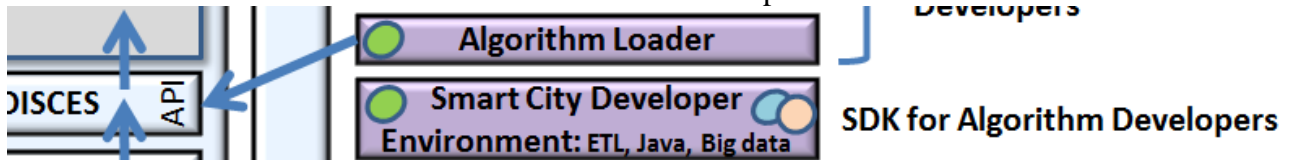
<b>Profilo Tool/Algoritmo</b> bigliettazione integrato: tariffazione dinamica	
Nome responsabile	
Partner responsabile	NEGENTIS
Descrizione Tool/algoritmo	Il sistema si baserà su listini che definiranno modalità di calcolo delle tariffe sulla base di parametri legati alla raccolta di punti/bonus. La tariffazione di un percorso potrà quindi variare sulla base dei punti/bonus accumulati dall'utente. Inoltre, il modulo di tariffazione premierà (sempre tramite la raccolta di punti/bonus) i comportamenti dell'utente che premieranno i suggerimenti delle amministrazioni indirizzate ad un utilizzo eco-sostenibile dei mezzi di trasporto. A tal fine dovranno essere concordate tra Sii-Mobility ed i diversi gestori politiche di sconto e/o agevolazione degli utenti virtuosi.
Dati primari in ingresso	Identificativo utente Sii-Mobility, percorso, data e modalità di trasporto
Dati prodotti in Uscita	Informazioni relative al costo del tragitto sulla base dell'uso suggerito dalle amministrazioni.
Principi (metodo, basi teoriche, etc.)	
Casi di test (presenti/assenti)	
Posizione casi di test	
Principali problemi non risolti	
Principali requisiti pendenti	
Aspetti Tecnologici	
Stato (proposto/approvato)	
Implementato/non implementato	

Stato implementazione, percentuale		
Eseguibile/libreria/web app		
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	Java, etc	
Piattaforme supportate	Linux	
Posizione del codice sorgente		
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

### 5.18 Specifica: Integrazione con sistemi di gestione traffico (MIZAR)



And for the ETL development



Sottoattività e gruppi di lavoro per questi requisiti		FCM	Mizar	UNIEI	IN20	TIME	Negentis	Effknow	liberologico	ataf	busitalia	ctnord	tiemme	argos	elfi	calamai	Midra	Proiect	GEOIN	OLJESTIT	SOFTEC	FWINGS	
5.2.1	Integrazione con sistemi di gestione traffico: Integrazione con gestore traffico Firenze		X							X	X												
5.2.2	Integrazione con sistemi di gestione traffico: Integrazione con gestore traffico Prato, Pistoia		X									X											
5.2.3	Integrazione con sistemi di gestione traffico: Integrazione con gestore traffico Siena arezzo		X										X										
5.2.4	Integrazione con sistemi di gestione traffico: Integrazione con gestore traffico Autostrada																X						
5.2.5	Integrazione con sistemi di gestione traffico: Sviluppo API info Sii-mobility verso gestori traffico				X					X	X	X	X										

Principali Sottosistemi da descrivere in questa Subsubsection, sono i sottosistemi primari della specifica					
	Sotto Attività	Partner coord	Sottosistema	Tool/Level	Descrizione sommaria
5.2.1	Integrazione con sistemi di gestione traffico: Integrazione con gestore traffico Firenze	MIZAR	ETL	processi di Acquisizione dati, Integrazione con gestore traffico Firenze (PETL11)	Acquisizione dati relativi alle informazioni di alto livello del gestore del traffico: dati dei flussi, dati delle temporizzazioni o profili attuati, ordinanze. I dati relativi alle informazioni su preferenziali, direzioni, etc. provengono dalla regione o dai comuni

5.2.2	Integrazione con sistemi di gestione traffico: Integrazione con gestore traffico Prato, Pistoia	MIZAR	ETL	processi di Acquisizione dati, Integrazione con gestore traffico Prato, Pistoia (PETL12)	Acquisizione dati relativi alle informazioni di alto livello del gestore del traffico: dati dei flussi, dati delle temporizzazioni o profili attuati, ordinanze. I dati relativi alle informazioni su preferenziali, direzioni, etc. provengono dalla regione o dai comuni
5.2.3	Integrazione con sistemi di gestione traffico: Integrazione con gestore traffico Siena Arezzo	MIZAR	ETL	processi di Acquisizione dati, Integrazione con gestore traffico Siena Arezzo (PETL13)	Acquisizione dati relativi alle informazioni di alto livello del gestore del traffico: dati dei flussi, dati delle temporizzazioni o profili attuati, ordinanze. I dati relativi alle informazioni su preferenziali, direzioni, etc. provengono dalla regione o dai comuni
5.2.4	Integrazione con sistemi di gestione traffico: Integrazione con gestore traffico Autostrada	MIDRA	ETL	processi di Acquisizione dati, Integrazione con gestore traffico Autostrada (PETL14)	Dovranno essere sviluppati interfaccia fra il sistema SII Mobility sia in Input che in Output in modo da garantire sia la corretta acquisizione dei dati dal gestore che di fornire a quest'ultimo i dati elaborati dal sistema
5.2.5	Integrazione con sistemi di gestione traffico: Sviluppo API info SII-mobility verso gestori traffico	IN20	Smart City API	Smart City API verso gestori traffico (API20)	Fornitura dati riguardo ad azioni di alto livello determinate dal Smart Decision Support.

### 5.18.1 Processo ETL: Integrazione con gestore traffico Firenze (PETL11) (MIZAR)

<b>Profilo Tool/Algoritmo</b>	
ETL: Integrazione con gestore traffico Firenze	
Nome responsabile	
Partner responsabile	MIZAR
Descrizione Tool/algoritmo	Acquisizione dati relativi alle informazioni di alto livello del gestore del traffico: dati dei flussi, dati delle temporizzazioni o profili attuati, ordinanze. I dati relativi alle informazioni su preferenziali, direzioni, etc. provengono dalla regione o dai comuni
Dati primari in ingresso	Informazioni di alto livello del gestore del traffico
Dati prodotti in Uscita	Informazioni di alto livello del gestore del traffico
Principi (metodo, basi teoriche, etc.)	Le Informazioni di alto livello del gestore del traffico vengono lette nel formato proprietario del sistema di Firenze e trasformate in modelli di dato normalizzati per la piattaforma SII
Casi di test (presenti/assenti)	nessuno

Posizione casi di test	n.a.	
Principali problemi non risolti	Specifiche di interfaccia del sistema di gestione del traffico di Firenze mancanti; specifiche di modello dati verso SII mancanti	
Principali requisiti pendenti	nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	ETL, Java	
Piattaforme supportate	Unix	
Posizione del codice sorgente	Repository di progetto	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.a.	
Indirizzo/i accesso via web (se presenti)	n.a.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n.a.	n.a.	n.a.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
REST API	n.a.	n.a.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
n.a.	n.a.	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
n.a.	ETL, Java	n.a.
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
n.a.	n.a.	GPL

## 5.18.2 Processo ETL: Integrazione con gestore traffico Prato, Pistoia (PETL12) (MIZAR)

<b>Profilo Tool/Algoritmo</b>		
ETL: Integrazione con gestore traffico Prato, Pistoia		
Nome responsabile		
Partner responsabile	MIZAR	
Descrizione Tool/algoritmo	Acquisizione dati relativi alle informazioni di alto livello del gestore del traffico: dati dei flussi, dati delle temporizzazioni o profili attuati, ordinanze. I dati relativi alle informazioni su preferenziali, direzioni, etc. provengono dalla regione o dai comuni	
Dati primari in ingresso	Informazioni di alto livello del gestore del traffico	
Dati prodotti in Uscita	Informazioni di alto livello del gestore del traffico	
Principi (metodo, basi teoriche, etc.)	Le Informazioni di alto livello del gestore del traffico vengono lette nel formato proprietario del sistema di Prato, Pistoia e trasformate in modelli di dato normalizzati per la piattaforma SII	
Casi di test (presenti/assenti)	nessuno	
Posizione casi di test	n.a.	
Principali problemi non risolti	Specifiche di interfaccia del sistema di gestione del traffico di Prato, Pistoia mancanti; specifiche di modello dati verso SII mancanti	
Principali requisiti pendenti	nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	ETL, Java	
Piattaforme supportate	Unix	
Posizione del codice sorgente	Repository di progetto	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.a.	
Indirizzo/i accesso via web (se presenti)	n.a.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n.a.	n.a.	n.a.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON	n.a.	n.a.
<b>Protocolli usati</b>	<b>Condiviso con</b>	<b>Nome protocollo o</b>

(aggiungere una riga per ogni protocollo usato)	tool/modulo	riferimento a sezione definizione
REST API	n.a.	n.a.
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
n.a.	n.a.	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
n.a.	n.a.	n.a.
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
n.a.	n.a.	GPL

### 5.18.3 Processo ETL: Integrazione con gestore traffico Siena Arezzo (PETL13) (MIZAR)

<b>Profilo Tool/Algoritmo</b>	
ETL: Integrazione con gestore traffico Siena Arezzo	
Nome responsabile	
Partner responsabile	MIZAR
Descrizione Tool/algoritmo	Acquisizione dati relativi alle informazioni di alto livello del gestore del traffico: dati dei flussi, dati delle temporizzazioni o profili attuati, ordinanze. I dati relativi alle informazioni su preferenziali, direzioni, etc. provengono dalla regione o dai comuni
Dati primari in ingresso	Informazioni di alto livello del gestore del traffico
Dati prodotti in Uscita	Informazioni di alto livello del gestore del traffico
Principi (metodo, basi teoriche, etc.)	Le Informazioni di alto livello del gestore del traffico vengono lette nel formato proprietario del sistema di Siena e Arezzo e trasformate in modelli di dato normalizzati per la piattaforma SII
Casi di test (presenti/assenti)	nessuno
Posizione casi di test	n.a.
Principali problemi non risolti	Specifiche di interfaccia del sistema di gestione del traffico di Siena e Arezzo mancanti; specifiche di modello dati verso SII mancanti
Principali requisiti pendenti	nessuno
<b>Aspetti Tecnologici</b>	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0%
Eseguibile/libreria/web app	Eseguibile

Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	ETL, Java	
Piattaforme supportate	Unix	
Posizione del codice sorgente	Repository di progetto	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.a.	
Indirizzo/i accesso via web (se presenti)	n.a.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n.a.	n.a.	n.a.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
REST API	n.a.	n.a.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
n.a.	n.a.	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
n.a.	n.a.	n.a.
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
n.a.	n.a.	GPL

#### 5.18.4 Processo ETL: Integrazione con gestore traffico Autostrada (PETL14) (MIDRA)

<b>Profilo Tool/Algoritmo</b>	
ETL: Integrazione con gestore traffico Autostrada	
Nome responsabile	
Partner responsabile	MIDRA
Descrizione Tool/algoritmo	Dovranno essere sviluppati interfaccia fra il sistema SII Mobility sia in Input che in Output in modo da garantire sia la corretta acquisizione dei dati dal gestore che di fornire a quest'ultimo i dati



	elaborati dal sistema. Processo di integrazione in SII Mobility delle informazioni di traffico e QoS su rete Autostrade	
Dati primari in ingresso	Statistiche di utilizzo del servizio e dati reportistica traffico, provenienti da sistema di bigliettazione Autostrada (MIDRA) e processi acquisizione dati Autostrade (MIDRA).	
Dati prodotti in Uscita	Stime di QoS (situazione traffico, velocità media di percorrenza) utilizzate come coefficienti decisionali per rispondere alle query di navigazione assistita intra/inter sistema (solo Autostrade, con ibridazione Autostrade e rete ferroviaria)	
Principi (metodo, basi teoriche, etc.)	Non applicabile	
Casi di test (presenti/assenti)	Assenti	
Posizione casi di test	n/d	
Principali problemi non risolti	Assenti	
Principali requisiti pendenti	Interfaccia di scambio con il sistema SII Mobility	
Aspetti Tecnologici		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0	
Eseguibile/libreria/web app	Eseguibile + web app	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	C, HTML, PHP, SQL, ETL	
Piattaforme supportate	Windows Server, Linux	
Posizione del codice sorgente	Git	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/d	
Indirizzo/i accesso via web (se presenti)	n/d	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
Apache Web Server	APACHE	HTTP/CGI
MySQL connector for PHP	MYSQL	SQL
Pentaho Kettle per la produzione di ETL		
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione
Da definire		
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
HTTP		Hypertext Transfer Protocol
TCP/IP		
ad-hoc AUTOSTRADE		TELEPASS
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	

MySQL	Database relazionale	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
Web	HTML, PHP, JavaScript, CSS	
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
MySQL		GPL

### 5.18.5 API verso gestori traffico (API20) (IN20)

<b>Profilo Tool/Algoritmo</b>	
Smart City API verso gestori traffico	
Nome responsabile	
Partner responsabile	IN20
Descrizione Tool/algoritmo	Fornitura dati riguardo ad azioni di alto livello determinate dal Smart Decision Support.
Dati primari in ingresso	Dati provenienti dalla KB SiiMobility riguardanti la situazione del traffico o modifiche alla stessa
Dati prodotti in Uscita	Previsione di modifica dei flussi di traffico ed azioni suggerite per ottimizzarne l'impatto
Principi (metodo, basi teoriche, etc.)	In base ai dati raccolti di gestori di traffico, arricchiti con informazioni aggiuntive presenti in SiiMobility, si forniscono suggerimenti sulle azioni di alto livello da far intraprendere a ciascun gestore, affinché i flussi di traffico vengano ottimizzati in relazione allo stato attuale o ad eventi futuri noti.
Casi di test (presenti/assenti)	assenti
Posizione casi di test	
Principali problemi non risolti	
Principali requisiti pendenti	
<b>Aspetti Tecnologici</b>	
Stato (proposto/approvato)	proposto
Implementato/non implementato	non implementato
Stato implementazione, percentuale	0%
Eseguibile/libreria/web app	Endpoint API SiiMobility
Single thread/Multithread	multithread
Linguaggio di sviluppo (java, Php, ETL)	Java o Php
Piattaforme supportate	
Posizione del codice sorgente	Repository git
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se	

necessarie)		
Indirizzo/i accesso via web (se presenti)		
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
5.18.1 – 5.18.2 – 5.18.3 – 5.18.4		
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTP		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
REST API (solo dati)		
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
PHP – Laravel		GPL
JAVA - Atmosphere		GPL

#### 5.18.5.1 Protocollo, API: verso gestori mobilità

Verificare la possibilità di utilizzare preferibilmente un export in DATEX II.

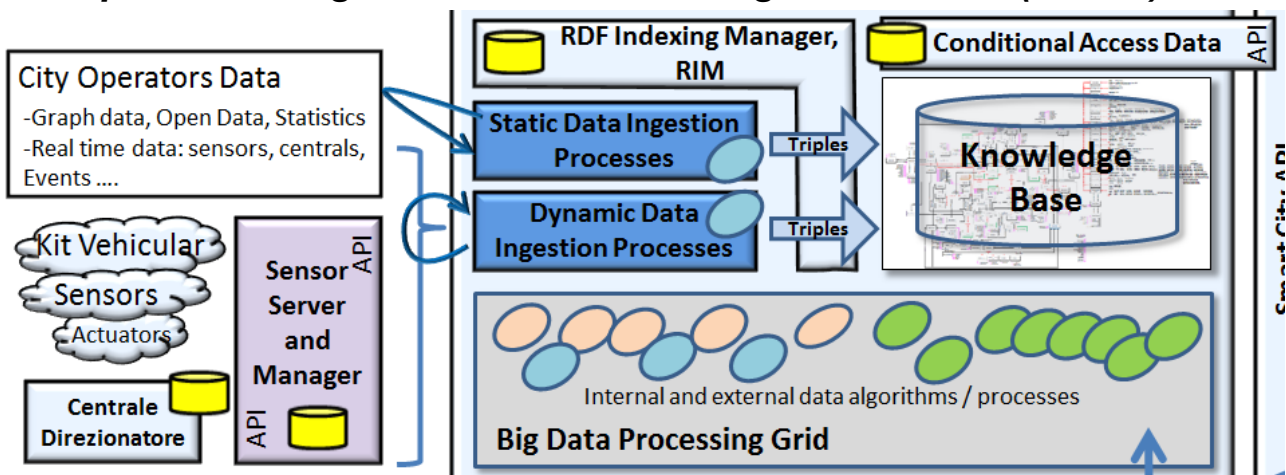
<b>Nome: TrafficEvolution</b>	
<b>URL: /trafficmanagers/evolution</b>	
<b>Descrizione</b>	
Questa API consente di conoscere quali modifiche ai flussi di traffico si sono verificate in un certo arco temporale ed in una area specifica	
Modalità: GET	
<b>Parametri</b>	
area	Descrizione dell'area di interesse
time_start	Periodo iniziale da analizzare
time_end	Periodo finale da analizzare
types	Tipologie di eventi da estrarre

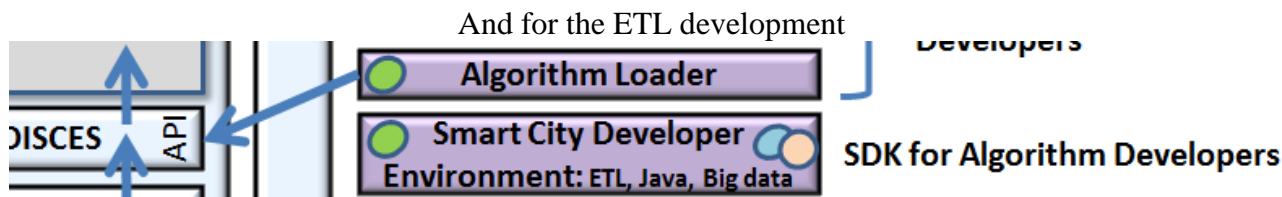
--

<b>Nome: TrafficForecast</b>	
<b>URL: /trafficmanagers/forecast</b>	
<b>Descrizione</b>	
Questa API consente di conoscere quali modifiche ai flussi di traffico sono previste un certo arco temporale ed in una area specifica	
Modalità: GET	
Parametri	
Area	Descrizione dell'area di interesse
Time_start	Periodo iniziale da analizzare
Time_end	Periodo finale da analizzare
types	Tipologie di eventi da estrarre

<b>Nome: ActionSuggestion</b>	
<b>URL: /trafficmanager/suggestion</b>	
<b>Descrizione</b>	
Questa API consente di richiedere suggerimenti sulle azioni da intraprendere per ottimizzare il flusso di traffico in una certa area	
Modalità: GET	
Parametri	
Area	Descrizione dell'area di interesse
Time_start	Orario/Data da cui far iniziare l'azione correttiva
Time_end	Orizzonte temporale lungo il quale l'azione dovrà avere effetto
types	Tipologie di azioni da poter intraprendere
Esempi	

### 5.19 Specifica: Integrazione con sistemi di gestione flotte (MIZAR)





Sottoattività e gruppi di lavoro per questi requisiti		ECM	Mizar	UNIFEL	IN20	TIME	Negentis	EffKnow	liberologico	ataf	busitalia	cttnord	tiemme	argos	elfi	calamai	Midra	Proiect	GEOIN	OUJSTIT	SOETEC	FWINGS	
5.3.1	Integrazione con sistemi di gestione flotte: Firenze		X							X													
5.3.2	Integrazione con sistemi di gestione flotte: Firenze provincia		X								X												
5.3.3	Integrazione con sistemi di gestione flotte: Prato, Pistoia, Pisa		X									X											
5.3.4	Integrazione con sistemi di gestione flotte: Arezzo, Siena		X										X										
5.3.5	Integrazione con sistemi di gestione flotte: Merci e logistica					X																	

**Principali Sottosistemi da descrivere in questa Subsubsection, sono i sottosistemi primari della specifica**

	Sotto Attività	Partner coord	Sottosistema	Tool/Level	Descrizione sommaria
5.3.1	Integrazione con sistemi di gestione flotte: Firenze	MIZAR	ETL	processi di Acquisizione dati, sistemi di gestione flotte: Firenze ATAF (PETL15)	Acquisizione dati relativi ad orari ed in tempo reale per i mezzi ATAF. Invio informazioni di alto livello a sistemi di gestione flotte.
5.3.2	Integrazione con sistemi di gestione flotte: Firenze provincia	MIZAR	ETL	processi di Acquisizione dati, sistemi di gestione flotte: Firenze provincia, BUSITALIA (PETL16)	Acquisizione dati relativi ad orari ed in tempo reale per i mezzi BUSITALIA Invio informazioni di alto livello a sistemi di gestione flotte.
5.3.3	Integrazione con sistemi di gestione flotte: Prato, Pistoia, Pisa	MIZAR	ETL	processi di Acquisizione dati, gestione flotte: Prato, Pistoia, CTTNORD (PETL17)	Acquisizione dati relativi ad orari ed in tempo reale per i mezzi CTTNORD Invio informazioni di alto livello a sistemi di gestione flotte.

5.3.4	Integrazione con sistemi di gestione flotte: Arezzo, Siena	MIZAR	ETL	processi di Acquisizione dati, sistemi di gestione flotte: Arezzo, Siena, TIEMME (PETL18)	Acquisizione dati relativi ad orari ed in tempo reale per i mezzi TIEMME Invio informazioni di alto livello a sistemi di gestione flotte.
5.3.5	Integrazione con sistemi di gestione flotte: Merci e logistica	TIME	ETL	processi di Acquisizione dati, gestione flotte: Merci e logistica (PETL19)	Acquisizione dati relativi ad orari ed in tempo reale per i mezzi trasporto Merci e logistica. Invio informazioni di alto livello a sistemi di gestione flotte.

### 5.19.1 Processo ETL: sistemi di gestione flotte: Firenze ATAF (PETL15) (MIZAR)

<b>Profilo Tool/Algoritmo</b>	
ETL: sistemi di gestione flotte: Firenze	
Nome responsabile	
Partner responsabile	MIZAR
Descrizione Tool/algoritmo	Acquisizione dati relativi ad orari ed in tempo reale per i mezzi ATAF. Invio informazioni di alto livello a sistemi di gestione flotte.
Dati primari in ingresso	Dati relativi ad orari ed in tempo reale per i mezzi ATAF
Dati prodotti in Uscita	Dati relativi ad orari ed in tempo reale per i mezzi ATAF
Principi (metodo, basi teoriche, etc.)	I dati relativi ad orari ed in tempo reale per i mezzi ATAF vengono letti nel formato proprietario del sistema ATAF e trasformati in modelli di dato normalizzati per la piattaforma SII
Casi di test (presenti/assenti)	nessuno
Posizione casi di test	n.a.
Principali problemi non risolti	Specifiche di interfaccia del sistema di gestione flotte ATAF mancanti; specifiche di modello dati verso SII mancanti
Principali requisiti pendenti	nessuno
Aspetti Tecnologici	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0%
Eseguibile/libreria/web app	Eseguibile
Single thread/Multithread	Multithread
Linguaggio di sviluppo (java, Php, ETL)	ETL, java
Piattaforme supportate	Unix
Posizione del codice sorgente	Repository di progetto
Indirizzo/i web services (se presenti) con	n.a.

indicazione credenziali accesso (se necessarie)		
Indirizzo/i accesso via web (se presenti)	n.a.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n.a.	n.a.	n.a.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
REST API	n.a.	n.a.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
n.a.	n.a.	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
n.a.	n.a.	n.a.
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
n.a.	n.a.	GPL

### 5.19.2 Processo ETL: sistemi di gestione flotte: Firenze provincia, BUSITALIA (PETL16) (MIZAR)

<b>Profilo Tool/Algoritmo</b>	
ETL: sistemi di gestione flotte: Firenze provincia	
Nome responsabile	
Partner responsabile	MIZAR
Descrizione Tool/algoritmo	Acquisizione dati relativi ad orari ed in tempo reale per i mezzi BUSITALIA. Invio informazioni di alto livello a sistemi di gestione flotte.
Dati primari in ingresso	Dati relativi ad orari ed in tempo reale per i mezzi BUSITALIA
Dati prodotti in Uscita	Dati relativi ad orari ed in tempo reale per i mezzi BUSITALIA
Principi (metodo, basi teoriche, etc.)	I dati relativi ad orari ed in tempo reale per i mezzi BUSITALIA vengono letti nel formato proprietario del

	sistema BUSITALIA e trasformati in modelli di dato normalizzati per la piattaforma SII	
Casi di test (presenti/assenti)	nessuno	
Posizione casi di test	n.a.	
Principali problemi non risolti	Specifiche di interfaccia del sistema di gestione flotte BUSITALIA mancanti; specifiche di modello dati verso SII mancanti	
Principali requisiti pendenti	nessuno	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	ETL, Java	
Piattaforme supportate	Unix	
Posizione del codice sorgente	Repository di progetto	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.a.	
Indirizzo/i accesso via web (se presenti)	n.a.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n.a.	n.a.	n.a.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
REST API	n.a.	n.a.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
n.a.	n.a.	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
n.a.	n.a.	n.a.
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
n.a.	n.a.	GPL



--	--	--

### 5.19.3 Processo ETL: gestione flotte: Prato, Pistoia, Pisa, CTTNORD (PETL17) (MIZAR)

Profilo Tool/Algoritmo		
ETL: gestione flotte: Prato, Pistoia, Pisa		
Nome responsabile		
Partner responsabile	MIZAR	
Descrizione Tool/algoritmo	Acquisizione dati relativi ad orari ed in tempo reale per i mezzi CTTNORD. Invio informazioni di alto livello a sistemi di gestione flotte	
Dati primari in ingresso	Dati relativi ad orari ed in tempo reale per i mezzi CTTNORD	
Dati prodotti in Uscita	Dati relativi ad orari ed in tempo reale per i mezzi CTTNORD	
Principi (metodo, basi teoriche, etc.)	I dati relativi ad orari ed in tempo reale per i mezzi CTTNORD vengono letti nel formato proprietario del sistema CTTNORD e trasformati in modelli di dato normalizzati per la piattaforma SII	
Casi di test (presenti/assenti)	nessuno	
Posizione casi di test	n.a.	
Principali problemi non risolti	Specifiche di interfaccia del sistema di gestione flotte CTTNORD mancanti; specifiche di modello dati verso SII mancanti	
Principali requisiti pendenti	nessuno	
Aspetti Tecnologici		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	ETL, Java	
Piattaforme supportate	Unix	
Posizione del codice sorgente	Repository di progetto	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.a.	
Indirizzo/i accesso via web (se presenti)	n.a.	
Nomi tool/moduli usati (aggiungere una riga per ogni tool/modulo usato)	Interfacce API usate	Modello di comunicazione e formato
n.a.	n.a.	n.a.
Formati usati (aggiungere una riga per ogni formato usato)	Condiviso con tool/modulo	Nome formato o riferimento a sezione definizione

JSON (verso SII)	n.a.	n.a.
Protocolli usati (aggiungere una riga per ogni protocollo usato)	Condiviso con tool/modulo	Nome protocollo o riferimento a sezione definizione
REST API (verso SII)	n.a.	n.a.
Nomi database usati (aggiungere una riga per ogni db usato)	Descrizione	
n.a.	n.a.	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
n.a.	n.a.	n.a.
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
n.a.	n.a.	GPL

#### 5.19.4 Processo ETL: sistemi di gestione flotte: Arezzo, Siena (PETL18) (MIZAR)

<b>Profilo Tool/Algoritmo</b>	
ETL: sistemi di gestione flotte: Arezzo, Siena	
Nome responsabile	
Partner responsabile	MIZAR
Descrizione Tool/algoritmo	Acquisizione dati relativi ad orari ed in tempo reale per i mezzi TIEMME. Invio informazioni di alto livello a sistemi di gestione flotte.
Dati primari in ingresso	Dati relativi ad orari ed in tempo reale per i mezzi TIEMME
Dati prodotti in Uscita	Dati relativi ad orari ed in tempo reale per i mezzi TIEMME
Principi (metodo, basi teoriche, etc.)	I dati relativi ad orari ed in tempo reale per i mezzi TIEMME vengono letti nel formato proprietario del sistema TIEMME e trasformati in modelli di dato normalizzati per la piattaforma SII
Casi di test (presenti/assenti)	nessuno
Posizione casi di test	n.a.
Principali problemi non risolti	Specifiche di interfaccia del sistema di gestione flotte TIEMME mancanti; specifiche di modello dati verso SII mancanti
Principali requisiti pendenti	nessuno
Aspetti Tecnologici	
Stato (proposto/approvato)	Proposto

Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Eseguibile	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	ETL, Java	
Piattaforme supportate	Unix	
Posizione del codice sorgente	Repository di progetto	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.a.	
Indirizzo/i accesso via web (se presenti)	n.a.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
n.a.	n.a.	n.a.
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON	n.a.	n.a.
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
REST API	n.a.	n.a.
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
n.a.	n.a.	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
n.a.	n.a.	n.a.
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
n.a.	n.a.	GPL

### 5.19.5 Processo ETL: gestione flotte: Merci e logistica (PETL19) (TIME)

Questo tool consente l'invio ai sistemi informativi dei corrieri dei dati relativi alle consegne /ritiri effettuati da ogni mezzo/utente (scarico merci), fornendo il report ed il tracciamento delle stesse a ciascun corriere da parte di ciascun utente.

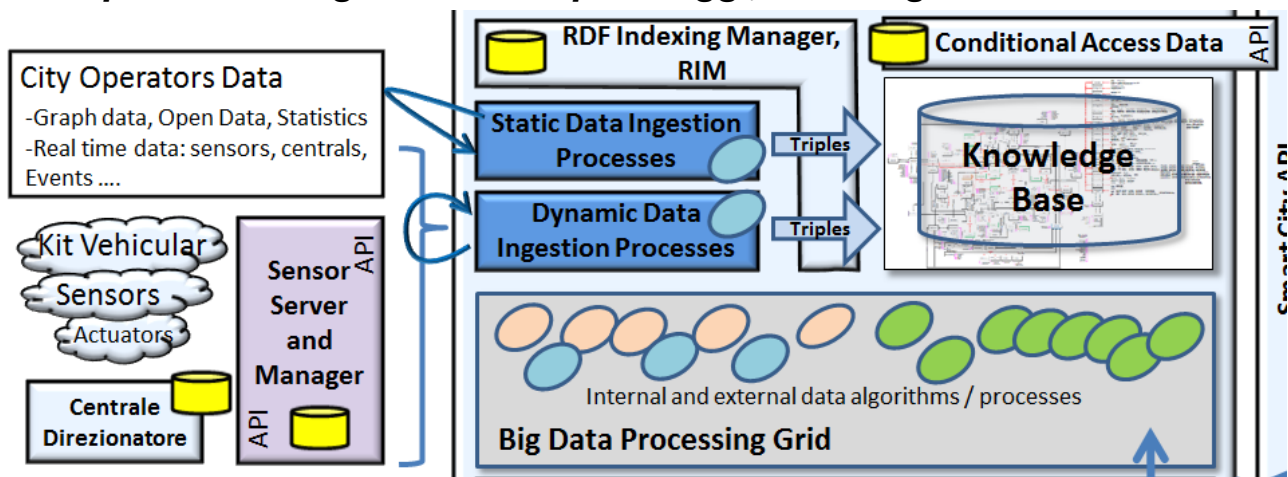
## Profilo Tool/Algoritmo

ETL: gestione flotte: Merci e logistica

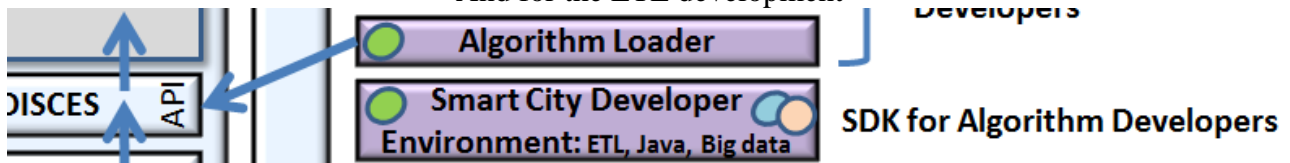
Nome responsabile		
Partner responsabile	TIME	
Descrizione Tool/algoritmo	<p>Il tool gestisce l'acquisizione dei dati di consegna/ritiro forniti da ciascun utente mobile attraverso l'APP di cui Al modulo MAPP14, v. 5.10.13</p> <p>Il Tool si basa sull'utilizzo della piattaforma Pentaho Data Integration (Open Source) che consente lo scambio dati con i sistemi informativi dei corrieri e produce report di consegna/ritiro per ciascun corriere, da inviare attraverso messaggio XML ai sistemi informativi dei corrieri</p>	
Dati primari in ingresso	Dati da APP di cui a 5.10.13	
Dati prodotti in Uscita	Report consegne/ritiri effettuati da ciascun mezzo/utente a ciascun corriere	
Principi (metodo, basi teoriche, etc.)	XML	
Casi di test (presenti/assenti)	Da definire	
Posizione casi di test	Da definire	
Principali problemi non risolti	Nessuno	
Principali requisiti pendenti	I/F web services o messaggistica XML con sistemi informativi corrieri, che necessitano di protocolli di agreement	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Eseguibile/web app	
Single thread/Multithread	Single Thread	
Linguaggio di sviluppo (java, Php, ETL)	Java, Php	
Piattaforme supportate	Windows, Linux	
Posizione del codice sorgente	n.d.	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.d.	
Indirizzo/i accesso via web (se presenti)	n.d.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Pentaho data integration	Java API	XML
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
Da definire		
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
http, EDI, XML		
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	

MySQL	RDBMS	
Tipo interfaccia utente (web/applicazione)	Modello sviluppo, linguaggio	Libreria usata per UI
WEB, EXE		
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.
Pentaho Data Integration (Open Source)		GPLv2

### 5.20 Specifica: integrazione con parcheggi, ed altri gestori servizi



And for the ETL development



		ECM	Mizar	UNIFI	IN2O	TIME	Negentis	Effknow	liberologico	ataf	busitalia	ctnord	tiemme	areos	elfi	calamai	Midra	Proiect	GEOIN	OUJSTIT	SOETEC	FWINGS
5.4.1	integrazione con parcheggi, ed altri gestori servizi: parcheggi vari								X											X		
5.4.2	integrazione con parcheggi, ed altri gestori servizi: car and bike sharing					X														X		
5.4.3	integrazione con parcheggi, ed altri gestori servizi: taxi, car service																X			X		
5.4.4	integrazione con parcheggi, ed altri gestori servizi: Nettezza e servizi																X			X		

Principali Sottosistemi da descrivere in questa Subsubsection, sono i sottosistemi primari della specifica

Sotto Attività	Partner coord	Sotto sistema	Tool/Level	Descrizione sommaria
----------------	---------------	---------------	------------	----------------------

5.4.1	integrazione con parcheggi, ed altri gestori servizi: parcheggi vari	LIBEROL OGICO	ETL	processi di Acquisizione dati, gestori servizi: parcheggi vari (PETL20)	Sistema di API/WS che espongono dati statici (per es. geolocalizzazione degli stalli di sosta sensorizzati, tipologia di stalli, piano tariffario della sosta cittadino, dislocazione delle rastrelliere del bike sharing, dislocazione dei gate RFID a cintura della città, ecc.) e dinamici (per es. stato libero/occupato/fuori servizio degli stalli monitorati dai sensori di sosta, flusso veicoli transitati da ogni gate RFID, ecc.) relativi ai sistemi di mobilità e sosta in disponibilità dei gestori della mobilità e sosta di Pisa e Firenze al Sii-Mobility e a sistemi di terze parti registrati. Questo modulo include sia il sistema di acquisizione dati da installare presso i gestori che il processo ETL per l'acquisizione dati statici (struttura, posti potenziali, posizioni, etc.) che dinamici (posti vuoti, posizione di tali posti, etc.)
5.4.2	integrazione con parcheggi, ed altri gestori servizi: car and bike sharing	QuestIT	ETL	processi di Acquisizione dati, gestori servizi: car and bike sharing (PETL21)	Sistema di API/WS da installare presso i gestori di servizi 'car & bike sharing' che consente di esporre i dati grezzi del gestore per la conseguente acquisizione tramite un processo ETL custom per ciascun gestore. Il sistema di API/WS esporrà anche i metodi per l'action triggering di Sii-Mobility verso i sottosistemi periferici. Questo modulo include sia il sistema di acquisizione dati da installare presso i gestori che il processo ETL per l'acquisizione dati statici (struttura, posti potenziali, posizioni, etc.) che dinamici (bike disponibili, posizione di tali bike, etc.)
5.4.3	integrazione con parcheggi, ed altri gestori servizi: taxi, car service	QuestIT	ETL	processi di Acquisizione dati, gestori servizi: taxi, car service (PETL22)	Sistema server di API/WS da installare presso i gestori di servizi 'taxi & car service' che consente di esporre i dati grezzi del gestore per la conseguente acquisizione tramite un processo ETL custom per ciascun gestore. Il sistema di API/WS esporrà anche i metodi per l'action triggering di Sii-Mobility verso i sottosistemi periferici. Questo modulo include sia il sistema di acquisizione dati da installare presso i gestori che il processo ETL per l'acquisizione dati statici (struttura, posti potenziali, posizioni, etc.) che dinamici (car disponibili, posizione di tali car, etc.)
5.4.4	integrazione con parcheggi, ed altri gestori servizi: Nettezza e servizi	QuestIT	ETL	processi di Acquisizione dati, gestori servizi: Nettezza e servizi (PETL23)	Sistema server di API/WS da installare presso i gestori di servizi 'nettezza' che consente di esporre i dati grezzi del gestore per la conseguente acquisizione tramite un processo ETL custom per ciascun gestore. Il sistema di API/WS esporrà anche i metodi per l'action triggering di Sii-Mobility verso i sottosistemi periferici. Questo modulo include sia il sistema di acquisizione dati da installare presso i gestori che il processo ETL per l'acquisizione dati statici (struttura, percorsi pianificati, etc.) che dinamici (posizione attuale mezzi, etc.)

### 5.20.1 Processo ETL: gestori servizi: parcheggi vari (PETL20) (LIBEROLOGICO)

<b>Profilo Tool/Algoritmo</b>	
ETL: gestori servizi: parcheggi vari	
Nome responsabile	
Partner responsabile	LIBEROLOGICO
Descrizione Tool/algoritmo	Sistema di API/WS che espongono dati statici (per es. geolocalizzazione degli stalli di sosta sensorizzati, tipologia di stalli, piano tariffario della sosta cittadino, dislocazione delle rastrelliere del bike sharing, dislocazione dei gate RFID a cintura della città, ecc.) e dinamici (per es. stato libero/occupato/fuori servizio degli stalli monitorati dai sensori di sosta, flusso veicoli transitati da ogni gate RFID, ecc.) relativi ai sistemi di mobilità e sosta in disponibilità dei gestori della mobilità e sosta di Pisa e Firenze al Sii-Mobility e a sistemi di terze parti registrati. Questo modulo include sia il sistema di acquisizione dati da installare presso i gestori che il processo ETL per l'acquisizione dati statici (struttura, posti potenziali, posizioni, etc.) che dinamici (posti vuoti, posizione di tali posti, etc.). Il processo viene gestito dal DISCES.
Dati primari in ingresso	Accesso ai Dati del gestore tramite chiamate API Rest
Dati prodotti in Uscita	Triple RDF relative ai dataset ingeriti
Principi (metodo, basi teoriche, etc.)	Gli ETL vengono messi in esecuzione da DISCES in modo periodico o aperiodico.
Casi di test (presenti/assenti)	Assenti
Posizione casi di test	n/d
Principali problemi non risolti	Assenti
Principali requisiti pendenti	Nessuno
<b>Aspetti Tecnologici</b>	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato
Stato implementazione, percentuale	0%
Eseguibile/libreria/web app	Eseguibile
Single thread/Multithread	Multithread
Linguaggio di sviluppo (java, Php, ETL)	ETL
Piattaforme supportate	LINUX, Windows
Posizione del codice sorgente	git
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n/a
Indirizzo/i accesso via web (se presenti)	n/a

<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
Ingestion	n/a	REST API, JSON
Triplification	n/a	n/a
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
JSON		JavaScript Object Notation
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
HTTPS		HyperText Transfer Protocol over Secure Socket Layer
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
MySQL	Database relazionale	
HBase	Database NoSql distribuito su filesystem Hadoop	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
n/a		
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>
n/a		

## 5.20.2 Processo ETL: gestori servizi: car and bike sharing (PETL21) (QUESTIT)

<b>Profilo Tool/Algoritmo</b>	
ETL: gestori servizi: car and bike sharing	
<b>Nome responsabile</b>	
<b>Partner responsabile</b>	QuestIT s.r.l.
<b>Descrizione Tool/algoritmo</b>	<p>Il modulo fornisce delle API REST che consentono di esporre i dati grezzi del gestore in formato JSON per la conseguente acquisizione tramite un processo ETL (uno per ciascun gestore). Inoltre espone l'API per l'action triggering per i sottosistemi del gestore.</p> <p>Sistema di API/WS da installare presso i gestori di servizi 'car &amp; bike sharing' che consente di esporre i dati grezzi del gestore per la conseguente acquisizione tramite un processo ETL custom per ciascun gestore.</p> <p>Il sistema di API/WS esporrà anche i metodi per l'action</p>



	triggering di Sii-Mobility verso i sottosistemi periferici. Questo modulo include sia il sistema di acquisizione dati da installare presso i gestori che il processo ETL per l'acquisizione dati statici (struttura, posti potenziali, posizioni, etc.) che dinamici (bike disponibili, posizione di tali bike, etc.)	
Dati primari in ingresso	Stringhe	
Dati prodotti in Uscita	Dati in HBASE	
Principi (metodo, basi teoriche, etc.)	L'API fornirà, oltre ai dati grezzi, l'identificativo del sistema/gestore che li genera.	
Casi di test (presenti/assenti)	n.d.	
Posizione casi di test	n.d.	
Principali problemi non risolti	n.d.	
Principali requisiti pendenti	n.d.	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Processo ETL	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	ETL	
Piattaforme supportate	Quelle supportate da ETL	
Posizione del codice sorgente	n.d.	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.d.	
Indirizzo/i accesso via web (se presenti)	n.d.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo,</b>	<b>Libreria usata per UI</b>

	linguaggio	
Libreria usata (aggiungere una riga per ogni libreria usata)	Nome e versione usata	Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.

### 5.20.3 Processo ETL: gestori servizi: taxi, car service (PETL22) (QUESTIT)

<b>Profilo Tool/Algoritmo</b>	
ETL: gestori servizi: taxi, car service	
Nome responsabile	
Partner responsabile	QuestIT s.r.l.
Descrizione Tool/algoritmo	<p>Il modulo fornisce delle API REST che consentono di esporre i dati grezzi del gestore in formato JSON per la conseguente acquisizione tramite un processo ETL (uno per ciascun gestore). Inoltre espone l'API per l'action triggering per i sottosistemi del gestore.</p> <p>Sistema server di API/WS da installare presso i gestori di servizi 'taxi &amp; car service' che consente di esporre i dati grezzi del gestore per la conseguente acquisizione tramite un processo ETL custom per ciascun gestore.</p> <p>Il sistema di API/WS esporrà anche i metodi per l'action triggering di Sii-Mobility verso i sottosistemi periferici. Questo modulo include sia il sistema di acquisizione dati da installare presso i gestori che il processo ETL per l'acquisizione dati statici (struttura, posti potenziali, posizioni, etc.) che dinamici (car disponibili, posizione di tali car, etc.)</p>
Dati primari in ingresso	Stringhe
Dati prodotti in Uscita	Dati in HBASE
Principi (metodo, basi teoriche, etc.)	L'API fornirà, oltre ai dati grezzi, l'identificativo del sistema/gestore che li genera.
Casi di test (presenti/assenti)	n.d.
Posizione casi di test	n.d.
Principali problemi non risolti	n.d.
Principali requisiti pendenti	n.d.
Aspetti Tecnologici	
Stato (proposto/approvato)	Proposto
Implementato/non implementato	Non implementato

Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Processo ETL	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	ETL	
Piattaforme supportate	Quelle supportate da ETL	
Posizione del codice sorgente	n.d.	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.d.	
Indirizzo/i accesso via web (se presenti)	n.d.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>
<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

#### 5.20.4 Processo ETL: gestori servizi: Nettezza e servizi (PETL23) (QUESTIT)

<b>Profilo Tool/Algoritmo</b>	
ETL: gestori servizi: Nettezza e servizi	
Nome responsabile	
Partner responsabile	QuestIT s.r.l.
Descrizione Tool/algoritmo	Il modulo fornisce delle API REST che consentono di esporre i dati grezzi del gestore in formato JSON

	<p>per la conseguente acquisizione tramite un processo ETL (uno per ciascun gestore). Inoltre espone l'API per l'action triggering per i sottosistemi del gestore.</p> <p>Sistema server di API/WS da installare presso i gestori di servizi 'nettezza' che consente di esporre i dati grezzi del gestore per la conseguente acquisizione tramite un processo ETL custom per ciascun gestore.</p> <p>Il sistema di API/WS esporrà anche i metodi per l'action triggering di Sii-Mobility verso i sottosistemi periferici. Questo modulo include sia il sistema di acquisizione dati da installare presso i gestori che il processo ETL per l'acquisizione dati statici (struttura, percorsi pianificati, etc.) che dinamici (posizione attuale mezzi, etc.)</p>	
Dati primari in ingresso	Stringhe	
Dati prodotti in Uscita	Dati in HBASE	
Principi (metodo, basi teoriche, etc.)	L'API fornirà, oltre ai dati grezzi, l'identificativo del sistema/gestore che li genera.	
Casi di test (presenti/assenti)	n.d.	
Posizione casi di test	n.d.	
Principali problemi non risolti	n.d.	
Principali requisiti pendenti	n.d.	
<b>Aspetti Tecnologici</b>		
Stato (proposto/approvato)	Proposto	
Implementato/non implementato	Non implementato	
Stato implementazione, percentuale	0%	
Eseguibile/libreria/web app	Processo ETL	
Single thread/Multithread	Multithread	
Linguaggio di sviluppo (java, Php, ETL)	ETL	
Piattaforme supportate	Quelle supportate da ETL	
Posizione del codice sorgente	n.d.	
Indirizzo/i web services (se presenti) con indicazione credenziali accesso (se necessarie)	n.d.	
Indirizzo/i accesso via web (se presenti)	n.d.	
<b>Nomi tool/moduli usati</b> (aggiungere una riga per ogni tool/modulo usato)	<b>Interfacce API usate</b>	<b>Modello di comunicazione e formato</b>
<b>Formati usati</b> (aggiungere una riga per ogni formato usato)	<b>Condiviso con tool/modulo</b>	<b>Nome formato o riferimento a sezione definizione</b>

<b>Protocolli usati</b> (aggiungere una riga per ogni protocollo usato)	<b>Condiviso con tool/modulo</b>	<b>Nome protocollo o riferimento a sezione definizione</b>
<b>Nomi database usati</b> (aggiungere una riga per ogni db usato)	<b>Descrizione</b>	
<b>Tipo interfaccia utente</b> (web/applicazione)	<b>Modello sviluppo, linguaggio</b>	<b>Libreria usata per UI</b>
<b>Libreria usata</b> (aggiungere una riga per ogni libreria usata)	<b>Nome e versione usata</b>	<b>Licenza: GPL, LGPL, PEK, proprietaria, commerciale, etc.</b>

## 6 Specifica e scenari della Sperimentazione: Validazione e Sperimentazione sul campo delle applicazioni specifiche e del SII (TIME)

### 6.1 Azioni della sperimentazione (DISIT)

In **Tabella 2**, viene riportata una **descrizione delle tipologie di dati** che saranno integrati e sfruttati nell'infrastruttura **Sii-Mobility**. Per i dati si faccia riferimento al deliverable dei requisiti e tabella allegata.

<b>Tabella 2: Mappatura dei sistemi in riferimento alle PA coinvolte, le Celle evidenziate in celeste con SM mostrano l'inserimento/ampliamento di servizi con servizi/prodotti evoluti di Sii-Mobility</b>								
<b>Funzionalità / informazioni</b>	<b>Comune di Arezzo</b>	<b>Provincia di Firenze</b>	<b>Comune di Firenze</b>	<b>Comune di Pisa</b>	<b>Comune di Pistoia</b>	<b>Provincia di Prato</b>	<b>Comune di Prato</b>	<b>Regione Toscana</b>
Interoperabilità fra centrali Sii-Mobility	SM	SM	SM	SM	SM	SM	SM	SM
Capacità deduttive, supporto alle decisioni Sii-Mobility	SM	SM	SM	SM	SM	SM	SM	SM
Applicazioni mobili Sii-Mobility		SM	SM	SM	SM	SM	SM	SM
Sensori parcheggio	x		x	xSM nuovi	x			
Sensori, sistema di monitoraggio Sii-Mobility		x (S.I.Mo. Ne)	x(S.I.Mo. Ne – SIGMA+)				x	
Sensori evoluti Sii-Mobility			SM	SM	SM		SM	
Monitoraggio traffico	x	x(S.I.Mo. Ne)	x(SI.Mo. GMA+)	x			x	x
Dati territorio	x	x	x			x(Inf Turistico)		x
Stradari dettagliati	x		x	x		X		X
rete ferroviaria, informazioni di vario tipo		x(Memorario)		x	x			
AVM trasporto pubblico	x		x	SM nuovi	x	X	X	
Kit bici (sharing-privati)			SM		SM		SM	
Kit auto (sharing, privati, taxi)			SM		SM		SM	
UTC (gestione semafori)	x		x(SIGMA+)		x			
social media, Partecipazione e sensibilizzazione, Web, mobile e Totem (Sii-Mobility)		SM	x SM	SM	SM	SM	SM	SM
Infomobility	x	x(S.I.Mo. Ne)	x(S.I.Mo. Ne)	x(PisaPass)	x(PIM)			x
varchi telematici, ZTL Attuatori (e.g., direzione, velocità) Sii-Mobility	x		x(SI.Mo. Ne)SM		SM		SM	
ordinanze, eventi pubblici		x	x					
parametri ambientali		x	x(Clean)		SM	X	X	x

			SM				SM	
servizi ed enti		x	x(Unplugged)				x	
emergenze polizia e 118	x		x					x
merci			x(Catalist)					

La seguente tabella 5 riporta il modello di valutazione qualitativo come descritto nel capitolato.

<b>Tabella 5: Relazioni fra attività di ricerca, obiettivi generali, ricadute possibili, validazione e risultati</b>				
<b>Obiettivo Generale</b>	<b>Attività di Ricerca</b>	<b>Ricadute Possibili in termini di impiego delle risorse e qualità della vita</b>	<b>Validazione ed evidenze</b>	<b>Originalità e utilità delle conoscenze acquisibili in riferimento allo stato dell'arte internazionale</b>
1. ridurre i costi sociali della mobilità	Infrastruttura per data integration, algoritmi di analisi, sensori ed attuatori, ottimizzazioni e, applicazioni mobili per il cittadino, totem informativi. Soluzioni di interoperabilità con altri sistemi di gestione.	<ul style="list-style-type: none"> <li>○ minori disagi, maggiore efficienza, maggiore informazione e maggiore integrazione dei servizi al cittadino, minori emissioni;</li> <li>○ percorsi informativi in modo che il cittadino cambi le abitudini non virtuose;</li> <li>○ ridurre i costi di trasporto ed i tempi di percorrenza per gli utenti, per i gestori e le amministrazioni;</li> </ul>	sperimentazione del sistema Sii-Mobility nelle province e comuni di Firenze, Prato, Pistoia, nei comuni di Arezzo e Pisa.	<p>Nuovi algoritmi per la pianificazione di percorsi in tempo reale anche con applicazioni verso il Turismo.</p> <p>Nuove soluzioni di integrazione fra Treni, BUS, sharing, taxi, trasporto merci, e privati.</p> <p>Riduzione dei tempi di reazione a casi critici, aumento della velocità di adattamento del sistema di viabilità.</p>
2. semplificare e l'uso dei sistemi di mobilità per il cittadino	Infrastruttura per data integration, sensori ed attuatori, algoritmi di analisi, ottimizzazioni e, Sistemi di presentazione e di sensibilizzazione	<ul style="list-style-type: none"> <li>○ minori disagi, maggiore efficienza</li> <li>○ Sistemi di pagamento parcheggio ed entrate direttamente basati su trasponder e sensori sul veicolo, pagamento senza click;</li> <li>○ servizi integrati innovativi al cittadino e alla pubbliche amministrazioni,</li> <li>○ applicazioni mobili e totem che possano informare e formare;</li> </ul>	sperimentazione del sistema Sii-Mobility nelle province e comuni di Firenze, Prato, Pistoia, nei comuni di Arezzo e Pisa.	<p>Nuove applicazioni per semplificare la mobilità.</p> <p>Riduzione dei costi di mobilità per il cittadino e di gestione per la PA.</p> <p>Nuove soluzioni di sensibilizzazione capaci di modificare il comportamento utente non virtuoso.</p> <p>Maggiore</p>

				interoperabilità fra sistemi di gestione diversi
3. sviluppare applicazioni e servizi aggiuntivi funzionanti e porre le basi per farne molte altre permettendo la sostenibilità	Definizione di API, opendata, applicazioni mobili per il cittadino, totem informativi. Sperimentare queste soluzioni con le PA e PMI per arrivare ad una validazione.	<ul style="list-style-type: none"> <li>○ rendere possibile l'accesso alle informazioni correlate alla mobilità verso le PA e Industria;</li> <li>○ aiutare le industrie a sviluppare applicazioni specifiche innovative, aprendo in questo modo il mercato dei servizi innovativi.</li> <li>○ apertura del servizio verso altri gestori in modo da poter incrementare il numero di flotte, mezzi, reti di sensori ed aspetti che vengono tenuti in considerazione durante le fasi di analisi dei dati;</li> </ul>	sperimentazione del sistema Sii-Mobility nelle province e comuni di Firenze, Prato, Pistoia, nei comuni di Arezzo e Pisa. con il supporto dei Poli di Innovazione POLIS e POLITER, oltre 800 industrie e	<p>Piattaforma di sviluppo collaborativa.</p> <p>Applicazioni innovative nei campi applicativi mobilità integrata con: turismo, lavoro, eventi, gestione crisi, gestione incidenti. Integrazione con Autostrade, ferrovie, TPL, e sistemi di gestione.</p>
4. Contribuire al miglioramento degli standard Contribuire all'avanzamento della ricerca del settore	Tutte le attività di ricerca	Ricadute a livello internazionale su tutti i sistemi di mobilità. Leadership Italiana nel settore, riconosciuta a livello internazionale. Nuovi standard di interoperabilità condivisi	sperimentazione del sistema Sii-Mobility e dei suoi componenti di ricerca	Nuovi algoritmi, soluzioni e metodi nel campo dei sistemi ITS, delle tematiche di ricerca di base, e nelle tematiche di ricerca applicata.
	Data modeling, ITS system, AVM system	Soluzioni a basso costo per AVM per offrire vantaggi al cittadino	Contributi agli enti di standardizzazione	Incremento dell'interoperabilità dei sistemi di mobilità.

La seguente tabella 4 riporta il modello di valutazione quantitativo come descritto nel capitolato.

<b>Tabella 4: Parametri quantitativi del progetto in riferimento agli Obiettivi Generali per azioni di verifica</b>			
<b>Obiettivo Generale</b>	<b>Descrizione della metrica</b>	<b>Valore atteso intermedio M14, Febbraio 2017</b>	<b>Valore atteso finale, M30, Giugno 2018</b>
1. ridurre i costi sociali	Valutazione generale dell'uso della soluzione proposta e dei benefici da parte del singolo, delle PA e di PMI. Valutazione fatta su un campione	10%	20%



della mobilità	ridotto ma significativo per caratteristiche.		
	Crescita nella frequenza di utilizzo delle tecnologie avanzate nella mobilità urbana da parte delle pubbliche amministrazioni.	3-5%	8-10%
	Percentuale di cittadini, esterni alla sperimentazione (fuori dal campione), che si sono interessati al progetto Sii-Mobility partecipando alla discussione sul portale o coinvolti mediante social network. La percentuale sarà riferita al numero di abitanti delle zone interessate dalla sperimentazione.	0,5-1,0%	1,2-1,0%
	Numero di nuovi servizi avanzati al cittadino	2	8
	Numero di centrali integrate con Sii-Mobility: TPL, semaforiche, parcheggi, meteo, ZTL, accessi, etc.	3	15
	Numero di nuovi algoritmi di data analytic per questioni di mobilità e trasporti	7	20-25
	Numero di Algoritmi e soluzioni integrate per la misurazione dei flussi di utenti in mobilità	2	3-5
	Livello di soddisfazione degli utenti che viaggiano sulla rete di trasporto a seguito dell'utilizzo dei servizi offerti da Sii-Mobility	Il 70% del campione deve percepire un miglioramento di almeno 3 (scala da 1 a 5)	L' 85% del campione deve percepire un miglioramento di almeno 3 (scala da 1 a 5)
2. semplificare l'uso dei sistemi di mobilità per il cittadino	Aggiunta di funzionalità non presenti sul mercato dei sistemi ITS.	15	25-30
	Livello di apprezzamento della soluzione da parte del gruppo di utenti esperti finali.	Buono	Ottimo
	Percentuale di accessi sui pannelli informativi di Sii-Mobility rispetto alla popolazione in transito	0,5-1,0%	1,2-1,0%
	Numero di applicazioni dimostrative di Sii-Mobility scaricate al mese (su piattaforma Android)	700-950	950-1600
	Numero di applicazioni dimostrative di Sii-Mobility scaricate al mese (su piattaforma iOS)	700-950	950-1600
	Numero di commenti positivi ricevuti tramite applicazioni di Sii-Mobility da parte dell'utenza finale.	50% (in assoluto # commenti 3%)	70% (in assoluto # commenti 5%)
	Numero di totem informativi installati ed utilizzati	5	12-20
	Numero di apparati sensori/attuatori innovativi sviluppati e/o utilizzati nelle sperimentazioni	--	100 + 200
	Numero di sistemi di bigliettazione e/o pagamento integrati (considerando che si parte da un sistema integrato in Toscana)	2	5-10
	Livello di comprensione della soluzione e dell'uso delle API da parte di utenti sviluppatori	Buono	Ottimo
3. svil	Numero di applicazioni (e/o moduli) innovative per	5	30

appare applicazioni e servizi aggiuntivi funzionanti e porre le basi per farne molte altre permettendo la sostenibilità	mobile che utilizzeranno Sii-Mobility		
	Numero di PMI e PA che avranno aderito a Sii-Mobility come affiliati	10-20	20-40
	Numero di apparati sensori/attuatori innovativi sviluppati e/o utilizzati nelle sperimentazioni	--	100 + 200
	Numero di applicazioni che vanno usare Sii-Mobility	7	20
	Numero di algoritmi per il supporto alla decisioni sulla forma di aree ad accesso controllato e direzionamenti	2	4-9
	Numero di PA coinvolte nella sperimentazione	4	10-15
	Numero di modelli ed algoritmi di guida connessa	1	3
	Numero partecipanti ai congressi e organizzazione di meeting di formazione del progetto incluso gli Hackathon	35-50	170-250
4. Contribuire al miglioramento degli standard nazionali ed internazionali Contribuire all'avanzamento della ricerca del settore	Numero di contributi a corpi di standardizzazione	1	2
	Percentuale di soluzioni di integrazione innovative introdotte negli standard nazionali ed internazionali (riferita al totale delle soluzioni realizzate e validate)	1-3%	3-5%
	Percentuale di soluzioni conformi agli standard nazionali ed internazionali (riferita al totale delle soluzioni realizzate e validate )	70-80%	80-90%
	Numero di pubblicazioni a livello internazionale con referee	8-10	15-30
	Numero di pubblicazioni di livello nazionale con referee	3-5	8-10
	Numero di accessi sul portale divulgativo della piattaforma Sii-Mobility	35.000-55.000	90.000-160.000
	Numero di pubblicazioni nazionali divulgative	10-20	27-40

## 6.2 Luoghi e scenari potenziali della sperimentazione

Le Regione Toscana è uno dei territori più complessi riguardo alla mobilità perché è contraddistinto da peculiarità orografiche, geomorfologiche e urbanistiche, e storicamente composto in innumerevoli piccoli insediamenti produttivi, che danno luogo a significativi movimenti di pendolarismo (e.g., Firenze ha il 51.6% di spostamenti interni ed il 38% di spostamenti in entrata, con solo il 10% di uscite). Si ha pertanto una forte esigenza di soluzioni ampie ed efficienti di multimodalità ed intermodalità dei trasporti, con una incisiva razionalizzazione del trasporto stradale, nelle aree urbane e metropolitane (si veda **Figura 8**). L'area metropolitana di Firenze-Prato-Pistoia, viene vista come un unicum nazionale; per i fattori descritti in precedenza, per aspetti multiculturali (turismo, studenti esterni, immigrati).

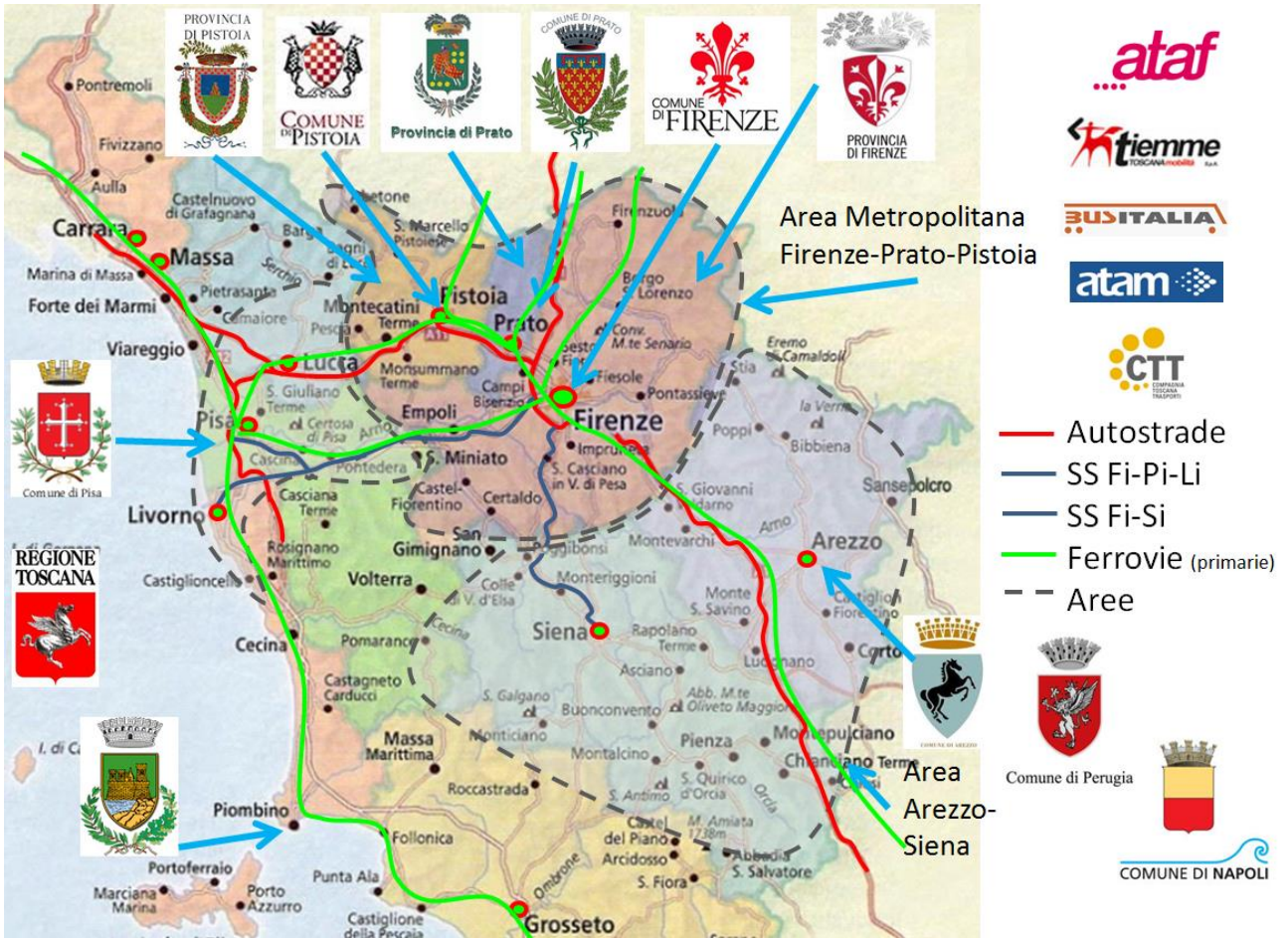


Figura 8: Aree di sperimentazione primaria del progetto Sii-Mobility

### 6.3 Specifica: dimostratori sul campo, attuatori e sensori, etc. HW/SW (TIME, MIZAR)

Sottoattività e gruppi di lavoro per questi requisiti		FCM	Mizar	UNIEI	IN20	TIME	Neentis	EffKnow	liberologico	ataf	bustalia	cttnord	tiemme	argos	elfi	calamai	Midra	Project	GEOIN	OUJSTIT	SOFTEC	FWINGS
2.7.1	Supporto alla sperimentazione con tutti i Comuni: Smart City Integrata		X		X	X			X	X	X											
2.7.2	Supporto alla sperimentazione con tutti i Comuni: Gestione Accessi e ZTL	X										X										
2.7.3	Supporto alla sperimentazione con tutti i Comuni: Interoperabilità		X						X	X	X	X	X									
2.7.4	Supporto alla sperimentazione con tutti i Comuni: metodi pagamento						X		X	X	X	X	X									
2.7.5	Supporto alla sperimentazione con tutti i Comuni: demo integrazione		X	X			X			X	X	X	X									
2.7.6	Supporto alla sperimentazione con tutti i Comuni: demo algoritmi		X	X		X	X			X												

Installazione dei dimostratori sul campo, attuatori e sensori, etc. HW/SW		Mizar	IN20	Soft ec	TIM E	liber olog ico	MID RA	ataf	ata m	busi t	ctt	tiem me	eWi ngs	argo s	elfi	cala mai	proj ect
6.1.1	piano con gestori: avm, kit bike, car, etc.	X						X	X	X	X	X		X	X	X	X
6.1.2	piano con PA: sensori, attuatori, totem	X		X	X		X						X				X
6.1.3	messa in opera: kit, totem, sensori, car, evoluti,...	X	X	X	X	X	X	X	X	X	X	X					
6.1.4	test e collaudo	X			X			X	X	X	X	X		X	X	X	X
6.1.5	documentazione	X	X	X	X	X	X					X	X	X	X	X	X

Principali Sottosistemi da descrivere in questa Subsubsection, sono i sottosistemi primari della specifica					
	Sotto Attività	Partner coord	Sottosistema	Tool/Level	Descrizione sommaria
2.7.1	Supporto alla sperimentazione e con tutti i Comuni: Smart City Integrata	MIZAR	Sperimentazione	Smart City Integrata	Acquisizione dati relativi alla mobilità, servizi integrati, meteo, etc. Acquisizione dati dagli utenti, produzione di suggerimenti e sperimentazione di suggerimenti agli utenti, valutazione del numero di suggerimenti e navigazioni prodotte ed accettate dagli utenti. Installazione Totem, distribuzione APP, promozione per la sperimentazione, installazione Kit veicolari, etc.
2.7.2	Supporto alla sperimentazione e con tutti i Comuni: Gestione Accessi e ZTL	ECM	Sperimentazione	Gestione Accessi e ZTL	Installazione delle soluzioni per il controllo accessi e direzione, sperimentazione. Acquisizione dati per controllo accessi e ZTL, integrazione con scenari di partecipazione del cittadino, etc. Comunicazione sistemi di controllo accesso con centrale, controllo del direzionatore da remoto.

2.7.3	Supporto alla sperimentazione e con tutti i Comuni: Interoperabilità	MIZAR	Sperimentazione	Interoperabilità	Interscambio dati fra comuni o meglio fra operatori e Sii-mobility e fra questo verso gli operatori che possono essere interessati ad avere dati di sintesi, di navigazione o di controllo. Integrazione del sistema Sii-Mobility con gestore TPL, gestore traffico, operatori, open data, etc.
2.7.4	Supporto alla sperimentazione e con tutti i Comuni: metodi pagamento	NEGENTIS	Sperimentazione	metodi pagamento	Sperimentazione sui metodi di pagamento, bigliettazione integrata, anche in connessione alle soluzioni multimodali, percorsi integrati fra TPL diverse, treno, etc. Acquisto di biglietti, calcolo delle tratte, distribuzione dei ricavi, calcolo di bigliettazione dinamica, gestione dei bonus.
2.7.5	Supporto alla sperimentazione e con tutti i Comuni: demo integrazione	MIZAR	Sperimentazione	demo integrazioni	Sperimentazione integrata degli scenari che coinvolgono comuni limitrofi, per esempio per i sistemi di navigazione, per i suggerimenti sui punti di interesse, per la bigliettazione integrata, per i kit veicolari, etc. Installazione Totem, distribuzione APP, promozione per la sperimentazione, installazione Kit veicolari, etc.
				demo integrazioni	Sperimentazione di guida connessa in Firenze
2.7.6	Supporto alla sperimentazione e con tutti i Comuni: demo algoritmi	UNIFI	Sperimentazione	demo algoritmi	Sperimentazione riguardo agli algoritmi di data ingestion, riconciliazioni, navigazione, navigazione multimodale, suggerimenti per il risparmio energetico della città, etc.

### 6.3.1 Specifica delle sperimentazioni come scenari e sotto scenari di sperimentazione (MIZAR, UNIFI)

Di seguito è riportata la lista degli Scenari di sperimentazione (questa stessa tabella viene riportata anche in Section 6) che include per ogni scenario:

- Scenario e suo nome;
- Coordinatore dello scenario;
- Section in cui viene descritto nel documento dei requisiti De1.1;
- Lista dei sottoscenari che include per ogni sottoscenario:
  - Nome come descrizione breve;
  - Identificativo univo del sottoscenario come riferito anche nella tabella Moduli nella relativa colonna;
  - Partner di riferimento per il sottoscenario;
  - Partner primari coinvolti. Tutti i partner coinvolti ne sottoscenario sono quelli i cui moduli sono coinvolti come da colonna corrispondente nella Tabella Moduli
  - Tool principali;
  - Area o aree di sperimentazione;
  - Key Indicator: indicatori primari per la valutazione quantitativa dello sviluppo e della sperimentazione dello scenario;

scenario	coordinatore	requisiti	sottoscenario	ID	partner coordinatore	partners	major tools	area	key indicators
Guida Connessa	SWARCO	5.2	veicolo e semaforo	GC01	SWARCO	UNIFI, ECM	Gestore, Sii-Mobility backoffice, APP, KIT veicolari vari, algoritmi e routing, sensori	Firenze	#utenti, #eventi, #app, #kit
			eventi traffico	GC02	SWARCO	UNIFI, ECM	Gestore, Sii-Mobility backoffice, APP, KIT veicolari vari, algoritmi e routing	Firenze	#utenti, #eventi, #app, #kit
			bike	GC03	UNIFI	ECM	Gestore, Sii-Mobility backoffice, APP, KIT bike vari, algoritmi e routing, sensori	Firenze	#utenti, #eventi, #app, #kit
			pedone	GC04	UNIFI	ECM	Gestore, Sii-Mobility backoffice, APP, algoritmi e routing	Firenze	#utenti, #eventi, #app, #kit
zione e sensibiliz	GEOIN	5.3	SP Web	PS01	GEOIN	TIME, UNIFI, SOFTEC	WEB, APP, Server Sii-Mobility, dashboard, crowd sourcing	Firenze, Prato, Pistoia, Siena,	#utenti, #usabilità, #accessi distinti nel

							Pisa	tempo,	
			SP Totem	PS02	TIME	GEOIN UNIFI, SOFTEC	Totem, APP, Server Sii-Mobility, dashboard	Firenze, Prato, Pistoia, Siena, Pisa	#utenti, #totem, #usabilità, #accessi su totem distinti nel tempo,
			SP App	PS03	UNIF I	....GEOIN,	APP, Server Sii-Mobility, dashboard	Firenze, Prato, Pistoia, Siena, Pisa	#utenti, #app, #usabilità, #accessi distinti nel tempo,
Politiche incentivazione		5.4	Politiche Integ	PI01	UNIF I	ATAF, CTTNord, TIEMME, BUSITALIA, NEGENTIS, SWARCO	Server SII, APP, rete TPL, dashboard, sistema di engagement e recommendation	Firenze, (Prato), (Pistoia), Arezzo, Siena, Pisa-Livorno-Lucca	#utenti, #app, #bonus
Interoperabilità ed integrazione	UNIFI	5.5	II con gestori	II01	SWARCO	ATAF, CTTNord, TIEMME, BUSITALIA, UNIFI	Server SII, rete TPL, rete gestori, applicativi di gestione, dashboard, smart city API	Firenze, (Prato), (Pistoia), Arezzo, Siena, Pisa-Livorno-Lucca	#gestori, #flussi dati, #accessi, #pa coinvolte, #mezzi, #app
			II smart city	II02	UNIF I	ATAF, CTTNord, TIEMME, BUSITALIA, SWARCO	Server SII, APP, rete open data, applicativi di gestione, dashboard, smart city API, data analytics	Firenze, (Prato), (Pistoia), Arezzo, Siena, Pisa-Livorno-Lucca, regione toscana	#open data, #accessi chiamate, #view su dashboard
			II IOT	II03	SWARCO	UNIFI, ECM	Server SII, applicativi di gestione, dashboard, smart city API, social media, data analytics, sensoristica autonoma, crowd sourcing	Firenze	#dati e sensori, flussi dati, livello di integrazione

bigliettazione integrata	NEGENTIS	5.6	BI Operatori Mob	BI01	NEGENTI S	ATAF, CTTNord, TIEMME, BUSITALIA, LIBEROLOGICO, QUESTIT	Server Sii, parking	Firenze	#system integrati, #biglietti, #utenti
			BI Parcheggi	BI02	LIBEROL OGIC O	QUESTIT	Server Sii, parking	Firenze	#system integrati, #biglietti, #utenti
Gestione dinamica confini	ECM	5.7	gestore ZTL	GD01	ECM	Pubbliche amministrazioni: Pistoia. UNIFI, ELFI, ARGOS, CALAMAI, PROJECT, ...	Sensori, attuatori SII, direzionatore, info pannelli, etc.	Pistoia	#dimostrazioni, tempi di reazione, affidabilità, .. #accessi e transazioni
			Direzionatore	GD02	ECM	Pubbliche amministrazioni: Pistoia. UNIFI, ELFI, ARGOS, CALAMAI, PROJECT, ...	Sensori, attuatori SII, direzionatore, info pannelli, etc.	Pistoia	#dimostrazioni, tempi di reazione, affidabilità, .. #accessi e transazioni
			rastrelliera bike	GD03	ECM	Pubbliche amministrazioni: Pistoia. UNIFI, ELFI, ARGOS, CALAMAI, PROJECT, ...	SII, sensori, app rastrelliera, APP bike,	Pistoia	#numero di bike, #kit, numero transazioni ,
Monitoraggio domanda e offerta	ATAF	5.8	DO TPL	DO01	ATAF	CTTNord, TIEMME, BUSITALIA, SWARCO, UNIFI, Pubbliche amministrazioni	server Sii, gestori mobilità, dashboard, data analytics	Firenze, (Prato), (Pistoia) , Arezzo, Siena, Pisa-Livorno-Lucca	#flussi persone, #matrici OD, flussi mezzi, pannelli di monitoraggio
			DO persone	DO02	UNIFI	CTTNord, TIEMME, BUSITALIA, SWARCO, UNIFI, Pubbliche amministrazioni	server Sii, gestori mobilità, dashboard, data analytics	Firenze, (Prato), (Pistoia) , Arezzo, Siena, Pisa-Livorno-Lucca	#flussi persone, #matrici OD, flussi mezzi, pannelli di monitoraggio



### 6.3.2 Specifica della procedura per lo sviluppo delle sperimentazioni (MIZAR, UNIFI)

Gli scenari Sii-Mobility presentati nel deliverable DE1.1 verranno implementati sul campo e valutati sia da un punto di vista qualitativo che quantitativo, seguendo la metodologia di cui sotto, attraverso un processo diviso in 3 fasi:

- Una prima fase di progettazione del pilota, che prevede l'identificazione dell'area di sperimentazione e degli attori coinvolti definendo l'architettura fisica dello stesso
- Una seconda fase dedicata alla definizione della procedura di sperimentazione, che per il sistema definito al punto uno, sulla base dei requisiti di progetto identifica i criteri di valutazione qualitativa e quantitativa
- La terza e l'ultima fase prevede l'installazione del sistema sul campo ed il monitoraggio della fase operativa con rispettiva raccolta dati a supporto della valutazione qualitativa e quantitativa come derivato dal punto precedente

<p><b>Fase 1:</b> <b>Piano del pilota</b></p>	<ol style="list-style-type: none"> <li>1. Identificazione del sito della demo: descrizione fisica dell'area (geografia, dimensioni), caratteristiche, infrastruttura esistente;</li> <li>2. Identificazione di componenti SiiMobility principali e secondari da coinvolgere nel pilota;</li> <li>3. Identificazione delle persone/ partner da coinvolgere e definizione di uno User Group;</li> <li>4. Descrizione dello/degli scopo/i della sperimentazione;</li> <li>5. Definizione dell'architettura fisica del progetto esecutivo, condivisione tra i partner coinvolti (es. PA);</li> <li>6. Identificazione e valutazione dei rischi e sviluppo piano di mitigazione;</li> <li>7. Definizione di un Piano del pilota</li> </ol> <p><b>Output: Report interno contenente la pianificazione del pilota</b></p>
<p><b>Fase 2:</b> <b>Piano di sperimentazione</b></p>	<ol style="list-style-type: none"> <li>1. Identificazione dei requisiti per ogni componente coinvolto e degli indicatori di valutazione relativi (documentazione Attività 1.1.);</li> <li>2. Identificazione del periodo di installazione, prevalidazione e sperimentazione; Vedasi pianificazione generale e le durate minime di 10/12 mesi di sperimentazione.</li> <li>3. Identificazione dei parametri di valutazione quantitativa e qualitativa;</li> <li>4. Definizione del flusso/procedura di sperimentazione e di eventuali scenari alternativi;</li> <li>5. Identificazione dei dati in ingresso e output;</li> <li>6. Definizione di un Piano di sperimentazione e di validazione;</li> </ol> <p><b>Output: Report interno contenente il piano di test</b></p>
<p><b>Fase 3:</b> <b>Implementazione e validazione</b></p>	<ol style="list-style-type: none"> <li>1. Installazione dei componenti e verifica tecnica in sede di test;</li> <li>2. Verifica tecnica delle interfacce di connessione verso il sistema Sii-Mobility;</li> <li>3. Verifica funzionale del sistema;</li> <li>4. Apertura del sito di sperimentazione allo User Group; pianificazione di eventi dimostrativi;</li> <li>5. Produzione del rapporto di validazione quantitativa (derivata dagli indicatori dell'analisi dei requisiti) e qualitativa (derivata dall'analisi soggettiva dello User Group, es. questionari)</li> </ol> <p><b>Output: Report interno contenente i risultati della validazione</b></p>

## 6.4 Specifica: Connessione verso il SII, validazione di integrazione (MIZAR)

Sottoattività e gruppi di lavoro per questi requisiti		ECM	Mizar	UNI FI	IN20	TIME	Negentis	Fffknow	liberologico	ataf	busitalia	cttnord	tiemme	argos	elfi	calamai	Midra	Proiect	GFOIN	QUESTIT	SOETEC	FWINGS
2.4.1	Validazione rispetto ai casi di test e metriche definite: Controllo e monitoraggio	X	X		X		X															
2.4.2	Validazione rispetto ai casi di test e metriche definite: centrale operativa		X																			
2.4.3	Validazione rispetto ai casi di test e metriche definite: simulazione	X					X															
2.4.4	Validazione rispetto ai casi di test e metriche definite: bigliettazione						X															
2.4.5	Validazione rispetto ai casi di test e metriche definite: Merci e logistica	X				X																
2.4.6	Validazione rispetto ai casi di test e metriche definite: pianificazione	X	X				X															
2.4.7	Validazione rispetto ai casi di test e metriche definite: big data			X																		

Sottoattività e gruppi di lavoro per questi requisiti		Mizar	UNIFI	ataf	atam	busit	ctt	tiemme
6.2.1	V&V dati in ingresso a Sii-Mobility	X	X	X	X	X	X	X
6.2.2	V&V dati in uscita da Sii-Mobility	X	X	X	X	X	X	X
6.2.3	V&V interoperabilità delle API	X	X	X	X	X	X	X

### 6.4.1 Specifica: Prototipi applicativi verticali nei vari comuni (ATAF)

Sottoattività e gruppi di lavoro per questi requisiti		ECM	Mizar	UNI FI	Qu estl T	IN20	Soft ec	ataf	atam	bus it	ctt	tie mm e	arg os	elfi	cala mai	pro ject
6.3.1	Firenze		X	X	X	X	X	X								X
6.3.2	Firenze provincia		X	X	X	X	X	X		X						X
6.3.3	Prato, Pistoia, Pisa	X	X	X	X	X	X				X		X	X	X	
6.3.4	Arezzo, Siena (Piombino)		X	X	X	X	X		X			X				

Il prototipo deve garantire almeno la possibilità di:

- interfacciarsi col sistema AVM già presente a bordo o, in alternativa, essere comunque monitorabile/interrogabile dall'azienda di trasporto;
- rilevare il numero di utenti in ingresso/uscita da un veicolo pubblico, così come il numero di utenti presenti a bordo di un veicolo pubblico;
- rilevare condizioni ambientali (temperatura, umidità) sia internamente che esternamente ai bus;
- rilevare i dati relativi all'utilizzo dei titoli di viaggio provenienti dal SBE/tornelli degli utenti a bordo di un veicolo pubblico;

- rilevare il flusso di veicoli (pubblici e privati) e la relativa velocità di transito.

## 6.5 Specifica: usabilità e percezione delle soluzioni da parte del cittadino (UNIFI CSL)

Sottoattività e gruppi di lavoro per questi requisiti		UNIFI. CSL	Quest IT	Softec	TIME	ataf	atam	busit	ctt	tiem me	EffKn ow
6.4.1	X app web e mobile	X		X		X	X	X	X	X	X
6.4.2	X totem	X			X	X	X	X	X	X	
6.4.3	X sistema di partecipazione	X		X		X	X	X	X	X	X
6.4.4	X Accessi servizi esterni		X								

L'usabilità del sistema Sii-Mobility e la percezione delle soluzioni da parte di cittadini e utenti saranno analizzate attraverso un mix di interventi basati sull'applicazione di strumenti di ricerca sociale e su test di usabilità appositamente perfezionati sulla base della metodologia generativa e delle specifiche esigenze del progetto. Si farà quindi uso di questionari strutturati, focus group generativi, interviste semistrutturate e test di usabilità in laboratorio e sul campo. Per realizzare queste ricerche sarà da valutare la necessità di fornire ricompense ai partecipanti (ad esempio un carnet di biglietti per il trasporto pubblico o un buono per il parcheggio). I risultati (anche parziali) di ciascun intervento porteranno elementi conoscitivi funzionali a ottimizzare tutti gli altri interventi, indirizzando la ricerca in direzione delle eventuali criticità che potranno emergere. Di seguito la specifica di ogni intervento.

### Percezione delle soluzioni da parte del cittadino

#### Invito alla recensione

Dopo 5 utilizzi dell'App un messaggio invita a recensire l'app sullo store (Google Play Store per la versione Android, App Store per la versione iOS, Store per la versione Windows Mobile). La recensione non fornisce nessun premio all'utente.

#### Breve questionario in app (con premio)

Si può inserire nell'App un messaggio che invita l'utente a compilare un breve questionario da mobile. La compilazione del questionario garantisce un premio all'utente, come se si trattasse dell'adozione di un comportamento virtuoso. Il questionario può essere compilato soltanto una volta.

Una bozza del breve questionario sarà condivisa con tutti i partner di progetto per raccogliere consigli e sollecitazioni. Sulla base di questo il CSL realizzerà la versione definitiva del questionario.

Il questionario sarà progettato dal CSL ma l'implementazione all'interno dell'App sarà gestita dal DISIT.

#### Questionario strutturato a campioni rappresentativi di utenti

Saranno realizzati dei questionari strutturati per ogni categoria di utenti del Sii-mobility. I questionari saranno autosomministrati e compilati tramite piattaforme digitali.

La realizzazione dei questionari si articola in una serie di fasi, che possono essere raggruppate nei tre momenti del prima, del durante e del dopo.

#### Prima della somministrazione

- Fase 1. Laboratorio di Comunicazione Generativa (LCG) per la definizione dei questionari  
Un momento di confronto strutturato che il CSL gestirà coinvolgendo i partner del progetto. Il Laboratorio, che si svolgerà presso la sede del CSL, servirà per presentare una prima bozza dei questionari e per far emergere dall'ascolto, dal dialogo e dalle relazioni dirette tra le persone coinvolte elementi di conoscenza utili a rafforzare l'intero processo. I risultati del LCG saranno sistematizzati dal CSL, formalizzati e successivamente incorporati nei questionari.

- Fase 2. Pre-somministrazione dei questionari  
I questionari, sviluppati a seguito del LCG, saranno pre-somministrati ad un gruppo ristretto di utenti dell'applicazione, per valutarne l'efficacia comunicativa, i punti di forza e di debolezza. La somministrazione avverrà attraverso la stessa piattaforma digitale che sarà utilizzata poi nella fase successiva. La presomministrazione riguarderà un gruppo di 5 utenti per ogni categoria individuata di utenti dell'applicazione. I risultati della pre-somministrazione saranno sistematizzati in un report che sarà condiviso con il gruppo di progetto e saranno utilizzati per re-indirizzare opportunamente i questionari.

#### Durante la somministrazione

- Fase 3. Somministrazione del questionario
- I questionari saranno autosomministrati e compilati tramite piattaforme digitali. L'obiettivo dei questionari sarà quello di accertare i punti di forza e i punti di debolezza per quanto riguarda l'usabilità e le funzionalità del sistema Sii-Mobility, nella percezione degli utenti di tale sistema.
- Per assicurare la rappresentatività del campione si procederà per due sotto-fasi successive:
  - 3.1: autoselezione del campione;
  - 3.2: selezione mirata del campione, finalizzata ad assicurare la rappresentatività dello stesso.
- Durante la sotto-fase 3.1 i questionari saranno distribuiti ai potenziali utenti tramite tutti i canali di comunicazione del CSL, del progetto Sii-Mobility e dei partner di progetto. In questo modo il campione di partecipanti sarà autoselezionato.
- In seguito, i ricercatori del CSL provvederanno ad una prima analisi finalizzata ad individuare le caratteristiche del campione e il suo livello di rappresentatività dell'universo di riferimento.
- Sulla base di questi risultati si provvederà ad una successiva sotto-fase (3.2) di somministrazione mirata dei questionari ad un campione in grado di compensare gli squilibri rilevati nella prima fase, tenendo presente le categorie di utenti ai quali l'app sarà rivolta.

#### Dopo la somministrazione

- Fase 4. Analisi dei dati  
Al termine di questa seconda fase i ricercatori del CSL analizzeranno i risultati emersi. I dati dei questionari relativi alle domande chiuse saranno processati su piattaforma IBM SPSS 20 Statistics Data Editor per l'analisi statistica, e incrociati e sistematizzati con i risultati delle domande aperte riportate a parte. Dopo l'analisi da parte del CSL, i risultati saranno riportati in un report che sarà condiviso con i partner di progetto.

#### Output previsto

Al termine della somministrazione e dell'analisi dei questionari sarà realizzato un report in cui saranno presentati la metodologia utilizzata e i risultati dell'indagine, con la descrizione delle criticità e dei punti di forza.

### **Focus group generativi a campioni rappresentativi di utenti**

I focus group generativi (FGG) coinvolgeranno tutte le diverse categorie di utenti del Sii-mobility. La realizzazione dei focus group generativi si articola in una serie di fasi, che possono essere raggruppate nei tre momenti del prima, del durante e del dopo.

#### Prima dell'incontro

- Verrà individuata una scaletta di argomenti relativa a ciascuna categoria coinvolta; ogni FGG, infatti, verterà oltre che su una parte introduttiva comune a tutte le categorie, sulle peculiari aspettative/bisogni di ciascuna categoria di utenti del Sii-mobility;
- Verrà identificato un luogo adatto alla realizzazione del FGG, che in linea di massima corrisponderà alla sede del CSL a Firenze;
- Per ogni FGG sarà individuato un gruppo di partecipanti appartenenti alla categoria individuata (6-8 persone per ciascun focus). In questa fase potranno essere coinvolti altri partner di progetto. Nella selezione dei partecipanti si terranno in considerazione criteri di omogeneità (categoria) e di varietà (genere, età, ecc.);
- Si individueranno il giorno e l'ora dell'incontro in presenza sulla base delle disponibilità dei partecipanti;
  - prima dell'incontro in presenza, ai partecipanti verrà inviato tramite email del materiale preparatorio (tra cui la scaletta degli argomenti), in modo che possano arrivare all'incontro con un'idea di massima delle modalità e degli argomenti che saranno oggetto dell'incontro;
  - prima dell'incontro in presenza sarà stabilito un contatto telefonico con i partecipanti, in modo che possano esporre e trovare risposta a eventuali dubbi e richieste di chiarimento.

#### Durante l'incontro

- durante l'incontro in presenza, il FGG sarà diviso in tre parti:
  - breve presentazione del progetto, dei suoi obiettivi e della logica del sistema;
  - prima fase di discussione generale sul Sii-mobility;
  - seconda fase di discussione specifica sulle aspettative e i bisogni della categoria oggetto del FGG;
- l'incontro sarà moderato dai ricercatori del CSL. Uno degli obiettivi della moderazione consisterà nel facilitare l'emersione di esperienze e vissuti personali che possano essere paradigmatici e rappresentativi dell'intera categoria di utenti;
- durante l'incontro un ricercatore del CSL scriverà un instant report, proiettato nella sala e visibile a tutti i partecipanti. Quando un argomento della scaletta sarà stato affrontato in maniera esaustiva, il ricercatore che si occupa dell'instant report rileggerà ai partecipanti il punto in questione, così da verificare che essi si riconoscano in quanto emerso dalla discussione e non abbiano altro da aggiungere; a questo punto si passerà all'argomento successivo;
- il FGG sarà ripreso con una telecamera a 360°, così da permettere l'analisi degli atteggiamenti dei partecipanti oltre che dei risultati espliciti dell'incontro. Il materiale video sarà utilizzato esclusivamente dai ricercatori del CSL in fase di analisi e poi cancellato.

#### Dopo l'incontro

- alcuni giorni dopo l'incontro, i partecipanti saranno ricontattati uno ad uno per permettergli di condividere considerazioni e impressioni a freddo, che non erano emerse nella fase di incontro;

- dopo gli incontri in presenza sarà realizzato un report per ciascun FGG e un report consuntivo. Nel report consuntivo sarà possibile evidenziare, oltre alle aspettative e ai bisogni di ciascuna categoria, le differenti prospettive con cui esse si relazionano al Sii-mobility. Una versione rimaneggiata dei report sarà inviata ai partecipanti, per rafforzare il senso di coinvolgimento e gratificarli per la partecipazione.

#### Output previsto

Dopo la realizzazione dei FGG sarà realizzato un report in cui saranno presentati la metodologia utilizzata e i risultati dell'indagine, con la descrizione delle criticità e dei punti di forza.

#### **Interviste semistrutturate a campioni rappresentativi di utenti**

In base ai risultati dei FGG e dei questionari saranno realizzate alcune interviste semistrutturate a campioni rappresentativi di utenti. Queste interviste sono funzionali ad approfondire le criticità emerse nel corso dei FGG.

Anche in questo caso, la realizzazione delle interviste si articola in una successione di fasi che possono essere suddivise in un prima, un durante e un dopo.

#### Prima dell'intervista

- saranno individuate alcune tematiche e criticità da approfondire tramite le interviste;
- saranno individuati rappresentanti delle diverse categorie da intervistare, sulla base delle tematiche e delle criticità;
- per ogni intervista sarà realizzata una scaletta di argomenti da affrontare;
- i partecipanti saranno contattati e gli verrà messo a disposizione del materiale preparatorio, tra cui la scaletta degli argomenti;
- con ogni partecipante sarà concordato un giorno per la realizzazione dell'intervista e un luogo, che nella maggior parte dei casi dovrebbe essere la sede del CSL a Firenze.

#### Durante l'intervista

- le interviste saranno condotte da due ricercatori del CSL;
- le interviste saranno registrate con il consenso dei partecipanti. Il materiale video sarà utilizzato esclusivamente dai ricercatori del CSL Lab in fase di analisi e poi cancellato;

#### Dopo l'intervista

- alcuni giorni dopo l'intervista, i partecipanti saranno ricontattati uno ad uno per permettergli di condividere considerazioni e impressioni a freddo, che non erano emerse durante l'intervista;
- saranno realizzati un report per ogni intervista e un report consuntivo. Il report della singola intervista sarà condiviso con il partecipante, per verificare la sua reazione e i suoi riscontri.

#### Output previsto

Dopo la realizzazione delle lezioni intervista sarà realizzato un report in cui saranno presentati la metodologia utilizzata e i risultati dell'indagine, con la descrizione delle criticità e dei punti di forza.

#### **Test di usabilità**

I test di usabilità saranno effettuati sia in laboratorio che sul campo sugli strumenti comunicativi del progetto Sii-mobility:

1. la piattaforma di partecipazione e sensibilizzazione, che include anche la control room e dashboard;
2. l'App mobile;
3. i totem.

Attraverso la combinazione di strumenti di registrazione dell'esperienza dell'utente e interviste semi-strutturate, saranno monitorati diversi parametri, con l'obiettivo di valutare:

- l'interazione tra utente e sistema (e viceversa);
- l'accessibilità e la facilità di utilizzo;
- la leggibilità dei contenuti;
- la struttura degli strumenti di comunicazione;
- il senso di utilità e la soddisfazione da parte dei fruitori.

Per ottenere tali dati il tester tenterà di risolvere alcune task presenti all'interno di scenari tratti da situazioni reali, appositamente progettati per far emergere le criticità e i punti di forza degli strumenti di comunicazione del progetto Sii-Mobility. L'analisi video e l'uso di diverse tipologie di videocamere avrà un ruolo importante nella lettura dei comportamenti degli utenti. Ogni test di usabilità prevede due livelli di videoregistrazione realizzati con differenti tecnologie in base alla natura dello strumento di comunicazione analizzato. Le videoregistrazioni sono necessarie per integrare le osservazioni effettuate dai ricercatori e rispondere efficacemente agli obiettivi del test.

L'usabilità dei tre strumenti comunicativi sarà misurata attraverso tre diverse tipologie di test: tutti i risultati confluiranno in un unico report composto dai risultati generali e da quelli dei singoli report dei diversi test di usabilità. L'obiettivo è quello di fornire una visione unitaria della relazione tra gli utenti di Sii-Mobility e i suoi strumenti di comunicazione.

Come già visto per le attività precedenti, anche la realizzazione dei test si articola in una successione di fasi che possono essere suddivise in un prima, un durante e un dopo. Il tutto sarà videoregistrato a due livelli (comportamento e attività su schermo) e che il gruppo di lavoro sarà composto sempre da almeno due persone a contatto con il tester (conduttore e co-conduttore) e da un gruppo di elaborazione dati.

## **Test di usabilità in laboratorio**

### **Test di usabilità sulla piattaforma di partecipazione e sensibilizzazione**

Il test di usabilità in laboratorio ha l'obiettivo di analizzare il comportamento degli utenti e la reazione che hanno di fronte ad un primo utilizzo della piattaforma di partecipazione e sensibilizzazione di Sii-Mobility.

Nello specifico, saranno presentati ai tester alcuni possibili scenari con obiettivi reali da raggiungere.

La finalità del test è quella di comprendere la facilità di utilizzo, così come i punti di forza e le possibili criticità della piattaforma di partecipazione e sensibilizzazione.

Gli scenari proposti saranno sviluppati tenendo in considerazione tre dinamiche principali:

1. Criticità: compiti e scenari che potrebbero essere fonte di errore.
2. Frequenza: compiti e scenari tipici dell'utilizzo dell'applicazione.
3. Rappresentatività: compiti e scenari caratterizzanti del servizio.

In particolare, attraverso questo test si intende ottenere dati relativi a:

- Semplicità: la facilità di utilizzo immediato della piattaforma.
- Efficienza ed efficacia: capacità dell'utente di svolgere compiti con successo.
- Memoria: capacità dell'utente di ricordare le funzioni della piattaforma.
- Errori: le difficoltà e gli errori effettuati dall'utente nell'uso della piattaforma.
- Soddisfazione: la sensazione finale dell'utente di fronte alla piattaforma.

Il test di usabilità sarà effettuato all'interno di due diverse stanze del CSL.

- La prima stanza, dove saranno presenti il tester e il conduttore, sarà dotata di:
  - una postazione computer con connessione internet;
  - una videocamera fissa che riprenderà il comportamento del tester.
- Nella seconda stanza sarà presente un gruppo di analisi che osserverà l'esperienza del tester tramite lo streaming video proveniente dalla prima stanza.
- Il tester sarà guidato dal conduttore presente accanto a lui, il quale presenterà le diverse task da superare.
- L'esperienza dell'utente sarà registrata attraverso due tecniche differenti:
  - videoregistrazione del comportamento del tester con videocamera fissa;
  - videoregistrazione delle azioni effettuate dal tester sullo schermo tramite la registrazione dello schermo.

Di seguito il dettaglio delle tre fasi in cui è suddiviso il test di laboratorio.

#### Prima del test

- Definizione degli scenari e delle task che il tester dovrà affrontare attraverso la condivisione di questi con un gruppo selezionato all'interno del team di lavoro di Sii Mobility per ricevere informazioni, feedback ed apportare eventuali integrazioni. Tali task saranno in parte comuni e in parte specifiche per ogni categoria di utente;
- Individuazione e contatto dei possibili partecipanti al test di usabilità tenendo presente le diverse categorie di utenti del progetto;
- Profilazione dei possibili partecipanti attraverso un questionario online prima del test, così da avere un quadro completo delle esperienze di fruizione e navigazione con gli strumenti di comunicazione digitale e, in generale, del grado di consapevolezza degli utenti rispetto all'uso di tali strumenti. Sulla base del campione individuato verrà effettuata una selezione con l'obiettivo di comprendere nell'analisi diversi livelli di alfabetizzazione digitale.

#### Durante il test

- Incontro e briefing.
  - L'utente sarà accolto presso il CSL per effettuare il test di usabilità.
  - Il conduttore illustrerà il progetto e i passaggi che il tester si troverà ad affrontare.
- Esecuzione del test
  - Tramite la metodologia del *think aloud*, il tester racconterà i passaggi che sta eseguendo, le difficoltà riscontrate e le altre caratteristiche della propria esperienza di navigazione.
  - Il gruppo di ricerca, nella seconda stanza, osserverà l'esperienza del tester tramite la ripresa in streaming del suo schermo.

#### Dopo il test

Breve intervista al tester con l'obiettivo di:

- ripercorrere i momenti critici che il tester ha trovato nell'utilizzo dell'applicazione, la soddisfazione che ha raggiunto nel portare a termine certi obiettivi e l'efficienza del servizio;
- fornire al tester la possibilità di esprimere la propria opinione, la soddisfazione relativa all'esperienza di fruizione dell'App e le difficoltà che ha incontrato;
- dare la possibilità al tester di suggerire soluzioni ed esprimere pareri e giudizi relativi al progetto e all'applicazione mobile.

#### Output previsto



Al termine del test sarà realizzato un report in cui saranno presentati la metodologia utilizzata, i casi analizzati e i risultati dello studio, con la descrizione delle criticità e dei punti di forza. Tali risultati dovranno essere considerati come risorse qualitative a supporto della verifica della struttura e dell'interfaccia della piattaforma di partecipazione e sensibilizzazione.

## Test di usabilità sul campo

### Test di usabilità sull'app

Il test di usabilità sull'App *mobile* di Sii-mobility effettuato sul campo ha come obiettivo l'analisi del comportamento degli utenti e della loro reazione all'utilizzo dell'applicazione stessa.

L'analisi sarà effettuata su device forniti dal gruppo di ricerca con piattaforma Android, iOS e Windows Mobile in base al sistema operativo abitualmente utilizzato dall'utente.

Nello specifico, si andranno ad individuare una lista di scenari specifici per l'App mobile e task a difficoltà crescente per simulare esperienze reali di mobilità urbana: su questi scenari il test di usabilità misurerà la facilità di esecuzione di compiti specifici e la facilità di comprensione della struttura dell'App.

Come per il test precedente, si andranno a proporre scenari che tengono in considerazione tre dinamiche principali.

- Criticità: compiti e scenari che potrebbero essere fonte di errore.
- Frequenza: compiti e scenari tipici dell'utilizzo dell'applicazione.
- Rappresentatività: compiti e scenari caratterizzanti del servizio.

I test qualitativi hanno l'obiettivo di fornire indicazioni puntuali e significative su vari aspetti dell'App.

Il test sull'App *mobile* sarà strutturato per ottenere dati relativi a:

- Semplicità: la facilità di utilizzo immediato dell'applicazione.
- Efficienza ed efficacia: capacità dell'utente di svolgere compiti con successo.
- Memoria: capacità dell'utente di ricordare le funzioni dell'App.
- Errori: le difficoltà e gli errori effettuati dall'utente nell'uso dell'App.
- Soddisfazione: la sensazione finale dell'utente di fronte all'App.

Alcuni test saranno effettuati su device in cui l'app è preinstallata, in altri la ricerca e l'installazione dell'app sul device rappresenteranno le prime task che i tester dovranno affrontare. Di seguito il dettaglio delle fasi:

### Prima del test

- Definizione degli scenari e delle task che il tester dovrà affrontare attraverso la condivisione di questi con un gruppo selezionato all'interno del team di lavoro di Sii Mobility per ricevere informazioni, feedback ed apportare eventuali integrazioni. Tali task saranno in parte comuni e in parte specifiche per ogni categoria di utente.
- Individuazione e contatto dei possibili partecipanti al test di usabilità tenendo presente le diverse categorie di utenti del progetto;
- Profilazione dei possibili partecipanti attraverso un questionario online prima del test, così da avere un quadro completo delle esperienze di fruizione e navigazione con gli strumenti di comunicazione digitale e, in generale, del grado di consapevolezza degli utenti rispetto all'uso di tali strumenti. Sulla base del campione individuato verrà effettuata una selezione con l'obiettivo di comprendere nell'analisi diversi livelli di alfabetizzazione digitale.

### Durante il test

- Incontro e briefing
  - Incontro tra il conduttore, il co-conduttore e il tester in zone selezionate della città sulla base delle task e delle categorie di utenti. Effettuare il test in tali location ha come obiettivo quello di simulare esperienze reali di mobilità urbana.
  - Fase di briefing in cui il conduttore illustrerà il progetto e i passaggi che saranno effettuati
- L'esperienza di fruizione dell'App.
  - L'utente attraverserà gli scenari selezionati assieme al conduttore e al co-conduttore che monitoreranno i passaggi effettuati.
  - Il tester, tramite la metodologia del *think aloud*, descriverà i passaggi che sta compiendo, le riflessioni inerenti alle task richieste e alla loro difficoltà di superamento.
  - I tester dovranno superare le task proposte dal conduttore, il quale osserverà il comportamento dell'utente e annoterà i passaggi e i risultati a cui è giunto, le criticità e la soddisfazione generale di fronte all'utilizzo dell'App.
  - Ogni test sarà videoregistrato tramite applicazioni preinstallate sui device con cui è possibile registrare lo schermo, i gesti touch degli utenti e l'audio.
  - Il co-conduttore avrà uno strumento di videoripresa indossabile (occhiali dotati di videocamera o action cam) per registrare il test di usabilità. In questo modo sarà possibile analizzare tutti i passaggi effettuati dall'utente, le incertezze e gli errori.

#### Dopo il test

Breve intervista al tester con l'obiettivo di:

- ripercorrere i momenti critici che il tester ha trovato nell'utilizzo dell'applicazione, la soddisfazione che ha raggiunto nel portare a termine certi obiettivi e l'efficienza del servizio;
- fornire al tester la possibilità di esprimere la propria opinione, la soddisfazione relativa all'esperienza di fruizione dell'App e le difficoltà che ha incontrato;
- dare la possibilità al tester di suggerire soluzioni ed esprimere pareri e giudizi relativi al progetto e all'applicazione mobile.

#### Output

L'output del test sarà un report contenente la metodologia utilizzata, i casi analizzati e i risultati dello studio, con la descrizione delle criticità e dei punti di forza. Tali risultati dovranno essere considerati come risorse qualitative a supporto della verifica della struttura e interfaccia dell'App.

#### Test di usabilità sui totem

Il test di usabilità sui totem di Sii-mobility ha come obiettivo:

- l'analisi del comportamento degli utenti;
- l'analisi dell'interazione che gli utenti hanno con l'interfaccia grafica.

Nello specifico, saranno individuate task a difficoltà crescente su cui si incentrerà il test di usabilità e zone della città in cui effettuare i test, così da comprendere la facilità di esecuzione di compiti specifici e la facilità di utilizzo dei totem in reali condizioni di mobilità urbana. I test hanno l'obiettivo di fornire indicazioni su diversi aspetti relativi alla relazione che si va ad instaurare tra il soggetto e i totem.

Per questo motivo il test sarà strutturato per ottenere dati relativi a:

- Semplicità: la facilità di utilizzo immediato del totem.
- Efficienza ed efficacia: capacità dell'utente di svolgere compiti con successo.
- Memoria: capacità dell'utente di ricordare le funzioni dei totem.
- Errori: le difficoltà e gli errori effettuati dall'utente nell'uso dei totem.
- Soddisfazione: la sensazione finale dell'utente di fronte ai totem.

Di seguito il dettaglio delle tre fasi alla base del test:

#### Prima del test

- Definizione degli scenari e delle task che il tester dovrà affrontare attraverso la condivisione di questi con un gruppo selezionato all'interno del team di lavoro di Sii Mobility per ricevere informazioni, feedback ed apportare eventuali integrazioni. Tali task saranno in parte comuni e in parte specifiche per ogni categoria di utente.
- Individuazione e contatto dei possibili partecipanti al test di usabilità tenendo presente le diverse categorie di utenti del progetto;
- Profilazione dei possibili partecipanti attraverso un questionario online prima del test, così da avere un quadro completo delle esperienze di fruizione e navigazione con gli strumenti di comunicazione digitale e, in generale, del grado di consapevolezza degli utenti rispetto all'uso di tali strumenti. Sulla base del campione individuato verrà effettuata una selezione con l'obiettivo di comprendere nell'analisi diversi livelli di alfabetizzazione digitale.

#### Durante il test

- Incontro e briefing.
  - Incontro tra il conduttore, il co-conduttore e il tester in zone selezionate della città sulla base delle task e delle categorie di utenti.
  - Fase di briefing in cui il conduttore illustrerà il progetto e i passaggi che saranno effettuati
- L'esperienza di fruizione dei totem.
  - I tester dovranno superare le task proposte dal conduttore, il quale osserverà il comportamento dell'utente e annoterà con l'aiuto del co-conduttore i passaggi e i risultati a cui l'utente è giunto, le criticità e la soddisfazione generale di fronte all'utilizzo del totem.
  - Il tester, tramite la metodologia del *think aloud*, descriverà i passaggi che sta compiendo, le riflessioni inerenti alle task richieste e alla loro difficoltà di superamento.
  - Ogni test sarà videoregistrato tramite:
    - Una videocamera fissa per registrare lo schermo del totem e i gesti dell'utente.
    - Una videocamera a 360° che registrerà l'intera scena con l'obiettivo di verificare l'interazione che il tester ha con il totem e con l'ambiente circostante.

#### Dopo il test

Breve intervista al tester con l'obiettivo di:

- Ripercorrere i momenti critici che il tester ha trovato nell'utilizzo dell'applicazione, la soddisfazione che ha raggiunto nel portare a termine certi obiettivi e l'efficienza del servizio.
- Fornire al tester la possibilità di esprimere la propria opinione, la soddisfazione relativa all'esperienza di fruizione del totem e le difficoltà che ha incontrato
- Dare la possibilità al tester di suggerire soluzioni ed esprimere pareri e giudizi relativi al progetto e all'applicazione mobile.

#### Output

I risultati del test di usabilità sull'interazione tra l'utente e i totem confluiranno in un report contenente la metodologia utilizzata, i casi analizzati e i risultati dello studio con la descrizione delle criticità e dei punti di forza. Tali risultati dovranno essere considerati come risorse qualitative a supporto della verifica della struttura e dell'interfaccia dei totem.

## 7 Acronimi

- API: Application Program Interface
- AVL: Automatic vehicle location
- AVM: Automatic Vehicle Monitoring
- BDaaS: Big Data as a Service
- CAP principle: Consistency Availability Partition Tolerance principle
- CBB: Content Based Billing
- CBB: Content Based Billing
- CEN: European Committee for Standardization
- DBMS: database management system
- FCD: Floating Cellular Data
- GPRS: General packet radio service
- GPS: Global positioning System
- GSM: Global System for Mobile
- ICT: Information and Communication Technologies
- ITS: Intelligent Transport Systems
- LCD: liquid-crystal display
- LOD: linked open data
- MC: Mobile Collector
- MMS: Multimedia Messaging Service
- NLP: Natural Language Processing
- NoSQL: no SQL database
- OD: open data
- OD: Open Data
- OGC: Open Geospatial Consortium
- OWL: Web Ontology Language
- PA: Pubblica Amministrazione
- PMI: Piccola e Media Impresa
- PMS: Private Mobile Systems
- PMV: Pannelli a Messaggio Variabile
- POS: part-of-speech
- RDF: Resource Description Framework
- RFID: Radio Frequency IDentification o Identificazione a radio frequenza
- RTTI: Real-time Travel & Traffic Information
- SDI: Spatial Data Infrastructures
- SII: sistema di interoperabilità integrato
- SIMONE: progetto Simone
- SMS: Short Message Service
- SN: social networking, oppure sensor network
- SOA: Service Oriented Architecture
- SOAP: Simple Object Access Protocol
- SSAMM: Agenzia per la Mobilità Metropolitana strumenti di supporto, TOSCANA
- TMC: Traffic Message Channel
- TPEG: Transport Protocol Experts Group
- TPL: gestore trasporto pubblico locale
- UML: Unified Modeling Language
- UMTS: Universal Mobile Telecommunications System
- UTC: Urban Traffic Control
- UUDI: Universal Description Discovery and Integration
- V2I: Vehicle-to-Infrastructure
- V2V: vehicle-to-vehicle

- VMS: Variable Message Sign
- VWSN: Vehicular Wireless Sensor Networks
- W3C: World Wide Web Consortium
- WSD: Word Sense Disambiguation
- WSDL: Web Services Description Language
- WSN: Wireless Sensor Networks
- XMI: XML Metadata Interchange standard di OMG
- XML: Extensible Markup Language
- ZTL: Zona a Traffico Limitato