

Knowledge mining and Semantic Models: from Cloud to Smart City

x Dottorato DIST, Univ. Firenze

Pierfrancesco Bellini, Paolo Nesi

DISIT Lab

Dipartimento di Ingegneria dell'Informazione, DINFO

Università degli Studi di Firenze

Via S. Marta 3, 50139, Firenze, Italy

Tel: +39-055-2758511, fax: +39-055-2758570

<http://www.disit.dinfo.unifi.it> *alias*

<http://www.disit.org>

Pierfrancesco.bellini@unifi.it , Paolo.nesi@unifi.it

Knowledge mining and Semantic Models: from Cloud to Smart City

part 1: Ontology Engineering

x Dottorato DIST, Univ. Firenze

Pierfrancesco Bellini, Paolo Nesi

DISIT Lab

Dipartimento di Ingegneria dell'Informazione, DINFO

Università degli Studi di Firenze

Via S. Marta 3, 50139, Firenze, Italy

Tel: +39-055-2758511, fax: +39-055-2758570

<http://www.disit.dinfo.unifi.it> *alias*

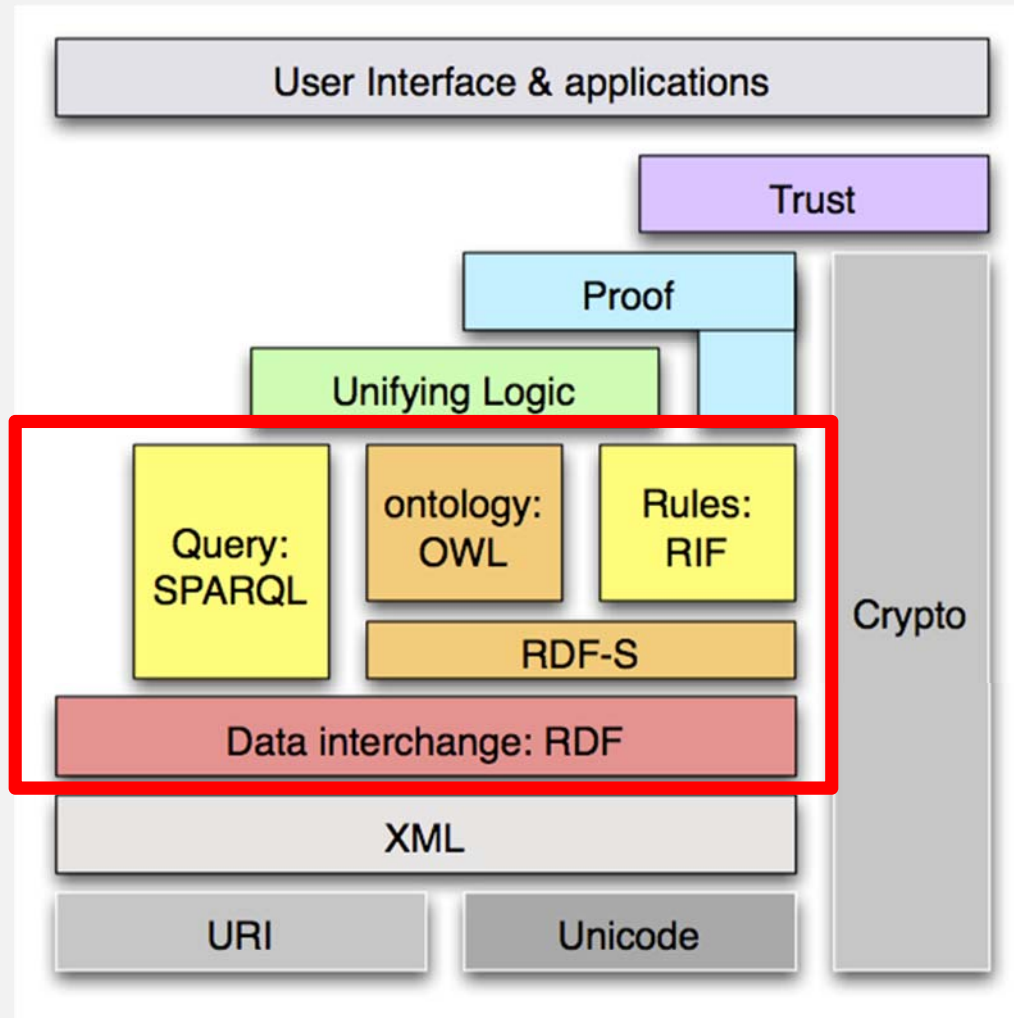
<http://www.disit.org>

Pierfrancesco.bellini@unifi.it , Paolo.nesi@unifi.it

Major topics addressed

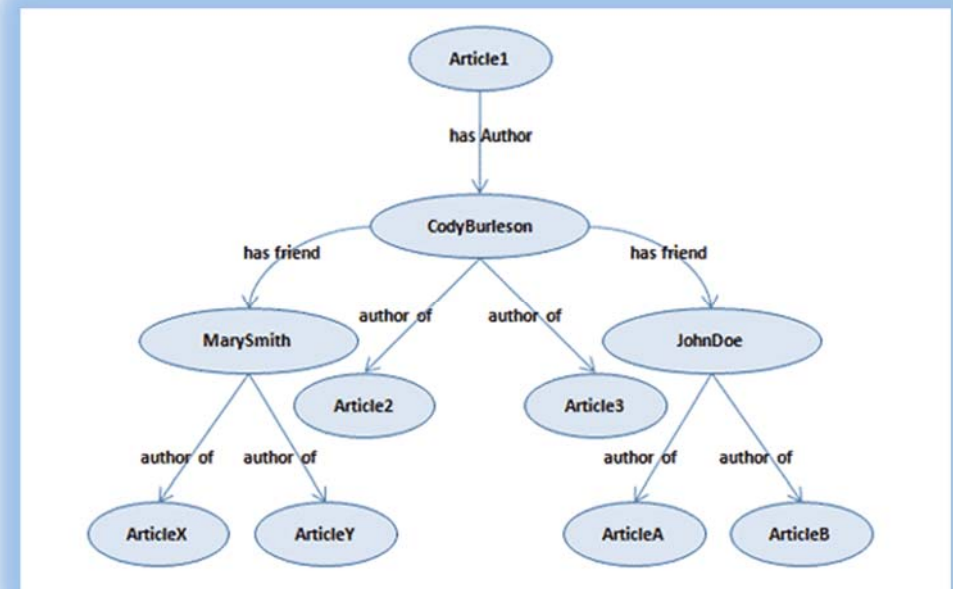
- From RDF to OWL ←
- Knowledge engineering for Beginners
- Smart Cloud Application (ICARO Case)
- Big Data Smart City Architecture
- Smart-city Ontology
- Data Ingestion and Mining
- Distributed and real time processes
- RDF processing
- Smart City Engine
- Development Interfaces
- Sii-Mobility

W3C Semantic Web



RDF

- Resource Description Framework
- Graph based
- Triples/Quadruples:
 - [Context]
 - Subject
 - Predicate
 - Object
- Everything identified by URI
- Many serialization formats: RDF/XML, Turtle, NTriples



W3C SPARQL 1.1

- Graph matching query language

```
SELECT ?x ?y ?h WHERE {  
  ?x rdf:type foaf:Person.  
  ?x foaf:knows ?y.  
  OPTIONAL {  
    ?x foaf:homepage ?h.  
  }  
}
```

Ontology

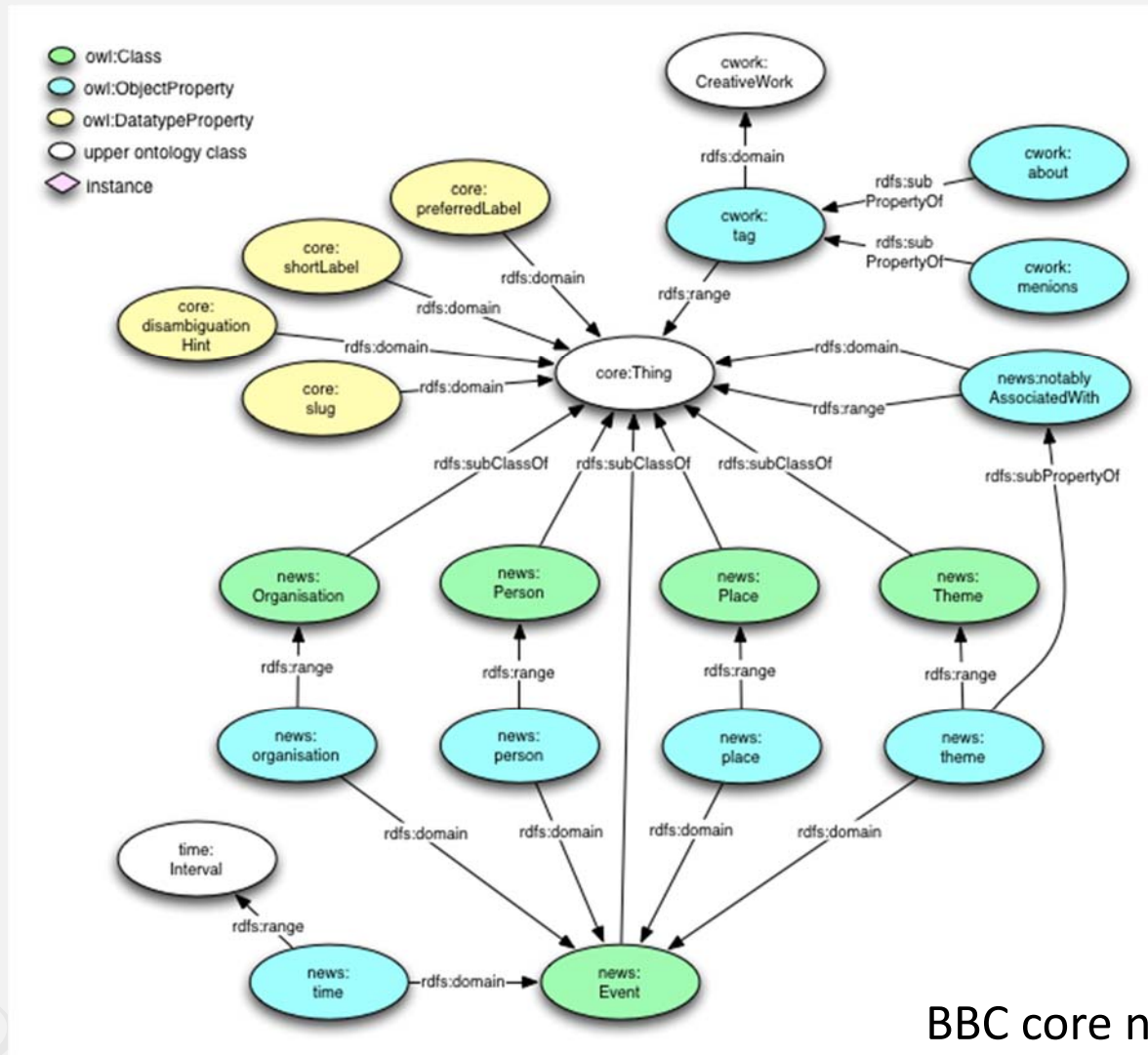
- An ontology is a formal explicit description of:
 - **Concepts/classes** in a domain of discourse,
 - **Properties/roles/slot** of each concept describing various features and attributes of the concept,
 - and restrictions on slots
- An ontology together with a set of individual instances of classes constitutes a **knowledge base**

Ontology

- In practical terms, developing an ontology includes:
 - defining classes in the ontology,
 - arranging the classes in a taxonomic (subclass–superclass) hierarchy,
 - defining properties and describing allowed values for these properties,
 - filling in the values for properties for instances.



Ontology Example



BBC core news data model

Why Ontologies?

- Why would someone want to develop an ontology? Some of the reasons are:
 - To share common understanding of the structure of information among people or software agents.
 - To enable reuse of domain knowledge.
 - To make domain assumptions explicit.
 - To separate domain knowledge from the operational knowledge
 - To analyze domain knowledge

Classification

- Foundational/Top level/Upper Level Ontologies
 - Generic ontologies applicable to many domains (DOLCE, BFO, ...)
- General Ontologies
 - Not dedicated to a specific domain (OpenCyc)
- Core reference ontologies
 - A standard used by different groups of users.
- Domain Ontologies
 - Applicable to a specific domain with a specific viewpoint.
- Local or Application Ontologies
 - Specific for a single user/application view point

Classification

- **Information Ontologies**
 - MindMap
- **Linguistic/Terminological Ontologies**
 - Thesauri, taxonomies (SKOS)
- **Software Ontologies**
 - UML, ER
- **Formal Ontologies**
 - OWL, Description Logic, FOL

W3C - OWL2

- The W3C language to define ontologies
- Many operators:
 - subClassOf, equivalentClass, disjointClasses
 - subObjectPropertyOf, domain, range
 - Union, Intersection
 - someValuesFrom, allValuesFrom, value
 - maxCardinality, minCardinality, exactCardinality
 - oneOf
 - inverseProperty, symmetric/asymmetricProperty, reflexive/irreflexiveProperty, functional/inverseFunctionalProperty, transitiveProperty

– ...

OWL/RDF Assumptions

- Due to the distributed nature of OWL and RDF,
Anyone can say Anything about Anything
 - **Open World Assumption (OWA)**
 - If something it is not explicitly stated or derived we cannot say it is true/false
 - **Not Unique Name Assumption**
 - The same resource can be identified with different URI



Major topics addressed

- From RDF to OWL
- Knowledge engineering for Beginners ←
- Smart Cloud Application (ICARO Case)
- Big Data Smart City Architecture
- Smart-city Ontology
- Data Ingestion and Mining
- Distributed and real time processes
- RDF processing
- Smart City Engine
- Development Interfaces
- Sii-Mobility

Ontology Engineering

- Many methodologies for Ontologies Engineering
 - METHONTOLOGY, NeOn, Cyc, etc.
 - A review in «An Analysis of Ontology Engineering Methodologies: A Literature Review»
- N. F. Noy, D. L. McGuinness, “Ontology Development 101: A Guide to Creating Your First Ontology”
 - http://protege.stanford.edu/publications/ontology_development/ontology101.pdf

Ontology Engineering

1. There is no one correct way to model a domain—there are always viable alternatives. The best solution almost always depends on the application that you have in mind and the extensions that you anticipate.
2. Ontology development is necessarily an iterative process.
3. Concepts in the ontology should be close to objects (physical or logical) and relationships in your domain of interest. These are most likely to be nouns (objects) or verbs (relationships) in sentences that describe your domain

Step 1

- What is the domain that the ontology will cover?
- For what we are going to use the ontology?
- For what types of questions the information in the ontology should provide answers?
- Who will use and maintain the ontology?



Competency questions

- One of the ways to determine the scope of the ontology is to sketch a list of questions that a knowledge base based on the ontology should be able to answer, **competency questions**.



Step 2. Reuse

- Consider reusing existing ontologies. It is almost always worth considering what someone else has done and checking if we can refine and extend existing sources for our particular domain and task.
 - Linked Open Vocabularies <http://lov.okfn.org/dataset/lov/>
 - Schema.org <http://schema.rdfs.org/>
 - Dbpedia ontology
 - ...

Step 3. Enumerate terms

- Enumerate important terms in the ontology. It is useful to write down a list of all terms we would like either to make statements about or to explain to a user.
 - What are the terms we would like to talk about?
 - What properties do those terms have?
 - What would we like to say about those terms?



Step 4. Define classes and class hierarchy

- There are several possible approaches in developing a class hierarchy:
 - Top-down
 - Bottom-up
 - A combination
- From the list created in Step 3, we select the terms that describe objects having independent existence rather than terms that describe these objects. These terms will be classes in the ontology and will become anchors in the class hierarchy.
- Organize the classes into a hierarchical taxonomy by asking if by being an instance of one class, the object will necessarily (i.e., by definition) be an instance of some other class.

Step 5. Define the properties of classes

- The classes alone will not provide enough information to answer the competency questions from Step 1.
- Once we have defined some of the classes, we must describe the internal structure of concepts.
- We have already selected classes from the list of terms we created in Step 3. Most of the remaining terms are likely to be properties of these classes.
- In general, there are several types of object properties that can become slots in an ontology:
 - “intrinsic” properties such as the flavor of a wine;
 - “extrinsic” properties such as a wine’s name, and area it comes from;
 - parts, if the object is structured; these can be both physical and abstract “parts” (e.g., the courses of a meal)
 - relationships to other individuals; these are the relationships between individual members of the class and other items

Step 6. Define the facets of the properties

- Properties can have different facets describing the value type, allowed values, the number of the values (cardinality), and other features of the values the property can take
- **Property cardinality** defines how many values a property can have. Some systems distinguish only between single cardinality (allowing at most one value) and multiple cardinality (allowing any number of values).
- **Props-value type** A value-type facet describes what types of values can fill in the property.
- **Domain and range of a property**
 - When defining a domain or a range for a property, find the most general classes or class that can be respectively the domain or the range for the properties .
- **Define Property hierarchy**
 - One property can be a sub property of another

Step 7. Create instances

- The last step is creating individual instances of classes in the hierarchy.
- Defining an individual instance of a class requires:
 - choosing a class,
 - creating an individual instance of that class, and
 - filling in the properties values.



Step 8. Test

- Use a tool (e.g. Protegè) to validate the ontology to see if there are inconcistencies.
- Check the inferred statements to see if they make sense.
- Try to make the «competencies query» over the KB using SPARQL and check the result.



Some Tips

- All the siblings in the hierarchy (except for the ones at the root) must be at the same level of generality
- If a class has only one direct subclass there may be a modeling problem or the ontology is not complete.
- If there are more than a dozen subclasses for a given class then additional intermediate categories may be necessary
- Subclasses of a class usually have additional properties that the superclass does not have, or restrictions different from those of the superclass, or participate in different relationships than the superclasses
- Classes in terminological hierarchies do not have to introduce new properties
- ...

Ontology Design Patterns

- Ontology Design Patterns and good practices
 - <http://ontologydesignpatterns.org>
 - <http://www.gong.manchester.ac.uk/odp/html/>
 - <http://www.mkbergman.com/911/a-reference-guide-to-ontology-best-practices/>
 - <http://www.w3.org/2001/sw/BestPractices/OEP/>
- Interesting also the patterns for Linked Data
 - Leigh Dodds, Ian Davis, «Linked Data Patterns»
<http://patterns.dataincubator.org>

Ontology Life-cycle

- Similar to Software life-cycle
 - Waterfall
 - Requirements → Design → Test → Maintenance
 - Iterative
 - Requirements₁ → Design₁ → Test₁ →
Requirements₂ → Design₂ → Test₂ →
...



Major topics addressed

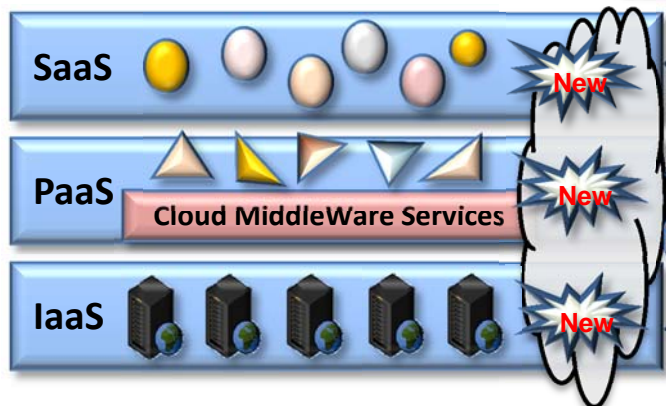
- From RDF to OWL
- Knowledge engineering for Beginners
- Smart Cloud Application (ICARO Case) ←
- Big Data Smart City Architecture
- Smart-city Ontology
- Data Ingestion and Mining
- Distributed and real time processes
- RDF processing
- Smart City Engine
- Development Interfaces
- Sii-Mobility



Architettura ICARO

Utenza Finale

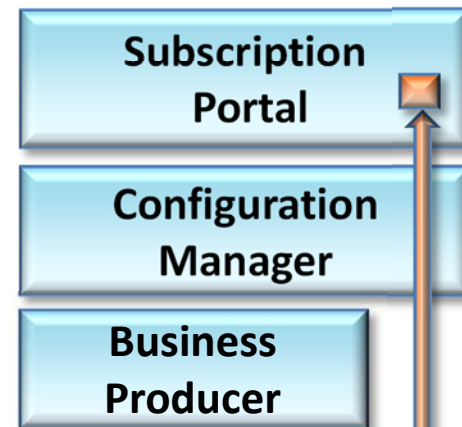
Application Access on
iCaro cloud



PMI

Access to BPaaS,
Services Purchase

SLA



Supervisor & Monitor

PMI-ICT

Developers
PaaS

CMW SDK

Smart Cloud

Knowledge Base

Cloud
Management

Cloud Simulator



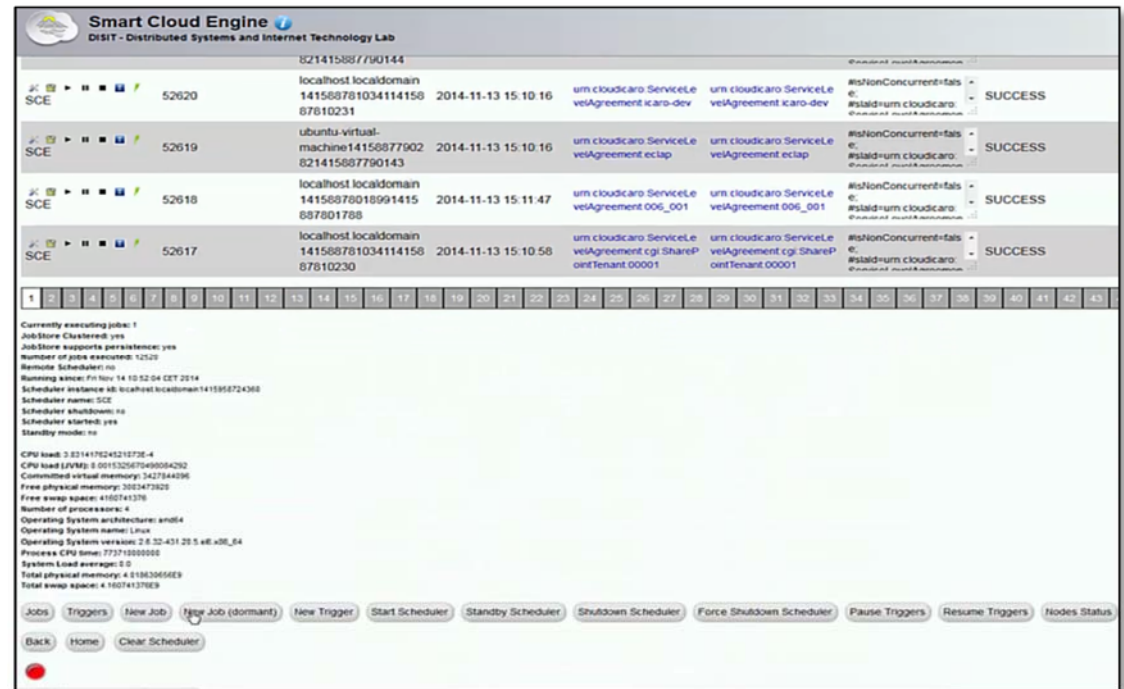
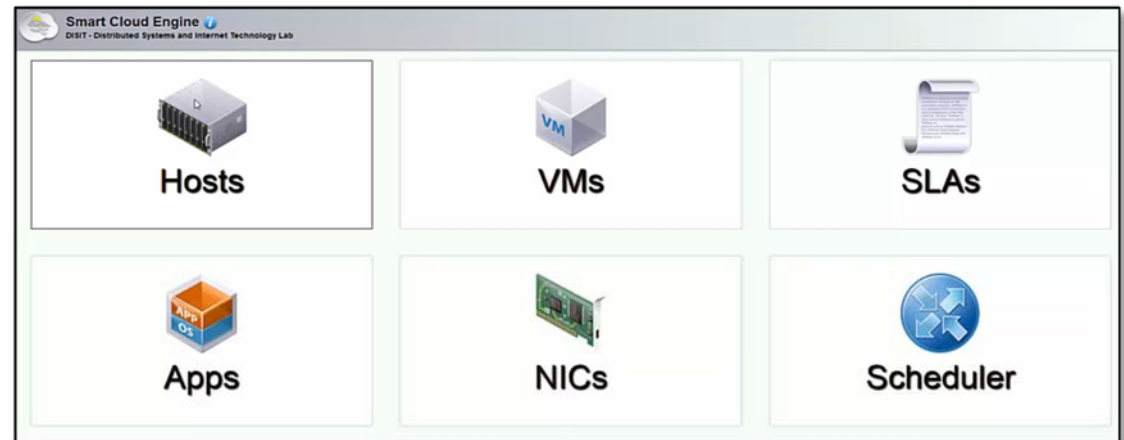
- ❑ **Gestisce Processi di Smart Cloud per:**
 - ❑ Il Configuration Manager, al quale comunica i risultati di analisi dello stato di salute ed eventuali situazioni di allarme, etc.
 - ❑ monitoraggio e identificazione attiva di situazioni critiche che possono dover produrre riconfigurazioni, allarmi, revisioni di contratto, etc., a livello di: Host, VM, SLA, Business, etc.
 - ❑ supporto alle decisioni come la generazione di suggerimenti, a fronte di simulazioni, e previsioni, anche tramite Cloud Simulator
- ❑ **Lo Smart Cloud usa la Knowledge Base che**
 - ❑ configura in modo automatico i moduli di monitoraggio e supervisione, che rimangono totalmente trasparenti per l'Service Portal, Configuration Manager e Business Producer.



Smart Cloud Engine

□ Motore di Cloud intelligence

- algoritmi di ottimizzazione della gestione del cloud
- algoritmi per il monitoraggio smart del comportamento di servizi e applicazioni: IaaS, PaaS, SaaS, BPaaS !!

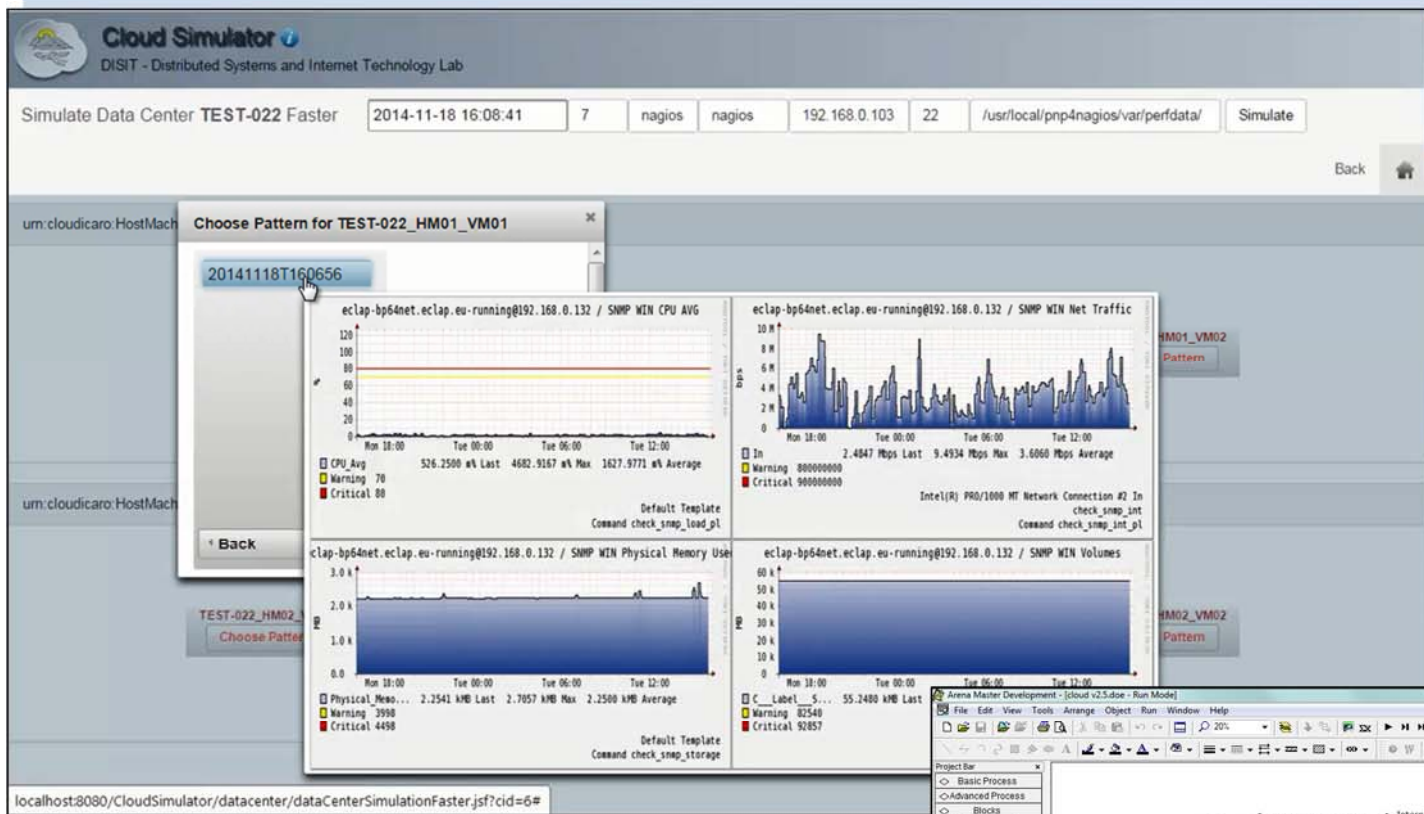




- ❑ **Permette di**
 - ❑ **Simulare** il comportamento di carico di datacenter complessi
 - ❑ **creare situazioni di carico** partendo da andamenti di carico reali dallo storico del sistema di monitoraggio
 - ❑ **studiare** gli effetti del carico sulle risorse di base a livello IaaS
- ❑ **Produce andamenti Simulati** accessibili e analizzabili da Supervisor & Monitor come dallo Smart Cloud Engine
- ❑ **Si integra** con
 - ❑ Lo Smart Cloud Engine per l'esecuzione di processi di controllo e valutazione e
 - ❑ la Knowledge Base per gestione delle configurazioni e dei dati, navigazione nella rappresentazione complessa del cloud
 - ❑ Il Supervisor & Monitor per l'accesso ai dati di monitoraggio, e la produzione di grafici

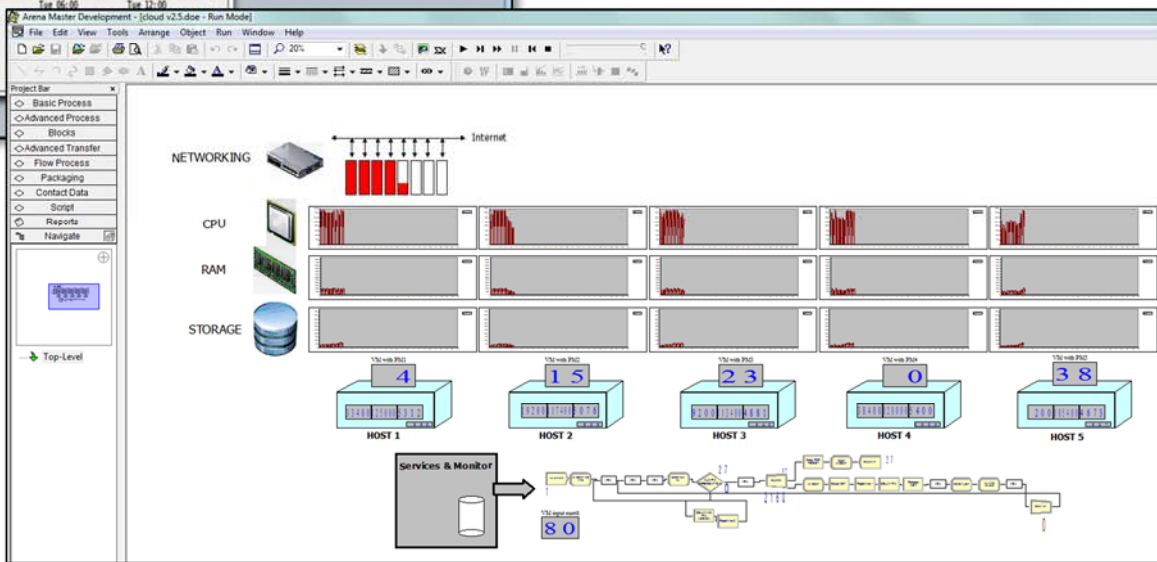


Cloud Simulator



Simulare il comportamento di carico di datacenter complessi

Identificare allocazioni ottime delle risorse





Knowledge Base & Tools

- ❑ La Knowledge Base **modella la conoscenza del cloud** (smart cloud ontology), viene alimentata con XML descrittivi con i quali
 - ❑ configura in modo automatico i moduli di monitoraggio e supervisione, che rimangono totalmente trasparenti per l'Service Portal, Configuration Manager e Business Producer.
- ❑ **Tramite i suoi Servizi**, la Knowledge Base permette di effettuare ragionamenti tenendo conto di modelli, e istanze dei processi allocati sul cloud e dei dati che provengono dal monitoraggio:
 - ❑ sullo stato del cloud, e la sua evoluzione
 - ❑ sulle configurazioni: coerenza e completezza
- ❑ KB ed i suoi Tool sono utilizzati dallo
 - ❑ **Smart Cloud Engine** per tutte le operazioni di data intelligence.
 - ❑ **Cloud Simulator** per ottimizzazioni e valutazioni

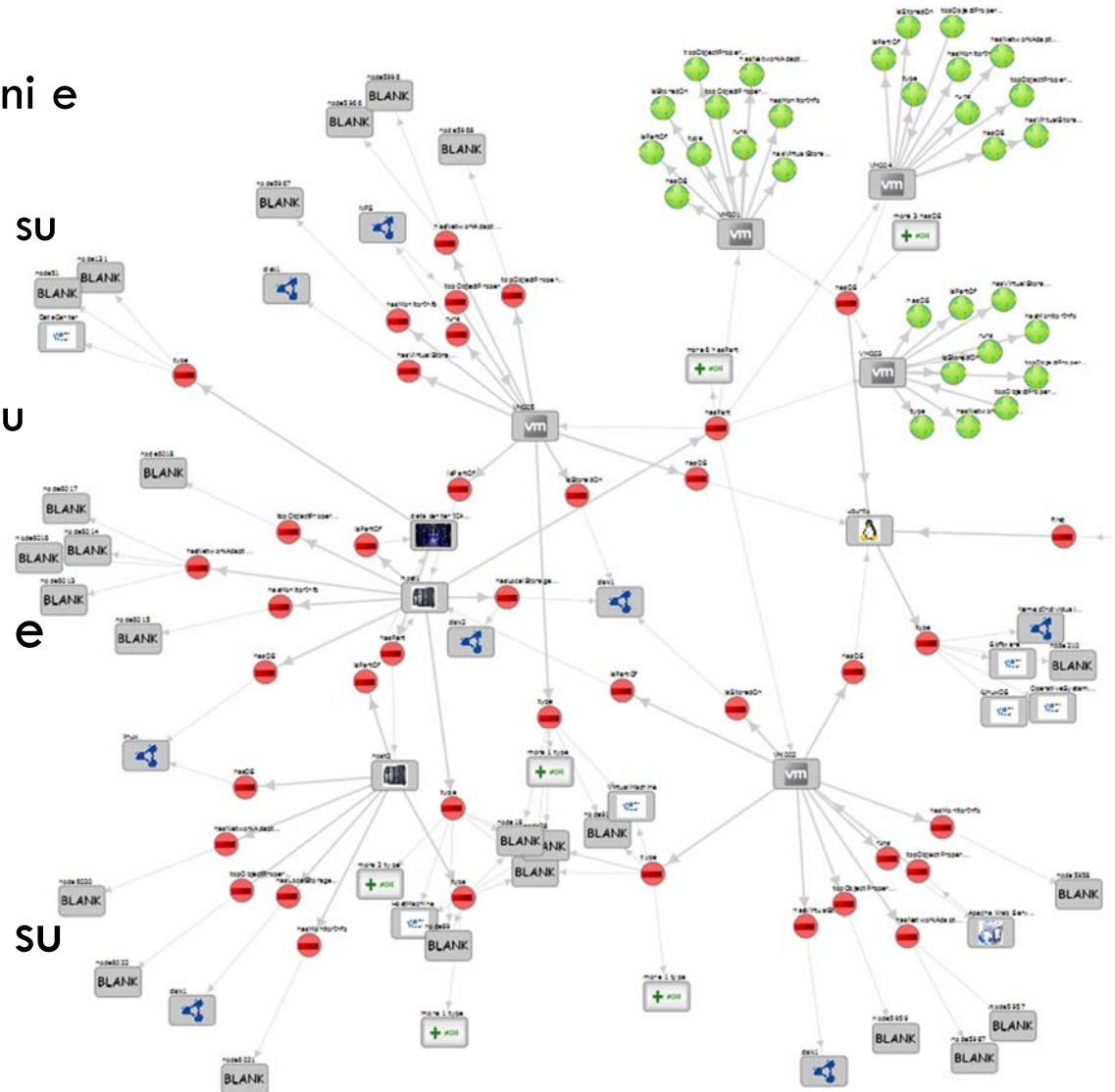


Modello di Cloud intelligence

- Formalizzazione di configurazioni e SLA (Service Level Agreement)
- reasoner supporto alle decisioni su configurazioni: consistenza e completezza
- adeguamento dell'architettura su alcune applicazioni

Tecnologia

- Knowledge base: RDF store e inference engine
- Smart Cloud Ontology: <http://www.disit.org/5604>
- Esempio di dato accessibile su <http://log.disit.org>





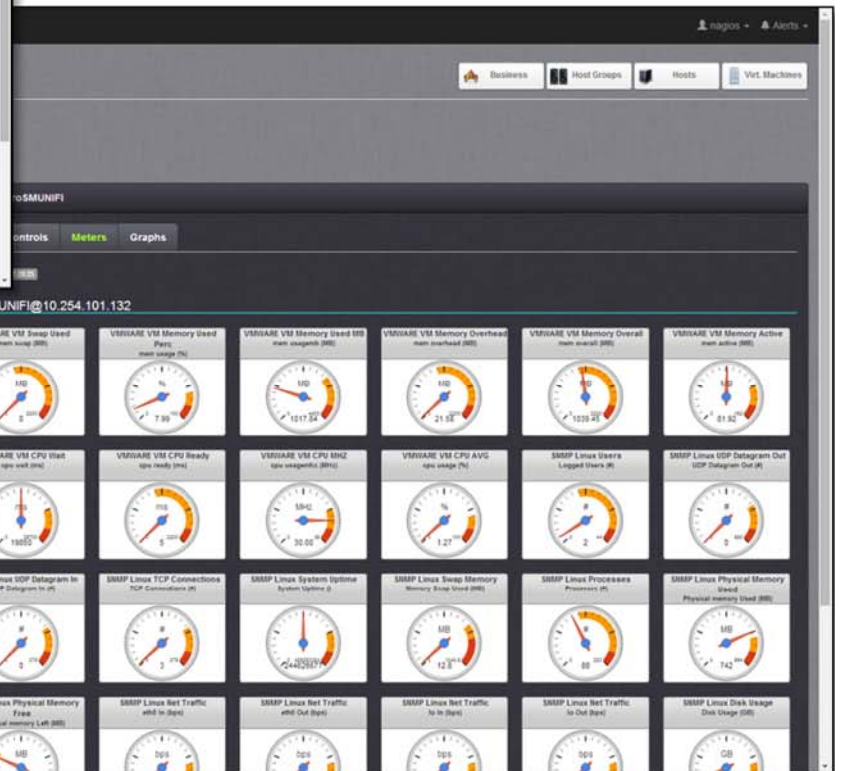
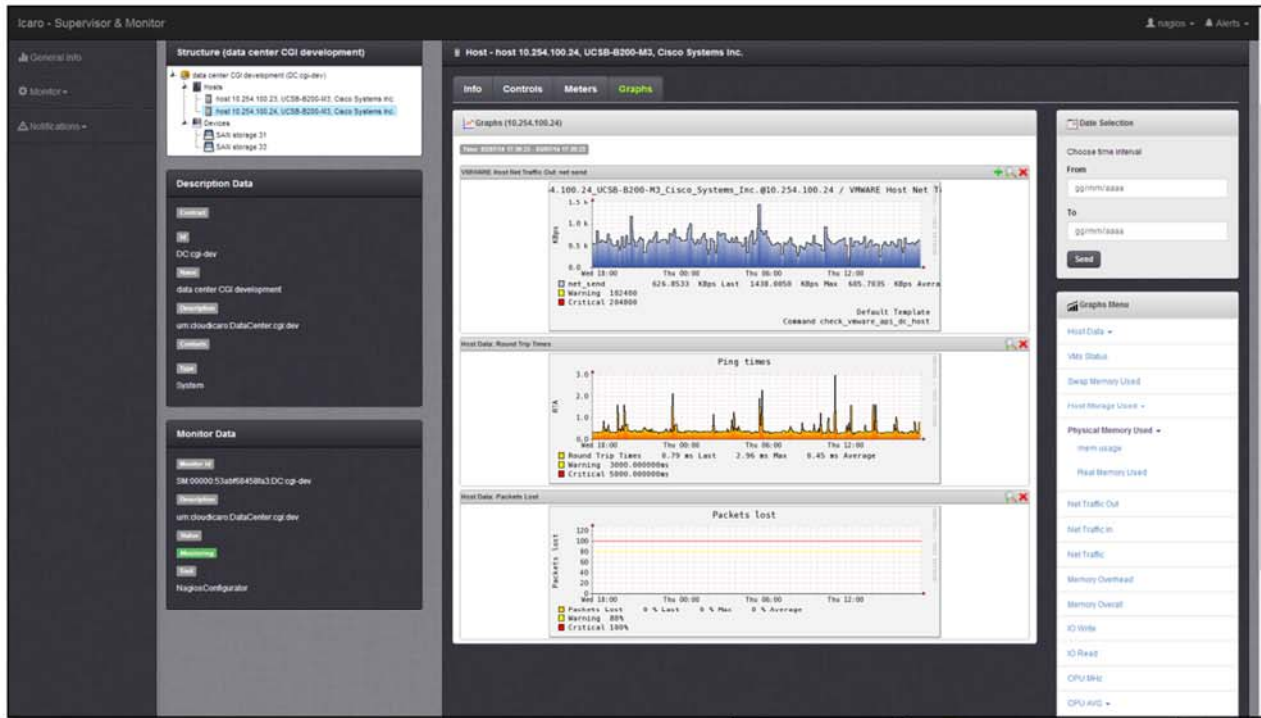
Supervisor & Monitor

- ❑ **Supervisione e monitoraggio delle risorse e dei consumi in modo integrato analizzando e tenendo sotto controllo:**
 - ❑ risorse cloud ai livelli: IaaS, SaaS, PaaS, BPaaS;
 - ❑ metriche applicative di Applicazioni e Servizi single/multi-tier: standard e caricati tramite il PaaS;
 - ❑ metriche definite in relazione alle SLA;
 - ❑ servizi interni ed esterni anche locati in altri cloud e sistemi, come supervisione dello stato dei processi: http, ftp, reti, server esterni, Web App Server, etc.

- ❑ **Il Supervisor & Monitor:**
 - ❑ è configurato in modo automatico dalla Knowledge Base
 - ❑ in ICARO utilizza il tool Nagios ma può essere esteso ad altri sistemi di monitoraggio di basso livello.
 - ❑ è in grado di controllare e configurare Nagios in modo automatizzato e di accedere in remoto alle funzionalità dei suoi componenti



Supervisor & Monitor



- Monitoraggio del Business

Cloud Ontology

- Developed for the ICARO project
- Focused on modelling the cloud aspects for validation and verification
- Competency questions:
 - «Is an instance of an application consistent with its definition?»
 - «Can Host machine X host VM Y in terms of CPU, memory, disk?»
 - «Which host machine can host VM X?»
 - «Which host machine is over-used?»

— ...

Cloud Ontology

- Represent the different aspects of cloud
 - Infrastructure
 - Host machine, virtual machine, network, network adapter, storage, local storage, external storage, firewall, router, ...
 - Applications & Services
 - Applications based on services (Tomcat, http server, dbms, mail server, ...)
 - An application can be deployed differently, all services on one VM, each service on a different VM
 - Business Configuration
 - Aggregate different applications to create a business, but also simple VMs and hosts
 - SLA & metrics
 - Define service level agreement for applications & VMs/hosts
 - Define metrics and metric values
 - Monitoring info
 - Information for monitoring services

Cloud Infrastructure

```
ex:datacenter1 rdf:type cld:DataCenter;  
  cld:hasName "production data center";  
  cld:hasPart ex:host1;  
  ...  
  cld:hasPart ex:host100;  
  cld:hasPart ex:storage1;  
  cld:hasPart ex:firewall1;  
  cld:hasPart ex:firewall2;
```



Cloud Infrastructure

```
ex:vm1 rdf:type cld:VirtualMachine
  cld:hasName "vm 1, windows xp";
  cld:hasCPUCount "2";
  cld:hasMemorySize "1";
  cld:hasVirtualStorage ex:vm1_disk;
  cld:hasNetworkAdapter ex:vm1_net1;
  cld:hasOS cld:windowsXP_Prof;
  cld:isStoredOn ex:host1_disk
  cld:isPartOf ex:host1;
```

```
ex:vm1_disk rdf:type cld:VirtualStorage;
  cld:hasDiskSize 10.
```

Cloud Infrastructure

```
ex:host1 rdf:type cld:HostMachine;  
  cld:hasName "host 1";  
  cld:hasCPUCount 16;  
  cld:hasCPUSpeed 2.2;  
  cld:hasCPUType "Intel Xeon X5660";  
  cld:hasMemorySize 16;  
  cld:hasDiskSize 300;  
  cld:hasLocalStorage ex:host1_disk;  
  cld:hasNetworkAdapter ex:host1_net1;  
  cld:hasNetworkAdapter ex:host1_net2;  
  cld:hasOS cld:vmware_esxi;  
  cld:isPartOf ex:datacenter1;
```

```
ex:host1_net1 rdf:type cld:NetworkAdapter;  
  cld:hasIPAddress "192.168.1.1";  
  cld:boundToNetwork ex:network1;
```

```
ex:host1_disk rdf:type cld:LocalStorage;  
  cld:hasDiskSize 300.
```

```
...  
ex:firewall1 rdf:type cld:Firewall;  
  cld:hasName "Firewall 1";  
  cld:hasNetworkAdapter ex:firewall1_net1;  
  cld:hasNetworkAdapter ex:firewall1_net2.
```

Cloud Applications

CloudApplication = *Software*

and (*hasIdentifier* exactly 1 string)

and (*hasName* exactly 1 string)

and (*developedBy* some *Developer*)

and (*developedBy* only *Developer*)

and (*createdBy* exactly 1 *Creator*)

and (*createdBy* only *Creator*)

and (*administeredBy* only *Administrator*)

and (*needs* only (*Service* or *CloudApplication* or *CloudApplicationModule*))

and (*hasSLA* max 1 *ServiceLevelAgreement*)

and (*hasSLA* only *ServiceLevelAgreement*)

and (*useVM* some *VirtualMachine*)

and (*useVM* only *VirtualMachine*)

Cloud Applications

JoomlaBalancedApp SubClassOf *CloudApplication*

and (needs exactly 1 *MySQLServer*)

and (needs exactly 1 *HttpBalancer*)

and (needs exactly 1 *NFSServer*)

and (needs min 1 (*ApacheWebServer* and (supportsLanguage value php_5))))



Cloud Applications

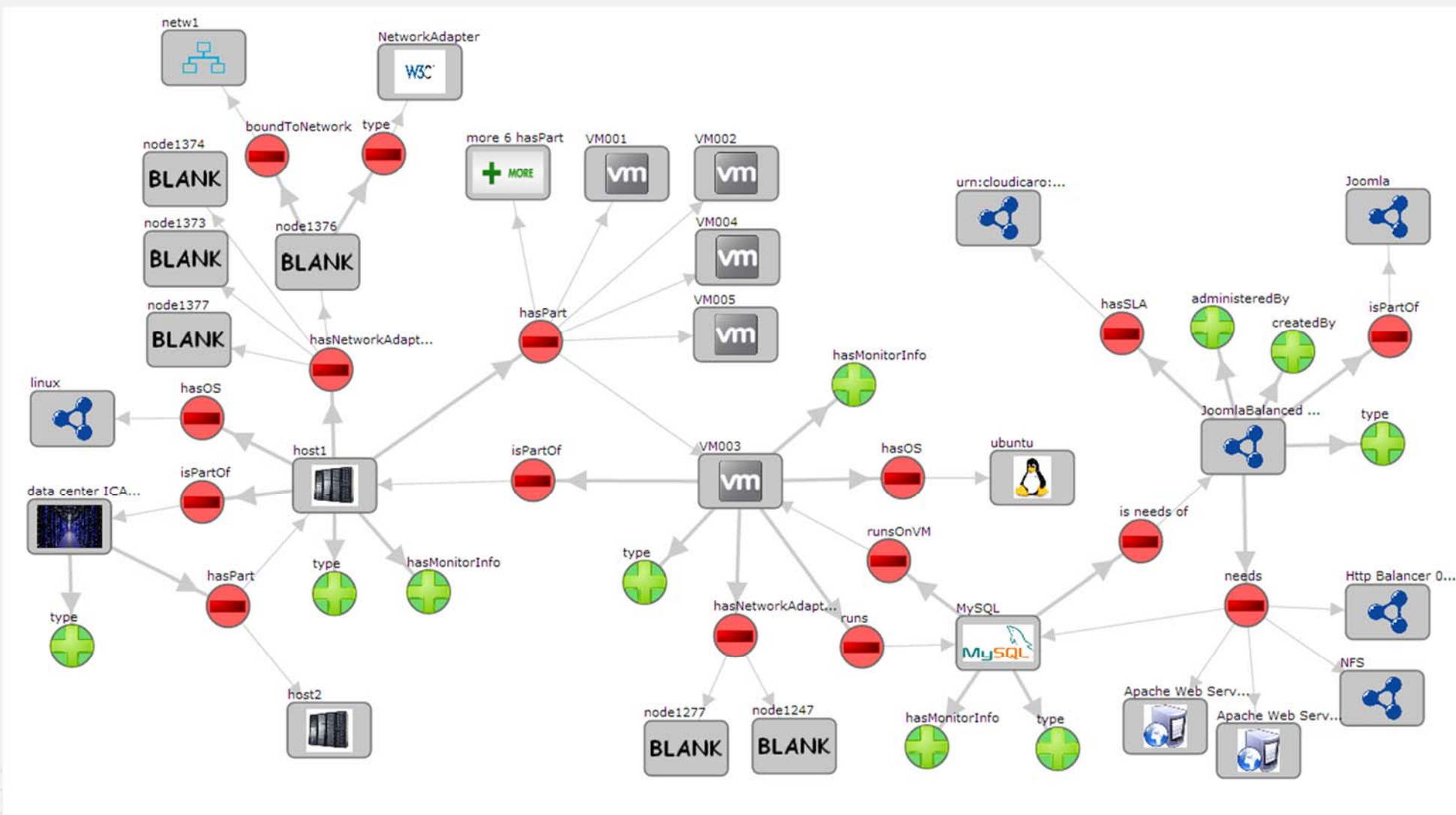
```
ex:Joomla1 rdf:type app:JoomlaBalancedApp;  
  cld:hasName "Joomla for my business";  
  cld:developedBy ex:user;  
  cld:createdBy ex:u1;  
  cld:needs ex:mysql1, ex:apache1, ex:apache2, ex:httpbalancer1,  
    ex:nfssserver1;  
  cld:hasSLA ex:sla1;  
  ...  
ex:mysql1 rdf:type cld:MySQLServer;  
  cld:runsOnVM ex:vm1;  
  ...  
ex:apache1 rdf:type cld:ApacheWebServer;  
  cld:runsOnVM ex:vm2;  
  cld:supportsLanguage cld:php_5;  
  ...
```

Business Configuration

```
ex:bc1 rdf:type cld:BusinessConfiguration;  
    cld:hasName "My business";  
    cld:createdBy ex:user1  
    cld:hasPart ex:joomla1;  
    cld:hasPart ex:crmTenant1;  
    ...
```

```
ex:crmTenant1 rdf:type cld:CloudApplicationTenant;  
    cld:hasName "My CRM tenant";  
    cld:hasIdentifier "crm:tenant:16373";  
    cld:createdBy ex:user1  
    cld:isTenantOf ex:crmApp1;
```


log.disit.org



SLA

- Service Level Agreement
- AND/OR Conditions on metric values with reference values
- «AVG responseTime 30Min»(apache1)<5s AND «LAST databaseSize»(mysql)<1GB



SLA

```
ex:sla1 rdf:type cld:ServiceLevelAgreement;  
  cld:hasSLObjective ex:slobj1;  
  cld:hasStartTime "2013-01-01T00:00:00";  
  cld:hasEndTime "2014-01-01T00:00:00".
```

```
ex:slobj1 rdf:type cld:ServiceLevelObjective;  
  cld:hasSLMetric ex:slmetric;  
  cld:hasSLAction ex:slaction1.
```

```
ex:slmetric rdf:type cld:ServiceLevelAndMetric;  
  cld:dependsOn ex:slmetric1;  
  cld:dependsOn ex:slmetric2.
```

```
ex:slmetric1 rdf:type cld:ServiceLevelSimpleMetric;  
  cld:hasMetricName "AVG responseTime 30Min";  
  cld:hasMetricValueLessThan "5";  
  cld:hasMetricUnit "seconds";  
  cld:dependsOn ex:apache1.
```

```
ex:slmetric2 rdf:type cld:ServiceLevelSimpleMetric;  
  cld:hasMetricName "LAST databaseSize";  
  cld:hasMetricValueLessThan "1";  
  cld:hasMetricUnit "GB";  
  cld:dependsOn ex:mysql.
```



Metrics

- Low Level Metrics
 - Defined by the monitoring tool for the VM/host/service/application
- High Level Metrics
 - Combine the low level metrics to produce a higher level indicator
- High Level Metric values



Metrics

- HighLevelMetric
 - Syntax tree with basic mathematic operators (+, -, /, *) that combine constants and temporal aggregations on low level metrics:
 - Max, min, avarage, last value over a temporal interval expressed in seconds, minutes, hours, days, months
 - In case the metric has multiple values for each time instant (e.g. Multiple disks) they can be combined with sum, min, max, avarage



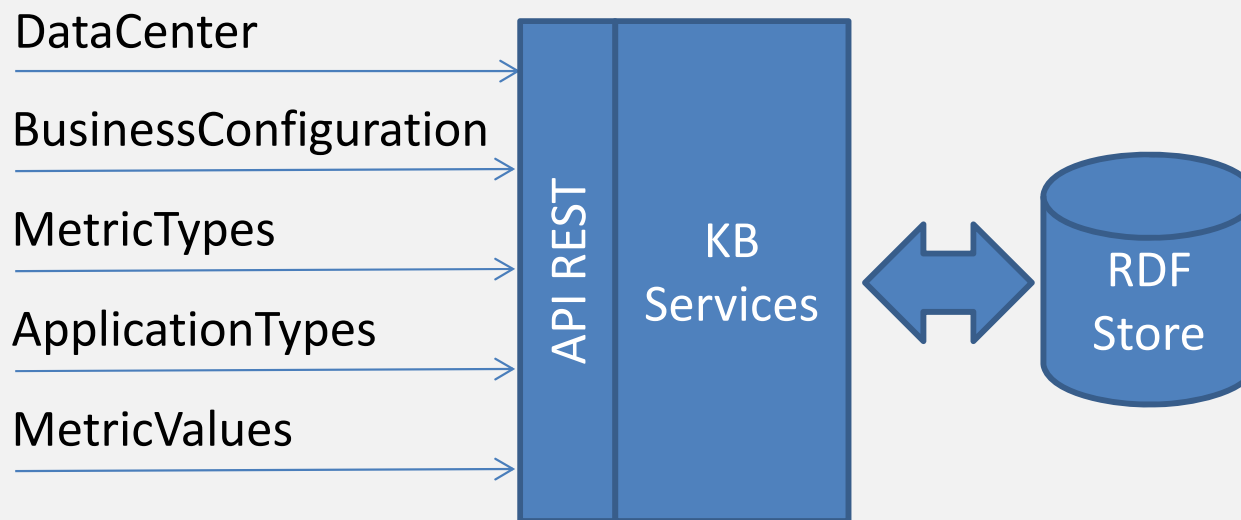
Monitoring information

```
ex:apache1 rdf:type cld:ApacheWebServer;  
  cld:runsOnVM ex:vm2;  
  cld:hasMonitorInfo ex:minfo1;
```

...

```
ex:minfo rdf:type cld:MonitorInfo;  
  cld:hasMetricName "responseTime";  
  cld:hasArguments "http://..."; #specific arguments to be  
provided to the plugin  
  cld:hasWarningValue 1;  
  cld:hasCriticalValue 4;  
  cld:hasMaxCheckAttempts 3;  
  cld:hasCheckInterval 5; #check every 5 min
```

Validation & Verification



RDF/XML format

Validation & Verification

- Use an OWL2 reasoner (e.g. Pellet) to check for inconcistencies
 - **Problem** with **Open World Assumption**. Some inconsistencies are not found.
 - An application needs one web server but none is specified, this is not inconsistent.
 - **Problem** with **not Unique Name Assumption**. Some inconsistencies not found.
 - An application needs exactly one MySQL instance, two are specified, no inconsistencies are found, they are assumed to be the same.

Validation & Verification

- Use SPARQL queries to check the configuration submitted
 - One query for each aspect, e.g. Is the OS valid?

```
SELECT ?vm ?os WHERE {  
  GRAPH <...> {  
    ?vm a cld:VirtualMachine;  
    cld:hasOS ?os.  
  }  
  FILTER NOT EXISTS {  
    ?os a cld:OperativeSystem.  
  }  
}
```

Validation & Verification

- We can have problems with inference
 - `cld:hasOS rdfs:range cld:OperativeSytem` (in ontology)
 - `ex:vm1 cld:hasOS ex:aWrongOS`
 - Inferred:
 - `ex:aWrongOS rdf:type cld:OperativeSystem`
 - The previous query will not identify *ex:aWrongOS* as not correct.



Validation & Verification

- Using SPARQL has the advantage that can be checked aspects that cannot be modeled with OWL (e.g. The host machine has now enough resources to host the VM?)
- SPARQL validation queries can be stored in a configuration and can be updated if the ontology change, without modifying the application.

RDF Blank nodes

- Blank nodes are nodes in the RDF graph that have a temporary local identifier.

ex:vm1 cld:hasNetworkAdapter **_:st1**.

_:st1 a cld:NetworkAdapter.

_:st1 cld:hasIPAddress "192.168.0.1".

- The blank nodes identifiers returned from a SPARQL query cannot be used in another SPARQL query to obtain information on this blank node...
- **They should be avoided**, especially if are present recursive references to blank nodes as in trees of expressions.

Major topics addressed

- From RDF to OWL
- Knowledge engineering for Beginners
- Smart Cloud Application (ICARO Case)
- Big Data Smart City Architecture ←
- Smart-city Ontology
- Data Ingestion and Mining
- Distributed and real time processes
- RDF processing
- Smart City Engine
- Development Interfaces
- Sii-Mobility

Knowledge mining and Semantic Models: from Cloud to Smart City

part 2: From Cloud to Smart City

x Dottorato DIST, Univ. Firenze

Pierfrancesco Bellini, Paolo Nesi

DISIT Lab

Dipartimento di Ingegneria dell'Informazione, DINFO

Università degli Studi di Firenze

Via S. Marta 3, 50139, Firenze, Italy

Tel: +39-055-2758511, fax: +39-055-2758570

<http://www.disit.dinfo.unifi.it> *alias*

<http://www.disit.org>

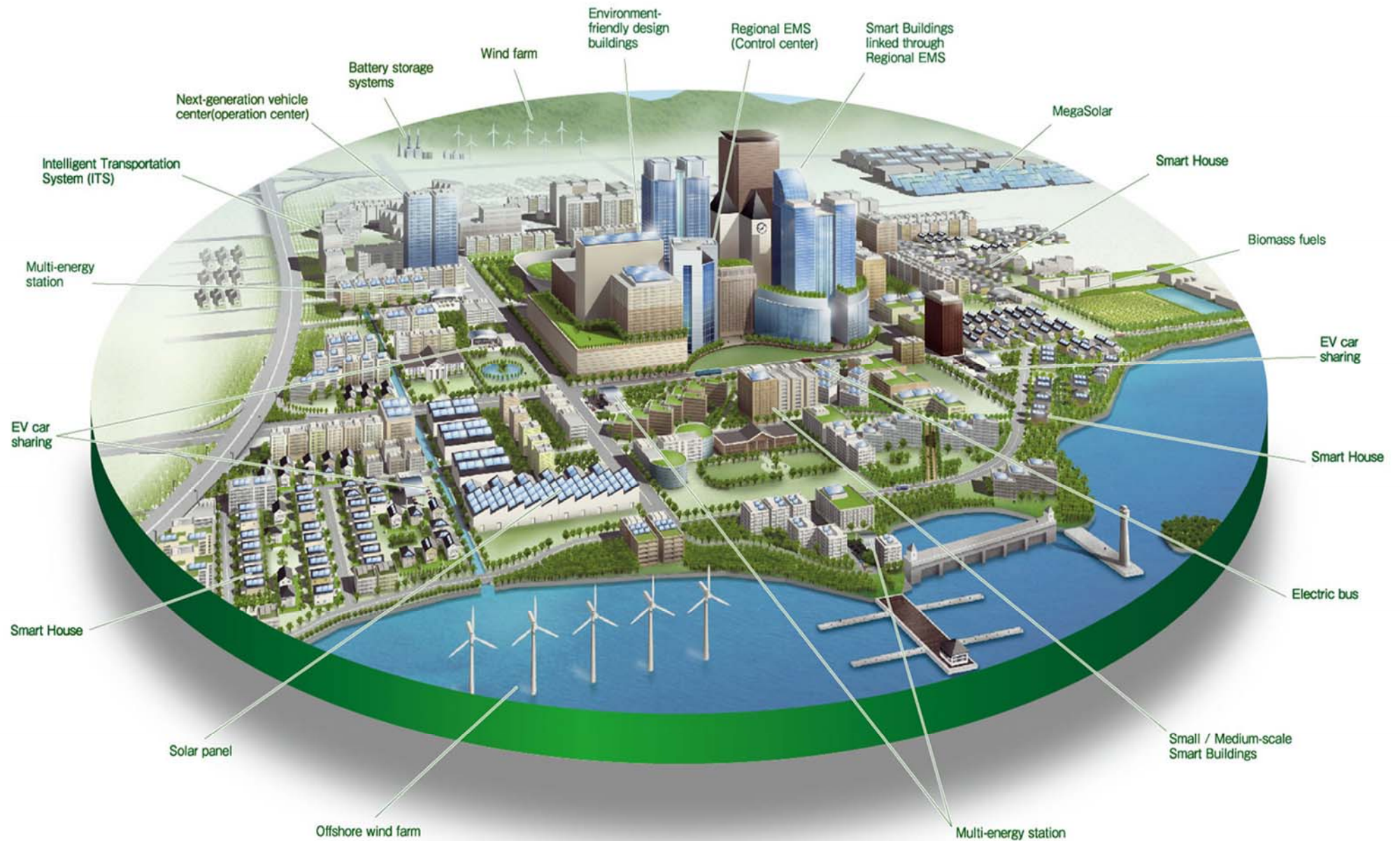
Pierfrancesco.bellini@unifi.it , Paolo.nesi@unifi.it

Smart City: Motivations

- **Societal challenge**

- We see a strong increment of population of our cities, since in the cities the life is simple and of higher quality in term of services and working opportunities
- The cities needs to be adapted to the increment of population, to new evolving ages, to the new technologies and expectations of population

- → **Sustainability of the growth**



Smartness, smart city needs 6 features

- Smart Health
- Smart Education
- Smart Mobility
- Smart Energy
- Smart Governmental
 - Smart economy
 - Smart people
 - Smart environment
 - Smart living
- Smart Telecommunication

Smart health

(can be regarded as smart governmental)

- Online accessing to health services:
 - booking and paying
 - selecting doctor
 - access to EPR (Electronic Patient Record)
- **Monitoring** services and users for,
 - learn people behavior, create collective profiles
 - personalized health
 - Inform citizens to the risks of their habits
 - Improve efficiency of services
 - redistribute workload, thus reducing the peak of consumption



Smart Education

(can be regarded as smart governmental)

- Diffusion of ICT into the schools:
 - LIM, PAD, internet connection, tables, ..
- Primary and secondary schools → university
→ industry & services
- **Monitoring** the students and quality of service,
 - learn student behavior, create collective profiles,
 - personalized education
- suggesting behavior to
 - Informing the families
 - moderate the peak of consumption
 - increase the competence in specific needed sectors, etc.
- Increase formation impact and benefits



Smart Mobility



- Public transportation:
 - bus, railway, taxi, metro, etc.,
- Public transport for services:
 - garbage collection, ambulances,
- Private transportation:
 - cars, delivering material, etc.
- New solutions (public and/or private):
 - electric cars, car sharing, car pooling, bike sharing, bicycle paths
- Online:
 - ticketing, monitoring travel, infomobility, access to RTZ, parking, etc.



Smart Mobility and urbanization

- **Monitoring** the city status,
 - learn city behavior on mobility
 - learn people behavior
 - create collective profiles
 - tracking people flows
- **Providing Info/service**
 - personalized
 - **Info** about city status to
 - help moving people and material
 - education on mobility,
 - moderate the peak of consumption
- **Reasoning to**
 - make services sustainable
 - make services accessible
 - Increase the quality of service



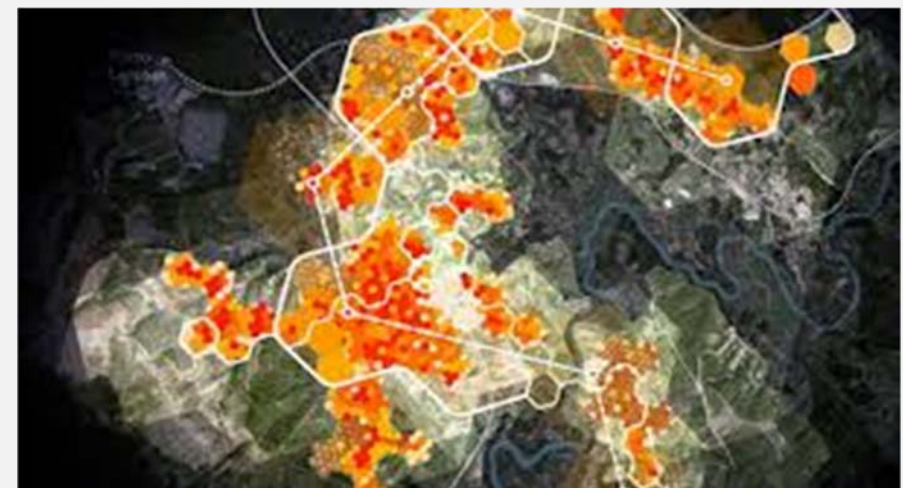
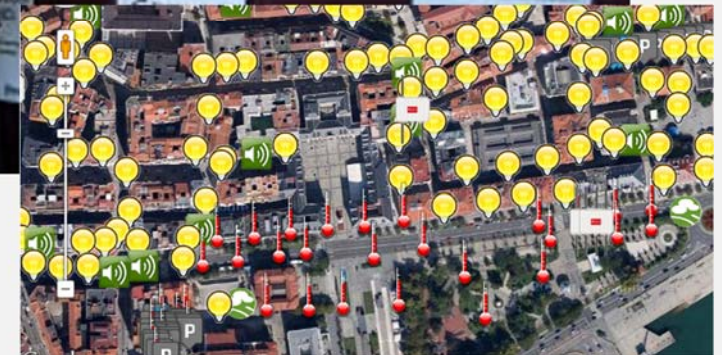
Smart Energy

- **Smart building:**
 - saving and optimizing energy consumption, district heating
 - renewable energy: photovoltaic, wind energy, solar energy, hydropower, etc.
- **Smart lighting:**
 - turning on/off on the basis of the real needs
- **Energy points for electric: c**
 - cars, bikes, scooters,
- **Monitoring** consumption, learn people/city behavior on energy consumption, learn people behavior, create collective profiles
- **Suggesting consumers**
 - different behavior for consumption: different time to use the washing machine
- **Suggesting administrations**
 - restructuring to reduce the global consumption,
 - moderate the peak of consumption



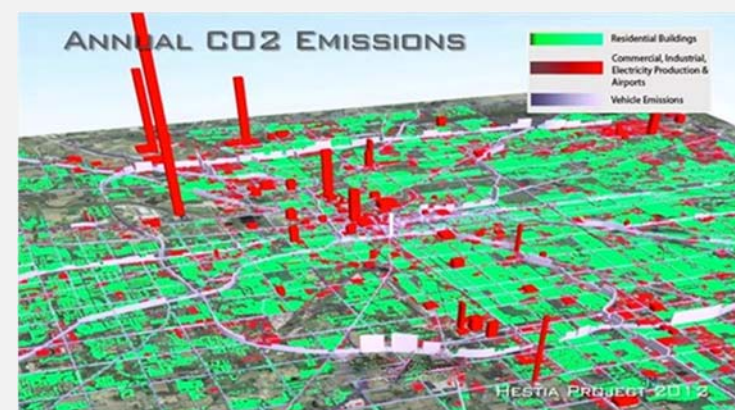
Smart Governmental Services

- Service toward citizens:
 - on-line services:
 - register, certification, civil services, taxes, use of soil, ...
 - Payments and banking:
 - taxes, schools, accesses
 - Garbage collection:
 - regular and exceptional
 - Quality of air:
 - monitoring pollution
 - Water control:
 - monitoring water quality, water dispersion, river status



Smart Governmental Services

- **Service toward citizens:**
 - **Cultural Heritage:** ticketing on museums,
 - **Tourism:** ticketing, visiting, planning, booking (hotel and restaurants, etc.)
 - **social networking:** getting service feedbacks, monitoring
- **Social sustainability of services:**
 - crowd services
- **Social recovering** of infrastructure,
 - New services, exploiting infrastructures
- **Monitoring** consumption and exploitation of services, learn people behavior, create collective profiles
 - Discovering problems of services,
 - Finding collective solutions and new needs...



Telecommunication, broadband

- **Fixed Connectivity:**
 - ADSL or more, fiber,
- **Mobile Connectivity:**
 - Public wifi, Services on WiFi, HSPDA, LTE
- **Monitoring** communication infrastructure
- Providing information and formation on:
 - how to exploit the communication infrastructure
 - Exploiting the communication for the other services,
 - moderate the peak of consumption



Privati Statici

- Codice fiscale
- Foto non condivise
- Aspetti legali
- Cartella clinica
- ..

- Movimenti personali non pubblicati
- Relazioni personali non pubblicate

- comportamenti social media
- contributi consumi

- Traffico personale
- Posizione mezzi, Parcheggi
- Posizione taxi
- Posizione CarSharing
- ...

Privati Tempo reale

Pubblici statici (open data)

statistiche: incidenti, censimenti, votazioni

- Statistiche accessi alla ZTL
- Strutture pubbliche UNIFI

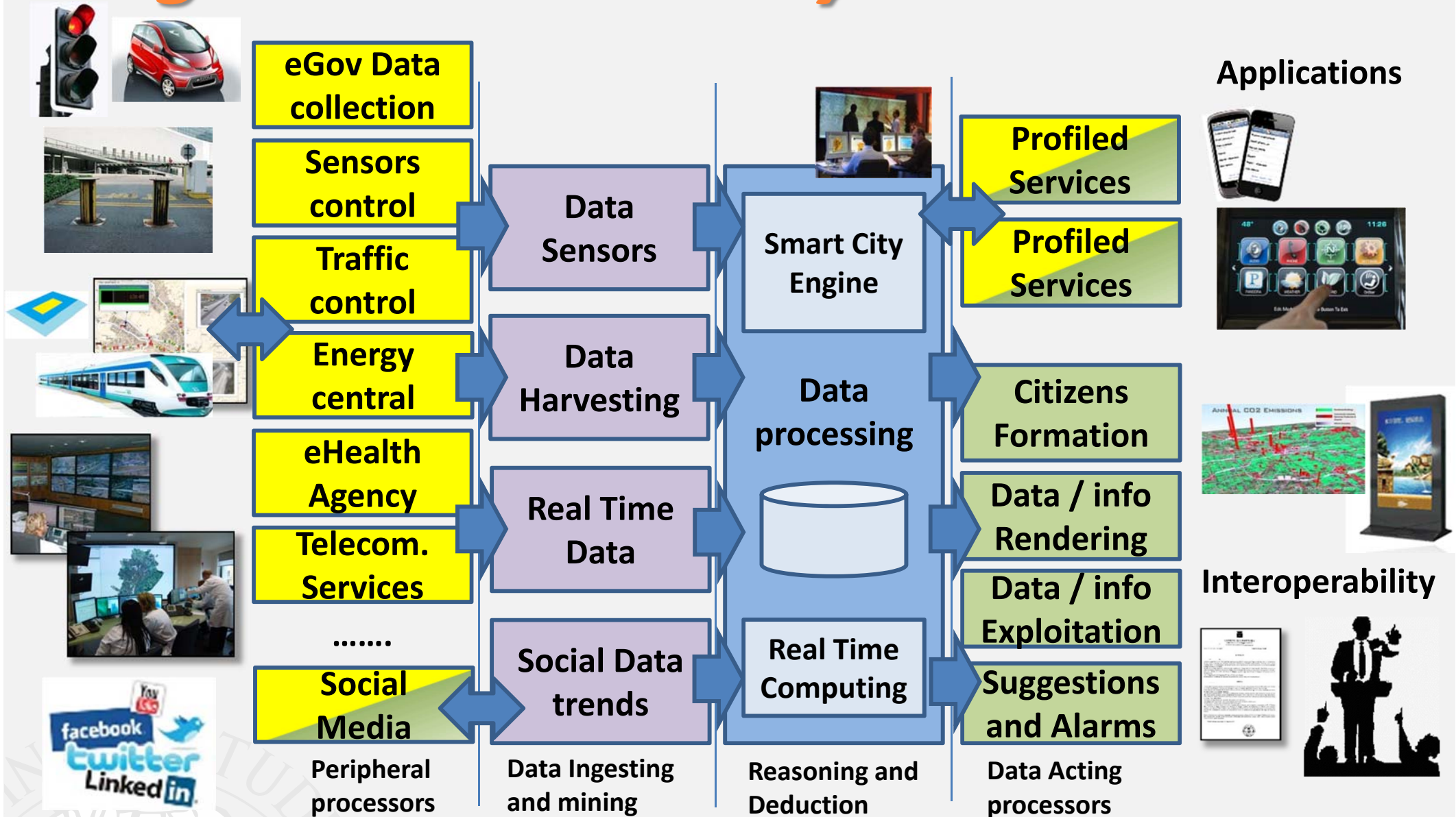
posizione dei punti di interesse

- Musei
- Strutture della città
- Servizi attivi

- Info traffico
- video camere
- Info Meteo
- Info Ambiente
- Code ai musei pubblici
- Terremoti
- Parcheggi
- Stato accessi alla ZTL
- Stato dei servizi

Pubblici Tempo reale (open data)

Big Data Smart City Architecture



Major topics addressed

- From RDF to OWL
- Knowledge engineering for Beginners
- Smart Cloud Application (ICARO Case)
- Big Data Smart City Architecture
- Smart-city Ontology ←
- Data Ingestion and Mining
- Distributed and real time processes
- RDF processing
- Smart City Engine
- Development Interfaces
- Sii-Mobility

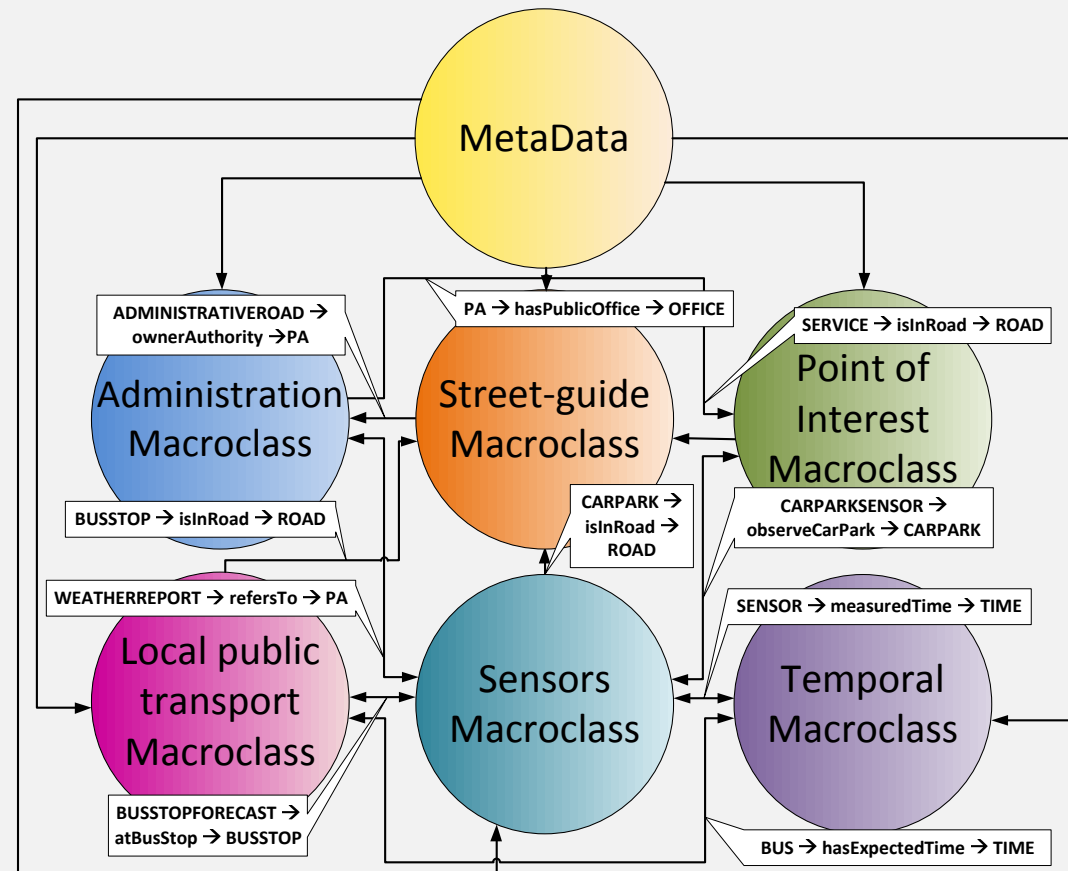
Smart-city Ontology objectives

- Create a unified knowledge base grounded on a common ontology that allows to combine all data coming from different sources making them semantically interoperable
- To.
 - Create coherent queries independently from the source, format, date, time, provider, etc.
 - Enrich the data, make it more complete, more reliable, more accessible
 - Enable to perform inference as triple materialization from some of the relations
 - to enable the implementation of new integrated services related to mobility
 - to provide repository access to SMEs to create new services

Smart-city Ontology

- The data model provided have been mapped into the ontology, it covers different aspects:

- Administration
- Street-guide
- Points of interest
- Local public transport
- Sensors
- Temporal aspects
- Metadata on the data



Smart-city Ontology

- **Metadata:** modeling the additional information associated with:
 - **Descriptor** of Data sets that produced the triples: data set ID, title, description, purpose, location, administration, version, responsible, etc..
 - **Licensing** information
 - **Process** information: IDs of the processes adopted for ingestion, quality improvement, mapping, indexing,.. ; date and time of ingestion, update, review, ...;

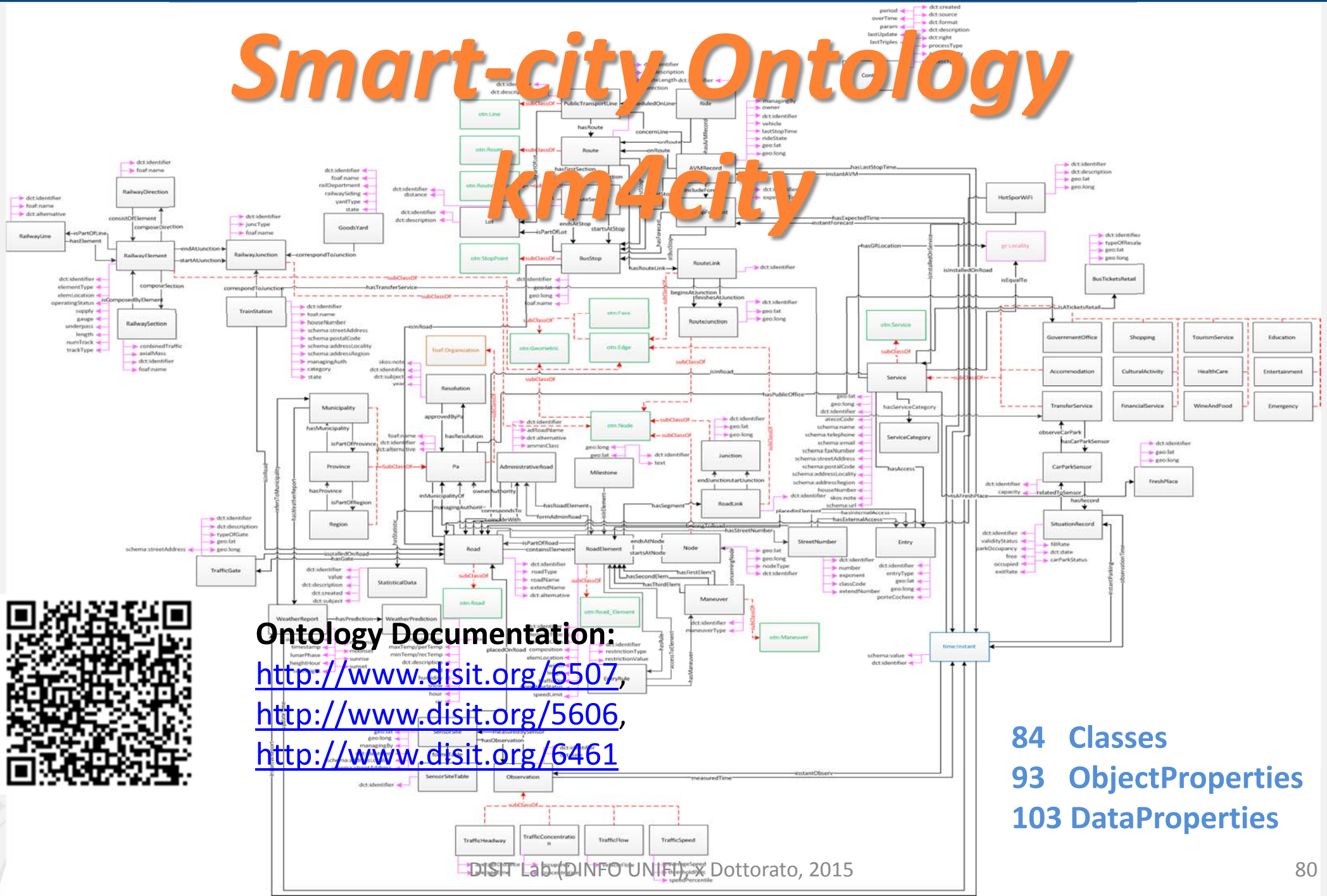
When a problem is detected, we have the information to understand when and how the problem has been included
- **Including basic ontologies as:**
 - *DC: Dublin core, standard metadata*
 - *OTN: Ontology for Transport Network*
 - *FOAF: for the description of the relations among people or groups*
 - *Schema.org: for a description of people and organizations*
 - *wgs84_pos: for latitude and longitude, GPS info*
 - *OWL-Time: reasoning on time, time intervals*
 - *GoodRelations: commercial activities models*

P. Bellini, M. Benigni, R. Billero, P. Nesi and N. Rauch, "Km4City Ontology Building vs Data Harvesting and Cleaning for Smart-city Services", *International Journal of Visual Language and Computing*, Elsevier,

<http://dx.doi.org/10.1016/j.jvlc.2014.10.023>

Smart-city Ontology

km4city



Ontology Documentation:

<http://www.disit.org/6507>,

<http://www.disit.org/5606>,

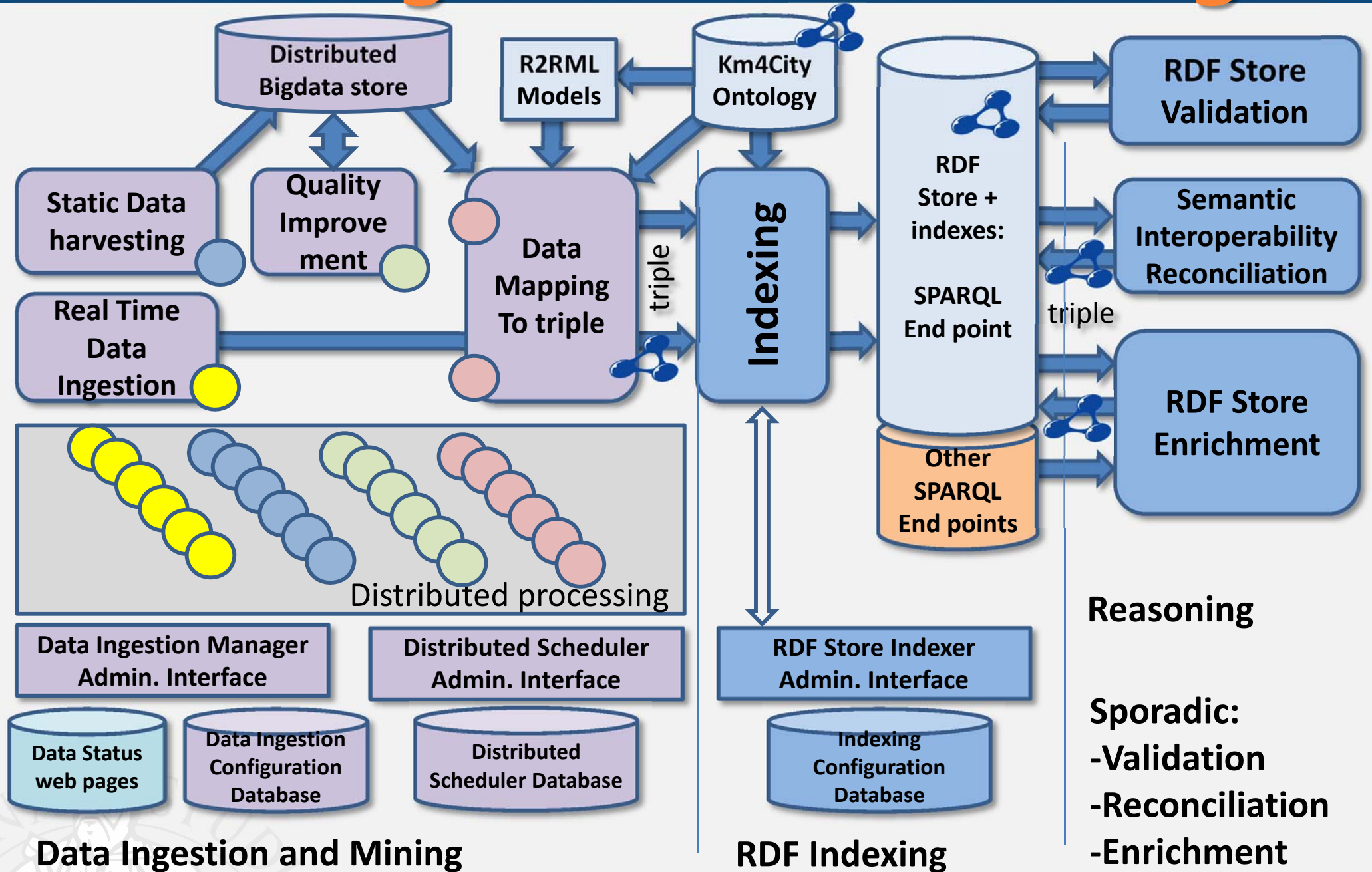
<http://www.disit.org/6461>

84 Classes
93 ObjectProperties
103 DataProperties

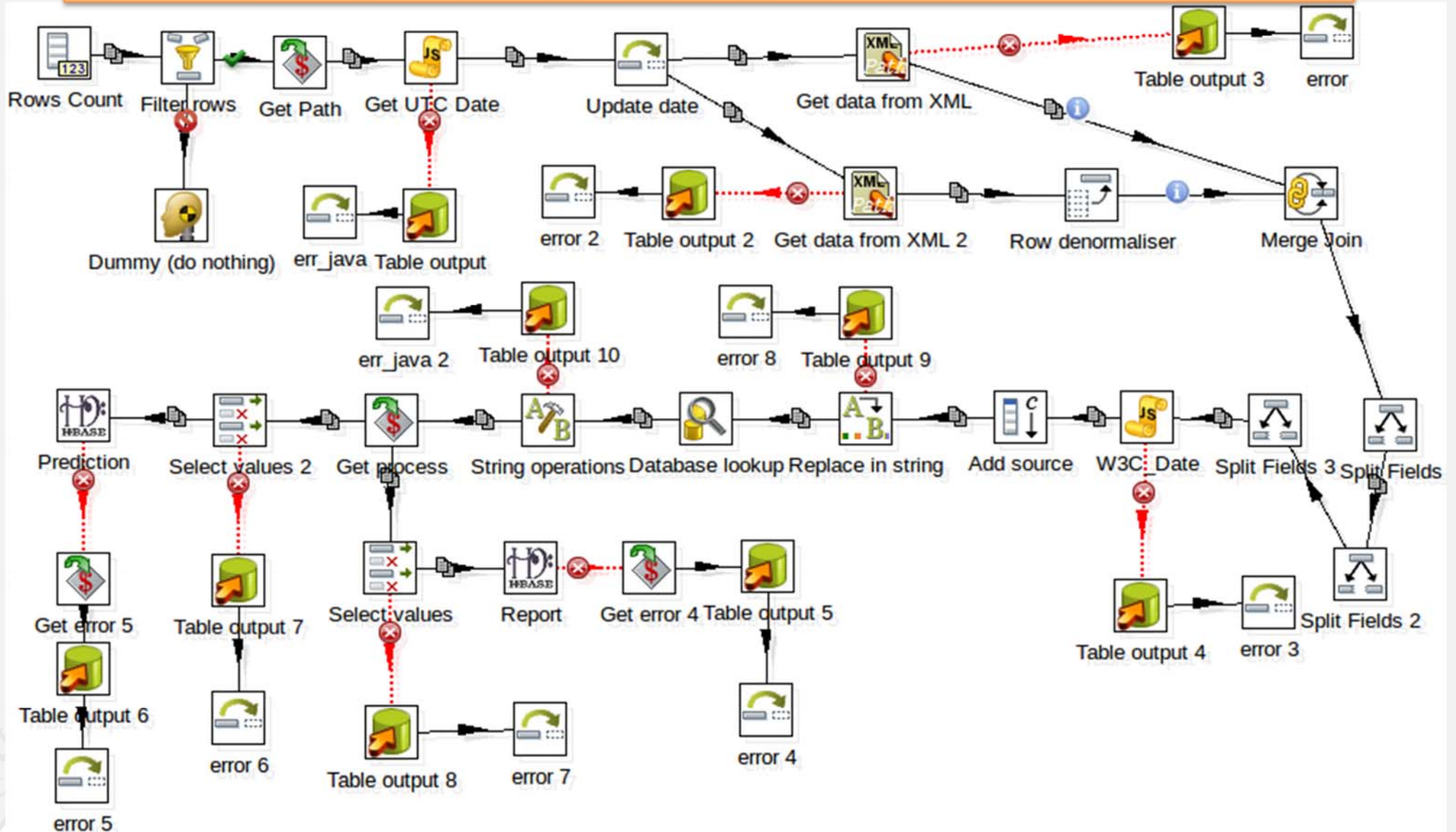
Major topics addressed

- From RDF to OWL
- Knowledge engineering for Beginners
- Smart Cloud Application (ICARO Case)
- Big Data Smart City Architecture
- Smart-city Ontology
- Data Ingestion and Mining ←
- Distributed and real time processes
- RDF processing
- Smart City Engine
- Development Interfaces
- Sii-Mobility

Data Ingestion and Mining



Example of Ingestion process



Data Quality Improvement

Data quality's aspect:

- **Completeness:** presence of all information needed to describe an object, entity or event (e.g. Identifying).
- **Consistency:** data must not be contradictory. For example, the total balance and movements.
- **Accuracy:** data must be correct, i.e. conform to actual values. For example, an email address must not only be well-formed [nome@dominio.it](#), but it must also be valid and working.

Data mapping to Triples

- Transforms the data from HBase to RDF triples
- Using **Karma Data Integration tool**, a mapping model from SQL to RDF on the basis of the ontology was created
 - Data to be mapped first temporarily passed from Hbase to MySQL and then mapped using Karma (in batch mode)
- The mapped data in triples have to be uploaded (and indexed) to the **RDF Store** (OpenRDF – sesame with OWLIM-SE)

Major topics addressed

- From RDF to OWL
- Knowledge engineering for Beginners
- Smart Cloud Application (ICARO Case)
- Big Data Smart City Architecture
- Smart-city Ontology
- Data Ingestion and Mining
- Distributed and real time processes ←
- RDF processing
- Smart City Engine
- Development Interfaces
- Sii-Mobility

Distributed Scheduler

- Use of a scheduler to manage periodic execution of ingestion and triple generation processes.
 - This tool throws the processes with predefined interval determined in phase of configuration.
- **Static Data: as Sporadic processes:**
 - scheduled every months or week
- **Real Time data** (car parks, road sensors, etc.)
 - ingestion and triple generation processes should be performed periodically (no for **static data**).

<http://192.168.0.72>

Distributed Scheduler

Smart Cloud Engine
DISIT - Distributed Systems and Internet Technology Lab

<p>192.168.0.14</p> <ul style="list-style-type: none"> LAST_CHECK: 2014-12-16 11:29:04 SCHEDULER_INSTANCE_ID: hadoopnode01d1418718662488 CPU_LOAD: 0.05322341999577256 FREE_PHYSICAL_MEMORY: 4686659584 JOBS_EXECUTED: 0 SCHEDULER_NAME: SCE CURRENT_TIME: 2014-12-16 11:29:58 JOBS/h: 0 RUNNING SINCE: 2014-12-16 09:31:02 CLUSTERED: 1 PERSISTENCE: 1 REMOTE_SCHEDULER: 0 CURRENTLY_EXECUTING_JOBS: 0 CPU_LOAD_JVM: 8.877615726062143E-4 SYSTEM_LOAD_AVERAGE: 0.0 OPERATING_SYSTEM_VERSION: 3.13.0.24-generic COMMITTED_VIRTUAL_MEMORY: 3679342592 OPERATING_SYSTEM_NAME: Linux FREE_SWAP_SPACE: 12860071936 PROCESS_CPU_TIME: 3977000000 TOTAL_PHYSICAL_MEMORY: 1.260085248E10 NUMBER_OF_PROCESSORS: 4 OPERATING_SYSTEM_ARCHITECTURE: amd64 TOTAL_SWAP_SPACE: 1.2881752064E10 IS_SCHEDULER_STANDBY: 0 IS_SCHEDULER_SHUTDOWN: 0 IS_SCHEDULER_STARTED: 1 TOTAL_DISK_SPACE: 2321541849088 UNALLOCATED_DISK_SPACE: 1937102204928 USABLE_DISK_SPACE: 1819795460096 PREV_FIRE_TIME: 2014-12-15 23:09:17 CPU: Intel(R) Xeon(R) CPU X3470 @ 2.93GHz 	<p>192.168.0.26</p> <ul style="list-style-type: none"> LAST_CHECK: 2014-12-16 11:29:04 SCHEDULER_INSTANCE_ID: hadoopnode06c1418718723312 CPU_LOAD: 0.048106517968803606 FREE_PHYSICAL_MEMORY: 10056519680 JOBS_EXECUTED: 0 SCHEDULER_NAME: SCE CURRENT_TIME: 2014-12-16 11:29:58 JOBS/h: 0 RUNNING SINCE: 2014-12-16 09:32:03 CLUSTERED: 1 PERSISTENCE: 1 REMOTE_SCHEDULER: 0 CURRENTLY_EXECUTING_JOBS: 0 CPU_LOAD_JVM: 8.425309630128908E-4 SYSTEM_LOAD_AVERAGE: 0.13 OPERATING_SYSTEM_VERSION: 3.13.0.24-generic COMMITTED_VIRTUAL_MEMORY: 3679342592 OPERATING_SYSTEM_NAME: Linux FREE_SWAP_SPACE: 12633550648 PROCESS_CPU_TIME: 18990000000 TOTAL_PHYSICAL_MEMORY: 1.260085248E10 NUMBER_OF_PROCESSORS: 4 OPERATING_SYSTEM_ARCHITECTURE: amd64 TOTAL_SWAP_SPACE: 1.2881752064E10 IS_SCHEDULER_STANDBY: 0 IS_SCHEDULER_SHUTDOWN: 0 IS_SCHEDULER_STARTED: 1 TOTAL_DISK_SPACE: 2321541775360 UNALLOCATED_DISK_SPACE: 1938265976832 USABLE_DISK_SPACE: 1820929695744 PREV_FIRE_TIME: 2014-12-15 23:14:19 CPU: Intel(R) Xeon(R) CPU E5-4620 @ 2.20GHz 	<p>192.168.0.40</p> <ul style="list-style-type: none"> LAST_CHECK: 2014-12-16 11:29:11 SCHEDULER_INSTANCE_ID: hadoopnode01d1418719522697 CPU_LOAD: 0.0013337223356812403 FREE_PHYSICAL_MEMORY: 10849054720 JOBS_EXECUTED: 20 SCHEDULER_NAME: SCE CURRENT_TIME: 2014-12-16 11:29:58 JOBS/h: 14.91 RUNNING SINCE: 2014-12-16 09:45:22 CLUSTERED: 1 PERSISTENCE: 1 REMOTE_SCHEDULER: 0 CURRENTLY_EXECUTING_JOBS: 0 CPU_LOAD_JVM: 5.001458758804651E-4 SYSTEM_LOAD_AVERAGE: 0.0 OPERATING_SYSTEM_VERSION: 3.13.0.24-generic COMMITTED_VIRTUAL_MEMORY: 3679342592 OPERATING_SYSTEM_NAME: Linux FREE_SWAP_SPACE: 12881752064 PROCESS_CPU_TIME: 18990000000 TOTAL_PHYSICAL_MEMORY: 1.260085248E10 NUMBER_OF_PROCESSORS: 4 OPERATING_SYSTEM_ARCHITECTURE: amd64 TOTAL_SWAP_SPACE: 1.2881752064E10 IS_SCHEDULER_STANDBY: 0 IS_SCHEDULER_SHUTDOWN: 0 IS_SCHEDULER_STARTED: 1 TOTAL_DISK_SPACE: 21252098688 UNALLOCATED_DISK_SPACE: 195266711552 USABLE_DISK_SPACE: 185156763648 PREV_FIRE_TIME: 2014-12-16 09:53:47 CPU: Intel(R) Xeon(R) CPU X5690 @ 3.47GHz 	<p>192.168.0.42</p> <ul style="list-style-type: none"> LAST_CHECK: 2014-12-16 11:29:35 SCHEDULER_INSTANCE_ID: hadoopnode061418718994664 CPU_LOAD: 0.16369819341126463 FREE_PHYSICAL_MEMORY: 1921798144 JOBS_EXECUTED: 0 SCHEDULER_NAME: SCE CURRENT_TIME: 2014-12-16 11:29:58 JOBS/h: 0 RUNNING SINCE: 2014-12-16 09:36:34 CLUSTERED: 1 PERSISTENCE: 1 REMOTE_SCHEDULER: 0 CURRENTLY_EXECUTING_JOBS: 0 CPU_LOAD_JVM: 7.651759904778099E-4 SYSTEM_LOAD_AVERAGE: 1.04 OPERATING_SYSTEM_VERSION: 3.13.0.24-generic COMMITTED_VIRTUAL_MEMORY: 3679342592 OPERATING_SYSTEM_NAME: Linux FREE_SWAP_SPACE: 12159328256 PROCESS_CPU_TIME: 29620000000 TOTAL_PHYSICAL_MEMORY: 1.260085248E10 NUMBER_OF_PROCESSORS: 4 OPERATING_SYSTEM_ARCHITECTURE: amd64 TOTAL_SWAP_SPACE: 1.2881752064E10 IS_SCHEDULER_STANDBY: 0 IS_SCHEDULER_SHUTDOWN: 0 IS_SCHEDULER_STARTED: 1 TOTAL_DISK_SPACE: 2321541775360 UNALLOCATED_DISK_SPACE: 1937021210624 USABLE_DISK_SPACE: 1819684929536 PREV_FIRE_TIME: 2014-12-15 23:09:17 CPU: Intel(R) Xeon(R) CPU E5-2640 v2 @ 2.00GHz 	<p>192.168.0.69</p> <ul style="list-style-type: none"> LAST_CHECK: 2014-12-16 11:29:56 SCHEDULER_INSTANCE_ID: hadoopnode021418718835378 CPU_LOAD: 0.081939516810272 FREE_PHYSICAL_MEMORY: 5102755840 JOBS_EXECUTED: 0 SCHEDULER_NAME: SCE CURRENT_TIME: 2014-12-16 11:29:58 JOBS/h: 0 RUNNING SINCE: 2014-12-16 09:33:55 CLUSTERED: 1 PERSISTENCE: 1 REMOTE_SCHEDULER: 0 CURRENTLY_EXECUTING_JOBS: 0 CPU_LOAD_JVM: 8.025004223686434E-4 SYSTEM_LOAD_AVERAGE: 0.6 OPERATING_SYSTEM_VERSION: 3.13.0.24-generic COMMITTED_VIRTUAL_MEMORY: 3683553280 OPERATING_SYSTEM_NAME: Linux FREE_SWAP_SPACE: 12881752064 PROCESS_CPU_TIME: 29770000000 TOTAL_PHYSICAL_MEMORY: 1.260085248E10 NUMBER_OF_PROCESSORS: 4 OPERATING_SYSTEM_ARCHITECTURE: amd64 TOTAL_SWAP_SPACE: 1.2881752064E10 IS_SCHEDULER_STANDBY: 0 IS_SCHEDULER_SHUTDOWN: 0 IS_SCHEDULER_STARTED: 1 TOTAL_DISK_SPACE: 1937131741184 UNALLOCATED_DISK_SPACE: 1819795460096 PREV_FIRE_TIME: 2014-12-15 23:09:16 CPU: Intel(R) Xeon(R) CPU E5-2640 v2 @ 2.00GHz 	<p>192.168.0.70</p> <ul style="list-style-type: none"> LAST_CHECK: 2014-12-16 11:29:43 SCHEDULER_INSTANCE_ID: hadoopnode01c141871882292 CPU_LOAD: 0.16330841042537914 FREE_PHYSICAL_MEMORY: 8749056560 JOBS_EXECUTED: 0 SCHEDULER_NAME: SCE CURRENT_TIME: 2014-12-16 11:29:58 JOBS/h: 0 RUNNING SINCE: 2014-12-16 09:34:42 CLUSTERED: 1 PERSISTENCE: 1 REMOTE_SCHEDULER: 0 CURRENTLY_EXECUTING_JOBS: 0 CPU_LOAD_JVM: 8.870865543023698E-4 SYSTEM_LOAD_AVERAGE: 0.89 OPERATING_SYSTEM_VERSION: 3.13.0.24-generic COMMITTED_VIRTUAL_MEMORY: 3679342592 OPERATING_SYSTEM_NAME: Linux FREE_SWAP_SPACE: 12508909568 PROCESS_CPU_TIME: 35000000000 TOTAL_PHYSICAL_MEMORY: 1.260085248E10 NUMBER_OF_PROCESSORS: 4 OPERATING_SYSTEM_ARCHITECTURE: amd64 TOTAL_SWAP_SPACE: 1.2881752064E10 IS_SCHEDULER_STANDBY: 0 IS_SCHEDULER_SHUTDOWN: 0 IS_SCHEDULER_STARTED: 1 TOTAL_DISK_SPACE: 2321541779456 UNALLOCATED_DISK_SPACE: 1937341145088 USABLE_DISK_SPACE: 1820004864000 PREV_FIRE_TIME: 2014-12-15 23:09:15 CPU: Intel(R) Xeon(R) CPU X3470 @ 2.93GHz 	<p>192.168.0.92</p> <ul style="list-style-type: none"> LAST_CHECK: 2014-12-16 11:29:23 SCHEDULER_INSTANCE_ID: hadoopnode01c1418718921761 CPU_LOAD: 0.09430552637108637 FREE_PHYSICAL_MEMORY: 7336054784 JOBS_EXECUTED: 0 SCHEDULER_NAME: SCE CURRENT_TIME: 2014-12-16 11:29:58 JOBS/h: 0 RUNNING SINCE: 2014-12-16 09:35:21 CLUSTERED: 1 PERSISTENCE: 1 REMOTE_SCHEDULER: 0 CURRENTLY_EXECUTING_JOBS: 0 CPU_LOAD_JVM: 0.0010086152553057364 SYSTEM_LOAD_AVERAGE: 0.46 OPERATING_SYSTEM_VERSION: 3.13.0.24-generic COMMITTED_VIRTUAL_MEMORY: 3679342592 OPERATING_SYSTEM_NAME: Linux FREE_SWAP_SPACE: 12312961024 PROCESS_CPU_TIME: 36300000000 TOTAL_PHYSICAL_MEMORY: 1.260085248E10 NUMBER_OF_PROCESSORS: 4 OPERATING_SYSTEM_ARCHITECTURE: amd64 TOTAL_SWAP_SPACE: 1.2881752064E10 IS_SCHEDULER_STANDBY: 0 IS_SCHEDULER_SHUTDOWN: 0 IS_SCHEDULER_STARTED: 1 TOTAL_DISK_SPACE: 2321541849088 UNALLOCATED_DISK_SPACE: 1938246713344 USABLE_DISK_SPACE: 1820910432256 PREV_FIRE_TIME: 2014-12-15 23:09:15 CPU: Intel(R) Xeon(R) CPU X3470 @ 2.93GHz
---	---	--	---	--	--	--

CPU: 18.01 GHz

CPU Load: 1.48 GHz (8.19%)

Mem Tot: 82.15 GB

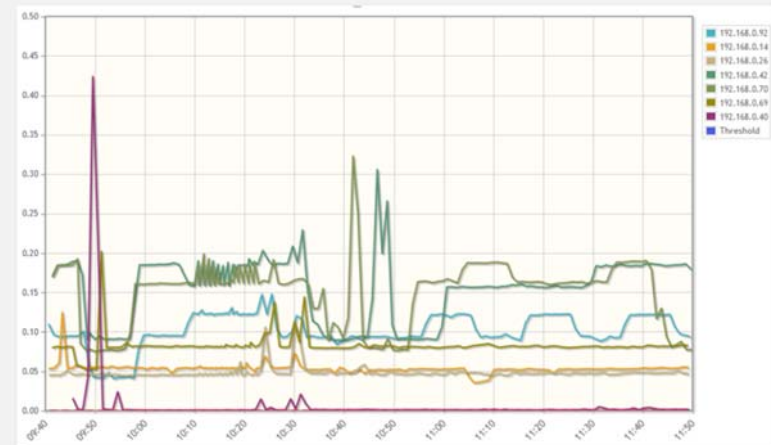
Mem Free: 45.47 GB

Cores: 28

Jobs/h: 14.91

192.168.0.72: (root@cs28) # cat /etc/passwd | grep OPERATING_SYSTEM_ARCHITECTURE | sed -n 1p | sed -e 's/192.168.0.72/

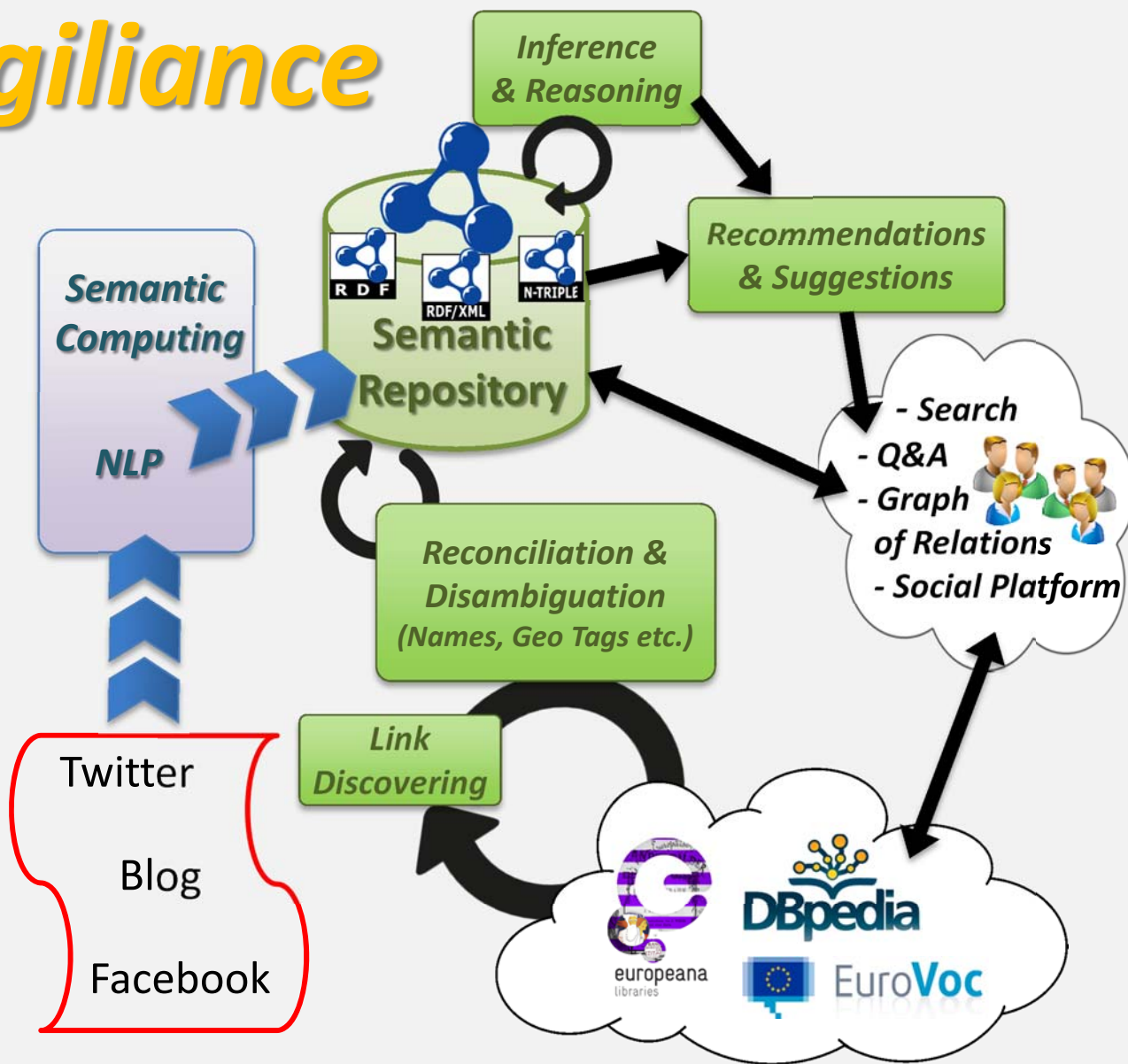
<http://192.168.0.72>



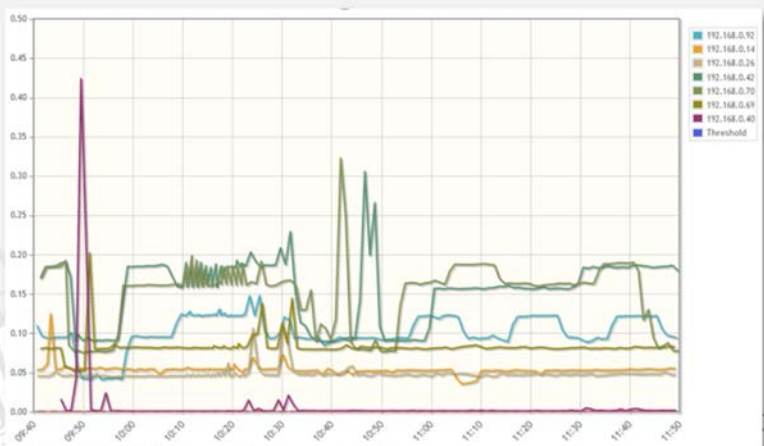
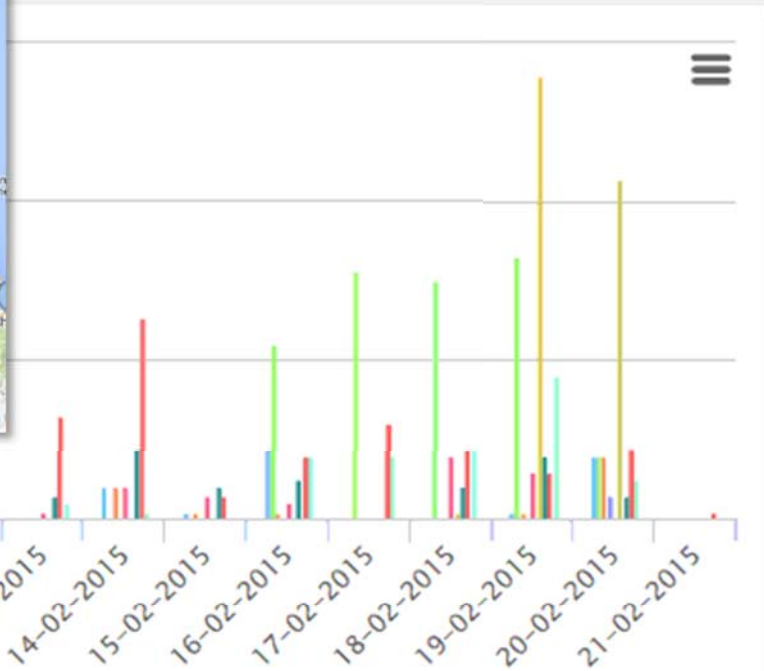
Macro Class	Static Triples	Reconciliation Triples	Real Time Triples Loaded	Total on 1.5 months
Administration	2.431	0	--	2.431
Metadata of DataSets	416	0	--	416
Point of Interest (35.273 POIs in Tuscany)	471.657	34.392	--	506.049
Street-guide (in Tuscany)	68.985.026	0	--	68.985.026
Local Public Transport (<5 lines of FI)	644.405	2.385	135.952 per line per day, to be filtered, read every 30 s, they respond in minutes	(static) 646.790
Sensors (<201 road sensors, 63 scheduled every two hours)	--	4.240	102 per sensor per read, every 2 hours, they are very slow in responding	51.111.078
Parking (<44 parkings, 12 scheduled every 30min)	--	1.240	7920 per park per day, 3 read per hour, they respond in seconds	
Meto (286 municipalities, all scheduled every 6 hours)	--	--	185 per location per update, 1-2 updates per day	
Temporal events, time stamp	--	--	6 for each event	
Total	70.103.935	42.257		122.966.893

NLP e Blog Vigilance

- Recuperare informazioni dagli utenti
- Validare le informazioni fornite da siti e utenti in relazione a quelle divulgate da siti istituzionali
- Inserire le informazioni estratte nella base di conoscenza semantica km4city per arricchire i dati
- Fornire le informazioni arricchite agli utenti attraverso il ServiceMap, un portale web, un blog o i social network come Twitter



Twitter Vigilance



- #fodd
- #ODDIT15#Firenze
- #previsionimeteo
- @flash_meteo
- #meteo
- #OpenDataDay#Firenze
- @comunefi
- @firenzedigitale
- #meteo#neve
- @UNI_FIRENZE

Major topics addressed

- From RDF to OWL
- Knowledge engineering for Beginners
- Smart Cloud Application (ICARO Case)
- Big Data Smart City Architecture
- Smart-city Ontology
- Data Ingestion and Mining
- Distributed and real time processes
- RDF processing ←
- Smart City Engine
- Development Interfaces
- Sii-Mobility

RDF Store Validation

- Some of the produced and addressed triples in indexing could not be loaded and indexed since the code can be wrong or for the presence of noise and process failure
- A set of queries applied to verify the consistency and completeness, after new re-indexing and new data integration
 - I.e.: the KB regression testing !!!!!
 - Success rate is presently of the 99,999%!
- Non loaded: 0,00000638314%

Comparing different reconciliation approaches based on

- SILK link discovering language
- SPARQL based reconciliation described above

Method	Precision	Recall	F1
SPARQL –based reconciliation	1,00	0,69	0,820
SPARQL -based reconciliation + additional manual review	0,985	0,722	0,833
Link discovering - Leveisthein	0,927	0,508	0,656
Link discovering - Dice	0,968	0,674	0,794
Link discovering - Jaccard	1,000	0,472	0,642
Link discovering + heuristics based on data knowledge + Leveisthein	0,925	0,714	0,806
QI - Link discovering – Dice	0,945	0,779	0,854
QI - Link discovering – Jaccard	1,000	0,588	0,740
QI - Link discovering + heuristics based on data knowledge + Leveisthein	0,892	0,839	0,865

Thus automation of reconciliation is possible and produces acceptable results!!



Localization via web crawling

- Using the **Ge(o)Lo(cator)** framework:
 - Mining, retrieving and geolocating web-domains associated to companies in Tuscany (thanks to a Distribute Web Crawler based on Apache Nutch + Hadoop)
 - Extraction of geographical information based on a hybrid approach (thanks to Open Source **GATE** Framework + using external gazetteers)
 - Validation in 2 steps: Evaluation of Complete Address Array Extraction, Evaluation of Geographic Coordinate Extraction
- New services found, can be transformed into RDF triples and added to the repository!

VIP names identification

- Searching RDF and/or MySQL stores looking for VIP names (citations) into strings:
 - *Via Leonardo Da Vinci*
 - *Piazza Lorenzo il Magnifico*
 - *Palazzo Medici Riccardi*
 - *Etc.*
- The idea is to link those entities with LD/LOD information as dbPedia information and



VIP names identification

Synonyms

Enter here a new name

a b c **D** e f g h i j k l m n o p q r s t u v w x y z X

'D' Pages: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#)
[22](#) [23](#) [24](#) [25](#) [26](#) [27](#)

All Dbpedia
 EclapU Both
 Norm

STATISTICHE
SETTINGS

Daria Guerrini	<input type="radio"/>	<input type="checkbox"/>
Daria Marušića	<input type="radio"/>	<input type="checkbox"/>
Daria Menichetti	<input type="radio"/>	<input type="checkbox"/>
Daria Panettieri	<input type="radio"/>	<input type="checkbox"/>
Darimonde Odette	<input type="radio"/>	<input type="checkbox"/>
Dario Abela	<input type="radio"/>	<input type="checkbox"/>
Dario Aggioli	<input type="radio"/>	<input type="checkbox"/>
Dario Antiseri	<input type="radio"/>	<input type="checkbox"/>
Dario Argento	<input type="radio"/>	<input type="checkbox"/>
Dario Benedetti	<input type="radio"/>	<input type="checkbox"/>
Dario Cincillà	<input type="radio"/>	<input type="checkbox"/>
Dario Di Donato	<input type="radio"/>	<input type="checkbox"/>
Dario Ferrara	<input type="radio"/>	<input type="checkbox"/>
Dario Fo	<input type="radio"/>	<input type="checkbox"/>
D. Fo	<input type="radio"/>	<input checked="" type="checkbox"/>
Dario Fo	<input type="radio"/>	<input checked="" type="checkbox"/>
Fo Dario	<input type="radio"/>	<input checked="" type="checkbox"/>
Dario Gessati	<input type="radio"/>	<input type="checkbox"/>
Dario Giannozzi	<input type="radio"/>	<input type="checkbox"/>
Dario Giannini	<input type="radio"/>	<input type="checkbox"/>

140544(20 mostrate) occorrenze di: Dario Fo

Axoid: urn:axmedis:00000:obj:e75313b5-90ab-4176-baab-aab19375eafc
Field: Group
Value: Dario Fo & Franca Rame Archive, CTFR, Milano, Italia
Link: <http://www.eclap.eu/urn:axmedis:00000:obj:e75313b5-90ab-4176-baab-aab19375eafc>

Axoid: urn:axmedis:00000:obj:e3edb41e-cb17-4ed5-a973-d516b000749b
Field: Group
Value: Dario Fo & Franca Rame Archive, CTFR, Milano, Italia
Link: <http://www.eclap.eu/urn:axmedis:00000:obj:e3edb41e-cb17-4ed5->

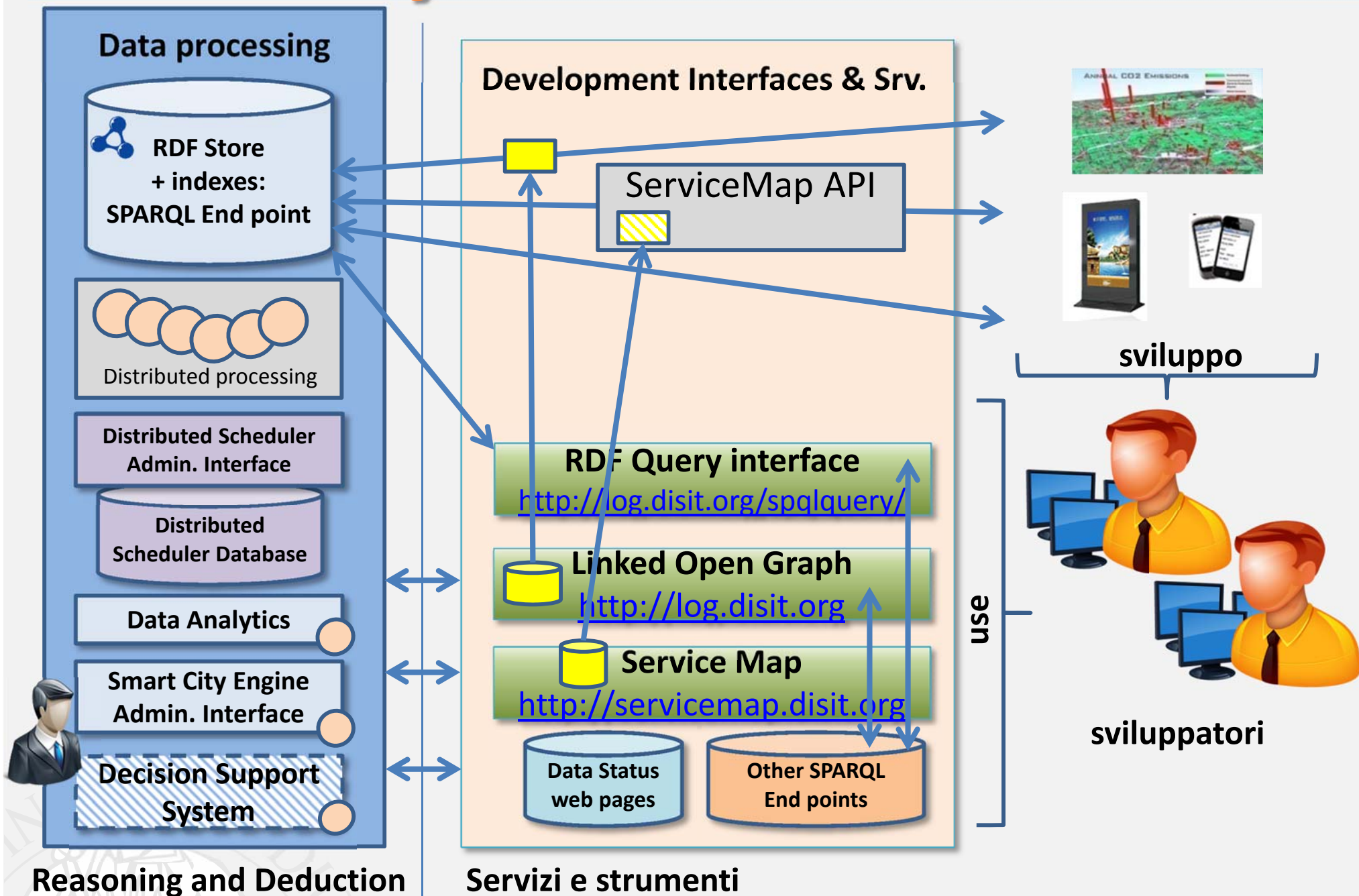
1 info di: Dario Fo

DBpedia Eclap

http://dbpedia.org/resource/Dario_Fo

Major topics addressed

- From RDF to OWL
- Knowledge engineering for Beginners
- Smart Cloud Application (ICARO Case)
- Big Data Smart City Architecture
- Smart-city Ontology
- Data Ingestion and Mining
- Distributed and real time processes
- RDF processing
- Smart City Engine ←
- Development Interfaces
- Sii-Mobility



<https://play.google.com/store/apps/details?id=org.disit.fodd>



Open Data Day App Menu

- Programma
- Servizi Vicini
- Previsioni Meteo
- Stato alla Pensilina
- Parcheggio Stazione
- Sensore Empoli
- Leggimi
- Exit

Servizi Vicini

Previsioni Meteo

FIRENZE
sereno
Ultimo aggiornamento 2

Prossimi giorni

- Martedì**
sereno
- Mercoledì**
sereno
- Giovedì**
sereno
- Venerdì**
nuvoloso

Sensore Empoli

Informazione Tempo Reale Sensore EM0100102
VIALE GIOVANNI BOCCACCIO - EMPOLI

Aggiornamento del 2015-02-21T01:00:00.000+01:00

Distanza Media (m)	585.90
Temp Medio (s)	63.20
Occupazione (%)	0.01
Concentrazione (auto/km)	1.00
Flusso (auto/h)	9.00
Velocità Media (Km/h)	35.22
Soglia (%)	0.00
Velocità Percentile (%)	Not Available

Ponte Vecchio (DL)

Ponte Vecchio

Costruito in epoca romana, il Ponte fu più volte danneggiato dalle alluvioni e ricostruito e fu l'unico ponte a non essere distrutto nel agosto 1944 dalle mine tedesche. La struttura fu rialzata nel 1345 con tre ampi valichi: arco ribassato e aveva il passaggio calpestio fiancheggiato da due file lunghe botteghe legate al commercio alimentare

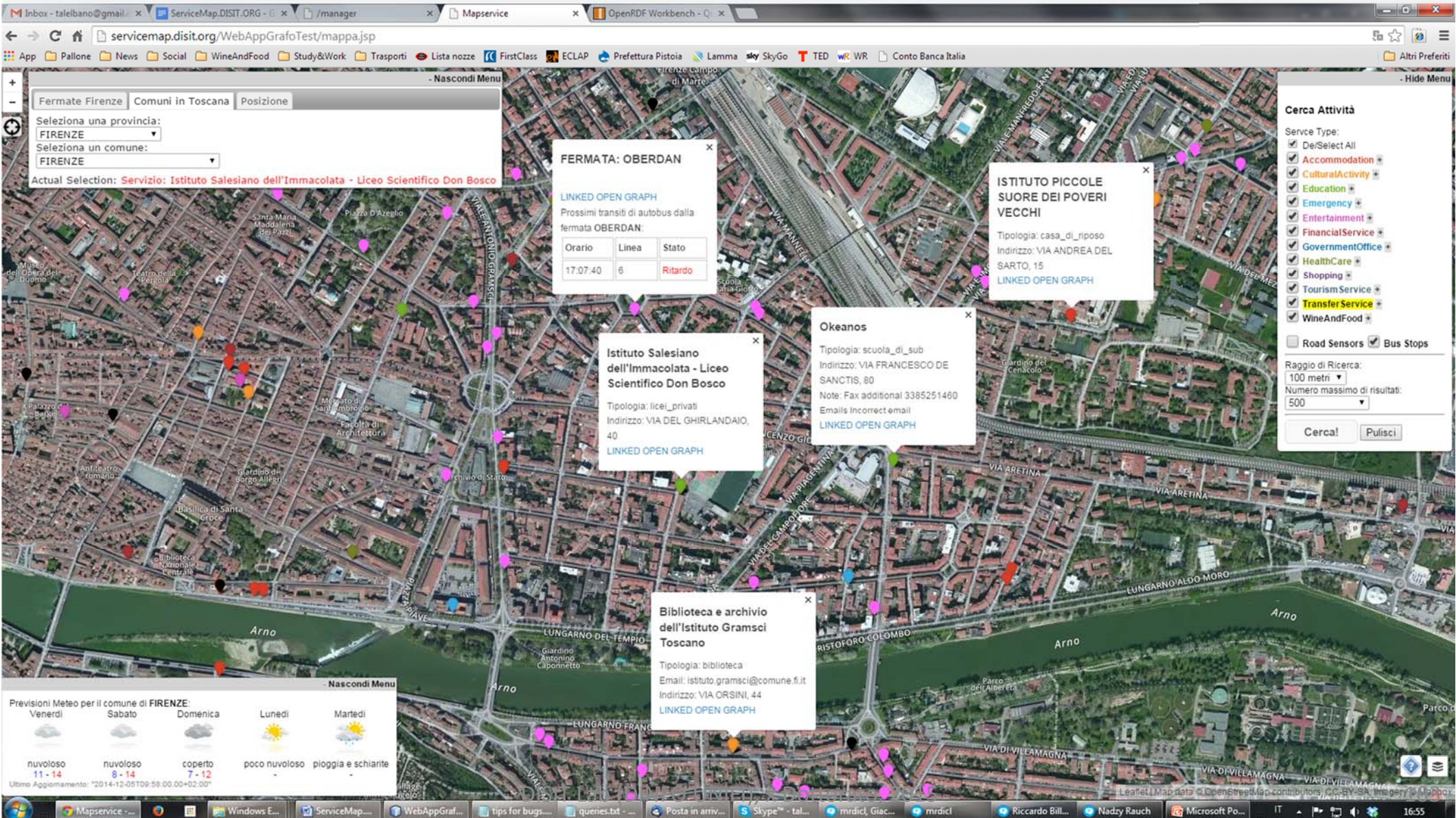
Major topics addressed

- From RDF to OWL
- Knowledge engineering for Beginners
- Smart Cloud Application (ICARO Case)
- Big Data Smart City Architecture
- Smart-city Ontology
- Data Ingestion and Mining
- Distributed and real time processes
- RDF processing
- Smart City Engine
- Development Interfaces ←
- Sii-Mobility

Development Interfaces

- **Service map:** <http://servicemap.disit.org>
 - service based on OpenStreetMaps that allows to search services available in a preset range from the selected bus stop.
- **Linked Open Graph:** <http://log.disit.org>
 - a tool developed to allow exploring semantic graph of the relation among the entities. It can be used to access to many different LOD repository.
 - To query Europeana, ECLAP, Getty, Camera, Senato, Cultura Italia, ...-> digital location
- **Ontology Documentation:** <http://www.disit.org/6507>,
 - <http://www.disit.org/5606>, <http://www.disit.org/6461>
- **Data Status Web pages:**
 - active, you have to be registered on www.disit.org, smart city group, send an email to info@disit.org with the request
- **LOD UNIFI as RDF Stores:**
 - OSIM: to access at UNIFI open data as RDF store on UNIFI competence: <http://osim.disit.org>
 - ECLAP: to provide access to Performing arts data <http://www.eclap.eu>
- **Visual Query Graph:** under development
- **SCE as Decision Support System:** under development

<http://servicemap.disit.org>



The screenshot shows the Service Map web application interface. The main map displays a satellite view of Florence, Italy, with various service markers (colored dots) overlaid. Several pop-up windows provide detailed information about specific services:

- FERMATA: OBERDAN**:
 - LINKED OPEN GRAPH
 - Prossimi transiti di autobus dalla fermata OBERDAN:

Orario	Linea	Stato
17.07.40	6	Ritardo
- ISTITUTO PICCOLE SUORE DEI POVERI VECCHI**:
 - Tipologia: casa_di_riposo
 - Indirizzo: VIA ANDREA DEL SARTO, 15
 - LINKED OPEN GRAPH
- Okeanos**:
 - Tipologia: scuola_di_sub
 - Indirizzo: VIA FRANCESCO DE SANCTIS, 80
 - Note: Fax additional 3385251460
 - Emails Incorrect email
 - LINKED OPEN GRAPH
- Istituto Salesiano dell'Immacolata - Liceo Scientifico Don Bosco**:
 - Tipologia: licei_privati
 - Indirizzo: VIA DEL GHIRLANDAIO, 40
 - LINKED OPEN GRAPH
- Biblioteca e archivio dell'Istituto Gramsci Toscano**:
 - Tipologia: biblioteca
 - Email: istituto.gramsci@comune.fi.it
 - Indirizzo: VIA ORSINI, 44
 - LINKED OPEN GRAPH

Other interface elements include:

- Search and Filter Panel (Cerca Attività)**:
 - Seleziona una provincia: FIRENZE
 - Seleziona un comune: FIRENZE
 - Actual Selection: Servizio: Istituto Salesiano dell'Immacolata - Liceo Scientifico Don Bosco
 - Search Type: De/Select All
 - Checked categories: Accommodation, CulturalActivity, Education, Emergency, Entertainment, FinancialService, GovernmentOffice, HealthCare, Shopping, TourismService, TransferService, WineAndFood.
 - Search radius: 100 metri
 - Maximum results: 500
 - Buttons: Cerca!, Pulisci
- Weather Forecast (Previsioni Meteo)**:
 - Venerdì: nuvoloso 11 - 14
 - Sabato: nuvoloso 8 - 14
 - Domenica: coperto 7 - 12
 - Lunedì: poco nuvoloso
 - Martedì: pioggia e schiarite
- Map Navigation and Controls**:
 - Map service: Mapservice
 - Map type: Satellite
 - Map scale: 1000m
 - Map coordinates: 43.7711, 11.2500

Linea 4

FERMATA : GUAISTI

[LINKED OPEN GRAPH](#)

Prossimi transiti di autobus dalla fermata GUAISTI:

Orario	Linea	Stato	Ride
09:08:02	4	Ritardo	4854072
09:08:26	4	Ritardo	4854042
09:14:16	4	Ritardo	4854112
09:15:50	4	In orario	4854077
09:25:18	4		
09:25:48	4		

FERMATA : STATUTO 04

[LINKED OPEN GRAPH](#)

Prossimi transiti di autobus dalla fermata STATUTO 04:

Orario	Linea	Stato	Ride
09:09:04	4	Ritardo	4854072
09:09:28	4	Ritardo	4854042
09:15:06	4	Ritardo	4854112
09:16:52	4	In orario	4854077
09:26:20	4	In orario	4854020
09:26:50	4	In orario	4853998

FERMATA : FABBRONI

[LINKED OPEN GRAPH](#)

Prossimi transiti di autobus dalla fermata FABBRONI:

Orario	Linea	Stato	Ride
09:09:24	4	In orario	4854020
09:10:42	4	In orario	4853998
09:19:24	4	Anticipo	4854133
09:20:48	4	In orario	4854126

FERMATA : GIOIA

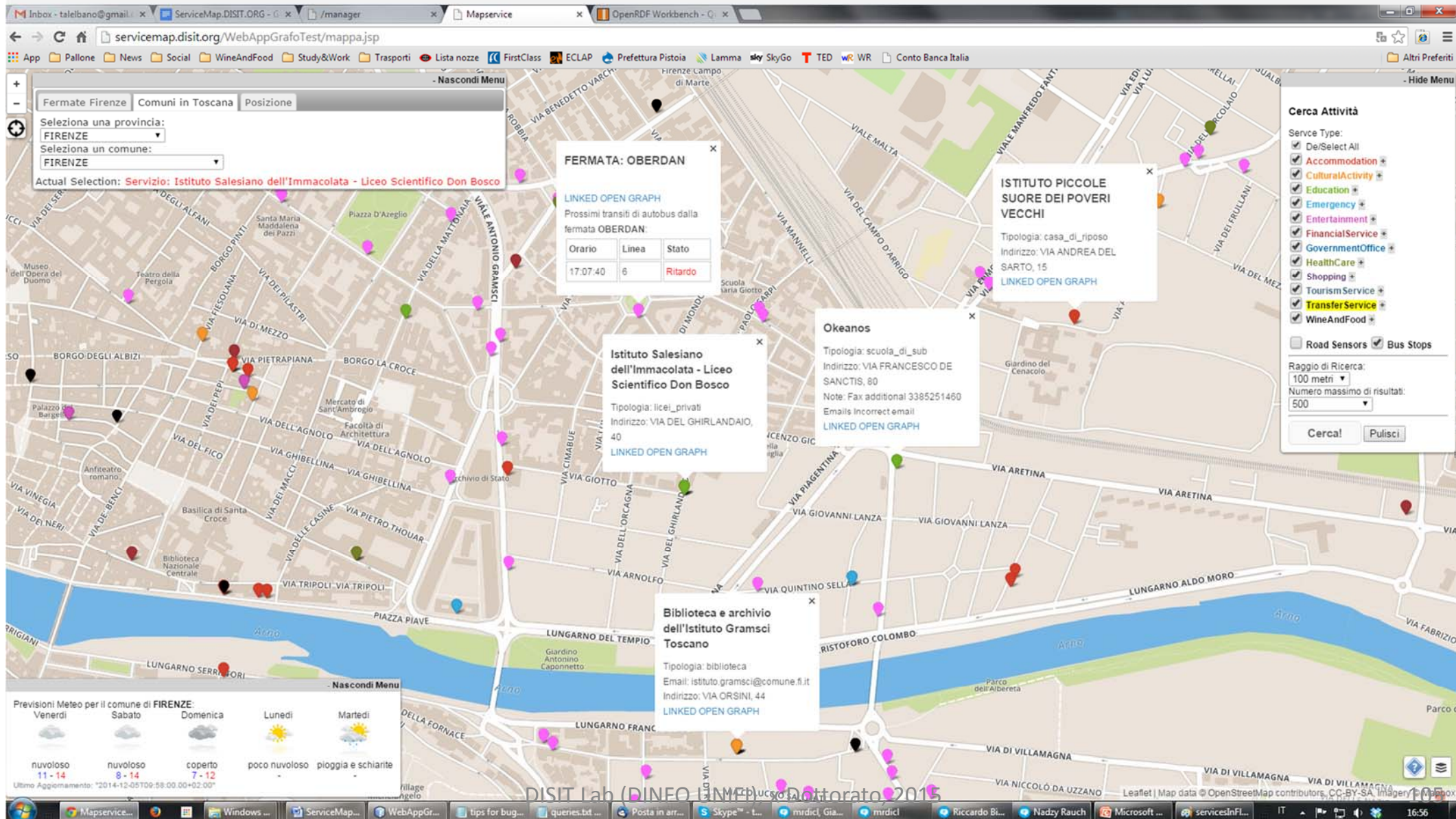
[LINKED OPEN GRAPH](#)

Prossimi transiti di autobus dalla fermata GIOIA:

Orario	Linea	Stato	Ride
09:10:22	4	In orario	4854020
09:11:40	4	In orario	4853998
09:20:22	4	Anticipo	4854133
09:21:46	4	In orario	4854126



<http://servicemap.disit.org>



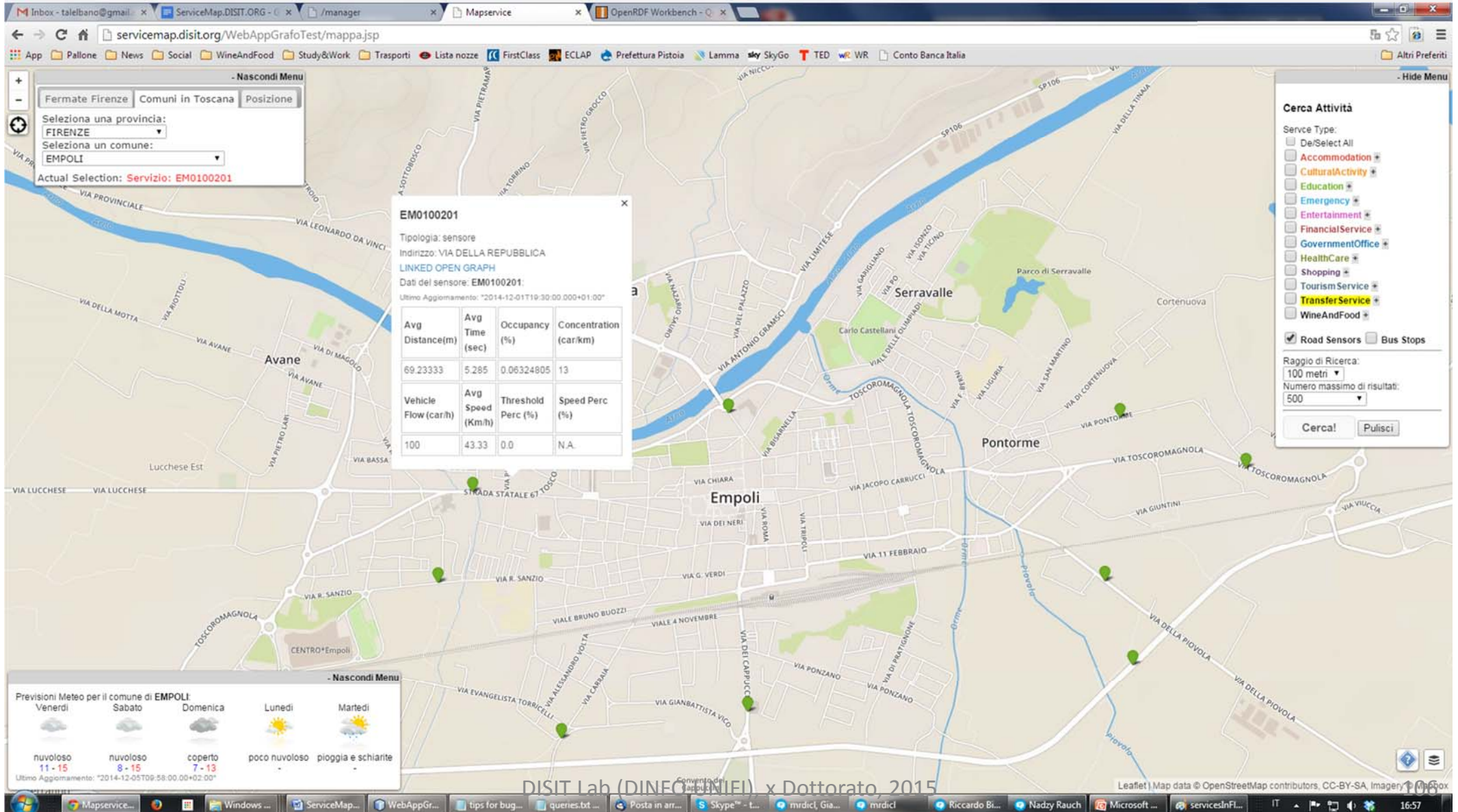
The screenshot shows the ServiceMap web application interface. The browser address bar displays `servicemap.disit.org/WebAppGrafoTest/mappa.jsp`. The main map area shows a street view of Florence, Italy, with various colored markers representing different services. Several pop-up information boxes are visible:

- FERMATA: OBERDAN**: A table showing bus transit information for the Oberdan stop.

Orario	Linea	Stato
17.07.40	6	Ritardo
- ISTITUTO PICCOLE SUORE DEI POVERI VECCHI**: Information for a house of prayer located at Via Andrea del Sarto, 15.
- Okeanos**: Information for a school located at Via Francesco de Sanctis, 80.
- Istituto Salesiano dell'Immacolata - Liceo Scientifico Don Bosco**: Information for a private high school located at Via del Ghirlandaio, 40.
- Biblioteca e archivio dell'Istituto Gramsci Toscano**: Information for a library located at Via Orsini, 44.

On the right side, there is a search panel titled "Cerca Attività" (Search Activities) with a list of service types and a search button. At the bottom left, there is a weather forecast widget for Florence.

<http://servicemap.disit.org>



The screenshot shows the ServiceMap web application interface. The main map displays the area around Empoli, Toscana, with various data points and overlays. A search menu is visible on the right, and a data popup is shown for a specific sensor.

- Nascondi Menu

Fermate Firenze Comuni in Toscana Posizione

Seleziona una provincia:
 FIRENZE

Seleziona un comune:
 EMPOLI

Actual Selection: Servizio: EM0100201

EM0100201

Tipologia: sensore
 Indirizzo: VIA DELLA REPUBBLICA
 LINKED OPEN GRAPH
 Dati del sensore: EM0100201:
 Ultimo Aggiornamento: "2014-12-01T19:30:00.000+01:00"

Avg Distance(m)	Avg Time (sec)	Occupancy (%)	Concentration (car/km)
69.23333	5.285	0.06324805	13

Vehicle Flow (car/h)	Avg Speed (Km/h)	Threshold Perc (%)	Speed Perc (%)
100	43.33	0.0	N.A.

Cerca Attività

Service Type:

- De/Select All
- Accommodation
- CulturalActivity
- Education
- Emergency
- Entertainment
- FinancialService
- GovernmentOffice
- HealthCare
- Shopping
- TourismService
- TransferService
- WineAndFood

Road Sensors Bus Stops

Raggio di Ricerca:
 100 metri

Numero massimo di risultati:
 500

Cerca! Pulisci

Previsioni Meteo per il comune di EMPOLI

Venerdì	Sabato	Domenica	Lunedì	Martedì
nuvoloso 11 - 15	nuvoloso 8 - 15	coperto 7 - 13	poco nuvoloso	pioggia e schiarite

Ultimo Aggiornamento: "2014-12-05T09:58:00.000+02:00"

DISIT Lab (DINFO-UNIFI) x Dottorato 2015

Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox

<http://log.disit.org>

Linked Open Graph

log.disit.org/service/?graph=df5b467075d50005562768e...

Linked Open Graph

SiiMobility (by DISIT)

- Examples:
- VIA_GIACOMO_MATTEOTTI
 - Bagno_a_ripoli
 - Florence

Choose a class:

Search for keyword

keyword:

uri:

Your data

sparql endpoint: (optional)

uri:

Request

Status

Requests:

Remove

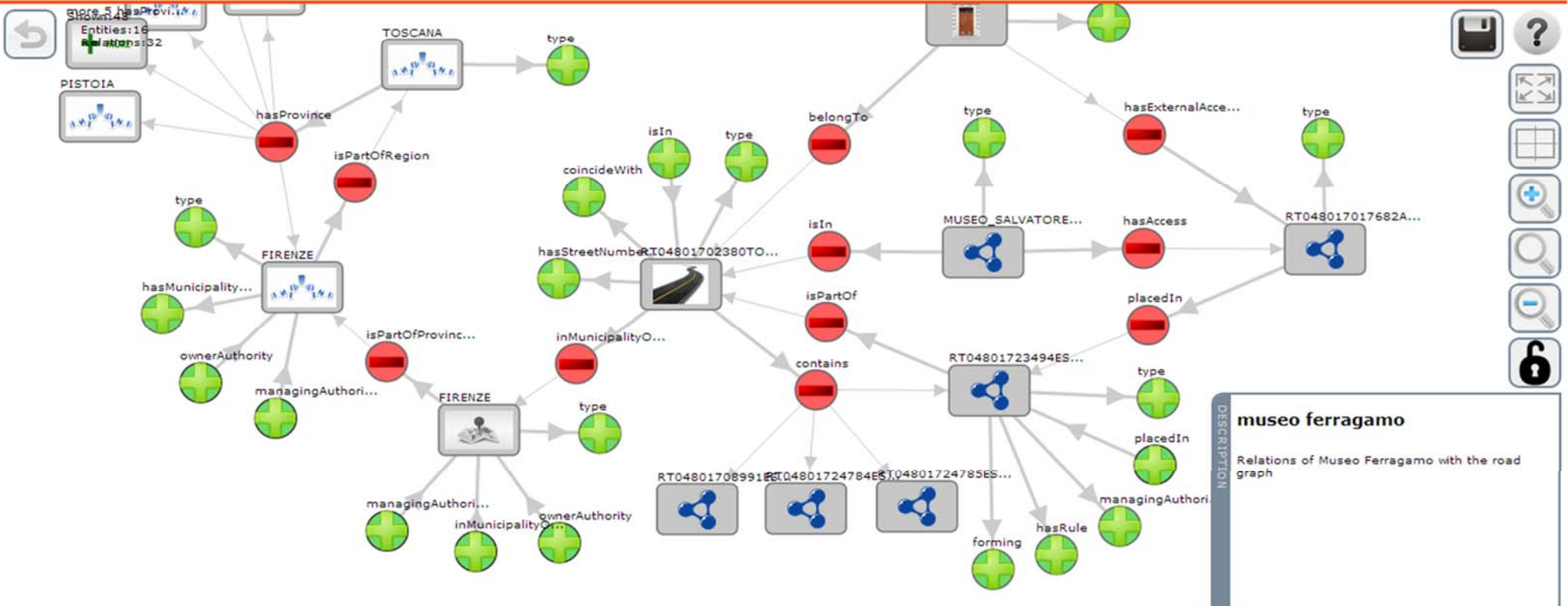
Clear

Type of relations

Select all Deselect all Invert Hide all inverse

- | | |
|---|--|
| <input checked="" type="checkbox"/> belongTo | <input checked="" type="checkbox"/> coincideWith |
| <input checked="" type="checkbox"/> contains | <input type="checkbox"/> depiction |
| <input type="checkbox"/> ends | <input checked="" type="checkbox"/> forming |
| <input type="checkbox"/> has | <input checked="" type="checkbox"/> hasAccess |
| <input checked="" type="checkbox"/> hasExternalAccess | <input checked="" type="checkbox"/> hasMunicipality |
| <input checked="" type="checkbox"/> hasProvince | <input checked="" type="checkbox"/> hasRule |
| <input checked="" type="checkbox"/> hasStreetNumber | <input checked="" type="checkbox"/> inMunicipalityOf |
| <input checked="" type="checkbox"/> isIn | <input checked="" type="checkbox"/> isPartOf |
| <input checked="" type="checkbox"/> isPartOfProvince | <input checked="" type="checkbox"/> isPartOfRegion |
| <input checked="" type="checkbox"/> managingAuthority | <input checked="" type="checkbox"/> ownerAuthority |
| <input checked="" type="checkbox"/> placedIn | <input type="checkbox"/> sameAs |
| <input checked="" type="checkbox"/> seeAlso | <input type="checkbox"/> starts |

Linked Open Graph



Linked Open Graph

<http://log.disit.org>

A bus stop info....

Linked Open Graph

Select a SPARQL endpoint:

Km4City SmartCity Ontology (by DISIT)

- dbpedia live
- British Museum
- FactForge live
- LinkedGeoData
- Europeana
- Cultura Italia
- Comune di Firenze
- Senato, Italiano
- Camera dei deputati, Italiano
- Getty Vocabularies
- Open Link SW
- IEEE Video Stanford representation
- Km4City SmartCity Ontology (by DISIT)**
- ICARO Smart Cloud Ontology (by DISIT)
- MyStory Player (by DISIT)
- OSIM UNIFI Competences (by DISIT)
- ECLAP Performing Arts Network (by DISIT)
- lodlaundromat.org
- geo.linkeddata.es

Relations:14

Linked Open Graph

Select a SPARQL endpoint:

Km4City SmartCity Ontology (by DISIT)

Examples:

- VIA GIACOMO MATTEOTTI
- Bagno a ripoli
- Florence
- Fermata di Piazza San Marco, real time status
- Empoli traffic flow sensor, real time status
- Florence, Parking at the station, real time status

Choose a class:

Search for keyword

keyword:

uri: <http://www.disit.org/km4city/resource/FM0084> Request

Multiple endpoint search

Your data

sparql endpoint: (c

uri: http://...

Multiple endpo

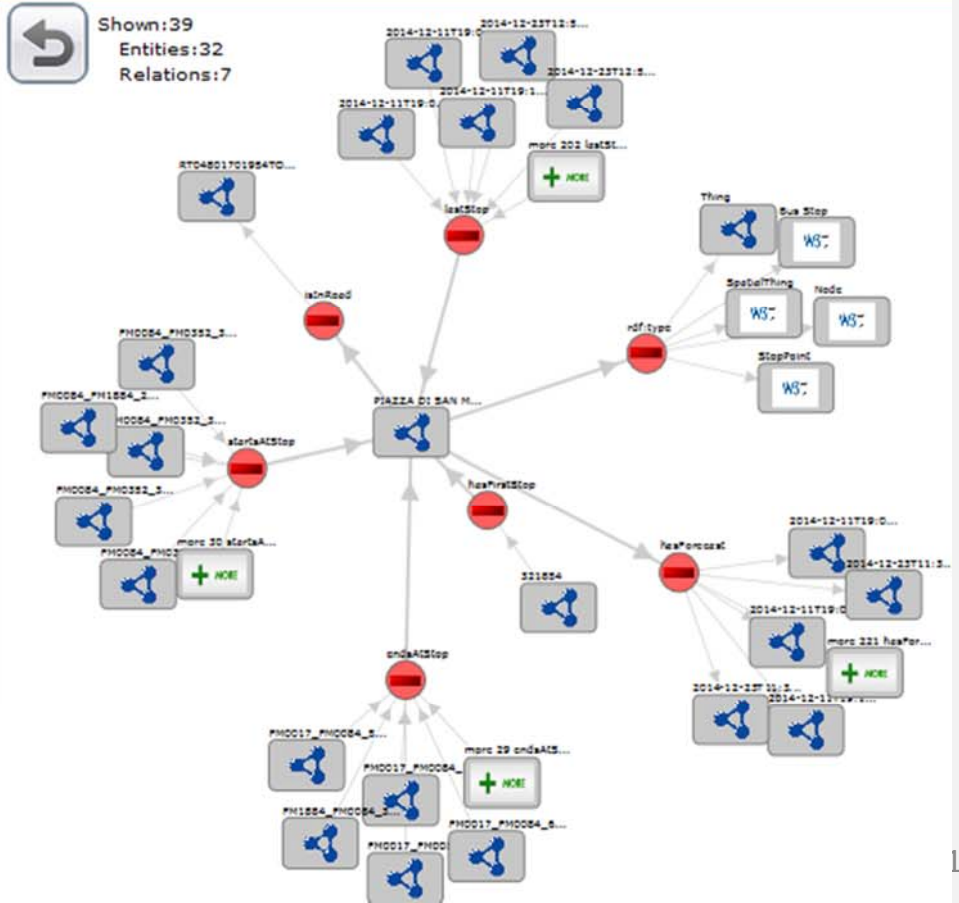
Status

Requests:

Fermata di Pi

Remove

Linked Open Graph



Strumenti: km4city e DISIT

- **Service Map:** <http://servicemap.disit.org>
- **Linked Open Graph, Multiple RDF Store Visual Browser:** <http://log.disit.org>
- **RDF Store SPARQL query tool:**
<http://log.disit.org/spqlquery/>
- **FODD 2015 applicazione dimostrativa:**
 - <http://www.disit.org/6595> (pagina)
 - Google Play
<https://play.google.com/store/apps/details?id=org.disit.fodd>
 - Sorgenti: <http://www.disit.org/6596>

Documentazione Km4City e DISIT

- API Service Map: <http://www.disit.org/6597>
- grafico: <http://www.disit.org/6507>
- articolo: <http://www.disit.org/6573>
- descrizione ITA, v2.3: <http://www.disit.org/6461>
- Descrizione ENG, v2.1:
<http://www.disit.org/5606>
- OWL e XML: <http://www.disit.org/6506>

Major topics addressed

- From RDF to OWL
- Knowledge engineering for Beginners
- Smart Cloud Application (ICARO Case)
- Big Data Smart City Architecture
- Smart-city Ontology
- Data Ingestion and Mining
- Distributed and real time processes
- RDF processing
- Smart City Engine
- Development Interfaces
- Sii-Mobility ←

Comments from
citizens, Social Media



AVM Public
transport

Materials



Sensors for
private transport

Parking
sensors



Traffic monitoring,
highways



Railways Environmental
parameters

Public Services



Ordinances: events,
large scale works
...



Sensors,
monitoring system



UTC



Infomobility



Telematic
passages, ZTL

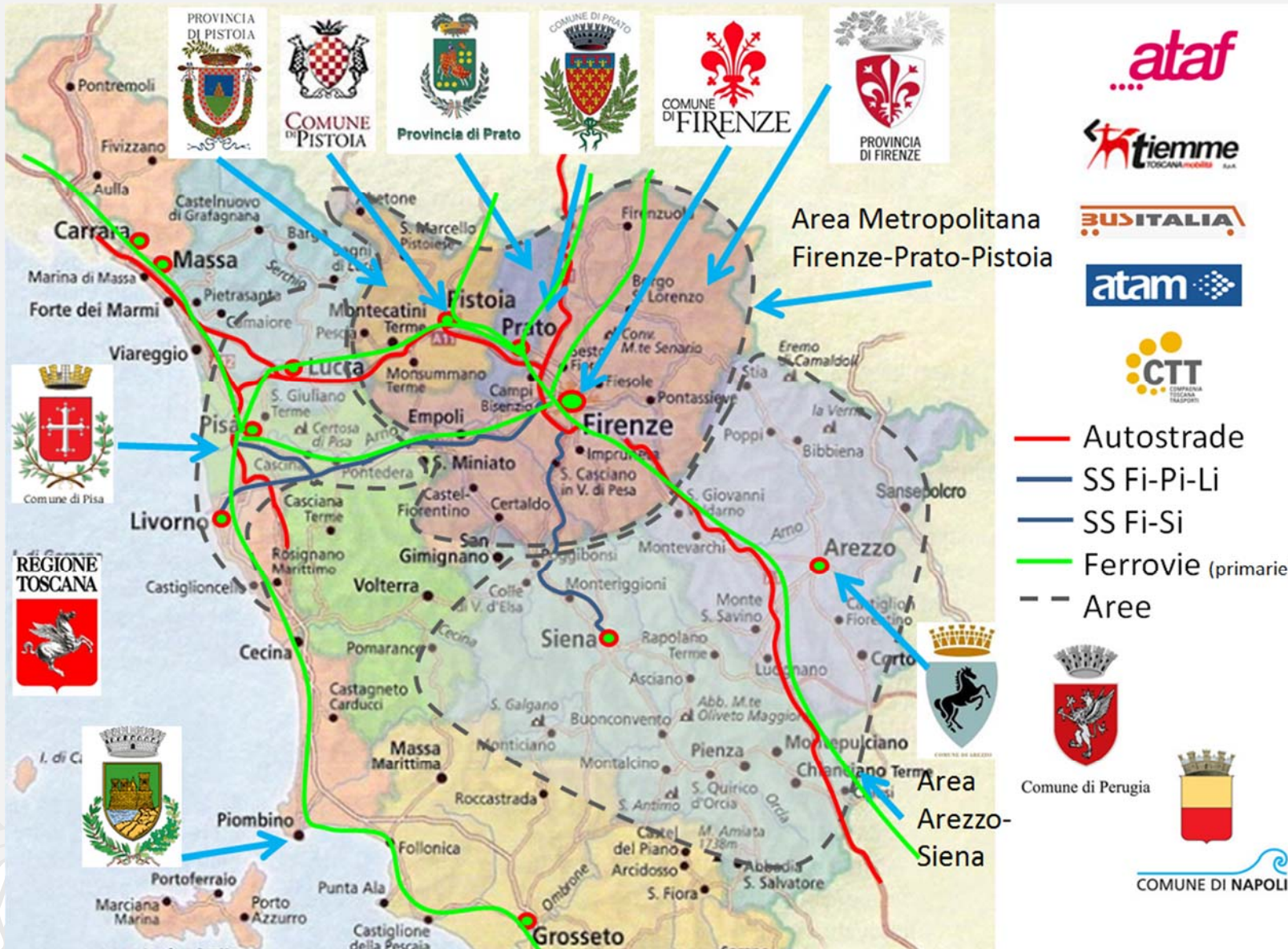
Emergency,
police, 118



Sii-Mobility: main scenarios

- solutions of connected guide / path
 - personalized services, alarms, vehicle / person receives commands and information in real time, personalized and contextualized;
- Platform of participation and awareness
 - to receive information from the citizen, the citizen as Intelligent Sensor, to inform and educate the citizen, through totem, mobile applications, web applications, etc .;
- personalized management of access policies
 - Incentive policies of deterrence and the use of the vehicle, Credit mobility, flow monitoring;
- interoperability and integration of management systems
 - contribution to standards, testing and data validation, data reconciliation, etc .;
- integration of methods of payment and identification
 - Political pay-per-use, monitoring user behavior;
- dynamic management of the boundaries of the areas controlled traffic
 - dynamic pricing and category of vehicles;
- management shared network data exchange between services (PA and private)
 - data reliability and separation of responsibilities, integration of open data, reconciliation,;
- monitoring of supply and demand of public transport in real time
 - solutions for the integration and processing of data.

- Experimentations and validation in Tuscany
- Integration with present central station and subsystems



- Smart City Group on DISIT and several slides: www.disit.org
- P. Bellini, M. Benigni, R. Billero, P. Nesi and N. Rauch, "Km4City Ontology Building vs Data Harvesting and Cleaning for Smart-city Services", International Journal of Visual Language and Computing, Elsevier, <http://dx.doi.org/10.1016/j.jvlc.2014.10.023>, P. Bellini, P. Nesi, A. Venturi, "Linked Open Graph: browsing multiple SPARQL entry points to build your own LOD views", International Journal of Visual Language and Computing, Elsevier, 2014, DOI information: <http://dx.doi.org/10.1016/j.jvlc.2014.10.003> ,
- A. Bellandi, P. Bellini, A. Cappuccio, P. Nesi, G. Pantaleo, N. Rauch, "ASSISTED KNOWLEDGE BASE GENERATION, MANAGEMENT AND COMPETENCE RETRIEVAL", [International Journal of Software Engineering and Knowledge Engineering, World Scientific Publishing Company](#), press, vol.32, n.8, pp.1007-1038, Dec. 2012, DOI: 10.1142/S021819401240013X
- P. Bellini, M. Di Claudio, P. Nesi, N. Rauch, "Tassonomy and Review of Big Data Solutions Navigation", as Chapter 2 in "Big Data Computing", Ed. Rajendra Akerkar, Western Norway Research Institute, Norway, Chapman and Hall/CRC press, ISBN 978-1-46-657837-1, eBook: 978-1-46-657838-8, july 2013, pp.57-101, DOI: 10.1201/b16014-4
- P. Nesi, G. Pantaleo and M. Tenti, "Ge(o)Lo(cator): Geographic Information Extraction from Unstructured Text Data and Web Documents", SMAP 2014, 9th International Workshop on Semantic and Social Media Adaptation and Personalization, November 6-7, 2014, Corfu/Kerkyra, Greece. technically co-sponsored by the IEEE Computational Intelligence Society and technically supported by the IEEE Semantic Web Task Force. www.smap2014.org
- P. Bellini, P. Nesi and N. Rauch, "Smart City data via LOD/LOG Service", [LOD2014](#), Workshop Linked Open Data: where are we?, organized by W3C Italy and CNR, Rome, 2014

Knowledge mining and Semantic Models: from Cloud to Smart City

x Dottorato DIST, Univ. Firenze

Pierfrancesco Bellini, Paolo Nesi

DISIT Lab

Dipartimento di Ingegneria dell'Informazione, DINFO

Università degli Studi di Firenze

Via S. Marta 3, 50139, Firenze, Italy

Tel: +39-055-2758511, fax: +39-055-2758570

<http://www.disit.dinfo.unifi.it> *alias*

<http://www.disit.org>

Pierfrancesco.bellini@unifi.it , Paolo.nesi@unifi.it