



SII-Mobility

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

Trasporti e Mobilità Terrestre, SCN_00112

Deliverable ID: DE5.11b

**Titolo: Progettazione dell'integrazione con gestori
parcheggi ed altre tipologie**

Data corrente	03/05/2017
Versione (solo il responsabile può cambiare versione)	1.6
Stato (draft, final)	Finale
Livello di accesso (solo consorzio, pubblico)	Pubblico quando completo
WP	OR5
Natura (report, report e software, report e HW)	Report
Data di consegna attesa	31/12/2017
Data di consegna effettiva	03/05/2017
Referente primario, coordinatore del documento	Leonardo Rigutini (QuestIT) rigutini@quest-it.com , Alessandro Gronchi (QuestIT) gronchi@quest-it.com
Contributor	TIME, Midra, Liberologico, UNIFI DISIT
Coordinatore responsabile del progetto	Paolo Nesi, UNIFI, paolo.nesi@unifi.it

Sommario

Sommario	2
1. Introduzione ed obiettivi	5
1.1 Descrizione	6
1.2 Tipologie dei dati	7
1.2.1 Dati statici	7
1.2.2 Dati dinamici	7
1.2.3 Dati statistici	7
1.3 Analisi funzionale	7
2. Tasks	9
2.1 Attività 5.4.1: parcheggi vari [PETL20]	10
2.1.1 Analisi del tipo di servizio e del tipo di dati	10
2.1.2 Integrazione con i gestori	11
2.1.2.1 Gestori i cui dati sono già disponibili	12
2.1.2.2 Gestori contattati dal partner del progetto QuestIT: Siena Parcheggi e Sistema Grosseto	12
2.1.2.3 Gestori contattati dal partner del progetto Liberologico	12
2.1.3 Stato della attività di integrazione con i gestori di parcheggi	14
2.1.4 Attività di realizzazione ETL	15
2.1.4.1 Siena Parcheggi (Quest-IT)	15
2.1.4.1.1 Dati input	15
2.1.4.1.2 Dati statici: schema e descrizione ETL	16
2.1.4.1.2.1 Ingestion	16
2.1.4.1.2.2 Quality Improvement	17
2.1.4.1.2.3 Triplification	17
2.1.4.1.3 Dati real time: schema e descrizione ETL	18
2.1.4.1.3.1 Ingestion	18
2.1.4.1.3.2 Triplification	18
2.1.4.2 Sistema Grosseto (Quest-IT)	19
2.1.4.2.1 Dati input	19
2.1.4.2.2 Dati statici: schema e descrizione ETL	20
2.1.4.2.2.1 Ingestion	20
2.1.4.2.2.2 Quality Improvement	20
2.1.4.2.2.3 Triplification	21
2.1.4.2.3 Dati real time: schema e descrizione ETL	21

2.1.4.2.3.1 Ingestion	22
2.1.4.2.3.2 Triplification	22
2.1.4.3 PisaMo (Liberologico)	22
2.1.4.3.1 Dati input	22
2.1.4.3.2 Dati statici: schema e descrizione ETL	24
2.1.4.3.2.1 Ingestion	24
2.1.4.3.2.2 Quality Improvement	25
2.1.4.3.2.3 Triplification	25
2.1.4.3.3 Dati real time: schema e descrizione ETL	26
2.1.4.3.3.1 Ingestion	26
2.1.4.3.3.2 Triplification	26
2.1.4.4 Firenze parcheggi (Liberologico)	27
2.1.4.4.1 Dati input	27
2.1.4.4.2 Dati statici: schema e descrizione ETL	29
2.1.4.4.2.1 Ingestion	29
2.1.4.4.2.2 Quality Improvement	29
2.1.4.4.2.3 Triplification	30
2.1.4.4.3 Dati real time: schema e descrizione ETL	30
2.1.4.4.3.1 Ingestion	30
2.1.4.4.3.2 Triplification	31
2.2 Attività 5.4.2: car and bike sharing [PETL21]	31
2.2.1 Analisi del tipo di servizio e del tipo di dati	31
2.2.2 Integrazione con i gestori	32
2.2.2.1 Gestori i cui dati sono già disponibili	33
2.2.2.2 Gestori contattati dal partner del progetto QuestIT : SIPedala (Siena)	33
2.2.2.3 Gestori contattati dal partner del progetto Liberologico: Bike sharing PISA	33
2.2.3 Stato della attività di integrazione con i gestori di servizi di car&bike sharing	33
2.2.4 Attività di realizzazione ETL	34
2.2.4.1 SiPedala – Siena (Quest-IT)	34
2.2.4.1.1 Dati input	34
2.2.4.1.2 Dati statici: schema e descrizione ETL	36
2.2.4.1.2.1 Ingestion	36
2.2.4.1.2.2 Quality Improvement	36
2.2.4.1.2.3 Triplification	36
2.2.4.1.3 Dati real time: schema e descrizione ETL	37
2.2.4.1.3.1 Ingestion	37
2.2.4.1.3.2 Triplification	38

2.2.4.2 PisaMo – Pisa (QuestIT)	38
2.2.4.2.1 Dati input	38
2.2.4.2.2 Dati statici: schema e descrizione ETL	40
2.2.4.2.2.1 Ingestion	40
2.2.4.2.2.2 Quality Improvement	40
2.2.4.2.2.3 Triplification	40
2.2.4.2.3 Dati real time: schema e descrizione ETL	41
2.2.4.2.3.1 Ingestion	41
2.2.4.2.3.2 Triplification	42
2.3 Attività 5.4.3.a: taxi e car service [PETL22]	43
2.3.1 Analisi del tipo di servizio e del tipo di dati	43
2.3.2 Integrazione con i gestori di Taxi	44
2.3.2.1 Gestori contattati dal partner del progetto QuestIT: CO.TAS Taxi Siena	44
2.3.2.2 Gestori contattati dal partner del progetto MIDRA: SOCOTA Taxi Firenze	44
2.3.2.3 Gestori contattati dal partner del progetto MIDRA: COTAFI Taxi Firenze	45
2.3.3 Stato della attività di integrazione con i gestori di servizi di taxi e car service	45
2.3.4 Attività di realizzazione ETL	46
2.3.4.1 COTAS – Siena (Quest-IT)	46
2.3.4.2 SOCOTA – Firenze (Midra)	46
2.3.4.3 COTAFI – Firenze (Midra)	46
2.4 Attività 5.4.3b Logistica Merci [PETL19]	47
2.4.1 Analisi del tipo di servizio e del tipo di dati	47
2.4.2 Integrazione con i gestori di Logistica Merci	49
2.4.3 Stato della attività di integrazione con i gestori di servizi di taxi e car service	49
2.4.4 Attività di realizzazione ETL	50
2.4.4.1 ASSO Tosca (Time)	50
2.4.4.2 Corriere01 (Time)	50
2.4.4.2.1 Dati input	50
2.4.4.2.2 Dati statici: schema e descrizione ETL	50
2.4.4.2.2.1 Ingestion	51
2.4.4.2.2.2 Quality Improvement	53
2.4.4.2.2.3 Triplification	53
2.4.4.2.3 Dati real time: schema e descrizione ETL	54
2.3.4.2.3.1 Ingestion	54
2.3.4.2.3.2 Triplification	55
2.4.4.3 Corriere02 (Time)	56
2.4.4.3.1 Dati input	56

2.4.4.3.2 Dati statici: schema e descrizione ETL	56
2.4.4.3.2.1 Ingestion	56
2.4.4.3.2.2 Quality Improvement	58
2.4.4.3.2.3 Triplification	59
2.4.4.3.3 Dati real time: schema e descrizione ETL	60
2.4.4.3.3.1 Ingestion	60
2.4.4.3.3.2 Triplification	61
2.5 Attività 5.4.5: nettezza e servizi [PETL23]	62
2.5.1 Analisi del tipo di servizio e del tipo di dati	62
2.5.2 Integrazione con i gestori di Nettezza e servizi	63
2.5.2.1 Gestori contattati dal partner del progetto QuestIT: SEI Toscana	63
2.5.2.2 Gestori contattati dal partner del progetto MIDRA: Quadrifoglio	64
2.5.3 Stato della attività di integrazione con i gestori di servizi di nettezza e servizi	64
2.5.4 Attività di realizzazione ETL	64
2.5.4.1 SEI (QuestIT)	64
2.5.4.2 Quadrifoglio (Midra)	64
3. Conclusioni	65
Acronimi	68

1. Introduzione ed obiettivi

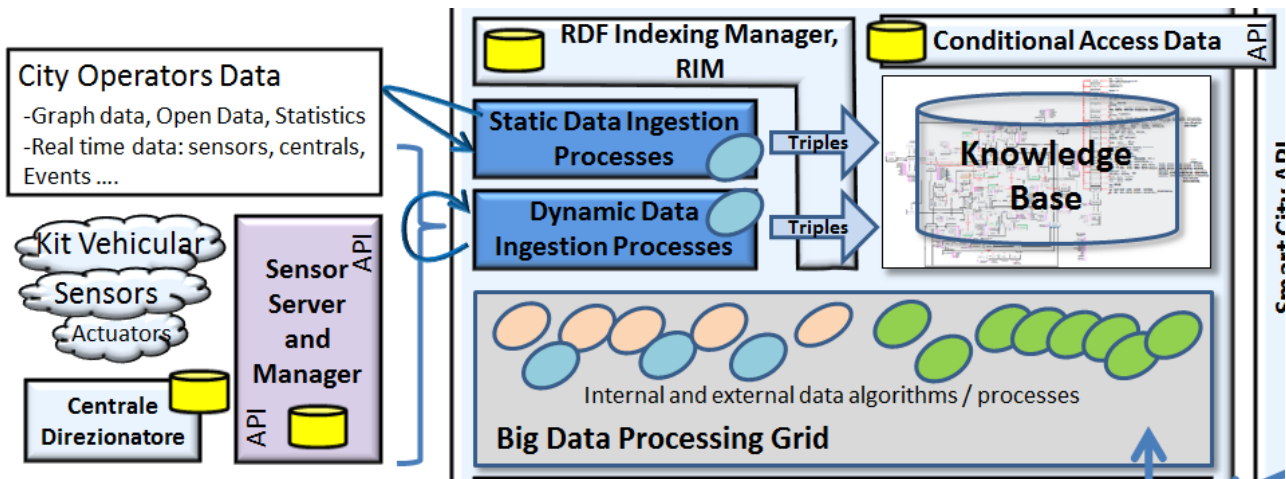
Il progetto SII-Mobility consiste nella realizzazione di un sistema centralizzato di data storage di dati relativi alla infomobilità. In particolare, il progetto prevede tre layer funzionali:

- un sistema centrale di data storage;
- un sistema periferico di data ingestion (acquisizione dati da gestori);
- un sistema periferico di utilizzo dei dati memorizzati nel sistema centrale (app, web, ecc...).

In questo schema, l'**Obiettivo Realizzativo OR5 - Integrazione ed Ottimizzazione dei sottosistemi** riguarda il layer di acquisizione dati da gestori (layer n. 2) ed in particolare l'**Attività 5.4 - Integrazione con parcheggi, ed altri gestori servizi** si concentra sull'acquisizione dei dati dalle seguenti tipologie specifiche di gestori:

- gestori di parcheggi;
- gestori di car and bike sharing;
- gestori taxi e car service;
- gestori nettezza urbana e servizi.

La seguente figura mostra i moduli dell'architettura generale di SII-Mobility interessati dalla attività 5.4 :



1.1 Descrizione

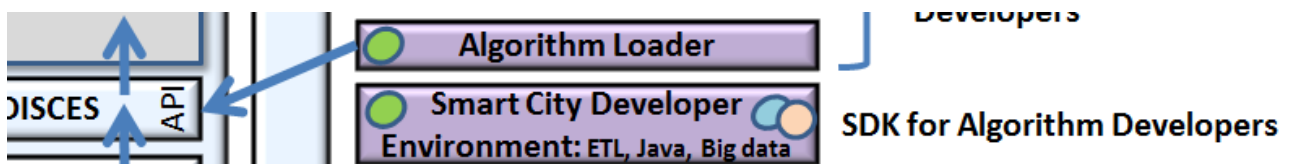
L'interconnessione tra i sistemi periferici (ovvero i gestori dei servizi) ed il sistema centrale è realizzata attraverso appositi software chiamati ETL (Extract, Transform, Load). Tali software costituiscono in pratica un layer di astrazione intermedio che permette di uniformare e normalizzare la enorme variabilità di memorizzazione dei dati dei gestori. In particolare, il recupero dei dati dei gestori è soggetti a due importanti problematiche:

1. tipologie di dati molto differenti in base al servizio gestito (es. dati relativi ai parcheggi piuttosto che al servizio di nettezza urbana)
2. in caso di dati relativi a stesse tipologie di servizio, le modalità di memorizzazione e/o le strutture di memorizzazione interna cambiano in base ai software ed ai sistemi DBMS utilizzati

Gli ETL consistono in appositi software progettati per accedere ai dati di ogni gestore ed attuare una serie di operazioni intermedie volte ad ottenere un output standard, compatibile con quello gestito dal sistema centrale:

1. selezione dei dati che sono di interesse per il sistema centrale
2. normalizzazione dei dati (per esempio eliminando i duplicati)
3. traduzione dei dati codificati
4. derivazione di nuovi dati calcolati
5. accoppiamenti (join) tra dati recuperati da differenti repository
6. raggruppamento dei dati

La figura seguente posiziona il livello di ETL all'interno della architettura generale del sistema:



Come appare evidente dalla precedente descrizione, il progetto e la realizzazione di un ETL è strettamente legato ai dati sottostanti e quindi richiede una approfondita fase di analisi dei dati e di progettazione dedicata. L'aggancio ai sistemi proprietari dei singoli gestori e la successiva acquisizione dei dati è fortemente vincolata da due fattori:

1. tipologia del servizio (parcheggi, car sharing, nettezza urbana, ecc...) che influenza la struttura

dei dati (sia di input che di output);

2. il sistema proprietario utilizzato dal gestore per memorizzare i dati internamente (DBMS, il software gestionale utilizzato, l'architettura di rete, ecc...)

La progettazione di tali moduli di interfaccia richiede quindi la disponibilità da parte dei gestori nel fornire il permesso all'uso degli stessi ma, soprattutto, il supporto per le procedure di accesso e di lettura dei dati stessi dai sistemi informativi interni proprietari.

1.2 Tipologie dei dati

Circa la natura dei dati dei gestori che si andranno a integrare al sistema SII-Mobility, si possono prevedere tre macro-categorie:

- dati statici;
- dati dinamici;
- dati statistici.

1.2.1 Dati statici

Per ogni tipologia di servizio si può individuare un sottoinsieme di dati che hanno una variabilità temporale molto ridotta ed in alcuni casi pressoché nulla. Ad esempio, la possibilità che un parcheggio in struttura modifichi la sua capienza è molto bassa in quanto tale modifica presuppone motivazioni eccezionali. Così come la mappa delle piste ciclabili di una città, che varia solamente nel caso il consiglio comunale deliberi in tale direzione. Data la loro natura quasi-statica, si può prevedere che l'acquisizione di tali dati possa avvenire attraverso un meccanismo meno invasivo e con requisiti meno stringenti che non necessita di accedere ai sistemi informativi del gestore. Tale procedura richiede che nel caso di variazioni, il gestore debba attivarsi per trasferire le variazioni anche sul sistema centrale, collegandosi al servizio e modificando le opportune informazioni.

1.2.2 Dati dinamici

Oltre a dati statici, i gestori dei servizi posseggono dati relativi allo stato in tempo reale. Tali dati hanno una elevata variabilità temporale e sono di solito riferiti come dati "dinamici". Il progetto dei relativi moduli ETL per la lettura di tali dati non può assolutamente esimersi da un stretto legame con i sistemi informativi proprietari utilizzati dai singoli gestori. In tali casi quindi è necessario stringere accordi con le società e gli enti gestori dei servizi affinché possa essere portata avanti un'analisi approfondita di tali sistemi informatici e progettare e sviluppare i relativi ETL. Data la elevata variabilità temporale dei dati, i moduli ETL per dati dinamici dovranno incorporare logiche di aggiornamento dei dati automatiche, ovvero meccanismi che permettano con una certa frequenza (dipendente dal tipo di dato) di aggiornare il dato sul sistema centrale allineandolo al dato del gestore.

1.2.3 Dati statistici

Per servizi per i quali vi è disponibilità di dati dinamici, è possibile prevedere anche un insieme di dati aggregati derivati dalle informazioni dinamiche in un certo periodo di tempo (medie, variazioni, ecc...). Tali dati possono non essere necessariamente recuperati dai gestori ma essere derivati dal modulo ETL memorizzando internamente (buffer interno) il range di interesse su cui tali medie e variazioni dovranno essere calcolate. E' evidente come tali statistiche abbiano senso solamente per i dati dinamici in quanto indicatori della loro variabilità e, se calcolati su dati statici, risulterebbero costanti (le medie) o nulli (le variazioni) e non fornirebbero alcun tipo di informazione addizionale. E' da sottolineare che tali informazioni sono recuperabili e derivabili successivamente dal sistema centrale SII-Mobility che permette analisi di tipo statistico e aggregazioni basate sui dati memorizzati.

1.3 Analisi funzionale

L'analisi relativa all'interfacciamento dei gestori verso SII-Mobility ha individuato tre principali metodologie di input dei dati:

1. recupero dati manuale - i dati del gestore vengono inseriti nel sistema attraverso una procedura manuale da parte di un operatore, come il caricamento di file in formato CSV/EXCEL o l'utilizzo di form web dedicati. Tale metodologia è chiaramente ammissibile esclusivamente per dati statici in quanto essi, data la loro bassa variabilità, richiedono rari interventi. E' evidente invece che per i dati dinamici, un tale metodo sia inusabile in quanto richiederebbe un continuo intervento da parte dell'operatore per aggiornare di continuo le variazioni. Si possono prevedere due possibili approcci:
 - progetto e realizzazione di un servizio web apposito per ogni tipologia di servizio in cui l'operatore del gestore del servizio possa inserire manualmente o tramite caricamento di file in formati standard (excel, csv, ecc...) le informazioni statiche richieste.
 - progetto e realizzazione di un servizio di estrazione dei dati da siti web (parsing o scraping) laddove tali dati siano pubblici (es. pubblicati sul sito web del gestore). In entrambi i casi, si ipotizza la realizzazione di semplici UI web per agevolare questa modalità.
2. recupero automatizzato - i dati del gestore vengono trasferiti sul sistema centrale tramite interazione diretta con i sistemi proprietari del singolo gestore. In tale approccio si possono prevedere due tipo di interconnessione: PUSH e PULL. Per ogni tipologia, sono prevedibili più soluzioni da seguire in base alle preferenze del gestore:
 - PUSH
 - a. disponibilità nel sistema SII-Mobility di apposite API invocate dai sistemi del gestore per l'invio dei dati richiesti. Tale soluzione prevede la realizzazione, nel sistema informativo del gestore, di apposite procedure e software che eseguono l'invio delle informazioni;
 - PULL
 - a. invocazione di API (disponibili o appositamente progettate e realizzate) dedicate all'esposizione dei dati residenti sull'infrastruttura proprietaria del gestore;
 - b. connessione diretta al DB o al FileSystem proprietario;
 - c. recupero di informazioni di pubblico dominio eventualmente disponibili come ad esempio scraping di pagine web di siti o portali, oppure recupero dai sistemi open data delle varie municipalità e degli enti coinvolti nell'attività.

Tale approccio prevede un forte accoppiamento con i sistemi proprietari dei gestori ed è utilizzabile sia per il recupero dei dati statici che per quelli dinamici. In particolare, in caso di recupero di dati dinamici, le due tipologie (PUSH e PULL) prevedono requisiti diversi. Nel caso di metodologie PUSH, l'aggiornamento del dato avviene in base ad un evento "trigger" che scatena lato gestore la chiamata alle API del sistema SII-Mobility per l'aggiornamento delle informazioni. Nel caso di metodologie di tipo PULL, è necessario definire la frequenza di "refresh" del dato che deve essere strettamente collegata alla frequenza con cui il dato varia (e quindi alla tipologia di informazione letta dal modulo ETL).

La metodologia manuale permette di approntare un modello iniziale per lo sviluppo e il testing del sistema e permette di considerare i casi di gestori il cui livello di automazione non è sufficiente, favorendo in ogni caso il caricamento di dati statici

2. Tasks

Il presente deliverable è conforme alle prescrizioni del manuale della qualità del progetto ([SIIM 3]) ed interessa lo studio e la progettazione del substrato tecnologico che farà da collante tra i sottosistemi periferici dei gestori (parcheggi, bike e car sharing, trasporto rifiuti urbani, trasporto merci, taxi, etc.) ed il sistema centrale SII. Il deliverable DE5.11 costituisce il documento di riferimento per la fase realizzativa dei moduli ETL di recupero dati dai gestori previsti:

- Task 5.4.1 - parcheggi;
- Task 5.4.2 - servizi di car & bike sharing;
- Task 5.4.3a - taxi e car-service;
- Task 5.4.3b - operatori della logistica (corrieri);
- Task 5.4.5 - nettezza e servizi vari (es. spazzamento strade).

Per ogni tipologia di servizio sono previsti contributi dai vari partner coinvolti come illustrato nella seguente tabella:

		TIME	liberologico	Midra	QUESTIT
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
5.4.3.a	integrazione con parcheggi, ed altri gestori servizi: taxi, car service			X	X
5.4.3.b	integrazione con car service e servizi di logistica merci dell'ultimo miglio	X			
5.4.4	integrazione con parcheggi, ed altri gestori servizi: nettezza e servizi			X	X
Tabella 1: Attività 5.4: integrazione con parcheggi, ed altri gestori servizi Tempistica: inizio attività mese: M7, M21 fine attività mese: M17, M28 Coordinatore: QUESTIT Partner: come da tabella					

Il documento prosegue riportando, con una sezione dedicata per task (ovvero tipologia di servizio), le attività di individuazione e coinvolgimento dei gestori da parte dei partner, e le attività tecniche di analisi, progetto ed integrazione dei dati con il sistema SII-Mobility (progetto e realizzazione degli ETL). In particolare, per ogni tipologia di servizio, viene dapprima riportata la lista dei gestori toscani individuati e contattati da ogni partner ai fini di ottenere dagli stessi una liberatoria all'utilizzo dei dati (statici e dinamici) e le relative informazioni tecniche sulla modalità con cui acquisire tali dati (API, scraping, DB ecc...). Tale attività si è scontrata fortemente con la resistenza di molti enti e aziende gestori a fornire i dati in loro possesso, sia per colpa di lunghi e burocratici di iter amministrativi interni, sia a causa di una certa reticenza dei gestori a rendere pubblici i propri dati (es. servizio Taxi). Inoltre, per ogni tipologia di servizio, viene riportata l'analisi e la progettazione dei tipi di dato individuando gli attributi necessari per il corretto funzionamento di un sistema di infomobilità quale è SII-Mobility. Infine, per i gestori che si sono dimostrati disponibili alla partecipazione al progetto e nei casi in cui i dati sono disponibili grazie al circuito degli OpenData,

vengono descritte le attività di progetto e realizzazione dei moduli di interfacciamento dei dati tra il gestore ed il sistema centrale, ovvero le attività di realizzazione degli ETL richiesti dalle specifiche SII-Mobility. Nella successiva sezione “Conclusioni” sono riportate le considerazioni finali relative agli aspetti trattati nelle precedenti sezioni. Infine, la sezione finale “Bibliography” riporta i riferimenti in letteratura individuati.

2.1 Attività 5.4.1: parcheggi vari [PETL20]

2.1.1 Analisi del tipo di servizio e del tipo di dati

L’analisi del servizio di gestione dei parcheggi e del relativo tipo di dato da gestire individua una prima importante classificazione:

1. parcheggi monitorati - Tali parcheggi sono attrezzati di un sistema automatico (o semiautomatico) di monitoraggio del numero di posti liberi/occupati che rende disponibile tale informazione in tempo reale. Normalmente questi parcheggi consistono in strutture ad accesso controllato con sbarra e/o telecamera, ma recentemente stanno emergendo dispositivi in grado di monitorare anche parcheggi lungo strada (sensori di presenza installati su ogni stallo oppure sistemi di telecamere in grado di identificare gli stalli liberi/occupati);
2. parcheggi non-monitorati - parcheggi per cui non è presente un sistema di acquisizione di informazioni relative allo stato degli stalli. Normalmente sono i parcheggi lungo strada per cui è difficile e costoso approntare sistemi di monitoraggio.

Tale suddivisione influisce fortemente sul tipo di dato fornito da queste due tipologie. Per entrambi i casi infatti sono prevedibili un consistente insieme di dati statici che caratterizzano il parcheggio, come ad esempio:

- nome e dati del gestore;
- geolocalizzazione;
- numero di stalli disponibili;
- tipo di parcheggio (lungo strada o in struttura)
- sistema di pagamento;
- eventuali giorni e/o orari di apertura e chiusura;
- ...

Per i parcheggi monitorati, sono disponibili anche le informazioni dinamiche sulla loro occupazione:

- numero di stalli liberi ed occupati;
- stato del parcheggio;
- ...

Si definisce uno schema di dato che possa incorporare entrambe le tipologie, gestendo così con un’unica struttura casi monitorati e non-monitorati. Di seguito si riporta lo schema di JSON strutturale da considerare come riferimento per la categoria dei gestori di parcheggi. Tale schema rimane come riferimento di base sia nel caso di nuovi gestori sia nel caso di gestori già affiliati e può essere soggetto a customizzazione. Al variare della modalità di inserimento del dato (manuale, automatizzato via API, open-data) è possibile immaginare una differenziazione fra dati statici e dati dinamici. La struttura JSON proposta include informazioni statiche e dinamiche e prevede, nel caso di parcheggi non-monitorati, la segnalazione della mancanza di informazioni dinamiche (monitored = false).

PARKING [QuestIT - Liberologico] Ver 0.1

```
{
  "company_id": 10,
  "company_name": "Company name",
  "company_description": "Company description",
  "city": "Parking city",
  "car_parkings": [{
    "id": 10,
    "name": "Parking lot/area name",
    "type_description": "Parking type description (off street parking, on street parking)",
    "tariff_system": "Tariff system",
    "opening_time": "Opening time",
    "address": "Parking address",
    "lat": 43.314272222222,
    "long": 11.3335388888889,
    "monitored": true/false
    "total_spaces": {"paid":500,"disabled":40, "loading/unloading":4},
    "used_spaces": {"paid":320,"disabled":32, "loading/unloading":2},
    "status": 80,
    "info": "Optional info"
    "parking_spot":{
      "map_position": 1,
      "recipients_description": "Parking's recipients description (paid, disabled,
loading/unloading)",
      "lat": 43.314272222222,
      "long": 11.3335388888889,
      "status": "parking spot status (empty, full, not available)"
    }
  ]
}
```

2.1.2 Integrazione con i gestori

All'interno del progetto SII-Mobility sono stati individuati diversi gestori di parcheggi del territorio Toscana. In altri casi i gestori sono stati individuati e contattati aziende che gestiscono parcheggi, è stato illustrato loro il progetto ed è stato chiesto loro una manifestazione di interesse e la liberatoria all'utilizzo dei dati in loro possesso nel progetto. In particolare, si ha la seguente lista:

- Gestori i cui dati sono già disponibili:
 - ZCS Zone parcheggio controllato FIRENZE
 - Parcheggi regione Toscana
 - Firenze Parcheggi
- Gestori contattati dal partner del progetto QuestIT:
 - Siena Parcheggi SPA (QuestIT)
 - Sistema Grosseto (QuestIT)
- Gestori contattati dal partner del progetto Liberologico:
 - PisaMo (Liberologico)
 - Parcheggi lungo strada PISA (Liberologico)
 - Parcheggi lungo strada FIRENZE (Liberologico)

2.1.2.1 Gestori i cui dati sono già disponibili

I dati relativi ai seguenti gestori e sistemi sono già stati integrati (o in fase di integrazione) nel sistema SII-Mobility e sono quindi già disponibili per l'uso verso terzi grazie alle API di ServiceMap di UNIFI:

- ZCS Zone parcheggio controllato FIRENZE
- Parcheggi regione Toscana
- Firenze Parcheggi

In particolare, per ZCS sono già disponibili le informazioni statiche solo sull'area di Firenze, per Firenze Parcheggi sono già presenti dati statici e dinamici, mentre per Parcheggi Regione Toscana l'integrazione dei dati statici e dinamici è al momento parziale e in fase di realizzazione.

2.1.2.2 Gestori contattati dal partner del progetto QuestIT: Siena Parcheggi e Sistema Grosseto

Sono state inviate email ai direttori tecnici e al direttore generale dei due enti contenenti la descrizione e gli scopi del progetto. Il direttore tecnico di Siena Parcheggi ha mostrato un forte interesse, mentre il direttore generale ha informato che è necessaria l'avallo da parte del consiglio di amministrazione. Per quel che riguarda Sistema Grosseto, il direttore generale ci ha dato risposta informale positiva relativamente all'interesse alla partecipazione al progetto ma ancora non è stata ricevuta risposta formale.

In particolare, per Siena Parcheggi si hanno disponibilità di dati statici e dinamici (incluso per questi ultimi i dati relativi ai posti riservati ai residenti in ZTL) per le seguenti strutture:

- Il Campo
- Il Duomo
- S. Caterina
- Stadio-Fortezza
- S. Francesco
- La stazione
- Eliporto
- FastPark

Sistema Grosseto fornisce dati statici e dinamici relativi ai quattro parcheggi monitorati:

- Galleria Oberdan
- Parcheggio Amiata
- Parcheggio Porta Corsica
- Parcheggio di Marina di Alberese

Per la integrazione di tali gestori, il partner QuestIT renderà disponibili API REST per la realizzazione del relativo ETL. L'output restituito da tali API REST sarà conforme al modello JSON in precedenza descritto.

2.1.2.3 Gestori contattati dal partner del progetto Liberologico

L'attività riguarderà la realizzazione di un sistema di API/WS che esponga dati statici (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 (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 di Pisa e Firenze sia al SII-Mobility che a sistemi di terze parti registrati.

Questo modulo includerà sia il sistema di acquisizione dati da installare presso i gestori sia il processo ETL per l'acquisizione dei dati statici (struttura, posti potenziali, posizioni, etc.) e dinamici (posti vuoti, posizione di tali posti, etc.).

Per quanto riguarda l'integrazione con i sottosistemi di mobilità e sosta urbani di Pisa e Firenze, il progetto SII-Mobility prevede di acquisire diversi tipi di informazioni di tipo statico e dinamico per le due città, che poi verranno messi a disposizione della piattaforma centrale per essere elaborati ed arricchiti in modo da essere poi essere utilizzati in:

- Sistema di supporto alle decisioni;
- Mobile app SII-Mobility;
- Applicazioni di terze parti.

Le due città già offrono sistemi di campo in grado di fornire dati utili al progetto, ma SII-Mobility provvederà ad installare anche sensori di sosta innovativi nella città di Firenze, in grado di rilevare lo stato di occupazione del singolo parcheggio, con cui alimentare i servizi di info-parking, sia per il gestore della sosta che per i cittadini e le imprese.

Nello specifico, per quanto riguarda l'integrazione con sistemi esistenti, verranno realizzate alcune componenti software per acquisire automaticamente le seguenti informazioni:

a) Città di Pisa

- Dati statici:

- geolocalizzazione degli stalli di sosta su strada sensorizzati;
- descrizione della tipologia di stalli (a pagamento, disabili, carico/scarico merci);
- piano tariffario della sosta su strada cittadino;
- geolocalizzazione delle rastrelliere del bike sharing;
- geolocalizzazione dei gate RFID a cintura della città, in grado di monitorare gli utenti dotati di PisaPass ed eCUDE che transitano sotto i gate.

- Dati dinamici:

- stato libero/occupato/fuori servizio degli stalli monitorati dai sensori di sosta;
- stato di attività dei gate RFID;
- flusso veicoli transitati da ogni gate RFID.

b) Città di Firenze

- Dati statici:

- geolocalizzazione degli stalli di sosta su strada sensorizzati;
- tipologia di stalli;
- piano tariffario della sosta su strada cittadina.

- Dati dinamici:

- stato libero/occupato/fuori servizio degli stalli monitorati dai sensori di sosta.

A tal fine verranno realizzate le seguenti componenti:

a) per la città di Pisa:

- potenziare, dove necessario, l'attuale sistema di API per la pubblicazione di dati e l'interoperabilità applicativa già in essere presso PisaMo, il gestore della mobilità e sosta del comune di Pisa;
- realizzare il modulo ETL che si interfacerà con il sistema di API del punto precedente per acquisire i dati statici e dinamici in SII-Mobility.

b) per la città di Firenze:

- per quanto riguarda il sistema di *sosta su strada* verrà realizzato:
 1. un apposito componente in grado di:
 - censire il sistema di monitoraggio automatico della sosta su strada (che verrà installato appositamente) basato sull'uso di sensori innovativi, allo stato dell'arte già in disposizione del partner Liberologico;
 - pubblicare le informazioni (statiche e dinamiche) relative al sistema di sosta su strada monitorato, sia su mappa web che per sistemi di terze parti, attraverso uno specifico sistema di API/WS;
 2. il modulo ETL che si interfacerà con il sistema di API del punto precedente per acquisire i dati statici e dinamici in SII-Mobility;
- per quanto riguarda il sottosistema di *parcheggi in struttura*, previo accordo con il gestore Firenze Parcheggi e un adeguato studio di fattibilità per ogni singola struttura (che presumibilmente sarà dotata di un proprio sistema tecnologico per la rilevazione della disponibilità di sosta e di API per la loro pubblicazione), si provvederà a:
 1. geolocalizzare i parcheggi in struttura;
 2. pubblicare le informazioni statiche e dinamiche di ciascun parcheggio in struttura con cui ci integreremo (Es: la capacità, il sistema tariffario, gli orari, la disponibilità di sosta in *near-real time*), sia su mappa web sia su sistemi di terze parti, attraverso uno specifico sistema di API/WS;
 3. realizzare il modulo ETL che si interfacerà con il sistema di API del punto precedente per importare i dati statici e dinamici in SII-Mobility.

2.1.3 Stato della attività di integrazione con i gestori di parcheggi

GESTORE	AREA GEO	TIPOLOGIA SERVIZIO	PARTNER DI RIFERIMENTO	LIVELLO AVALLO AMMINIST.	AVALLO UFFICIO TECNICO	DATI STATICI	DATI DINAMICI	METODOLOGIA INPUT DATI [manuale, API, open-data]
Siena parcheggi SPA	Siena	Parcheggi in struttura	QuestIT	In attesa di avallo	OK	✓	✓	API
Sistema Grosseto	Grosseto	Parcheggi in struttura	QuestIT	In attesa di avallo	OK	✓	✓	API
PisaMo	Pisa	Parcheggi in struttura	Liberologico	In attesa di avallo	OK	✓	In attesa di avallo	API
Parcheggi lungo strada PI	Pisa	Parcheggi lungo strada	Liberologico	Avallo ottenuto e dati già gestiti	OK	✓	✓	API

Firenze Parcheggi	Firenze	Parcheggi in struttura	SII-Mobility	dati già gestiti	OK	V	V	open-data area di Firenze
Parcheggi lungo strada FI	Firenze	Parcheggi lungo strada	Liberologico	Avallo ottenuto e dati già gestiti	OK	V	X	API
ZCS	Firenze	Parcheggio controllato	SII-Mobility	dati già gestiti	OK	V	X	open-data area di Firenze
Parcheggi Toscana	Toscana	Parcheggi in struttura	SII-Mobility (dati parzialmente gestiti su ServiceMap)	Avallo ottenuto	In fase di integrazione	V	V	MIIC area di Firenze

2.1.4 Attività di realizzazione ETL

2.1.4.1 Siena Parcheggi (Quest-IT)

2.1.4.1.1 Dati input

Siena Parcheggi espone i dati tramite un web-service REST all'indirizzo:

http://95.110.228.237/siena_parcheggi/sp_data.php?s=siimobility&f=json

Il servizio fornisce in output un json contenente le informazioni su 15 parcheggi gestiti da SienaParcheggi, di cui 4 non visibili ed uno in uno stato da non visualizzare. Il json restituito include in un'unica risposta sia i dati **statici** sia i dati **dinamici** (real-time) dei parcheggi ed è strutturato come un array associativo di elementi la cui chiave costituisce l'identificativo del parcheggio mentre il valore consiste in un oggetto strutturato che rappresenta il parcheggio e che contiene una serie di informazioni relative al parcheggio stesso (nome, georeferenziazione, dati sull'occupazione, ecc...). In particolare, i dati forniti per ogni parcheggio sono:

- id – identificativo
- nome e descrizione
- georeferenziazione
- informazioni sui posti totali e quelli occupati (da cui derivare il numero di posti disponibili)
- stato del parcheggio – un codice numerico che indica se il parcheggio è aperto (codice 80) o meno (es. chiuso, riservato, lavori, ecc...).
- timestamp dell'ultimo aggiornamento
- alcuni campi proprietari relativi all'eventuale impostazione manuale del parcheggio
- un campo che fornisce informazioni sulla visibilità o meno del parcheggio

Di seguito è riportato un'estratto di json relativo ad un singolo parcheggio:

```

1. {
2.   "10": {
3.     "id": "10",
4.     "name": "Il Campo",
5.     "description": "Via di Fontanella n.4",
6.     "lat": "43.314272222222",
7.     "lng": "11.3335388888889",
8.     "total": "544",
9.     "used": 335,
10.    "status": "80",
11.    "updated": "2017-03-23 10:55:54",
12.    "type": "auto",
13.    "current_mode": "auto",
14.    "free_m": "0",
15.    "used_m": 0,
16.    "total_m": "0",
17.    "status_m": "0",
18.    "visible": "yes"
19.  },
20.  ...
21. }

```

Da una analisi dei dati ritornati dal servizio, abbiamo selezionato per l'uso in SIIMobility un sottoinsieme dei campi disponibili ignorando i campi proprietari:

- type e current_mode
- free_m, used_m, total_m e status_m

Seguendo le linee guida per lo sviluppo degli ETL del deliverable 4.2 del protocollo Sii-Mobility, sono stati progettati e realizzati due moduli, uno per i dati statici, l'altro per i dati real time.

2.1.4.1.2 Dati statici: schema e descrizione ETL

Sono stati progettati e realizzati tre ETL dedicati alle tre fasi di recupero, quality improvement e trasformazione dei dati:

- Ingestion – recupero del dato statico attraverso API del gestore, selezione, processing, normalizzazioni ed infine memorizzazione interna;
- QI – modulo di quality improvement del dato acquisito nello step di ingestion;
- Triplication – trasformazione del dato in una tripla conforme alla ontologia specifica di SIIMobility (Km4City).

Di seguito sono riportati i dettagli per ognuno dei tre moduli ETL realizzati inclusi screenshot tratti dal software di progettazione utilizzato (Pentaho Spoon).

2.1.4.1.2.1 Ingestion

Il modulo di Ingestion è costituito da tre sottomoduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySql, database HBase, ecc.) da file csv
- **Database**: lettura dei parametri di processo dal database MySql
- **Job Ingestion**
 - **getAPI**: chiamata all'API REST e salvataggio dei dati su file csv
 - **Select values**: filtraggio dei parcheggi con status operativo e selezione dei dati statici
 - **HBase output**: scrittura dei dati statici sul database HBase (tabella dei dati raw)

Si riporta di seguito lo schema di dipendenza dei sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.1.4.1.2.2 Quality Improvement

Il modulo di Quality Improvement è costituito da tre sotto-moduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: lettura dei parametri di processo dal database MySQL
- **Trasformation QI**
 - **HBase input**: lettura dei dati statici dal database HBase (tabella dei dati raw)
 - **Manipulations**: operazioni di quality improvement
 - **HBase output**: scrittura dei dati statici sul database HBase (tabella dei dati QI)

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



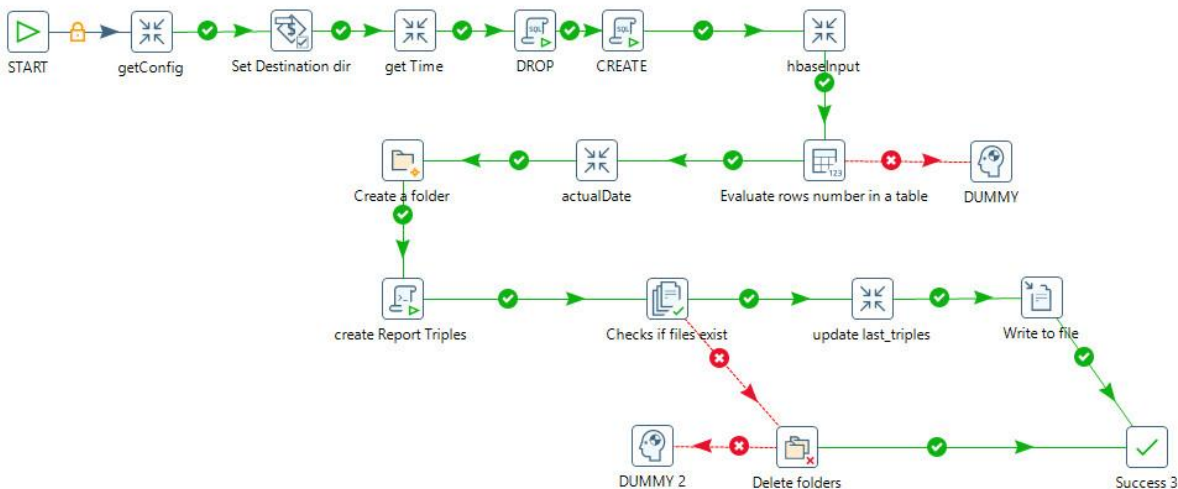
Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.1.4.1.2.3 Triplification

Il modulo di Triplification è costituito da quattro sotto-moduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: preparazione di una tabella dedicata del database MySQL per la scrittura dell'output finale
- **HBaseInput**: lettura dei dati statici dal database HBase (tabella dei dati QI) e scrittura degli stessi sulla tabella MySQL
- **create Report Triples**: modulo di generazione delle triple a partire dall'ontologia Sii-Mobility

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.1.4.1.3 Dati real time: schema e descrizione ETL

Sono stati progettati e realizzati due ETL dedicati alle due fasi di recupero e trasformazione dei dati real time:

- Ingestion – recupero del dato real time attraverso API del gestore, selezione, processing, normalizzazioni ed infine memorizzazione interna;
- Triplification – trasformazione del dato in una tripla conforme alla ontologia specifica di SII-Mobility (Km4City).

Di seguito sono riportati i dettagli per ognuno dei due moduli ETL realizzati inclusi screenshot tratti dal software di progettazione utilizzato (Pentaho Spoon).

2.1.4.1.3.1 Ingestion

Il modulo di Ingestion è costituito da tre sottomoduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: lettura dei parametri di processo dal database MySQL
- **Job Ingestion**
 - **getAPI**: chiamata all'API REST e salvataggio dei dati su file csv
 - **Select values**: filtraggio dei parcheggi con status operativo e selezione dei dati real time
 - **HBase output**: scrittura dei dati real time sul database HBase (tabella dei dati raw)

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



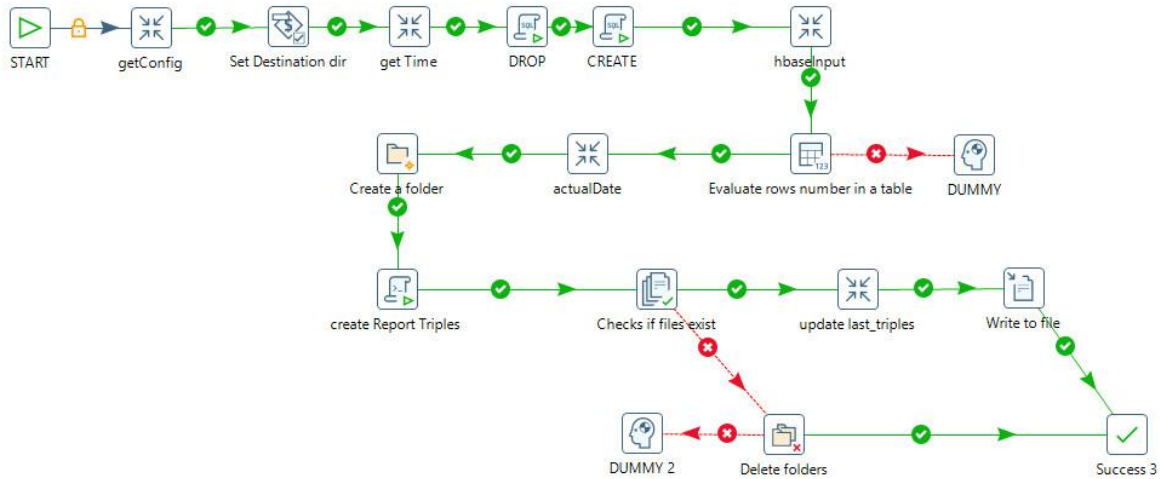
Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.1.4.1.3.2 Triplification

Il modulo di Triplification è costituito da quattro sotto-moduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: preparazione di una tabella dedicata del database MySQL per la scrittura dell'output finale
- **HBaseInput**: lettura dei dati real time dal database HBase (tabella dei dati raw) e scrittura degli stessi sulla tabella MySQL
- **create Report Triples**: modulo di generazione delle triple a partire dall'ontologia Sii-Mobility

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.1.4.2 Sistema Grosseto (Quest-IT)

2.1.4.2.1 Dati input

Sistema Grosseto espone i dati tramite un web-service REST all'indirizzo:

http://95.110.228.237/grp/sp_data.php?s=siimobility&f=json

Il servizio fornisce in output un json contenente le informazioni su 4 parcheggi gestiti da SistemaGrosseto, integrando in un'unica risposta sia i dati **statici** sia i dati **dinamici** (real-time) dei parcheggi. Il json restituito è strutturato come un array associativo di elementi la cui chiave costituisce l'identificativo del parcheggio mentre il valore consiste in un oggetto strutturato che rappresenta il parcheggio e che contiene una serie di informazioni relative al parcheggio stesso (nome, georeferenziazione, dati sull'occupazione, ecc...). In particolare, i dati forniti per ogni parcheggio sono:

- id – identificativo
- nome e descrizione
- georeferenziazione
- informazioni sui posti totali e quelli occupati (da cui derivare il numero di posti disponibili)
- stato del parcheggio – un codice numerico che indica se il parcheggio è aperto (codice 80) o meno (es. chiuso, riservato, lavori, ecc...).

Di seguito è riportato un estratto di json relativo ad un singolo parcheggio:

```

1. {
2.   "1": {
3.     "id": "1",
4.     "name": "Parcheggio Galleria Oberdan",
5.     "description": "Via Buoizzi",
6.     "lat": "42.7649",
7.     "lng": "11.11365",
8.     "total": "103",
9.     "used": 44,
10.    "status": 80
11.   },
12.   ...
13. }

```

Seguendo le linee guida per lo sviluppo degli ETL del deliverable 4.2 del protocollo Sii-Mobility, sono stati progettati e realizzati due moduli, uno per i dati statici, l'altro per i dati real time.

2.1.4.2.2 Dati statici: schema e descrizione ETL

Sono stati progettati e realizzati tre ETL dedicati alle tre fasi di recupero, quality improvement e trasformazione dei dati:

- Ingestion – recupero del dato statico attraverso API del gestore, selezione, processing, normalizzazioni ed infine memorizzazione interna;
- QI – modulo di quality improvement del dato acquisito nello step di ingestion;
- Triplification – trasformazione del dato in una tripla conforme alla ontologia specifica di SIIMobility (Km4City).

Di seguito sono riportati i dettagli per ognuno dei tre moduli ETL realizzati inclusi screenshot tratti dal software di progettazione utilizzato (Pentaho Spoon).

2.1.4.2.2.1 Ingestion

Il modulo di Ingestion è costituito da tre sottomoduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: lettura dei parametri di processo dal database MySQL
- **Job Ingestion**
 - **getAPI**: chiamata all'API REST e salvataggio dei dati su file csv
 - **Select values**: filtraggio dei parcheggi con status operativo e selezione dei dati statici
 - **HBase output**: scrittura dei dati statici sul database HBase (tabella dei dati raw)

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.1.4.2.2.2 Quality Improvement

Il modulo di Quality Improvement è costituito da tre sotto-moduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.)

da file csv

- **Database:** lettura dei parametri di processo dal database MySQL
- **Trasformation QI**
 - **HBase input:** lettura dei dati statici dal database HBase (tabella dei dati raw)
 - **Manipulations:** operazioni di quality improvement
 - **HBase output:** scrittura dei dati statici sul database HBase (tabella dei dati QI)

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



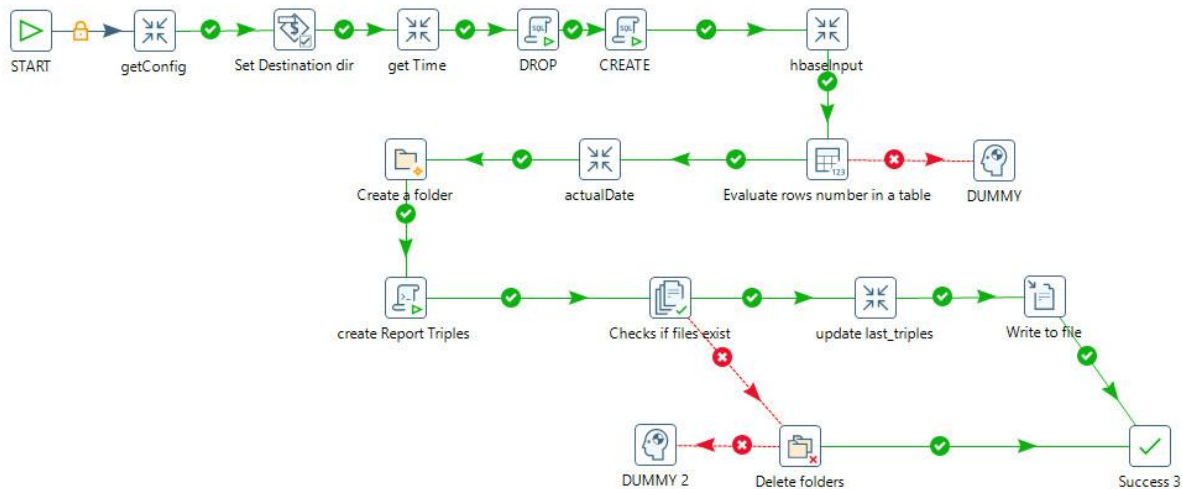
Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.1.4.2.2.3 Triplification

Il modulo di Triplification è costituito da quattro sotto-moduli principali:

- **getConfig:** lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database:** preparazione di una tabella dedicata del database MySQL per la scrittura dell'output finale
- **HBaseInput:** lettura dei dati statici dal database HBase (tabella dei dati QI) e scrittura degli stessi sulla tabella MySQL
- **create Report Triples:** modulo di generazione delle triple a partire dall'ontologia Sii-Mobility

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.1.4.2.3 Dati real time: schema e descrizione ETL

Sono stati progettati e realizzati due ETL dedicati alle due fasi di recupero e trasformazione dei dati real time:

- Ingestion – recupero del dato real time attraverso API del gestore, selezione, processing, normalizzazioni ed infine memorizzazione interna;
- Triplification – trasformazione del dato in una tripla conforme alla ontologia specifica di SII-Mobility (Km4City).

Di seguito sono riportati i dettagli per ognuno dei due moduli ETL realizzati inclusi screenshot tratti

dal software di progettazione utilizzato (Pentaho Spoon).

2.1.4.2.3.1 Ingestion

Il modulo di Ingestion è costituito da tre sottomoduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: lettura dei parametri di processo dal database MySQL
- **Job Ingestion**
 - **getAPI**: chiamata all'API REST e salvataggio dei dati su file csv
 - **Select values**: filtraggio dei parcheggi con status operativo e selezione dei dati real time
 - **HBase output**: scrittura dei dati real time sul database HBase (tabella dei dati raw)

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



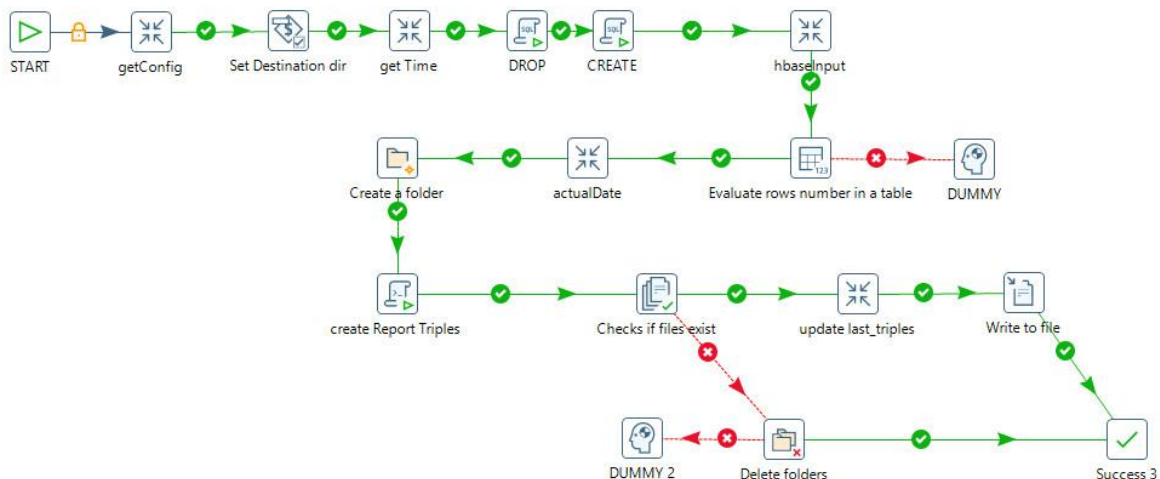
Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.1.4.2.3.2 Triplification

Il modulo di Triplification è costituito da quattro sotto-moduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: preparazione di una tabella dedicata del database MySQL per la scrittura dell'output finale
- **HBaseInput**: lettura dei dati real time dal database HBase (tabella dei dati raw) e scrittura degli stessi sulla tabella MySQL
- **create Report Triples**: modulo di generazione delle triple a partire dall'ontologia Sii-Mobility

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.1.4.3 PisaMo (Liberologico)

2.1.4.3.1 Dati input

PisaMo espone i dati tramite web-service SOAP all'indirizzo:

http://www.pisamo.it/ws_ps/index.php

Il servizio mette a disposizione diversi metodi per l'integrazione con le aree di sosta monitorate con sensori per la città di Pisa. Per quanto riguarda i servizi consultabili, questi forniscono in uscita degli XML che contengono sia i dati **statici** sia i dati **dinamici** (real-time).

Nello specifico per il Sii Mobility vengono utilizzate 3 richieste distinte:

- getParkingStateByCity
- getParkingAreas
- getParkingSpotTypes

La prima fornisce una serie di informazioni sia statiche che dinamiche, tra cui lo stato di occupazione delle aree di parcheggio gestite dal servizio mentre le altre due richieste forniscono informazioni statiche delle aree stesse:

- la sua posizione geografica (indirizzo, latitudine e longitudine, ecc.)
- la descrizione dei tipi di parcheggi.

Nello specifico, getParkingStateByCity fornisce le seguenti informazioni:

- areaCode: identificatore interno dell'area generale di parcheggio
- typeCode: identificatore specifico di ciascuna area (divise per tipo)
- freePlaces: numero posti disponibili

Estratto della risposta xml, come esempio:

```
<item xsi:type="tns:ParkingState">
  <areaCode xsi:type="xsd:string">PISAM001</areaCode>
  <description xsi:type="xsd:string">Parcheggio Piazza Carrara</description>
  <freePlaces xsi:type="xsd:int">7</freePlaces>
  <totalPlaces xsi:type="xsd:int">68</totalPlaces>
  <typeCode xsi:type="xsd:string">STPUB</typeCode>
  <typeDescription xsi:type="xsd:string">Stallo Pubblico</typeDescription>
  <latitude xsi:type="xsd:string">43.7167216</latitude>
  <longitude xsi:type="xsd:string">10.3970917</longitude>
</item>
```

Delle altre due richieste, getParkingAreas fornisce dati di localizzazione e il numero totale dei posti disponibili:

- code: identificatore interno dell'area, collegato ad areaCode nel metodo precedente
- description: descrizione
- latitude: latitudine
- longitude: longitudine
- address: indirizzo
- numero totale di stalli per tipologia

Estratto della risposta xml, come esempio:

```
<item xsi:type="tns:ParkingArea">
  <code xsi:type="xsd:string">PISAM001</code>
  <description xsi:type="xsd:string">Parcheggio Piazza Carrara</description>
  <numberSpaces xsi:type="xsd:integer">72</numberSpaces>
  <latitude xsi:type="xsd:string">43.7167216</latitude>
  <longitude xsi:type="xsd:string">10.3970917</longitude>
  <address xsi:type="xsd:string">Piazza Carrara 1, 56100, Pisa PI</address>
  <parkingSpotSensors xsi:type="SOAP-ENC:Array" SOAP-ENC:arrayType="tns:ParkingSpotSensor[72]">
    <item xsi:type="tns:ParkingSpotSensor">
      <code xsi:type="xsd:string">0004A3A274EB</code>
      <description xsi:type="xsd:string">
      </description>
      <latitude xsi:type="xsd:string">
      </latitude>
      <longitude xsi:type="xsd:string">
      </longitude>
      <positionOnMap xsi:type="xsd:string">1</positionOnMap>
      <parkingSpotType xsi:type="xsd:string">STPUB</parkingSpotType>
    </item>
  </parkingSpotSensors>
</item>
```

Infine, il metodo `getParkingSpotTypes` fornisce una descrizione per ogni possibile tipo di stallo parcheggio:

- `code`: identificatore e nome del tipo, riferito dal primo metodo come `typeCode`
- `description`: descrizione

Estratto della risposta xml, come esempio:

```
<item xsi:type="tns:ParkingSpotTypes">
  <code xsi:type="xsd:string">STPUB</code>
  <description xsi:type="xsd:string">Stallo Pubblico</description>
</item>
```

Seguendo le linee guida per lo sviluppo degli ETL del deliverable 4.2 del protocollo Sii-Mobility, sono stati progettati e realizzati due moduli, uno per i dati statici, l'altro per i dati real time.

2.1.4.3.2 Dati statici: schema e descrizione ETL

Sono stati progettati e realizzati tre ETL dedicati alle tre fasi di recupero, quality improvement e trasformazione dei dati:

- **Ingestion** – recupero del dato statico attraverso API del gestore, selezione, processing, normalizzazioni ed infine memorizzazione interna;
- **QI** – modulo di quality improvement del dato acquisito nello step di ingestion;
- **Triplification** – trasformazione del dato in una tripla conforme all'ontologia specifica di SII-Mobility (Km4City).

Di seguito sono riportati i dettagli per ognuno dei tre moduli ETL realizzati inclusi screenshot tratti dal software di progettazione utilizzato (Pentaho Spoon).

2.1.4.3.2.1 Ingestion

Il modulo di Ingestion è costituito da tre sottomoduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: lettura dei parametri di processo dal database MySQL
- **Ingestion: locations** e **Ingestion: stalls**: la struttura generale delle operazioni è la stessa, ripetuta per ciascuna richiesta SOAP:
 - **Connect and fetch xml**: chiamata all'API SOAP e salvataggio della risposta in un file xml
 - **Filter values**: filtraggio di risultati non rilevanti, se necessario

- o **Update databases:** scrittura dei dati statici sul database HBase (tabella dei dati raw) e aggiornamento storia nella tabella di gestione processi.

Si riporta di seguito lo schema di dipendenza dei sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.1.4.3.2.2 Quality Improvement

Il modulo di Quality Improvement è costituito da tre sotto-moduli principali:

- **getConfig:** lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database:** lettura dei parametri di processo dal database MySQL
- **Quality improvement**
 - o **HBase input:** lettura dei dati statici dal database HBase (tabella dei dati raw)
 - o **Manipulations:** operazioni di quality improvement
 - o **HBase output:** scrittura dei dati statici sul database HBase (tabella dei dati QI)

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



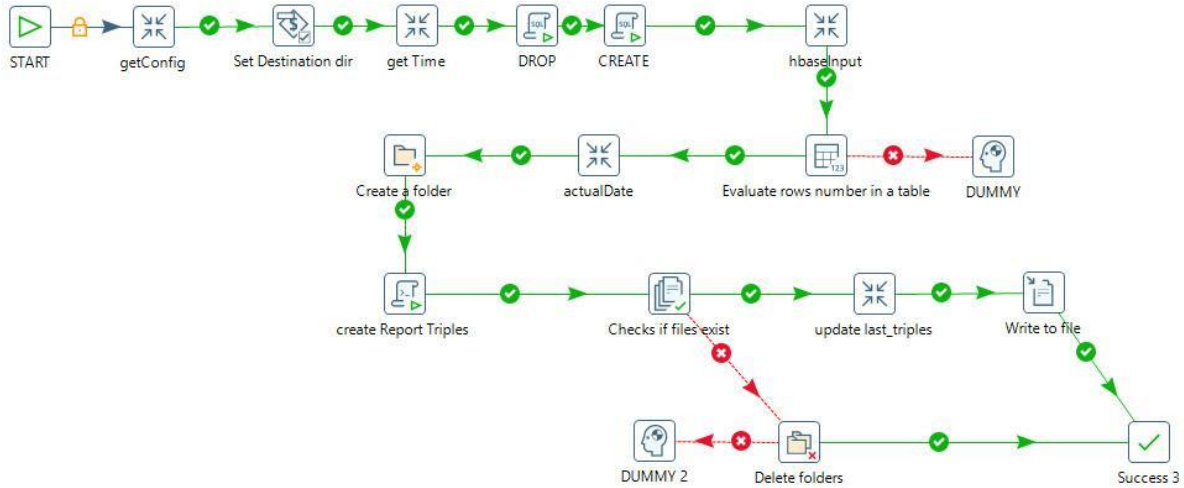
Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.1.4.3.2.3 Triplification

Il modulo di Triplification è costituito da quattro sotto-moduli principali:

- **getConfig:** lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database:** preparazione di una tabella dedicata del database MySQL per la scrittura dell'output finale
- **HBaseInput:** lettura dei dati statici dal database HBase (tabella dei dati QI) e scrittura degli stessi sulla tabella MySQL
- **create Report Triples:** modulo di generazione delle triple a partire dall'ontologia Sii-Mobility

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.1.4.3.3 Dati real time: schema e descrizione ETL

Sono stati progettati e realizzati due ETL dedicati alle due fasi di recupero e trasformazione dei dati real time:

- Ingestion – recupero del dato real time attraverso API del gestore, selezione, processing, normalizzazioni ed infine memorizzazione interna;
- Triplication – trasformazione del dato in una tripla conforme alla ontologia specifica di SIIMobility (Km4City).

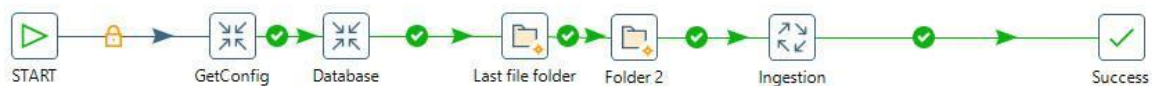
Di seguito sono riportati i dettagli per ognuno dei due moduli ETL realizzati inclusi screenshot tratti dal software di progettazione utilizzato (Pentaho Spoon).

2.1.4.3.3.1 Ingestion

Il modulo di Ingestion è costituito da tre sottomoduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: lettura dei parametri di processo dal database MySQL
- **Job Ingestion**
 - **getAPI**: chiamata all'API REST e salvataggio dei dati su file csv
 - **Select values**: filtraggio dei parcheggi con status operativo e selezione dei dati real time
 - **HBase output**: scrittura dei dati real time sul database HBase (tabella dei dati raw)

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.1.4.3.3.2 Triplication

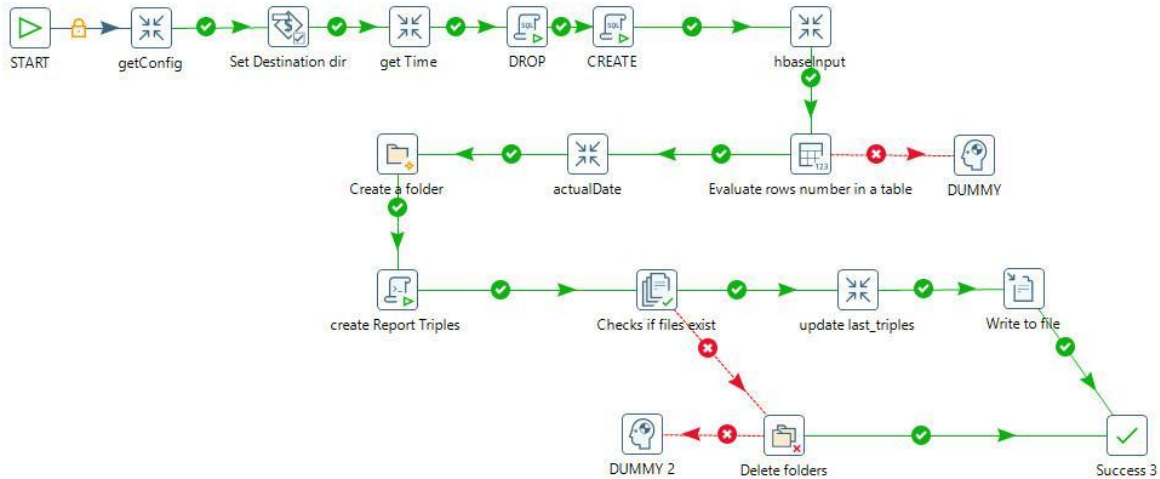
Il modulo di Triplication è costituito da quattro sotto-moduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: preparazione di una tabella dedicata del database MySQL per la scrittura dell'output

finale

- **HBaseInput**: lettura dei dati real time dal database HBase (tabella dei dati raw) e scrittura degli stessi sulla tabella MySql
- **create Report Triples**: modulo di generazione delle triple a partire dall'ontologia Sii-Mobility

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.1.4.4 Firenze parcheggi (Liberologico)

2.1.4.4.1 Dati input

SaS - Servizi alla Strada - espone i dati tramite web-service SOAP all'indirizzo:

https://servizionline.serviziallastrada.it/ws_ps/index.php

Il servizio mette a disposizione diversi metodi per l'integrazione con le aree di sosta monitorate con sensori per la città di Firenze. Per quanto riguarda i servizi consultabili, questi forniscono in uscita degli XML che contengono sia i dati **statici** sia i dati **dinamici** (real-time).

Nello specifico per il Sii Mobility vengono utilizzate 3 richieste distinte:

- getParkingStateByCity
- getParkingAreas
- getParkingSpotTypes

La prima fornisce una serie di informazioni sia statiche che dinamiche, tra cui lo stato di occupazione delle aree di parcheggio gestite dal servizio mentre le altre due richieste forniscono informazioni statiche delle aree stesse:

- la sua posizione geografica (indirizzo, latitudine e longitudine, ecc.)
- la descrizione dei tipi di parcheggi.

Nello specifico, getParkingStateByCity fornisce le seguenti informazioni:

- areaCode: identificatore interno dell'area generale di parcheggio
- typeCode: identificatore specifico di ciascuna area (divise per tipo)
- freePlaces: numero posti disponibili

Estratto della risposta xml, come esempio:

```
<item xsi:type="tns:ParkingState">
  <areaCode xsi:type="xsd:string">PISAM001</areaCode>
  <description xsi:type="xsd:string">Parcheggio Piazza Carrara</description>
  <freePlaces xsi:type="xsd:int">7</freePlaces>
  <totalPlaces xsi:type="xsd:int">68</totalPlaces>
  <typeCode xsi:type="xsd:string">STPUB</typeCode>
  <typeDescription xsi:type="xsd:string">Stallo Pubblico</typeDescription>
  <latitude xsi:type="xsd:string">43.7167216</latitude>
  <longitude xsi:type="xsd:string">10.3970917</longitude>
</item>
```

Delle altre due richieste, getParkingAreas fornisce dati di localizzazione e il numero totale dei posti disponibili.

- code: identificatore interno dell'area, collegato ad areaCode nel metodo precedente
- description: descrizione
- latitude: latitudine
- longitude: longitudine
- address: indirizzo
- numero totale di stalli per tipologia

Estratto della risposta xml, come esempio:

```
<item xsi:type="tns:ParkingArea">
  <code xsi:type="xsd:string">PISAM001</code>
  <description xsi:type="xsd:string">Parcheggio Piazza Carrara</description>
  <numberSpaces xsi:type="xsd:integer">72</numberSpaces>
  <latitude xsi:type="xsd:string">43.7167216</latitude>
  <longitude xsi:type="xsd:string">10.3970917</longitude>
  <address xsi:type="xsd:string">Piazza Carrara 1, 56100, Pisa PI</address>
  <parkingSpotSensors xsi:type="SOAP-ENC:Array" SOAP-ENC:arrayType="tns:ParkingSpotSensor[72]">
    <item xsi:type="tns:ParkingSpotSensor">
      <code xsi:type="xsd:string">0004A3A274EB</code>
      <description xsi:type="xsd:string">
      </description>
      <latitude xsi:type="xsd:string">
      </latitude>
      <longitude xsi:type="xsd:string">
      </longitude>
      <positionOnMap xsi:type="xsd:string">1</positionOnMap>
      <parkingSpotType xsi:type="xsd:string">STPUB</parkingSpotType>
    </item>
  </parkingSpotSensors>
</item>
```

Infine, il metodo getParkingSpotTypes fornisce una descrizione per ogni possibile tipo di stallo parcheggio:

- code: identificatore e nome del tipo, riferito dal primo metodo come typeCode
- description: descrizione

Estratto della risposta xml, come esempio:

```
<item xsi:type="tns:ParkingSpotTypes">
  <code xsi:type="xsd:string">STPUB</code>
  <description xsi:type="xsd:string">Stallo Pubblico</description>
</item>
```

Seguendo le linee guida per lo sviluppo degli ETL del deliverable 4.2 del protocollo Sii-Mobility, sono stati progettati e realizzati due moduli, uno per i dati statici, l'altro per i dati real time.

2.1.4.4.2 Dati statici: schema e descrizione ETL

Sono stati progettati e realizzati tre ETL dedicati alle tre fasi di recupero, quality improvement e trasformazione dei dati:

- Ingestion – recupero del dato statico attraverso API del gestore, selezione, processing, normalizzazioni ed infine memorizzazione interna;
- QI – modulo di quality improvement del dato acquisito nello step di ingestion;
- Triplification – trasformazione del dato in una tripla conforme all'ontologia specifica di SIIMobility (Km4City).

Di seguito sono riportati i dettagli per ognuno dei tre moduli ETL realizzati inclusi screenshot tratti dal software di progettazione utilizzato (Pentaho Spoon).

2.1.4.4.2.1 Ingestion

Il modulo di Ingestion è costituito da tre sottomoduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: lettura dei parametri di processo dal database MySQL
- **Ingestion: locations e Ingestion: stalls**: la struttura generale delle operazioni è la stessa, ripetuta per ciascuna richiesta SOAP:
 - **Connect and fetch xml**: chiamata all'API SOAP e salvataggio della risposta in un file xml
 - **Filter values**: filtraggio di risultati non rilevanti, se necessario
 - **Update databases**: scrittura dei dati statici sul database HBase (tabella dei dati raw) e aggiornamento storia nella tabella di gestione processi.

Si riporta di seguito lo schema di dipendenza dei sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.1.4.4.2.2 Quality Improvement

Il modulo di Quality Improvement è costituito da tre sotto-moduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: lettura dei parametri di processo dal database MySQL
- **Quality improvement**
 - **HBase input**: lettura dei dati statici dal database HBase (tabella dei dati raw)
 - **Manipulations**: operazioni di quality improvement

o **HBase output:** scrittura dei dati statici sul database HBase (tabella dei dati QI)

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



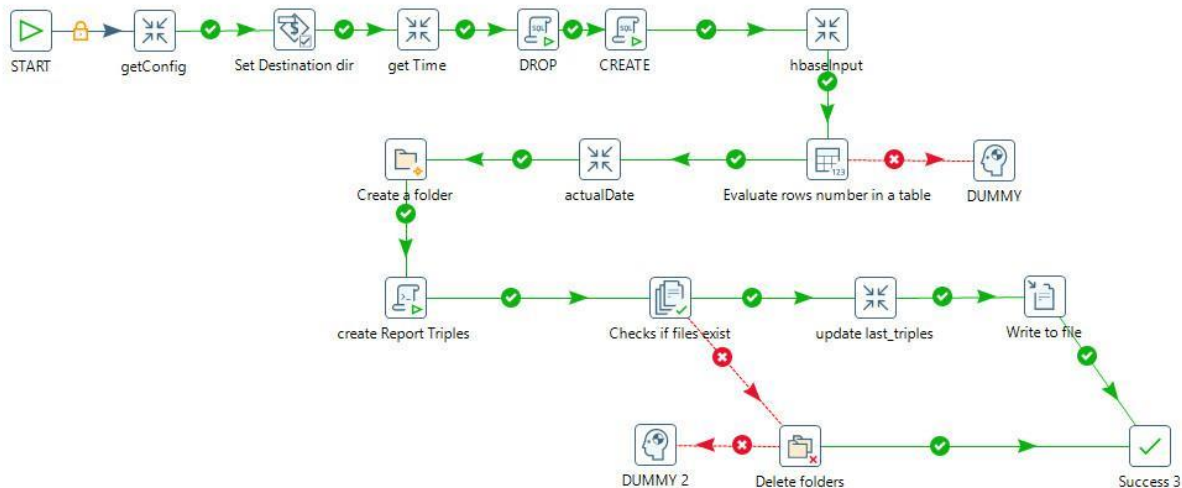
Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all’analisi del codice del progetto.

2.1.4.4.2.3 Triplification

Il modulo di Triplification è costituito da quattro sotto-moduli principali:

- **getConfig:** lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database:** preparazione di una tabella dedicata del database MySQL per la scrittura dell’output finale
- **HBaseInput:** lettura dei dati statici dal database HBase (tabella dei dati QI) e scrittura degli stessi sulla tabella MySQL
- **create Report Triples:** modulo di generazione delle triple a partire dall’ontologia Sii-Mobility

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all’analisi del codice del progetto.

2.1.4.4.3 Dati real time: schema e descrizione ETL

Sono stati progettati e realizzati due ETL dedicati alle due fasi di recupero e trasformazione dei dati real time:

- Ingestion – recupero del dato real time attraverso API del gestore, selezione, processing, normalizzazioni ed infine memorizzazione interna;
- Triplification – trasformazione del dato in una tripla conforme alla ontologia specifica di SIIMobility (Km4City).

Di seguito sono riportati i dettagli per ognuno dei due moduli ETL realizzati inclusi screenshot tratti dal software di progettazione utilizzato (Pentaho Spoon).

2.1.4.4.3.1 Ingestion

Il modulo di Ingestion è costituito da tre sottomoduli principali:

- **getConfig:** lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database:** lettura dei parametri di processo dal database MySQL
- **Job Ingestion**
 - **getAPI:** chiamata all'API REST e salvataggio dei dati su file csv
 - **Select values:** filtraggio dei parcheggi con status operativo e selezione dei dati real time
 - **HBase output:** scrittura dei dati real time sul database HBase (tabella dei dati raw)

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



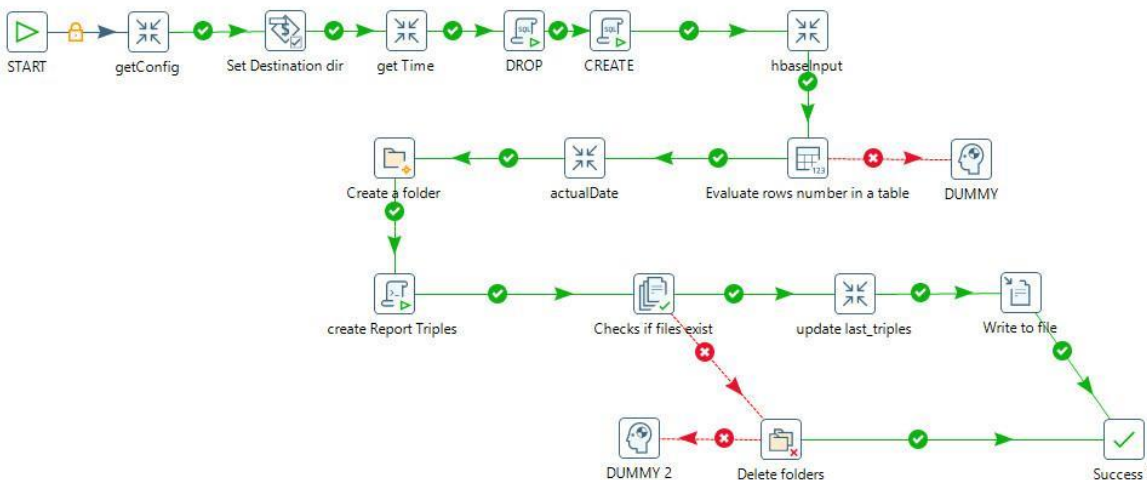
Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.1.4.4.3.2 Triplification

Il modulo di Triplification è costituito da quattro sotto-moduli principali:

- **getConfig:** lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database:** preparazione di una tabella dedicata del database MySQL per la scrittura dell'output finale
- **HBaseInput:** lettura dei dati real time dal database HBase (tabella dei dati raw) e scrittura degli stessi sulla tabella MySQL
- **create Report Triples:** modulo di generazione delle triple a partire dall'ontologia Sii-Mobility

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.2 Attività 5.4.2: car and bike sharing [PETL21]

2.2.1 Analisi del tipo di servizio e del tipo di dati

I servizi di car e bike sharing prevedono una serie di informazioni molto simili a quelle dei parcheggi. In particolare, ogni punto di prelievo/riconsegna è riferibile come “parcheggio” con informazioni

statiche (nome, geolocalizzazione ecc...) e dinamiche (disponibilità di auto e/o biciclette, stalli liberi per la riconsegna ecc...).

Di seguito si riporta lo schema di JSON strutturale da considerare come riferimento per la categoria dei gestori di car & bike sharing. Tale schema rimane come riferimento di base sia nel caso di nuovi gestori sia nel caso di gestori già affiliati e può essere soggetto a customizzazione. Al variare della modalità di inserimento del dato (manuale, automatizzato via API, open-data) è possibile immaginare una differenziazione fra dati statici e dati dinamici. La struttura JSON proposta include per ogni parcheggio le informazioni statiche e dinamiche e prevede, nel caso di parcheggi non-monitorati, la segnalazione della mancanza di informazioni dinamiche (monitored = false). Inoltre, per ogni parcheggio, viene specificato il tipo di noleggio disponibile: BIKE o CAR.

CAR&BIKE SHARING [QuestIT - Time] Ver 0.1

```
{
  "company_id": 10,
  "company_name": "Company name",
  "company_description": "Company description",
  "city": "Parking city",
  "pricing": "Pricing description",
  "parkings": [{
    "id": 10,
    "type": "Bike" / "Car",
    "name": "Parking name",
    "description": "Parking description",
    "address": "Parking address",
    "lat": 43.314272222222,
    "long": 11.333538888889,
    "monitored": true/false
    "total_spaces": 10,
    "spaces_available": 8,
    "items_available": 2,
    "status": 80,
    "info": "Optional info"
  }]
}
```

2.2.2 Integrazione con i gestori

All'interno del progetto SII-Mobility sono stati individuati diversi gestori di servizi di car&bike sharing del territorio Toscana. In alcuni casi i dati sono già disponibili nel sistema SII-Mobility centrale in quanto recuperati attraverso precedenti integrazioni e possono essere perciò utilizzati direttamente. In altri casi i gestori sono stati individuati e contattati aziende che gestiscono servizi di car&bike sharing, è stato illustrato loro il progetto ed è stato chiesto loro una manifestazione di interesse ed una liberatoria all'utilizzo dei dati in loro possesso nel progetto. In particolare, si ha la seguente lista:

- Gestori i cui dati sono già disponibili:
 - Percorsi ciclabili FIRENZE
- Gestori contattati dal partner del progetto QuestIT:
 - Bicincittà SIPedala SIENA (QuestIT)
- Gestori contattati dal partner del progetto Liberologico
 - Bike sharing PISA (Liberologico)

2.2.2.1 Gestori i cui dati sono già disponibili

I dati relativi ai seguenti gestori e sistemi sono già stati integrati nel sistema SII-Mobility e quindi disponibili per l'uso verso terzi grazie alle API di ServiceMap di UNIFI:

- Percorsi ciclabili FIRENZE
- Car sharing FIRENZE

2.2.2.2 Gestori contattati dal partner del progetto QuestIT : SIPedala (Siena)

E' stata inviata una email al direttore tecnico ed al direttore generale di Siena Parcheggi in qualità di ente gestore del servizio per la città di Siena. In tale email si illustra il progetto SII-Mobility e si chiede una manifestazione di interesse da parte di Siena Parcheggi alla partecipazione al progetto fornendo l'accesso ai dati, senza alcun onere per il gestore. In questo caso, il servizio di bike sharing è gestito da Siena Parcheggi per conto del comune di Siena e quindi il direttore generale ha chiesto formalmente di contattare il vicesindaco Fulvio Mancuso illustrando la proposta di affiliazione e gli scopi del progetto SII-Mobility e chiedendo quindi al comune di Siena il relativo via libera all'uso dei dati relativi al bike sharing.

2.2.2.3 Gestori contattati dal partner del progetto Liberologico: Bike sharing PISA

E' stata contattato PisaMo, in qualità di gestore del servizio di bike sharing CICLOPI della città di Pisa, e già a conoscenza del progetto SII-Mobility in quanto contattata anche per la disponibilità a fornire dati relativi ai parcheggi on/off street. PisaMo, a seguito di accordi contrattuali con il fornitore tecnologico che vincolano anche l'impiego dei dati del servizio, ha invitato il management del progetto SII-Mobility a contattare direttamente il fornitore tecnologico "Bicincittà Srl". Liberologico può comunque raccogliere e fornire i dati di tipo statico relativo al servizio, in quanto informazioni di pubblico dominio quali:

- nome del servizio
- n° rastrelliere
- geolocalizzazione delle rastrelliere del bike sharing
- n° max posti disponibili per ogni rastrelliera
- piano tariffario uso del servizio

Per l'integrazione di tali informazioni, Liberologico renderà disponibili API REST per la realizzazione del relativo ETL. L'output restituito da tali API REST sarà conforme al modello JSON in precedenza descritto.

2.2.3 Stato della attività di integrazione con i gestori di servizi di car&bike sharing

GESTORE	AREA GEO	TIPOLOGIA SERVIZIO	PARTNER DI RIFERIMENTO	LIVELLO AVALLO AMMINIST.	AVALLO UFFICIO TECNICO	NOTE	DATI STATICI	DATI DINAMICI	METODOLOGIA INPUT DATI [manuale, API, open-data]
Bicincittà SIPedala (dati statici)	Siena	Bike sharing	QuestIT	In attesa di risposta	In attesa di risposta	-	V	X	API
Bicincittà SIPedala (dati	Siena	Bike sharing	QuestIT	In attesa di risposta	In attesa di	-	X	V	API

dinamici)					risposta				
Bike sharing PI (dati statici)	Pisa	Bike sharing	QuestIT	In attesa di risposta	In attesa di risposta	-	V	X	API
Bike sharing PI (dati dinamici)	Pisa	Bike sharing	QuestIT	In attesa di risposta	In attesa di risposta	-	X	V	API
Percorsi ciclabili FI	Firenze	Percorsi ciclabili	<u>SII-Mobility</u>	Dati statici già gestiti	OK	-	V	X	open-data

2.2.4 Attività di realizzazione ETL

2.2.4.1 SiPedala – Siena (Quest-IT)

2.2.4.1.1 Dati input

Il Comune di Siena non ha risposto alle richieste di fornitura dei dati (sia statici che dinamici). Da un’analisi del sito web del servizio SIPedala e della relativa app è emerso che i dati sono disponibili tramite un URL pubblico in modalità REST all’indirizzo:

<http://servicegold.tobike.it/service.aspx/ElencoStazioniPerComune?UsernameRivenditore=arcnos&PasswordRivenditore=lumsar&CodiceComune=5072>

Il servizio fornisce in output un xml contenente un array di stringhe, ognuna delle quali corrisponde ad una rastrelliera e contiene le informazioni di tale rastrelliera in formato csv (con delimitatore ;).

In particolare, la stringa csv è costituita da 7 distinti campi:

1. id – identificativo della rastrelliera
2. nome
3. via
4. un campo proprietario di tipo numerico
5. latitudine
6. longitudine
7. un campo composto da 30 caratteri il cui significato è il seguente:
 - il carattere ‘x’ indica che lo stallo è mancante,
 - il carattere ‘0’ indica uno stallo presente con bicicletta mancante (libero),
 - il carattere ‘4’ indica uno stallo presente con bicicletta disponibile (occupato),
 - il carattere ‘5’ indica uno stallo presente con bicicletta danneggiata (inutilizzabile)

Per esempio, la stringa 4044445404440xxxxxxxxxxxxxxxxxxxx indica che vi sono 12 stalli presenti (numero di caratteri diversi da ‘x’) di cui:

- 3 stalli liberi (numero di caratteri ‘0’ presenti)
- 8 stalli con bicicletta disponibile (numero di caratteri ‘4’ presenti)
- 1 stallo con bicicletta danneggiata (numero di caratteri ‘5’ presenti)

Di seguito è riportato l'xml restituito dal web service:

```
1. <ArrayOfString xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="c://inetub/wwwroot/webse
rvice/Service.asmx">
2. <string>
3. 1295;01. Curtatone;Viale Curtatone;0;43.32159969615422;11.327948187171955;404444540440
xxxxxxxxxxxxxxxxxxxxx
4. </string>
5. <string>
6. 1296;02. S. Agostino;Prato di Sant'Agostino;0;43.314553575496504;11.331021341104474;40
404444xxxxxxxxxxxxxxxxxxxxx
7. </string>
8. <string>
9. 1297;03. S. Girolamo;Via di San Girolamo;0;43.31653903296188;11.337211648409948;444444
4xxxxxxxxxxxxxxxxxxxxx
10. </string>
11. <string>
12. 1298;04. Fortezza;Viale Vittorio Veneto;0;43.3205234076899;11.32322528552777;000004xxx
xxxxxxxxxxxxxxxxxxxxx
13. </string>
14. <string>
15. 1299;05. Antiporto;Viale Vittorio Emanuele II;0;43.32898332166825;11.321913356869572;0
000005xxxxxxxxxxxxxxxxxxxxx
16. </string>
17. <string>
18. 1300;06. Due Ponti;Parcheggio scambiatore Viale Pietro Toselli;0;43.317187429810794;11
.354256229005387;004444xxxxxxxxxxxxxxxxxxxxx
19. </string>
20. <string>
21. 1301;07. Napoli;Parcheggio scambiatore Via Napoli;0;43.34175662737405;11.3050104577068
85;000000xxxxxxxxxxxxxxxxxxxxx
22. </string>
23. <string>
24. 1302;08. Ospedale;Viale Mario Bracci;0;43.343243018858246;11.32673330024295;445444454x
xxxxxxxxxxxxxxxxxxxxx
25. </string>
26. <string>
27. 1303;09. Acquacalda;Via Bernardo Tolomei;0;43.33699934605541;11.300974983606011;0x040x
xxxxxxxxxxxxxxxxxxxxx
28. </string>
29. <string>
30. 1304;10. Ravacciano;Via Duccio di Boninsegna;0;43.323781022808305;11.338891018177037;0
0000xxxxxxxxxxxxxxxxxxxxx
31. </string>
32. <string>
33. 1305;11. S. Miniato;Piazza Palmiro Togliatti;0;43.34779988043242;11.326045898741882;44
4450xxxxxxxxxxxxxxxxxxxxx
34. </string>
35. <string>
36. 1306;12. Terminal Bus;Viale Riccardo Lombardi;0;43.33228840552265;11.32417581394725;00
004000000xxxxxxxxxxxxxxxxxxxxx
37. </string>
38. <string>
39. 1307;13. Taverne D'Arbia;Via Bartolomeo Renaldini;0;43.29380994104219;11.3983651318771
99;044400xxxxxxxxxxxxxxxxxxxxx
40. </string>
41. <string>
42. 1475;14. Vico Alto;Via Liguria;2;43.34144450071458;11.312348354700475;xxxxxxxxxxxxxxxxx
xxxxxxxxxxxxx
43. </string>
44. </ArrayOfString>
```


2.2.4.1.2 Dati statici: schema e descrizione ETL

Sono stati progettati e realizzati tre ETL dedicati alle tre fasi di recupero, quality improvement e trasformazione dei dati:

- Ingestion – recupero del dato statico attraverso API del gestore, selezione, processing, normalizzazioni ed infine memorizzazione interna;
- QI – modulo di quality improvement del dato acquisito nello step di ingestion;
- Triplification – trasformazione del dato in una tripla conforme alla ontologia specifica di SIIMobility (Km4City).

Di seguito sono riportati i dettagli per ognuno dei tre moduli ETL realizzati inclusi screenshot tratti dal software di progettazione utilizzato (Pentaho Spoon).

2.2.4.1.2.1 Ingestion

Il modulo di Ingestion è costituito da tre sottomoduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: lettura dei parametri di processo dal database MySQL
- **Job Ingestion**
 - **getAPI**: chiamata all'API REST e salvataggio dei dati su file csv
 - **Select values**: filtraggio dei parcheggi con status operativo e selezione dei dati statici
 - **HBase output**: scrittura dei dati statici sul database HBase (tabella dei dati raw)

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.2.4.1.2.2 Quality Improvement

Il modulo di Quality Improvement è costituito da tre sotto-moduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: lettura dei parametri di processo dal database MySQL
- **Trasformation QI**
 - **HBase input**: lettura dei dati statici dal database HBase (tabella dei dati raw)
 - **Manipulations**: operazioni di quality improvement
 - **HBase output**: scrittura dei dati statici sul database HBase (tabella dei dati QI)

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

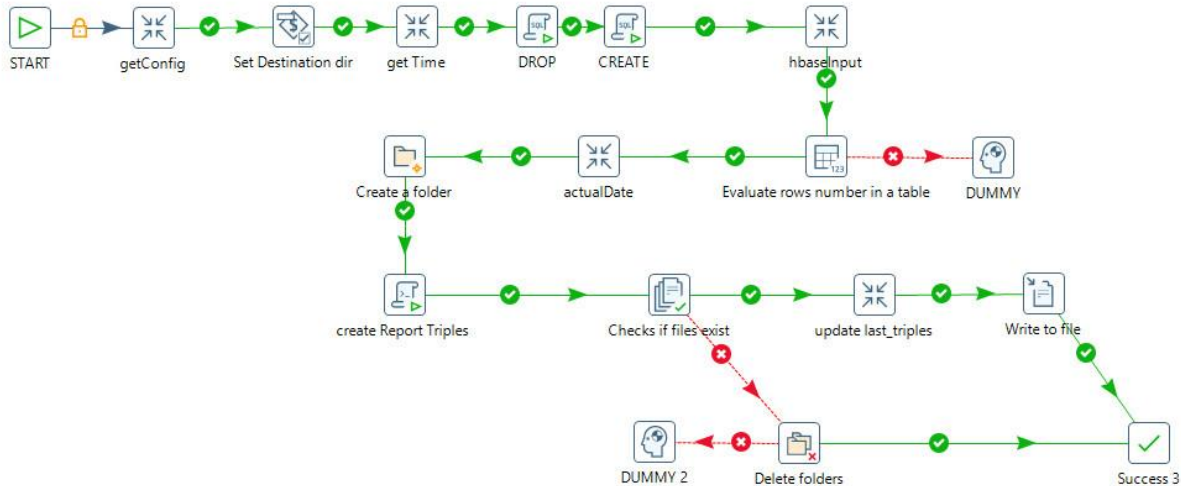
2.2.4.1.2.3 Triplification

Il modulo di Triplification è costituito da quattro sotto-moduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv

- **Database:** preparazione di una tabella dedicata del database MySql per la scrittura dell'output finale
- **HBaseInput:** lettura dei dati statici dal database HBase (tabella dei dati QI) e scrittura degli stessi sulla tabella MySql
- **create Report Triples:** modulo di generazione delle triple a partire dall'ontologia Sii-Mobility

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.2.4.1.3 Dati real time: schema e descrizione ETL

Sono stati progettati e realizzati due ETL dedicati alle due fasi di recupero e trasformazione dei dati real time:

- Ingestion – recupero del dato real time attraverso API del gestore, selezione, processing, normalizzazioni ed infine memorizzazione interna;
- TriPLICATION – trasformazione del dato in una tripla conforme alla ontologia specifica di SII-Mobility (Km4City).

Di seguito sono riportati i dettagli per ognuno dei due moduli ETL realizzati inclusi screenshot tratti dal software di progettazione utilizzato (Pentaho Spoon).

2.2.4.1.3.1 Ingestion

Il modulo di Ingestion è costituito da tre sottomoduli principali:

- **getConfig:** lettura dei parametri di configurazione (database MySql, database HBase, ecc.) da file csv
- **Database:** lettura dei parametri di processo dal database MySql
- **Job Ingestion**
 - **getAPI:** chiamata all'API REST e salvataggio dei dati su file csv
 - **Select values:** filtraggio dei parcheggi con status operativo e selezione dei dati real time
 - **HBase output:** scrittura dei dati real time sul database HBase (tabella dei dati raw)

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del

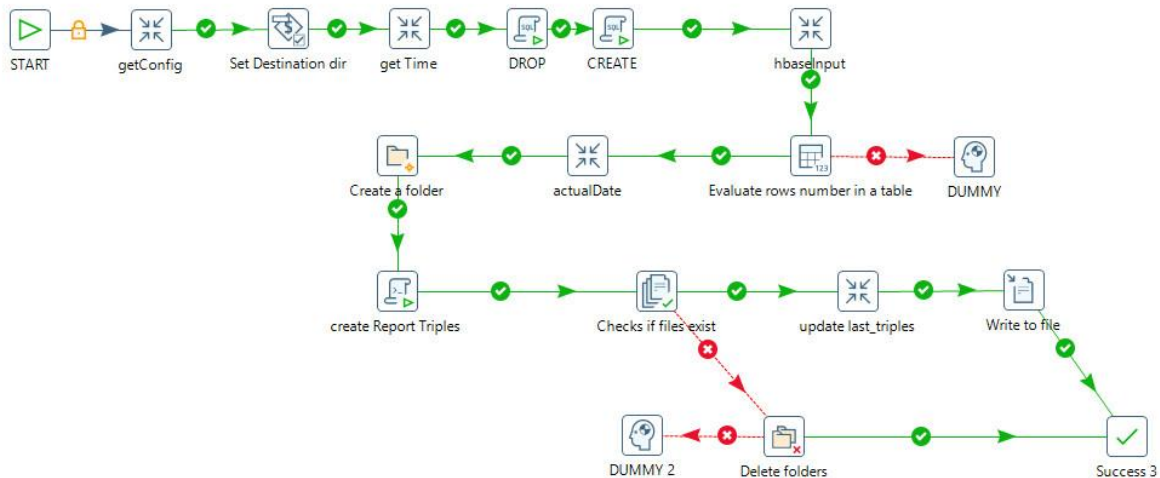
progetto.

2.2.4.1.3.2 Triplification

Il modulo di Triplification è costituito da quattro sotto-moduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: preparazione di una tabella dedicata del database MySQL per la scrittura dell'output finale
- **HBaseInput**: lettura dei dati real time dal database HBase (tabella dei dati raw) e scrittura degli stessi sulla tabella MySQL
- **create Report Triples**: modulo di generazione delle triple a partire dall'ontologia Sii-Mobility

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.2.4.2 PisaMo – Pisa (QuestIT)

2.2.4.2.1 Dati input

Il gestore PisaMo non ha risposto alle richieste di fornitura dei dati (sia statici che dinamici). Da un'analisi del sito web del servizio bike sharing e della relativa app è emerso che i dati sono disponibili tramite un URL pubblico in modalità REST all'indirizzo:

<http://servicegold.tobike.it/service.asmx/ElencoStazioniPerComune?UsernameRivenditore=arcnos&PasswordRivenditore=lumsar&CodiceComune=5037>

Il servizio fornisce in output un xml contenente un array di stringhe, ognuna delle quali corrisponde ad una rastrelliera e contiene le informazioni di tale rastrelliera in formato csv (con delimitatore ;). In particolare, la stringa csv è costituita da 6 distinti campi:

1. id – identificativo della rastrelliera
2. nome
3. via
4. un campo proprietario di tipo numerico
5. latitudine
6. longitudine

7. un campo composto da 30 caratteri il cui significato è il seguente:
- a. il carattere 'x' indica che lo stallo è mancante,
 - b. il carattere '0' indica uno stallo presente con bicicletta mancante (libero),
 - c. il carattere '4' indica uno stallo presente con bicicletta disponibile (occupato),
 - d. il carattere '5' indica uno stallo presente con bicicletta danneggiata (inutilizzabile)

Per esempio, la stringa 404444540440xxxxxxxxxxxxxxxxxxxx indica che vi sono 12 stalli presenti (numero di caratteri diversi da 'x') di cui:

- 3 stalli liberi (numero di caratteri '0' presenti)
- 8 stalli con bicicletta disponibile (numero di caratteri '4' presenti)
- 1 stallo con bicicletta danneggiata (numero di caratteri '5' presenti)

Di seguito è riportato l'xml restituito dal web service:

```
<?xml version="1.0" encoding="UTF-8"?> <ArrayOfString
xmlns="c://inetub/wwwroot/webservice/Service.asmx"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <string>922;Aeroporto;P.le C.
D'Ascanio;0;43.69889689157878;10.4007026553154;4444044040xxxxxxxxxxxxxxxxxxxx</string>
> <string>926;Borgo Stretto;Via
Oberdan;0;43.71850756685601;10.402157753705978;444000000xxxxxxxxxxxxxxxxxxxx</string>
> <string>1447;C. Marchesi;Via M.
Valgimigli;0;43.7149988319841;10.419548502516136;004000000004xxxxxxxxxxxxxxxxxxxx</stri
ng> <string>1444;Cavalieri;Via Consoli del
Mare;0;43.719862256627756;10.40019726309049;444004000040xxxxxxxxxxxxxxxxxxxx</string>
<string>1448;CNR-Praticelli;Via G.
Moruzzi;0;43.71937664670127;10.423707704299204;4440440400000xxxxxxxxxxxxxxxxxxxx</string>
> <string>924;Comune Palazzo Blu;Via
Toselli;0;43.715540542144296;10.400466620922089;x444440000044400xxxxxxxxxxxxxxxx</strin
g> <string>928;Duomo;P.zza
Manin;0;43.72288113973688;10.39179127129023;4444404444444444004500xxxxxxxx</string>
<string>1446;Galleria Gerace;Via Carlo
Matteucci;0;43.71081663441501;10.420468790351833;444444044444xxxxxxxxxxxxxxxxxxxx</stri
ng> <string>1392;Guerrazzi;P.zza D.
Guerrazzi;0;43.71035298602386;10.405330168647765;4040040004xxxxxxxxxxxxxxxxxxxx</stri
ng> <string>1443;Livornese;Via
Chiassatello;0;43.708118345328074;10.384021881753597;0000000040xxxxxxxxxxxxxxxxxxxx</
string> <string>1445;M. Libertà;P.zza Martiri della
Libertà;0;43.719864417699746;10.403020969311456;44400000040x0xxxxxxxxxxxxxxxxxxxx</strin
g> <string>932;Ospedale Cisanello;Via Pietro
Trivella;0;43.706835077467;10.440119425023681;40400045400000xxxxxxxxxxxxxxxxxxxx</string>
<string>935;Palacongressi;Via
Matteotti;0;43.71001320663703;10.410232543945312;0000000000040000400004044xxxxxxxx</stri
ng> <string>930;Paparelli;Via
Paparelli;0;43.724455713221055;10.410447120666504;1400000000040400000000040040</str
ing> <string>929;Pietrasantina;Via S.
Jacopo;0;43.729021341163644;10.392712354660034;xxxxxxx4444400440050004000000</string>
> <string>927;Polo Marzotto;Largo
Pontecorvo;0;43.71978699640267;10.407198965549469;40040000400x40xxxxxxxxxxxxxxxx0xxx</str
ing> <string>1116;Porta a Lucca;Via
Bianchi;0;43.724249448743706;10.402196310659065;0000000004xxxxxxxxxxxxxxxxxxxx</strin
g> <string>931;Pratale;Via di
Pratale;0;43.721251521991505;10.420242547988892;404000x4040400400000000000000</strin
g> <string>1391;San Rossore F.S.;Via A.
Pisano;0;43.722220128128825;10.388224964309757;4040000040044000xxxxxxxxxxxxxxxx</string>
> <string>933;Sms Biblioteca;L.go
Tibet;0;43.70658035883312;10.419128279762276;000040044404xxxxxxxxxxxxxxxxxxxx</string>
<string>1390;Solferino;P.zza
Solferino;0;43.71569279104801;10.395006805557273;0110004400xxxxxxxxxxxxxxxxxxxx</stri
ng> <string>923;Stazione F.S.;P.zza
Stazione;0;43.708474729164045;10.399020910263062;0000004040040444444040444444xx</stri
ng> <string>925;Teatro Tribunale;Via
Palestro;0;43.7163945097564;10.405159145593643;000000004044xxxxxxxxxxxxxxxxxxxx</string>
> <string>934;Vittorio Emanuele;P.zza V. Emanuele
II;0;43.71017025126551;10.398750007152557;4444004040440000xxxxxxxxxxxxxxxx</string>
</ArrayOfString>
```

2.2.4.2.2 Dati statici: schema e descrizione ETL

Sono stati progettati e realizzati tre ETL dedicati alle tre fasi di recupero, quality improvement e trasformazione dei dati:

- Ingestion – recupero del dato statico attraverso API del gestore, selezione, processing, normalizzazioni ed infine memorizzazione interna;
- QI – modulo di quality improvement del dato acquisito nello step di ingestion;
- Triplification – trasformazione del dato in una tripla conforme alla ontologia specifica di SIIMobility (Km4City).

Di seguito sono riportati i dettagli per ognuno dei tre moduli ETL realizzati inclusi screenshot tratti dal software di progettazione utilizzato (Pentaho Spoon).

2.2.4.2.2.1 Ingestion

Il modulo di Ingestion è costituito da tre sottomoduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: lettura dei parametri di processo dal database MySQL
- **Job Ingestion**
 - **getAPI**: chiamata all'API REST e salvataggio dei dati su file csv
 - **Select values**: filtraggio dei parcheggi con status operativo e selezione dei dati statici
 - **HBase output**: scrittura dei dati statici sul database HBase (tabella dei dati raw)

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.2.4.2.2.2 Quality Improvement

Il modulo di Quality Improvement è costituito da tre sotto-moduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: lettura dei parametri di processo dal database MySQL
- **Trasformation QI**
 - **HBase input**: lettura dei dati statici dal database HBase (tabella dei dati raw)
 - **Manipulations**: operazioni di quality improvement
 - **HBase output**: scrittura dei dati statici sul database HBase (tabella dei dati QI)

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

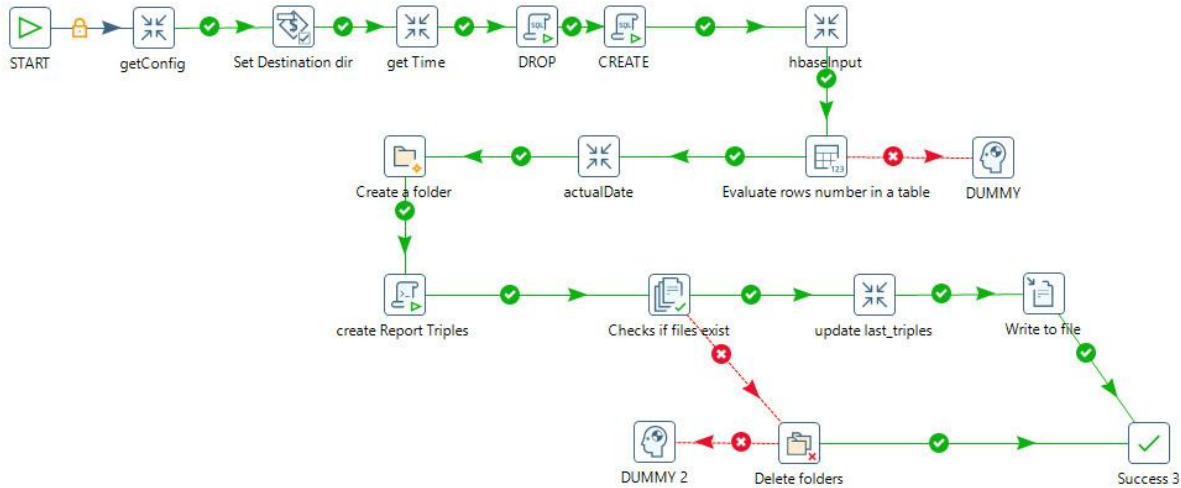
2.2.4.2.2.3 Triplification

Il modulo di Triplification è costituito da quattro sotto-moduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv

- **Database:** preparazione di una tabella dedicata del database MySQL per la scrittura dell'output finale
- **HBaseInput:** lettura dei dati statici dal database HBase (tabella dei dati QI) e scrittura degli stessi sulla tabella MySQL
- **create Report Triples:** modulo di generazione delle triple a partire dall'ontologia Sii-Mobility

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.2.4.2.3 Dati real time: schema e descrizione ETL

Sono stati progettati e realizzati due ETL dedicati alle due fasi di recupero e trasformazione dei dati real time:

- **Ingestion** – recupero del dato real time attraverso API del gestore, selezione, processing, normalizzazioni ed infine memorizzazione interna;
- **Triplication** – trasformazione del dato in una tripla conforme alla ontologia specifica di SII-Mobility (Km4City).

Di seguito sono riportati i dettagli per ognuno dei due moduli ETL realizzati inclusi screenshot tratti dal software di progettazione utilizzato (Pentaho Spoon).

2.2.4.2.3.1 Ingestion

Il modulo di Ingestion è costituito da tre sottomoduli principali:

- **getConfig:** lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database:** lettura dei parametri di processo dal database MySQL
- **Job Ingestion**
 - **getAPI:** chiamata all'API REST e salvataggio dei dati su file csv
 - **Select values:** filtraggio dei parcheggi con status operativo e selezione dei dati real time
 - **HBase output:** scrittura dei dati real time sul database HBase (tabella dei dati raw)

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del

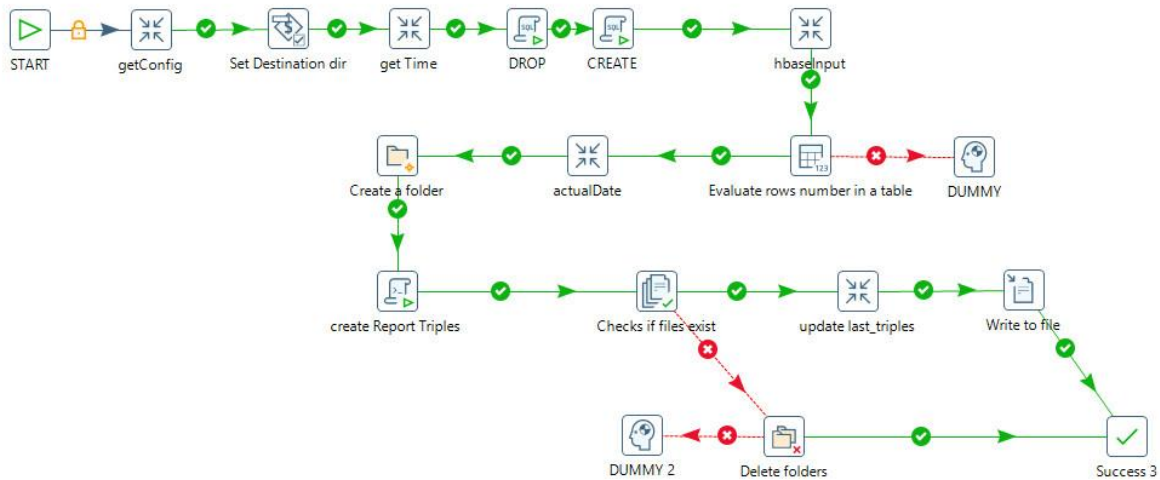
progetto.

2.2.4.2.3.2 Triplification

Il modulo di Triplification è costituito da quattro sotto-moduli principali:

- **getConfig**: lettura dei parametri di configurazione (database MySQL, database HBase, ecc.) da file csv
- **Database**: preparazione di una tabella dedicata del database MySQL per la scrittura dell'output finale
- **HBaseInput**: lettura dei dati real time dal database HBase (tabella dei dati raw) e scrittura degli stessi sulla tabella MySQL
- **create Report Triples**: modulo di generazione delle triple a partire dall'ontologia Sii-Mobility

Si riporta di seguito lo schema di dipendenza tra i sotto-moduli estratto da Spoon:



Per i dettagli dei singoli moduli ed i parametri di configurazione si rimanda all'analisi del codice del progetto.

2.3 Attività 5.4.3.a: taxi e car service [PETL22]

2.3.1 Analisi del tipo di servizio e del tipo di dati

Il progetto SII-Mobility prevede di mettere in grado il sistema di supporto alla mobilità SII-Mobility di fornire informazioni relative al servizio Taxi e Car Service. L'analisi delle informazioni relative al servizio taxi utili al sistema ha portato a definire le seguenti caratteristiche:

- dati statici
 - nome e descrizione del gestore
 - geolocalizzazione dei garage
 - numero di taxi
- dati dinamici (istantanei)
 - numero taxi disponibili
 - numero taxi disponibili per quartiere nelle varie fasce orarie (mattina, pomeriggio e notte)
 - numero taxi disponibili per quartiere
 - numero di taxi impegnati in trasporto
- dati statistici, ovvero informazioni dinamiche aggregate su definiti slot temporali:
 - consistenza Flotte Taxi
 - medie mensili di disponibilità flotte
 - medie giornaliere di disponibilità flotte
 - passeggeri trasportati annualmente
 - passeggeri trasportati mensilmente
 - passeggeri trasportati giornalmente
 - numero mensile di trasporti extraurbani
 - numero giornaliero di trasporti extraurbani
 - media numero dei taxi disponibili giornalmente
 - media di taxi disponibili per quartiere nelle varie fasce orarie (mattina, pomeriggio e notte)

Di seguito si riporta lo schema di JSON strutturale da considerare come riferimento per la categoria dei gestori di taxi. Tale schema rimane come riferimento di base sia nel caso di nuovi gestori sia nel caso di gestori già affiliati e può essere soggetto a customizzazione. Al variare della modalità di inserimento del dato (manuale, automatizzato via API, open-data) è possibile immaginare una differenziazione fra dati statici e dati dinamici e statistici. La struttura JSON proposta include informazioni statiche, dinamiche e statistiche.

TAXI SERVICE [QuestIT - Midra] Ver 0.1

```
{
  "taxi": [{
    "id": 10,
    "name": "Taxi service name",
    "description": "Taxi service description",
    "address": "Parking address",
    "lat": 43.314272222222,
    "long": 11.333538888889,
    "taxi_engaged": 20,
    "taxi_available_general": {
      "now": 30,
      "morning": 30,
      "afternoon": 30,
      "night": 30
    },
    "taxi_available_district": [{
      "district_id": "District id",
```



```
        "district_name": "District name",
        "now": 30,
        "morning": 30,
        "afternoon": 30,
        "night": 30
    }},
    "statistics": {
        "daily_taxi_available_average": 20.5,
        "taxi_fleet_consistency": 1.0,
        "daily_fleets_availability_average": 20.5,
        "monthly_fleets_availability_average": 200.5,
        "daily_transported_passengers_average": 20.5,
        "monthly_transported_passengers_average": 200.5,
        "annually_transported_passengers_average": 2000.5,
        "daily_suburban_transport_average": 12.5,
        "monthly_suburban_transport_average": 200.5
    },
    "status": 80
}
}
```

2.3.2 Integrazione con i gestori di Taxi

La positiva conclusione di tale attività richiede la disponibilità da parte dei gestori del servizio taxi di rendere fruibili dal sistema SII-Mobility una serie di informazioni e di dati (sia statistici che istantanei) disponibili presso i sistemi informatici di gestione delle flotte. È necessario mettere in evidenza che la disponibilità dei gestori a mettere a comune una serie di dati sensibili rappresenta la vera problematica del task. Alle società di gestione dei Taxi dovrà essere garantita la riservatezza dei dati grezzi provenienti dai loro sistemi di gestione nonché la sicurezza del dato e la sicurezza dell'accesso ai loro server. All'interno del progetto SII-Mobility sono stati individuati diversi gestori di taxi del territorio Toscana. In alcuni casi sono stati contattati per coinvolgerli chiedendo loro una manifestazione di interesse alla partecipazione al progetto priva di alcun onere mentre in altri casi i dati sono già disponibili nel sistema SII-Mobility centrale in quanto recuperati attraverso precedenti integrazioni. In particolare, si ha la seguente lista:

- Gestori i cui dati sono già disponibili:
 - Piazzole taxi FIRENZE
- Gestori contattati dal partner del progetto QuestIT :
 - CO.TAS Taxi Siena (QuestIT)
- Gestori contattati dal partner del progetto MIDRA :
 - SOCOTA Taxi (MIDRA)
 - COTAFI Taxi Firenze (MIDRA)

2.3.2.1 Gestori contattati dal partner del progetto QuestIT: CO.TAS Taxi Siena

E' stata inviata una email alla amministrazione in cui si illustra il progetto SII-Mobility e si chiede una manifestazione di interesse da parte di CO.TAS Taxi Siena alla partecipazione al progetto fornendo l'accesso ai dati, senza alcun onere per il gestore. A seguito di telefonata, il desk della amministrazione ci ha fatto sapere che la richiesta è all'ordine del giorno del prossimo Consiglio Di Amministrazione.

2.3.2.2 Gestori contattati dal partner del progetto MIDRA: SOCOTA Taxi Firenze

La procedura implementata è stata la seguente:

- a) E' stata effettuata una telefonata alla sede della cooperativa in cui è stata richiesta la modalità di contatto e richiesta la persona da contattare onde poter illustrare il progetto le sue finalità e la

possibilità di chiedere una partecipazione della stessa al progetto. A seguito di tale telefonata abbiamo avuto l'indicazione di contattare via mail la presidenza della cooperativa (presidenza@socota.it)

- b) A seguito della telefonata sono state inviate all'attenzione della Presidenza tre email illustrando il progetto e la richiesta di valutare una possibile collaborazione le seguenti Mail:
- 19/01/2017 @ presidenza
 - 26/01/2017 @ presidenza
 - 10/02/2017 @presidenza

Purtroppo non abbiamo ricevuto risposta.

Abbiamo provveduto a ricontattare la SOCOTA telefonicamente e ci hanno confermato che la procedura da seguire è quella di scrivere alla presidenza.

Pertanto siamo in attesa di ricevere una qualsiasi risposta da parte di detta Cooperativa.

2.3.2.3 Gestori contattati dal partner del progetto MIDRA: COTAFI Taxi Firenze

La procedura implementata è stata la seguente:

- a) E' stata effettuata una telefonata alla sede della cooperativa in cui è stata richiesta la modalità di contatto e richiesta la persona da contattare onde poter illustrare il progetto le sue finalità e la possibilità di chiedere una partecipazione della stessa al progetto. A seguito di tale telefonata abbiamo avuto l'indicazione di inviare apposita mail ai seguenti indirizzi: f.scarpelli@4390.it, info@4390.it)
- b) A seguito della telefonata sono state inviate email all'attenzione del Sig. Scarpelli e della generica casella di posta info@4390.it in cui veniva illustrato il progetto e la richiesta di valutare una possibile collaborazione. Le email sono state inviate nelle seguenti date
- 19/01/2017 @ scarpelli, @info
 - 26/01/2017 @ scarpelli, @info
 - 10/02/2017 @ scarpelli, @info

ma purtroppo non abbiamo ricevuto risposta. Abbiamo provveduto a ricontattare la COTAFI telefonicamente e ci ha confermato che la procedura da seguire è quella di scrivere agli indirizzi indicati. Pertanto siamo in attesa di ricevere una qualsiasi risposta da parte di detta Cooperativa.

2.3.3 Stato della attività di integrazione con i gestori di servizi di taxi e car service

GESTORE	AREA GEO	TIPOLOGIA SERVIZIO	PARTNER DI RIFERIMENTO	LIVELLO AVALLO AMMINIST.	AVALLO UFFICIO TECNICO	NOTE	DATI STATICI	DATI DINAMICI	METODOLOGIA INPUT DATI [manuale, API, open-data]
CO.TAS Taxi Siena	Siena	Taxi	QuestIT	In attesa di avallo	In attesa di avallo	-	V	X	API
Piazzole taxi FI	Firenze	Taxi	SII-Mobility	dati già gestiti	OK	-	V	X	open-data
SOCOTA	Firenze	Taxi	MIDRA	In attesa di risposta	In attesa di risposta	-	V	X	API
COTAFI	Firenze	Taxi	MIDRA	In attesa di risposta	In attesa di risposta	-	V	X	API

2.3.4 Attività di realizzazione ETL

2.3.4.1 COTAS – Siena (Quest-IT)

Il gestore non ha fornito alcun avallo all'uso dei dati nel progetto SII-Mobility e lo sviluppo dell'ETL non è ancora potuto iniziare.

2.3.4.2 SOCOTA – Firenze (Midra)

Si conferma nel presente paragrafo che il gestore, nonostante reiterati tentativi (l'ultimo effettuato in data 30/03/2017 mediante apposita mail), non ha fornito la disponibilità a collaborare al progetto rendendo disponibile allo stesso i dati necessari alla prosecuzione dell'attività.

Si precisa che al gestore è stato richiesto, telefonicamente, di formalizzare la non volontà / impossibilità a rendere disponibili i dati richiesti ma non è stato possibile ottenere tale dichiarazione. Pertanto non è stato possibile di avviare le attività necessarie allo sviluppo della relativa ETL

E' stata valutata brevemente la possibilità di creare un data base in grado di emulare la mancata fornitura dei dati da parte del gestore. Emulazione che teoricamente potrebbe consentire lo sviluppo della relativa ETL.

Tuttavia tale soluzione presenta tutta una serie di controindicazioni prima fra tutti il fatto che i dati essendo sintetizzati non rappresentano una situazione reale in quanto tale soluzione potrebbe ben simulare i dati statici mentre risulterebbe estremamente difficoltoso ricreare dei credibili set di dati che variano dinamicamente.

Altro grosso problema presentato da tale approccio è l'elevato costo in termini di risorse umane e finanziarie che sarebbe necessari sostenere da parte del singolo partner per finalizzare un tale tipo di emulazione.

Tale considerazione assieme alle controindicazioni sopra illustrate rende inapplicabile tale approccio alternativo.

2.3.4.3 COTAFI – Firenze (Midra)

Si conferma nel presente paragrafo che il gestore, nonostante reiterati tentativi (l'ultimo effettuato in data 30/03/2017 mediante apposita mail), non ha fornito la disponibilità a collaborare al progetto rendendo disponibile allo stesso i dati necessari alla prosecuzione dell'attività.

Si precisa che al gestore è stato richiesto, telefonicamente, di formalizzare la non volontà / impossibilità a rendere disponibili i dati richiesti ma non è stato possibile ottenere tale dichiarazione. Pertanto non è stato possibile di avviare le attività necessarie allo sviluppo della relativa ETL

E' stata valutata brevemente la possibilità di creare un data base in grado di emulare la mancata fornitura dei dati da parte del gestore. Emulazione che teoricamente potrebbe consentire lo sviluppo della relativa ETL.

Tuttavia tale soluzione presenta tutta una serie di controindicazioni prima fra tutti il fatto che i dati essendo sintetizzati non rappresentano una situazione reale in quanto tale soluzione potrebbe ben simulare i dati statici mentre risulterebbe estremamente difficoltoso ricreare dei credibili set di dati che variano dinamicamente.

Altro grosso problema presentato da tale approccio è l'elevato costo in termini di risorse umane e finanziarie che sarebbe necessari sostenere da parte del singolo partner per finalizzare un tale tipo di emulazione.

Tale considerazione assieme alle controindicazioni sopra illustrate rende inapplicabile tale approccio alternativo.

2.4 Attività 5.4.3b Logistica Merci [PETL19]

2.4.1 Analisi del tipo di servizio e del tipo di dati

Il modulo relativo la logistica merci mette in grado il sistema di supporto alla mobilità SII-Mobility di ottenere dai corrieri nazionali registrati sulla piattaforma SII-Mobility e dai vari operatori della logistica dell'ultimo miglio (padroncini), anche essi registrati sulla piattaforma SII-Mobility, le informazioni necessarie relative al servizio di recapito e ritiro di piccoli colli o plichi in ambito urbano e metropolitano.

Più in particolare, le informazioni acquisite dai Corrieri riguardano le consegne da effettuare giornalmente in una determinata area urbana (es. Firenze) e queste vengono elaborate sotto forma di liste di consegna/ritiro e fornite ai padroncini, a ciascuno dei quali è assegnata una zona (CAP) nella quale svolgerà le consegne/ritiri per conto di tutti i corrieri.

Il padroncino provvederà poi ad inviare la "spunta" delle consegne effettuate e delle mancate consegne, che verranno girate ai diversi Corrieri.

Il Modulo PETL19 è in particolare responsabile della acquisizione dei dati da parte dei Corrieri e dell'inoltro agli esiti degli esiti comunicati da ciascun padroncino.

La positiva conclusione di tale attività richiede la disponibilità da parte di ambedue le categorie di operatori a rendere fruibili dal sistema SII-Mobility una serie di informazioni statiche e dinamiche, che nel caso dei Corrieri provengono dai sistemi *Legacy* degli stessi, mentre nel caso dei padroncini sono fornite tramite apposito Modulo dell'APP SII Mobility (MAPP14).

Di seguito sono sintetizzati le tipologie di dati gestite.

Dati statici

- Anagrafica Corriere
 - Identificativo ID del corriere
 - Ragione sociale
 - P.IVA
 - Indirizzo sede legale
 - 1° Indirizzo magazzino
 - 2° Indirizzo magazzino
 - 3° Indirizzo magazzino
 - ...
- Anagrafica corriere ultimo miglio
 - Identificativo ID
 - Ragione Sociale
 - Indirizzo
 - P.IVA
 - Legale Rappresentante
- Anagrafica mezzi corriere ultimo miglio
 - ID Mezzo (targa)
 - Tipo mezzo (portata q.li, nr. Porte, 3 ruote, 2 ruote, bici)
 - Dimensioni
 - Capacità massima in termini di peso e volume
 - Tipo di carburante gasolio, benzina, gas, ibrido, elettrico
 - Tipologia di Euro: 0,1,2,3,4,5,6
- Anagrafica autisti corriere ultimo miglio:
 - ID autista (es. patente)
 - Cognome
 - Nome
 - Telefono cell.

- Tipo rapporto (titolare, dipendente, contratto ...)
- Data inizio rapporto

Come dati dinamici, si tratta di liste di colli da consegnare, per ciascuno dei quali verranno forniti i seguenti dati:

- Informazioni per singolo collo
 - Nome destinatario
 - Cognome destinatario
 - Indirizzo destinatario
 - CAP destinatario
 - Tipo di servizio (ritiro o consegna)
 - Peso
 - Volume
 - Orario (del ritiro/consegna, in tipologie fisse es: entro 9:00, 9:30, 10:00, 10:30, 12:00,18:00)
 - ID Lettera di vettura
 - BAR CODE del collo
 - Ritirato/Consegnato/Ritirato con riserva (*)
 - ID del corriere (*)
 - Data e ora del mancato ritiro/consegna presso utente finale (*)
 - Causa mancato ritiro/consegna presso utente finale (*)

Le informazioni evidenziate con (*) sono relativa alle informazioni che il padroncino fornisce al sistema tramite APP Mobile. Questo modulo include :

- il sistema di acquisizione dati statici tramite opportuni servizi WEB
- il sistema di acquisizione dati dinamici da parte dei corrieri tramite flussi XML gestiti tramite Pentaho
- il sistema di acquisizione dati dinamici da parte dei padroncini tramite APP mobile

Di seguito si riporta lo schema di JSON strutturale da considerare come riferimento per la categoria dei gestori di car service e logistica. Tale schema rimane come riferimento di base sia nel caso di nuovi gestori sia nel caso di gestori già affiliati e può essere soggetto a customizzazione. Al variare della modalità di inserimento del dato (manuale, automatizzato via API, open-data) è possibile immaginare una differenziazione fra dati statici e dati dinamici.

LOGISTICS [QuestIT - Time] Ver 0.1

```
{
  "logistics": [{
    "id": 10,
    "name": "Delivery name",
    "description": "Delivery description",
    "recipient_address": "Recipient address",
    "lat": 43.314272222222,
    "long": 11.333538888889,
    "parcels_to_be_delivered": [{
      "parcel_id": 10,
      "parcel_name": "Parcel name",
      "parcel_description": "Parcel description"
    }],
    "vehicle": {
      "vehicle_id": 3,
      "vehicle_characteristics": "Characteristics of the vehicle",
      "available_volume": "Available volume",
      "limitations": "Limitations on weight / volume of parcels"
    }
  ]
}
```

```

    "schedule_constraints": "Details of schedule constraints",
    "status": 80,
    "info": "Optional info"
  }}
}

```

2.4.2 Integrazione con i gestori di Logistica Merci

Il servizio presuppone la sottoscrizione di accordi di cooperazione sia da parte degli operatori ultimo miglio che da parte dei corrieri, che rappresenta la vera criticità del servizio stesso. L'integrazione con i gestori di logistica merci ed in particolare con i Corrieri è assai problematica, sia sotto il profilo amministrativo ed organizzativo, a causa delle forti chiusure e resistenze da parte di tali operatori a condividere informazioni riguardanti il loro servizio, che sotto il profilo tecnico, in quanto i sistemi *legacy* di tali operatori sono spesso obsoleti e sempre molto disomogenei tra di loro per quantità e qualità delle informazioni gestite.

Allo scopo di definire accordi di cooperazione, TIME si è rivolta in prima battuta ad ASSO.To.SCA, l'associazione di categoria relativa, allo scopo di impostare un accordo quadro che potesse essere sottoscritto se non da tutti da alcuni dei principali operatori.

Tuttavia gli incontri effettuati hanno evidenziato le difficoltà e le resistenze prima ricordate e l'Associazione ha declinato ogni possibilità di intervenire presso i propri associati.

Si sono così attivati contatti informali (favoriti da precedenti esperienze di collaborazione) con due operatori:

- BRT
- Barsanti

Parallelamente si sono avviati contatti formali con tali operatori e con altri del settore, prefigurando la sottoscrizione di convenzioni bilaterali, che sono attualmente all'esame degli organi delle Società, con tempi di valutazione ed eventuale approvazione che si prefigurano molto lunghi, almeno rispetto alle scadenze di progetto.

Per tale ragione, ed allo scopo di poter procedere con le fasi di implementazione e con le successive fasi di sperimentazione, verranno utilizzati i data set disponibili per i due operatori predetti, normalizzati attraverso la costruzione di un subset di dati comuni per quanto riguarda la gestione dei colli ed utilizzati tramite strumenti di simulazione, cioè generando flussi di dati simulati.

2.4.3 Stato della attività di integrazione con i gestori di servizi di taxi e car service

GESTORE	AREA GEO	TIPOLOGIA SERVIZIO	PARTNER DI RIFERIMENTO	LIVELLO AVALLO AMMINISTRATIVO	AVVALLO UFFICIO TECNICO	NOTE	DATI STATICI	DATI DINAMICI	METODOLOGIA INPUT DATI [manuale, API, open-data]
ASSO.To.S.C.A.	Toscana	Corrieri e Autotrasportatori	Time	nessuno	N.A.	-	N.A.	N.A.	N.A.
BRT	Nazionale	Corriere	Time	In attesa di avallo	In attesa di avallo	-	V	V	API
BARZANTI	Toscana	Corriere	Time	In attesa di avallo	In attesa di avallo	-	V	V	API

2.4.4 Attività di realizzazione ETL

In assenza di una manifestazione di interesse e di un avallo all'utilizzo dei dati da parte dei gestori, sono stati realizzati i relativi ETL anonimizzando i nomi dei gestori stessi.

2.4.4.1 ASSO Tosca (Time)

Asso.To.S.C.A. è una associazione di categoria che come tale non gestisce dati specifici, né servizi, ma esclusivamente gli aspetti associativi degli operatori. Pertanto non fornisce dati in input.

2.4.4.2 Corriere01 (Time)

2.4.4.2.1 Dati input

I dati acquisiti dal Corriere01 sono catalogati come dati statici per quanto riguarda la parte di anagrafica del corriere e dei magazzini presenti sul territorio di interesse mentre, per quanto riguarda le informazioni relative ai colli da gestire, questi sono stati catalogati come dati real-time. La suddivisione è data dal fatto che la prima classe di dati prevede eventuali variazioni a lungo termine mentre la seconda classe di dati variano giornalmente.

I dati relativi all'anagrafica del corriere contengono tutte le informazioni necessarie per l'individuazione su mappa della sede e altre informazioni aggiuntive di contorno utili per gli utenti finali, come, per esempio, numero telefonico, numero fax ecc. I dati vengono acquisiti tramite file .txt per limitare un impatto lato Corrieri nell'utilizzo della piattaforma.

I dati associati ai colli contengono tutte le informazioni base per eseguire la consegna o il recupero dello stesso, come ad esempio barcode, indirizzo di consegna/ritiro, orario ecc.

NOME CAMPO	DESCRIZIONE
Key	Identificativo del magazzino
RagioneSociale	Ragione Sociale del Corriere
PartitaIVA	Partita IVA del Corriere
Indirizzo	Indirizzo del magazzino
Longitudine	Longitudine del magazzino
Latitudine	Latitudine del magazzino
CAP	CAP del magazzino
Città	Città del magazzino
Provincia	Provincia del magazzino
NumeroTelefonico	Numero telefonico del magazzino
NumeroFAX	Numero FAX del magazzino
IndirizzoSedeLegale	Indirizzo della sede del Corriere
email	Email del Corriere
website	Website del Corriere
NumeroTelefonicoSedeLegale	Numero telefonico del Corriere
NumeroFAXSedeLegale	Numero FAX del Corriere

2.4.4.2.2 Dati statici: schema e descrizione ETL

Sono stati progettati e realizzati tre ETL dedicati alle tre fasi di recupero, quality improvement e trasformazione dei dati:

- Ingestion – recupero dei dati, processing, selezione ed infine memorizzazione interna;
- QI – modulo di quality improvement del dato acquisito nello step di ingestion;
- Triplication – trasformazione del dato in una tripla conforme alla ontologia specifica di

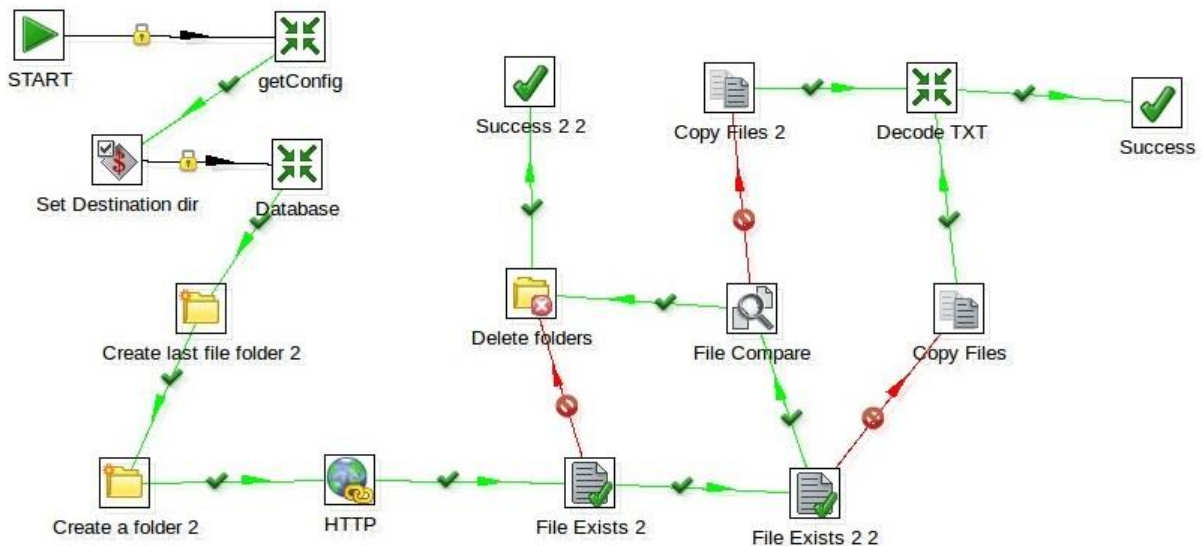
SIIMobility (Km4City).

2.4.4.2.2.1 Ingestion

Per quanto riguarda i dati statici, sotto riportiamo gli screenshot del relativo ETL.

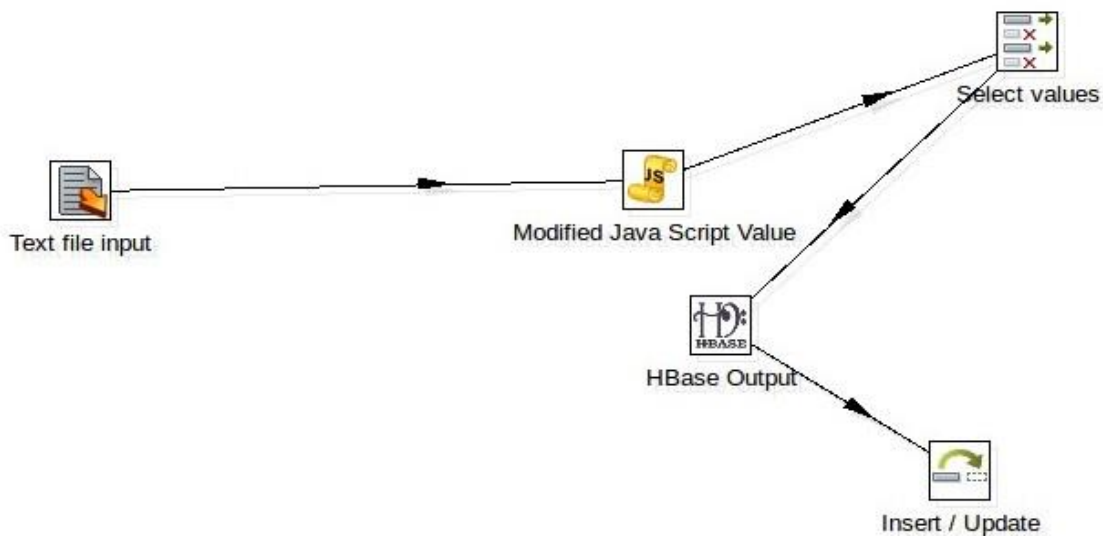


Nel main richiamiamo tante volte l'esecuzione del Job "Static_Corrieri" quanti sono i Corrieri.



Il Job prevede i sottomoduli principali:

- getConfig: lettura dei parametri di configurazione da csv
- Database: lettura dei parametri di processo dal database MySql
- Decode TXT (job Ingestion): elaborazione dei file txt ricevuti in input.



Questa trasformazione è personalizzata per il Corriere01. Il modulo “Text file input” recupera i campi del file separati da una ‘,’ per poi essere mappati con quelli da noi previsti per l’HBase. I campi memorizzati sono riportati nella seguente figura:

Select & Alter		Remove	Meta-data
Fields :			
#	Fieldname		
1	key		
2	RagioneSociale		
3	PartitaIVA		
4	Nome		
5	Indirizzo		
6	Longitudine		
7	Latitudine		
8	CAP		
9	Citta		
10	Provincia		
11	NumeroTelefonico		
12	NumeroFAX		
13	actualdate		
14	timestamp		
15	process		
16	indirizzoSedeLegale		
17	email		
18	website		
19	numeroTelefonicoSedeLegal		
20	numeroFAXSedeLegale		

2.4.4.2.2.2 Quality Improvement

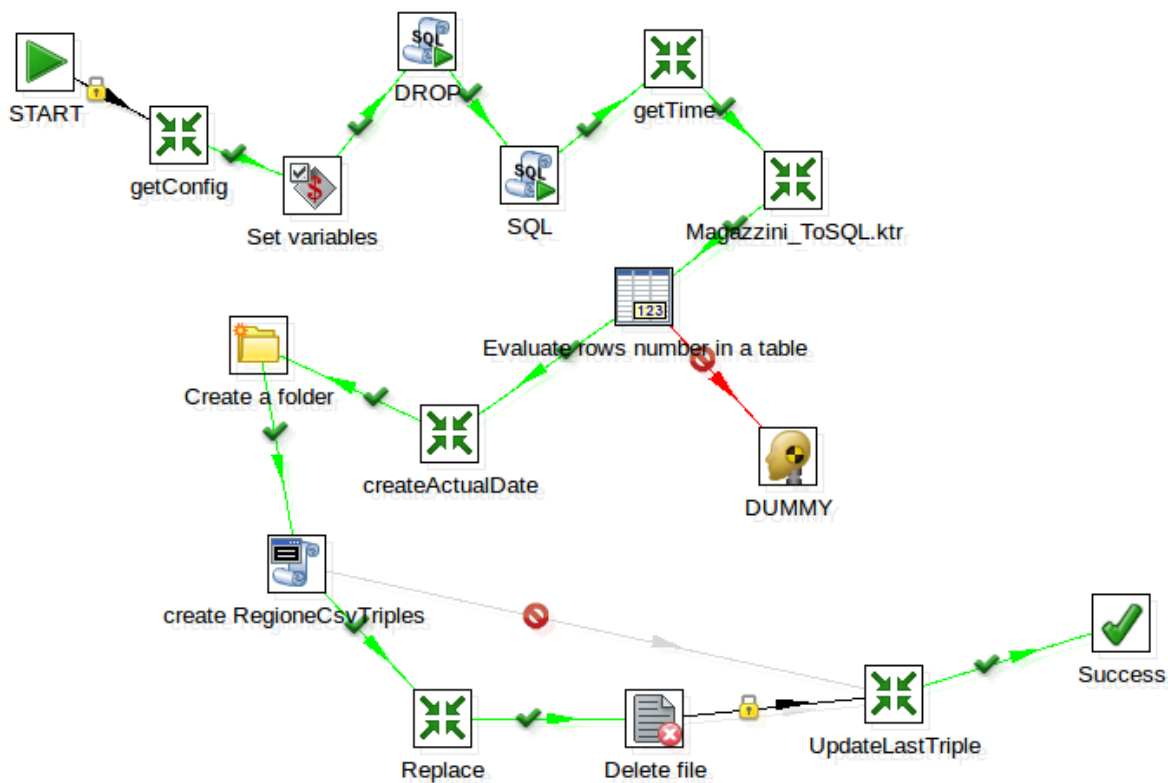
Il modulo di Quality Improvement è costituito dai seguenti sotto-moduli principali (vedi screenshot):

- getConfig: lettura dei parametri di configurazione da file csv
- Corrieri (Trasformation QI): questa trasformazione permette
 - la lettura dei dati statici dal database HBase (HBase input)
 - operazioni di quality improvement (Manipulations)
 - la scrittura dei dati statici sul database HBase (HBase output)



2.4.4.2.2.3 Triplification

In questa fase abbiamo mappato i dati memorizzati in HBase in triple appartenenti alle ontologie utilizzate da km4City.



Il modulo è costituito da quattro sotto-moduli principali:

- getConfig: lettura dei parametri di configurazione da file csv
- Database: preparazione di una tabella dedicata del database MySql per la scrittura dell'output finale
- HBaseInput: lettura dei dati statici dal database HBase (tabella dei dati QI) e scrittura degli stessi sulla tabella MySql
- create RegionesCsv Triples: modulo di generazione delle triple a partire dall'ontologia Sii-

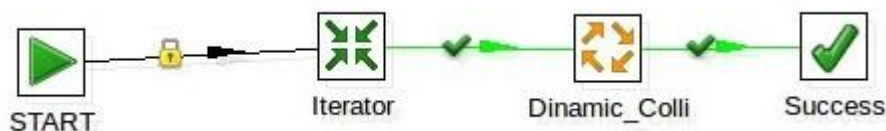
Mobility

2.4.4.2.3 Dati real time: schema e descrizione ETL

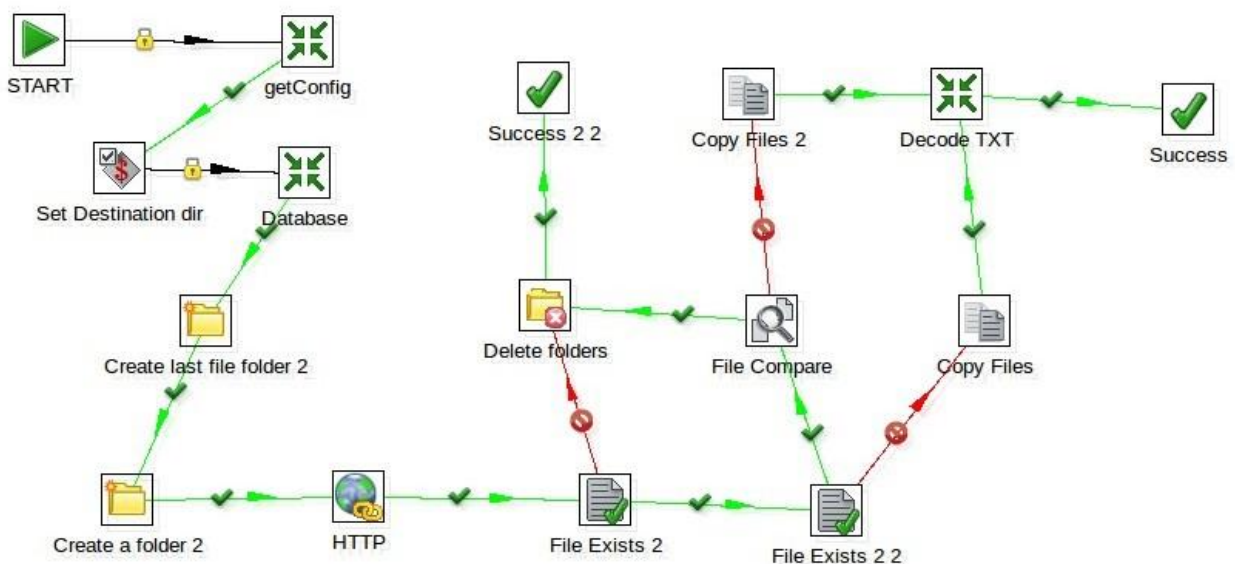
Sono stati progettati e realizzati due ETL dedicati alle due fasi di recupero e trasformazione dei dati real time:

- Ingestion – recupero dei dati, normalizzazioni, processing, selezione ed infine memorizzazione interna;
- Triplification – trasformazione del dato in una tripla conforme alla ontologia specifica di SIIMobility (Km4City).

2.3.4.2.3.1 Ingestion



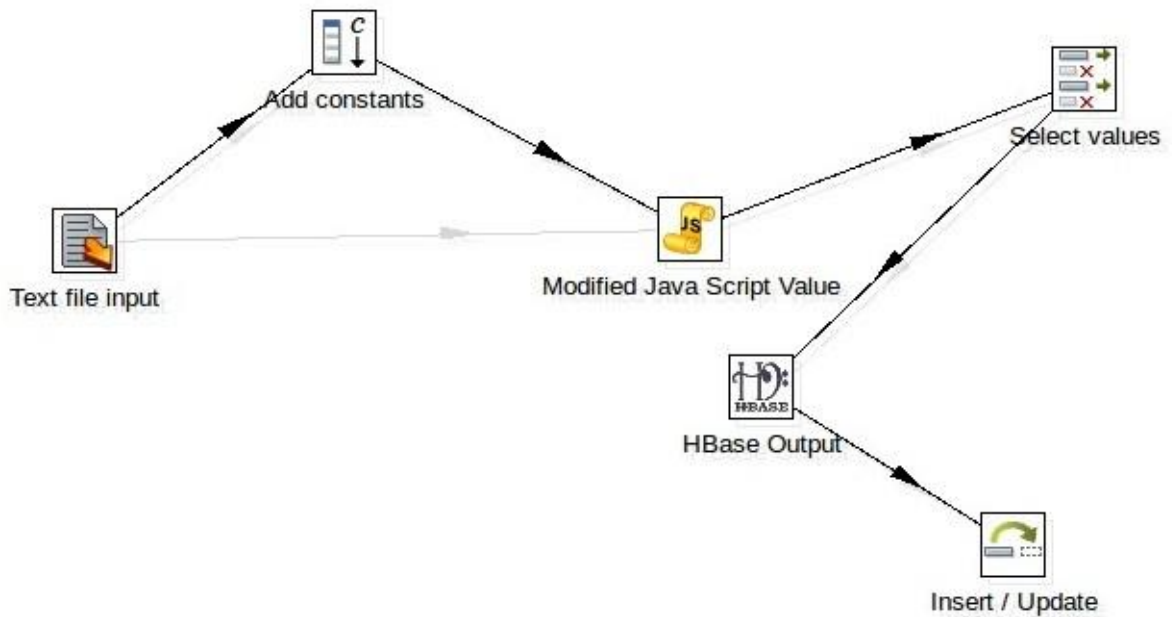
Nel main richiamiamo tante volte l'esecuzione del Job "Dinamic_Colli" quanti sono i Corrieri che necessitano del servizio (in un singolo txt sono presenti più ordini ed ogni corriere ha tipologie diverse di ordini).



Il Job prevede i sottomoduli principali:

- getConfig: lettura dei parametri di configurazione da csv
- Database: lettura dei parametri di processo dal database MySql

- Decode TXT (job Ingestion): elaborazione dei file txt ricevuti in input relativi ai Colli.



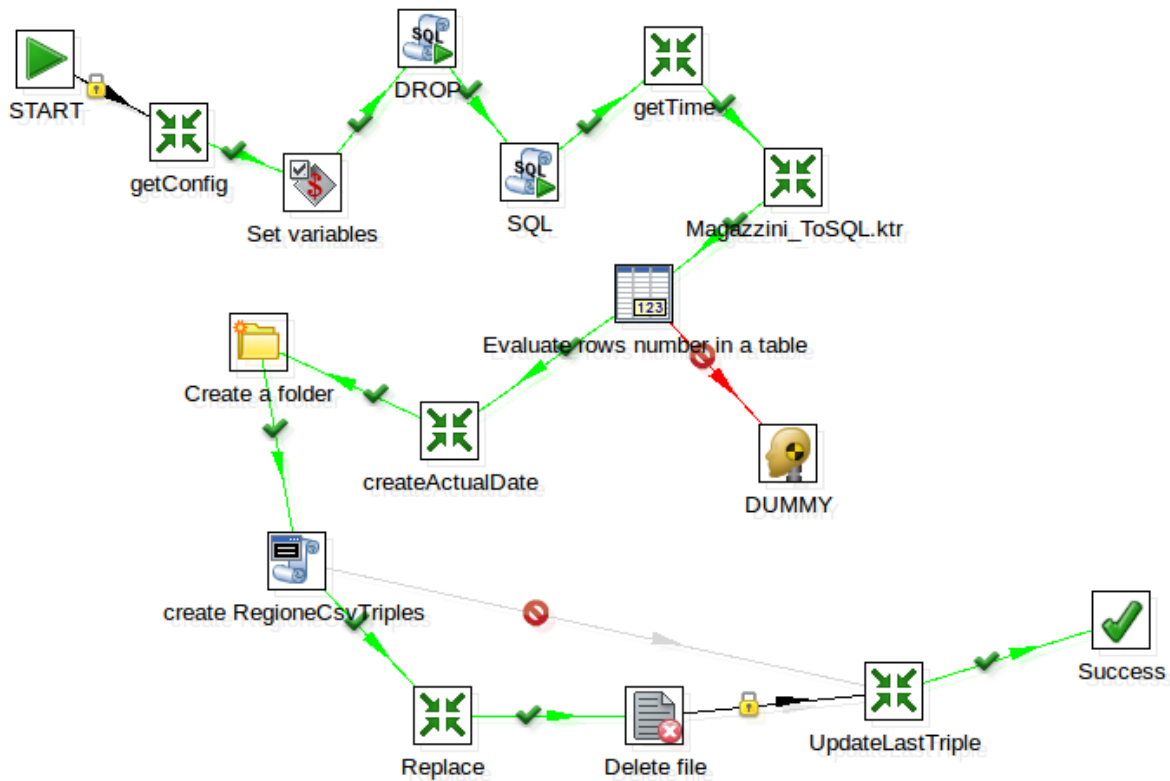
Nella fase di analisi abbiamo circoscritto un set di dati ritenuti utili per il servizio merci e logistica, alcuni dei quali indipendenti dal particolare Corriere (e.g. codiceAutista, objectType ecc.). I dati presenti nel file di input dei Corrieri sono stati mappati su questi dati selezionati cercando di raggiungere un insieme omogeneo e completo di dati da inserire nel DB. In questo modo, tutti i Colli dei Corrieri saranno caratterizzati dagli stessi attributi.

Nello specifico, per il Corriere01 abbiamo aggiunto i seguenti campi:

#	Name	Type
1	CodiceFiscaleAutista	String
2	dataFineServizio	String
3	stato	String
4	objectType	String
5	destinatarioAlternativo	String
6	tipoSpedizione	String
7	istruzioniConsegnaAggiuntive	String

2.3.4.2.3.2 Triplification

In questa fase abbiamo mappato i dati memorizzati in HBase in triple appartenenti alle ontologie utilizzate da km4City.



Il modulo è costituito da quattro sotto-moduli principali:

- getConfig: lettura dei parametri di configurazione da file csv
- Database: preparazione di una tabella dedicata del database MySql per la scrittura dell'output finale
- HBaseInput: lettura dei dati statici dal database HBase (tabella dei dati QI) e scrittura degli stessi sulla tabella MySql
- create RegioneCsv Triples: modulo di generazione delle triple a partire dall'ontologia Sii-Mobility

2.4.4.3 Corriere02 (Time)

2.4.4.3.1 Dati input

La procedura di acquisizione dei dati del Corriere02 segue la stessa logica prevista per il Corriere01 e descritta nel paragrafo "2.4.4.2.1 Dati input"

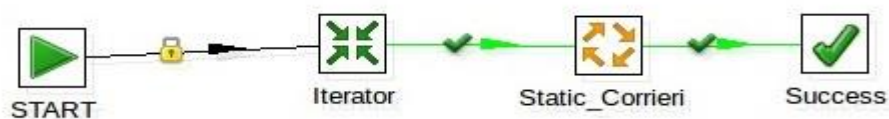
2.4.4.3.2 Dati statici: schema e descrizione ETL

Sono stati progettati e realizzati tre ETL dedicati alle tre fasi di recupero, quality improvement e trasformazione dei dati:

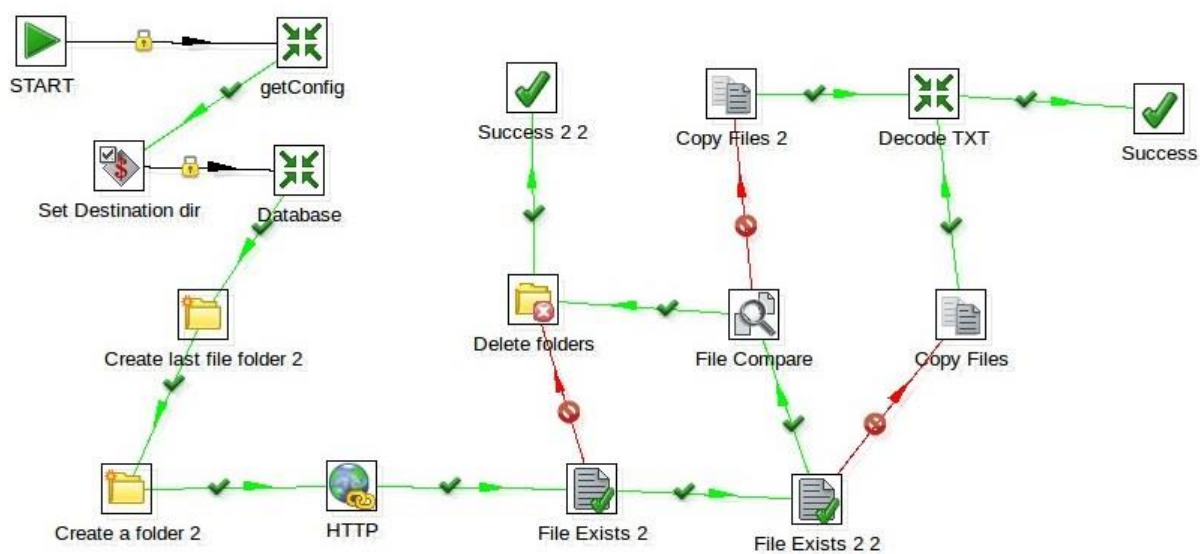
- Ingestion – recupero dei dati, processing, selezione ed infine memorizzazione interna;
- QI – modulo di quality improvement del dato acquisito nello step di ingestion;
- Triplication – trasformazione del dato in una tripla conforme alla ontologia specifica di SII-Mobility (Km4City).

2.4.4.3.2.1 Ingestion

Per quanto riguarda i dati statici, sotto riportiamo gli screenshot del relativo ETL.

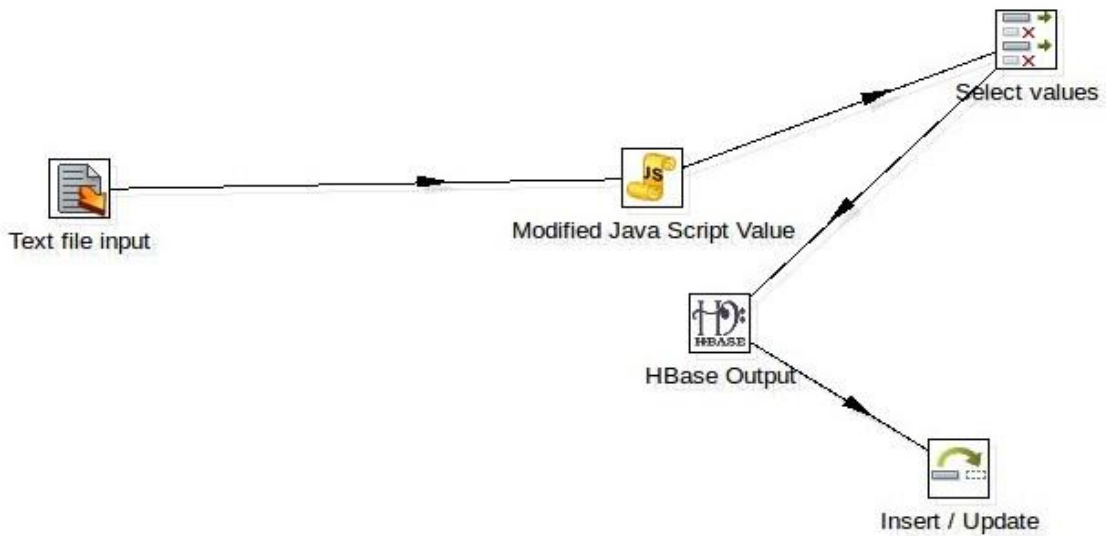


Nel main richiamiamo tante volte l'esecuzione del Job "Static_Corrieri" quanti sono i Corrieri.



Il Job prevede i sottomoduli principali:

- getConfig: lettura dei parametri di configurazione da csv
- Database: lettura dei parametri di processo dal database MySql
- Decode TXT (job Ingestion): elaborazione dei file txt ricevuti in input.



Questa trasformazione è personalizzata per il Corriere02. Il modulo “Text file input” recupera i campi del file separati da una ‘,’ per poi essere mappati con quelli da noi previsti per l’HBase. I campi memorizzati sono riportati nella seguente figura:

Select & Alter		Remove	Meta-data
Fields :			
#	Fieldname		
1	key		
2	RagioneSociale		
3	PartitaIVA		
4	Nome		
5	Indirizzo		
6	Longitudine		
7	Latitudine		
8	CAP		
9	Citta		
10	Provincia		
11	NumeroTelefonico		
12	NumeroFAX		
13	actualdate		
14	timestamp		
15	process		
16	indirizzoSedeLegale		
17	email		
18	website		
19	numeroTelefonicoSedeLegal		
20	numeroFAXSedeLegale		

2.4.4.3.2.2 Quality Improvement

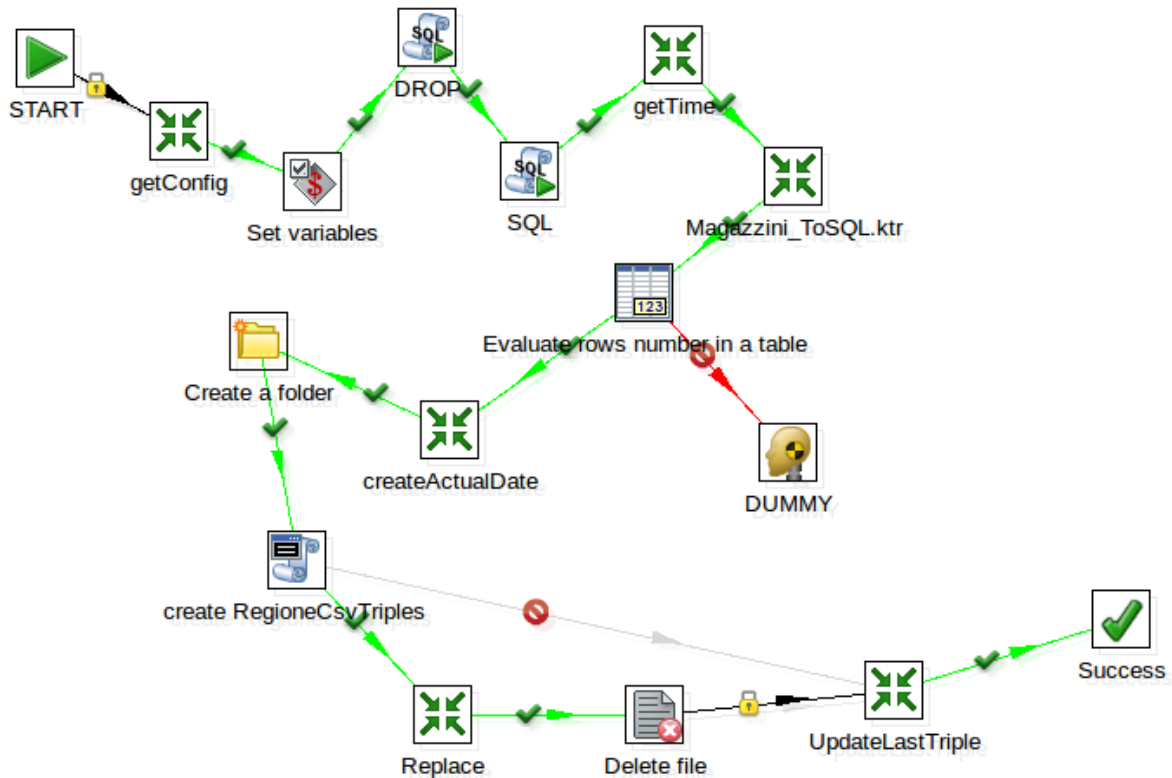


Il modulo di Quality Improvement è costituito dai seguenti sotto-moduli principali (vedi screenshot):

- getConfig: lettura dei parametri di configurazione da file csv
- Corrieri (Trasformazione QI): questa trasformazione permette
 - la lettura dei dati statici dal database HBase (HBase input)
 - operazioni di quality improvement (Manipulations)
 - la scrittura dei dati statici sul database HBase (HBase output)

2.4.4.3.2.3 Triplification

In questa fase abbiamo mappato i dati memorizzati in HBase in triple appartenenti alle ontologie utilizzate da km4City.



Il modulo è costituito da quattro sotto-moduli principali:

- getConfig: lettura dei parametri di configurazione da file csv
- Database: preparazione di una tabella dedicata del database MySql per la scrittura dell'output finale
- HBaseInput: lettura dei dati statici dal database HBase (tabella dei dati QI) e scrittura degli stessi sulla tabella MySql
- create RegionesCsv Triples: modulo di generazione delle triple a partire dall'ontologia Sii-Mobility

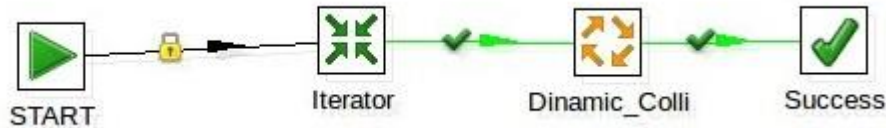
2.4.4.3.3 Dati real time: schema e descrizione ETL

Sono stati progettati e realizzati due ETL dedicati alle due fasi di recupero e trasformazione dei dati real time:

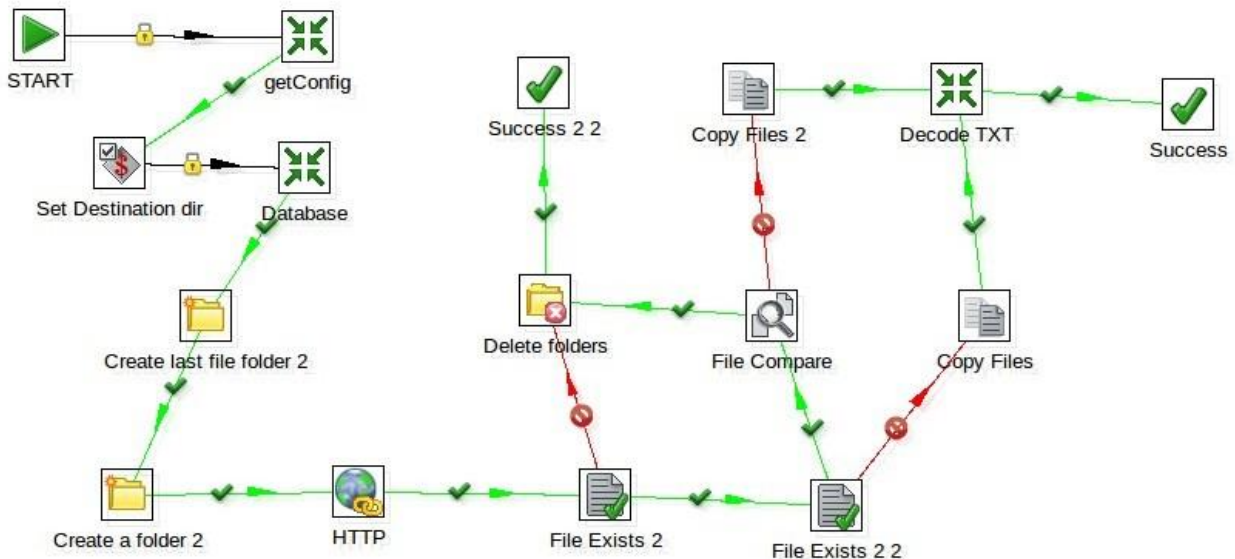
- Ingestion – recupero dei dati, normalizzazioni, processing, selezione ed infine memorizzazione interna;
- Triplification – trasformazione del dato in una tripla conforme alla ontologia specifica di SIIMobility (Km4City).

2.4.4.3.3.1 Ingestion

Lo schema del moduli Ingestion è il seguente.

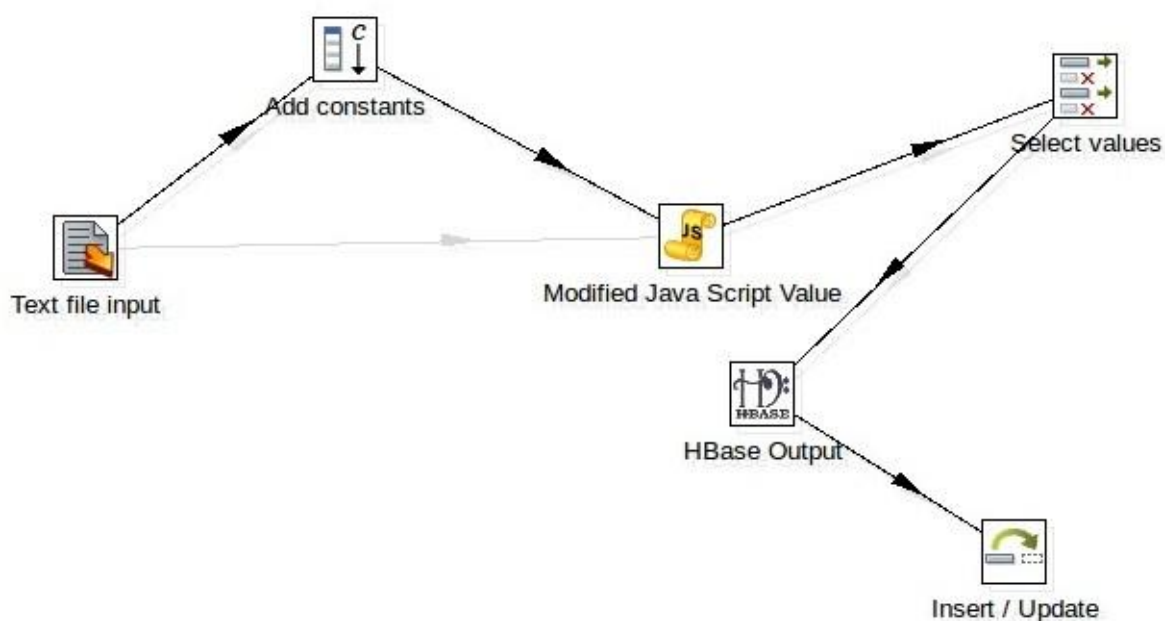


Nel main richiamiamo tante volte l'esecuzione del Job "Dinamic_Colli" quanti sono i Corrieri che necessitano del servizio (in un singolo txt sono presenti più ordini ed ogni corriere ha tipologie diverse di ordini).



Il Job prevede i sottomoduli principali:

- getConfig: lettura dei parametri di configurazione da csv
- Database: lettura dei parametri di processo dal database MySql
- Decode TXT (job Ingestion): elaborazione dei file txt ricevuti in input relativi ai Colli.



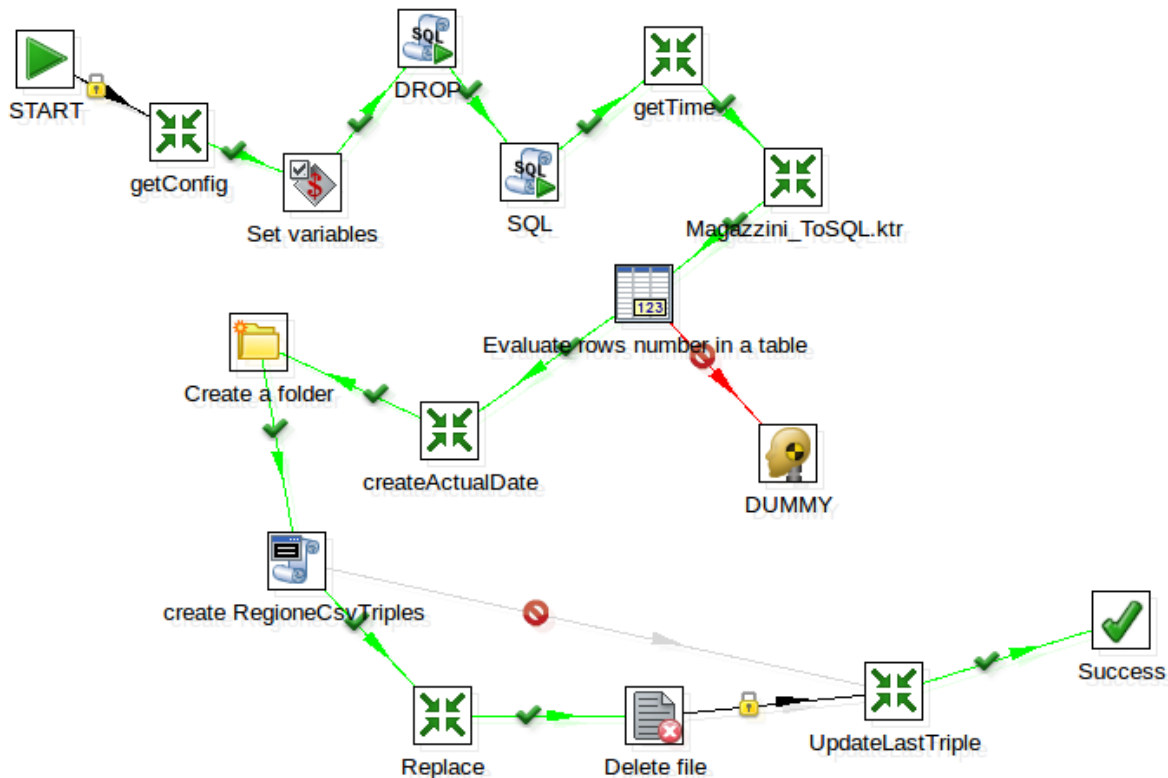
Come per il Corriere01, per raggiungere un insieme omogeneo e completo di dati da inserire del DB, abbiamo aggiunto dei campi:

Fields :

#	Name	Type
1	dataFineServizio	String
2	codiceAutista	String
3	stato	String
4	oraConsegnaRichiesta	String
5	objectType	String
6		

2.4.4.3.3.2 Triplification

In questa fase abbiamo mappato i dati memorizzati in HBase in triple appartenenti alle ontologie utilizzate da km4City.



Il modulo è costituito da quattro sotto-moduli principali:

- getConfig: lettura dei parametri di configurazione da file csv
- Database: preparazione di una tabella dedicata del database MySql per la scrittura dell'output finale
- HBaseInput: lettura dei dati statici dal database HBase (tabella dei dati QI) e scrittura degli stessi sulla tabella MySql
- create RegioneCsv Triples: modulo di generazione delle triple a partire dall'ontologia Sii-Mobility

2.5 Attività 5.4.5: nettezza e servizi [PETL23]

2.5.1 Analisi del tipo di servizio e del tipo di dati

Lo scopo dell'attività è quello di mettere in condizioni il sistema di supporto alla mobilità SII-Mobility di acquisire dal gestore dei servizi di igiene ambientale le informazioni necessarie ad integrare la mole dei dati necessari all'ottimizzazione della gestione della mobilità sia all'interno delle aree urbane sia fra le aree urbane e quelle extraurbane. È infatti evidente che le attività condotte dal gestore dei servizi di igiene ambientale impattano pesantemente sulla mobilità in termini di zone di occupazione della sede stradale, temporanei divieti di sosta o di parcheggio, rallentamento del traffico. L'analisi delle informazioni utilizzabili per questo tipo di servizio ha identificato come informazioni statiche:

- elenco giornaliero pulizia strade
- orario pulizia strade
- percorsi raccolta rifiuti
- orario raccolta rifiuti

e come dati dinamici:

- interventi straordinari

Di seguito si riporta lo schema di JSON strutturale da considerare come riferimento per la categoria dei gestori di taxi. Tale schema rimane come riferimento di base sia nel caso di nuovi gestori sia nel caso di gestori già affiliati e può essere soggetto a customizzazione. Al variare della modalità di inserimento del dato (manuale, automatizzato via API, open-data) è possibile immaginare una differenziazione fra dati statici e dati dinamici e statistici. La struttura JSON proposta include informazioni statiche, dinamiche e statistiche.

GARBAGE SERVICE [QuestIT - Midra] Ver 0.1

```
{
  "garbage": [{
    "id": 10,
    "name": "Garbage service name",
    "description": " Garbage service description",
    "address": "address",
    "lat": 43.314272222222,
    "long": 11.3335388888889,
    "consistency": 1.0,
    "daily_maid_roads_list": [{
      "road_id": 1,
      "road_name": "Road name",
      "maid_day": "Monday",
      "maid_time": "6.00-8.00"
    }],
    "waste_collection_routes_list": [{
      "route_id": 1,
      "route_name": "Route name",
      "route_roads": "List of roads",
      "route_day": "Monday, Friday",
      "route_time": "6.00-8.00"
    }],
    "extraordinary_measures": "Extraordinary measures"
  ]
}
```

2.5.2 Integrazione con i gestori di Nettezza e servizi

La positiva conclusione del presente task richiede la disponibilità da parte del gestore del servizio di igiene ambientale di rendere fruibili al sistema SII-Mobility una serie di informazioni e di dati (sia statistici che istantanei) disponibili presso i propri sistemi informatici. È necessario mettere in evidenza che la disponibilità dei gestori a mettere a comune una serie di dati sensibili rappresenta la vera problematica del task. All'interno del progetto SII-Mobility sono stati individuati diversi gestori di nettezza e servizi del territorio Toscano.

2.5.2.1 Gestori contattati dal partner del progetto QuestIT: SEI Toscana

Dal partner QuestIT è stata contattato la società SEI Toscana SPA, gestore del servizio di nettezza urbana per le provincie di Grosseto, Siena ed Arezzo. E' stata inviata una email al direttore tecnico di tale società in cui si illustra il progetto SII-Mobility e si chiede una manifestazione di interesse da parte di SEI Toscana alla partecipazione al progetto fornendo l'accesso ai dati, senza alcun onere per il gestore. Il responsabile si è dimostrato fortemente interessato al progetto e disponibile alla partecipazione ma tale decisione deve essere presa dal Consiglio Di Amministrazione della società. Purtroppo, recenti problemi giudiziari riguardanti l'intero consiglio di amministrazione ed alti

dirigenti di SEI Toscana hanno portato alla dimissioni dell'intero CDA e stanno portando la società verso il commissariamento esterno. Ciò sta quindi rallentando enormemente la valutazione da parte dell'organo competente della richiesta di partecipazione a SII-Mobility.

2.5.2.2 Gestori contattati dal partner del progetto MIDRA: Quadrifoglio

Il Partner MIDRA ha provveduto a contattare la Società Quadrifoglio Servizi Ambientali Area Fiorentina SpA responsabile dei servizi ambientali dell'Area Fiorentina. Si è provveduto ad effettuare una serie di telefonate per individuare un percorso per la presa di contatto con la società in modo da poter successivamente illustrare le finalità del progetto e successivamente richiedere la disponibilità a collaborare allo stesso. Dopo una serie di telefonate interlocutorie ci è stato suggerito di scrivere una mail al generico indirizzo quadrifoglio@quadrifoglio.org con le necessarie informazioni in modo da poter richiedere l'apertura di un contatto. Abbiamo pertanto provveduto ad inviare tale mail ma ancora non abbiamo ricevuto riscontro alla stessa.

E' tuttavia da considerare il fatto che per la sola parte di dati statici esiste sul sito della società la possibilità di ricavare le informazioni relativamente alle date in cui verranno effettuate le pulizie delle strade. Il sistema consente di selezionare il comune di interesse, la strada ed in base a tali informazioni viene ritornata la prima data per le prossime pulizie a la periodicità delle operazioni. Alternativamente è previsto che selezionando comune di interesse quartiere e data venga restituito l'elenco delle strade interessate al divieto di soste per l'esecuzione delle operazioni di pulizia. Risulta possibile quindi ricavare una parte delle informazioni necessarie anche in assenza della disponibilità della società ad attivare una canale di collaborazione.

2.5.3 Stato della attività di integrazione con i gestori di servizi di nettezza e servizi

GESTORE	AREA GEO	TIPOLOGIA SERVIZIO	PARTNER DI RIFERIMENTO	LIVELLO AVALLO AMMINISTRATIVO	AVALLO UFFICIO TECNICO	NOTE	DATI STATICI	DATI DINAMICI	METODOLOGIA INPUT DATI [manuale, API, open-data]
SEI	Siena, Arezzo, Grosseto	Servizi ecologici	QuestIT	In attesa di avallo	In attesa di avallo	-	V	X	API
Quadrifoglio	Firenze	Servizi ecologici	MIDRA	In attesa di risposta	in attesa di risposta	-	V	X	API

2.5.4 Attività di realizzazione ETL

2.5.4.1 SEI (QuestIT)

Il gestore non ha ad oggi fornito nessun avallo all'utilizzo dei dati e non sono disponibili informazioni di tipo OpenData. L'ETL per tale servizio non è al momento realizzabile.

2.5.4.2 Quadrifoglio (Midra)

Si conferma nel presente paragrafo che il gestore del servizio di igiene ambientale , nonostante

reiterati tentativi (l'ultimo effettuato in data 30/03/2017 mediante apposita mail), non ha fornito la disponibilità a collaborare al progetto rendendo disponibile allo stesso i dati necessari alla prosecuzione dell'attività.

Pertanto non è stato possibile di avviare le attività necessarie allo sviluppo della relativa ETL

Va sottolineato che in questo periodo è avvenuta la fusione per incorporazione di ASM SpA, Publiambiente SpA e CIS Srl in Quadrifoglio SpA che ha assunto la denominazione di Alia SpA.

In conseguenza di ciò a decorrere dal 13 marzo 2017 Alia SpA, ai sensi degli artt. 2504 e seguenti c.c. subentra a pieno titolo in tutto il patrimonio attivo e passivo ed in tutte le ragioni, azioni e diritti così come in tutti gli obblighi e passività di qualsiasi natura delle società partecipanti alla fusione alle scadenze e condizioni originariamente previste.

Alla luce di quanto sopra risulterà possibile tentare nuovamente di richiedere al nuovo soggetto giuridico la disponibilità a collaborare con il progetto.

3. Conclusioni

Il presente deliverable descrive le attività di studio e le scelte progettuali per l'integrazione dei gestori di servizi con il sistema SII-Mobility e le attività di coinvolgimento dei gestori ai fini di ottenere da essi i dati necessari. I servizi considerati sono:

- parcheggi;
- servizi di car & bike sharing;
- taxi e car-service;
- operatori della logistica merci;
- nettezza e servizi vari (es. spazzamento strade).

Per ogni tipologia di servizio, sono stati individuati tre categorie di dati:

- dati statici, ovvero informazioni sul servizio con variabilità temporale molto bassa e in molti casi nulla (es. capienza, percorsi, geolocalizzazione ecc...);
- dati dinamici, ovvero informazioni che variano in tempo reale (es. disponibilità di parcheggi, di auto per il noleggio, posizioni istantanee, ecc...);
- dati statistici, ovvero informazioni derivate da dati dinamici a seguito di analisi in determinati range temporali (es. medie, variazioni, ecc...).

Per il recupero dati, si possono prevedere due diversi metodi:

- “manuale”, ovvero la realizzazione di servizi in cui è richiesto l'intervento semi-manuale da parte dell'operatore del gestore del servizio per l'inserimento delle informazioni richieste: form web, caricamento di file in formati standard. Tale metodo è dedicato principalmente all'acquisizione di dati statici in quanto l'aggiornamento manuale di dati dinamici richiederebbe il continuo intervento dell'operatore.
- “automatico”, in cui è progettato e realizzato un software (ETL) direttamente connesso con il sistema informatico del gestore che legge i dati di interesse ed invia al sistema centrale. Tale metodo necessita di una stretta collaborazione con i sistemi software del gestore e richiede quindi una autorizzazione da parte del gestore del servizio.

Il documento riporta le analisi dettagliate svolte per ogni tipologia di servizio e propone i relativi tipi di dato individuati. Per ogni tipologia di servizio (task) sono quindi illustrati gli schemi JSON progettati, gli insiemi di dati statici e dinamici e le modalità di recupero (push e/o automatica).

Data la stretta dipendenza delle attività di integrazione con i sistemi informativi dei singoli gestori e quindi la conseguente necessità di autorizzazione esplicita da parte del gestore, sono riportate per

ogni task le attività messe in campo dai partner orientate alla individuazione di gestori del territorio toscano ed al loro coinvolgimento nel progetto. Tale attività è stata portata avanti tramite invio di email descrittive del progetto e delle sue finalità contenenti richieste di manifestazione di interesse alla partecipazione al progetto del tutto priva di oneri a carico del gestore stesso. Lo stato delle richieste di partecipazione e di integrazione inviate dai partner ai gestori individuati sono state riepilogate in tabelle suddivise per tipologia di servizio ed inserite nelle relative sezioni di competenza. Si riporta qui una tabella di sintesi dei gestori con i quali il partenariato ha cercato e sta cercando di giungere ad un accordo di affiliazione e con i quali i vari partner hanno definito o stanno definendo le specifiche tecniche di integrazione.

PROSPECT	AREA GEO	TIPOLOGIA SERVIZIO	PARTNER DI RIFERIMENTO
SIENA PARCHEGGI SPA	SIENA	PARCHEGGI IN STRUTTURA	QUESTIT
SISTEMA GROSSETO	GROSSETO	PARCHEGGI IN STRUTTURA	QUESTIT
BICINCITTA' (SIPEDALA)	SIENA	BIKE-SHARING	QUESTIT
SEI (SERVIZI ECOLOGICI INTEGRATI)	SIENA (AREZZO, GROSSETO)	NETTEZZA	QUESTIT
CO.TAS (TAXISIENA)	SIENA	TAXI	QUESTIT
SISTEMA GROSSETO (SISTEMA S.R.L.)	GROSSETO	PARCHEGGI IN STRUTTURA, ZTL CITTA'	QUESTIT
ASSO.To.S.C.A. (Associazione Toscana Corrieri e Autotrasportatori)	TOSCANA	CAR-SHARING, TRASPORTO E LOGISTICA	TIME
BRT	NAZIONALE	CORRIERE	TIME
BARSANTI SPA	TOSCANA	CORRIERE	TIME
PISAMO (PARCHEGGI)	PISA	PARCHEGGI IN STRUTTURA	LIBEROLOGICO
PARCHEGGI LUNGO STRADA	PISA	PARCHEGGI LUNGO STRADA	LIBEROLOGICO
PISAMO (BIKE SHARING)	PISA	BIKE-SHARING	QUESTIT
SOCOTA	FIRENZE	TAXI	MIDRA
COTAFI	FIRENZE	TAXI	MIDRA
FIRENZE PARCHEGGI	FIRENZE	PARCHEGGI IN STRUTTURA	SII-Mobility
PARCHEGGI LUNGO STRADA	FIRENZE	PARCHEGGI LUNGO STRADA	LIBEROLOGICO

ZCS Zone parcheggio controllato	FIRENZE	PARCHEGGIO CONTROLLATO	SII-Mobility
Parcheggi Toscana	TOSCANA	PARCHEGGI IN STRUTTURA	SII-Mobility
Percorsi ciclabili	FIRENZE	PERCORSI CICLABILI	SII-Mobility
Piazzole Taxi	FIRENZE	TAXI	SII-Mobility

Di seguito si riporta lo stato di avanzamento per la progettazione e realizzazione dei moduli ETL per le differenti attività. Sono evidenziati i casi in cui la realizzazione non è stata possibile in quanto:

- ancora mancante l'avallo del gestore all'utilizzo dei dati
- indisponibilità di un web-service per il recupero dei dati
- mancanza di dati in formato OpenData.

Attività 5.4.1 – Parcheggi Vari [PETL20]

Sono stati realizzati i moduli relativi a tutti i quattro gestori individuati:

- Sistema Grosseto (QuestIT) – realizzato ETL per l'interfacciamento con il sistema centrale dei dati relativi ai parcheggi monitorati del Comune di Grosseto;
- Siena Parcheggi (QuestIT) – realizzato ETL per l'interfacciamento con il sistema centrale dei dati relativi ai parcheggi monitorati del Comune di Siena;
- PisaMo (Liberologico) – realizzato ETL per l'interfacciamento con il sistema centrale dei dati relativi ai parcheggi lungostrada del comune di Pisa;
- Firenze Parcheggi (Liberologico) – realizzato ETL per l'interfacciamento con il sistema centrale dei dati relativi ai parcheggi lungostrada del comune di Firenze.

Attività 5.4.2 – Car and Bike Sharing [PETL21]

Sono stati realizzati i moduli ETL relativi ai due gestori individuati. In entrambi i casi, non avendo ricevuto l'avallo dai gestori, sono stati individuati ed utilizzati URL pubblici per ottenere i dati in formato xml:

- SIPedala (QuestIT) – realizzato ETL per l'interfacciamento con il sistema centrale dei dati relativi al servizio di BikeSharing del Comune di Siena.
- PisaMo (QuestIT) – realizzato il modulo ETL per l'interfacciamento con il sistema centrale dei dati relativi al servizio di bike sharing del comune di Pisa.

Attività 5.4.3a – Taxi e Car sharing [PETL22]

- COTAS (QuestIT) – servizio taxi nel comune di Siena. Non è stato possibile realizzare alcun modulo ETL in quanto ad oggi sono ancora mancanti le manifestazioni di interesse da parte del gestore e non sono disponibili OpenData.
- SOCOTA (Midra) – servizio taxi nel comune di Firenze. Non è stato possibile realizzare alcun modulo ETL in quanto ad oggi sono ancora mancanti le manifestazioni di interesse da parte del gestore.
- COTAFI (Midra) – servizio taxi nel comune di Firenze. Non è stato possibile realizzare alcun modulo ETL in quanto ad oggi sono ancora mancanti le manifestazioni di interesse da parte del gestore

Attività 5.4.3b – Logistica Merci [PETL19]

In assenza di una manifestazione di interesse e di un avallo all'utilizzo dei dati da parte dei gestori, sono stati realizzati i relativi ETL anonimizzando i nomi dei gestori stessi.

- Asso.To.S.C.A. (TIME) – AssoTosca è una associazione di categoria che come tale non

gestisce dati specifici, né servizi, ma esclusivamente gli aspetti associativi degli operatori. Pertanto non è possibile realizzare ETL.

- Corriere01 (TIME) – In assenza di avallo da parte del gestore all'utilizzo dei dati, è stato realizzato un ETL per l'interfacciamento dei dati con il sistema centrale SII-Mobility anonimizzando il nome del gestore stesso;
- Corriere02 (TIME) – In assenza di avallo da parte del gestore all'utilizzo dei dati, è stato realizzato un ETL per l'interfacciamento dei dati con il sistema centrale SII-Mobility anonimizzando il nome del gestore stesso;

Attività 5.4.5 – Nettezza e servizi [PETL23]

- SEI Toscana (QuestIT) – gestore del servizio di nettezza urbana di Grosseto, Siena ed Arezzo. Non è stato possibile realizzare alcun modulo ETL in quanto ad oggi sono ancora mancanti le manifestazioni di interesse da parte del gestore e non sono disponibili OpenData.
- Quadrifoglio (Midra) – gestore privato di nettezza urbana in area Firenze. Non è stato possibile realizzare alcun modulo ETL in quanto ad oggi sono ancora mancanti le manifestazioni di interesse da parte del gestore

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
- 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
- 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