Km4City Smart City API:
an integrated support for mobility services

C. Badii, P. Bellini, D. Cenni, G. Martelli, P. Nesi, M. Paolucci
University of Florence, Department of Information Engineering,
DISIT Lab, http://www.disit.org, http://www.sii-mobility.org , paolo.nesi@unifi.it

Abstract—**The main technical issues regarding smart city solutions are related todata gathering, aggregation, reasoning, access, and service delivering via Smart City APIs (Application Program Interfaces). Aggregated and re-conciliated data (open and private, static and real time) should be exploitable by reasoning/smart algorithms for enabling sophisticated service delivering. Different kinds of Smart City APIs enable Smart City Services and Applications, while their effectiveness depends on the architectural solutions to pass from data to services for city users and operators. To this end, a comparison of the state of the art solutions for data aggregation was performed, by putting in evidence the needs of semantic interoperable aggregated data, to provide smart services. This paper presents the work performed in the context of the Sii-Mobility national smart city project on mobility and transport integrated with services. Sii-Mobility is grounded on Km4City ontology and tools for smart city data aggregation and service production. To this end, Sii-Mobility/Km4City APIs have been compared to the state of the art solutions. Finally, the API consumption related data in the recent period are presented.**

Keywords —**smart city, smart city ontology, smart city API, smart mobility, multidomain smart city, smart services.**

I. INTRODUCTION

The major companies are proposing solutions to make city smarter, focusing on specific set of domains , such as IBM [1], [2] on services for citizens, business, transport, communication, water and energy; [3] on governmental, educational, e-health, safety, energy, transport and utilities, CISCO on people, thinks and data [4], etc. Most of these solutions present a multi-tier architecture ranging from 3 to 6 layers [5]. On the other hand, the number of tiers is partially relevant to the transformation of data in value for business, and thus to services for the city users, and in opportunities for the enterprises and city operators interested in creating innovative and effective services, while exploiting city data and information [6], [7], [8]. Also, the smart city ranking models are not suitable in putting in evidence these aspects, since they are mainly focused on counting the number of provided open data sets, smart services, solutions or energy results [9], [10].

As a general consideration, the main technical issues regarding smart city solutions are related to data access, aggregation, reasoning, access and delivering services via Smart City APIs. The final aim is serving city users in a smarter and more efficient manner, stimulating their participation to the city strategies. Therefore, collected and produced data are used for facilitating the creation of smart and effective services exploiting city data and information. Specific end user smart services should be developed and managed by enterprises and city operators, rather than by the municipality. On the other hand, the municipality has to provide a flexible data access and services. This means to make effective and efficient the data access with their semantics, the service delivering, the access to dashboard control, and the interoperability with any smart control systems that can be located in the city.

In the world, municipalities / cities, and public administrations are publishing huge amount of open data. These data can be coarsely aggregated for integration by using solutions such as CKAN [11], OpenDataSoft [12], ArcGIS and OpenData [13]. These solutions for open data collection and delivering are suitable for simple indexing data/metadata sources, by providing support for browsing and query data, by exploiting descriptive metadata. In some cases, they provide access to effective integrated data sets, by using some data integration and visualization tools which provide the possibility of creating graphic charts, such as distributions or pies. In the extreme case, they also provide access to data sets as Linked Data (LD), Linked Open Data (LOD), and even from an RDF store endpoint [14] to make SPARQL queries on the data, or only on metadata [15], exploiting some ontology. The access to RDF stores for data browsing can be performed by using visual browsers as in [16]. In the case of directly accessible LOD, we are in presence of the so called 5 stars open data [17]. On the other hand, in most cases the integrated LOD are not supported by multi-domain ontologies, as explained in the sequel. We could state that 6 stars data would also provide a data access and SPARQL queries exploiting a semantic ontology for the integrated data model and data inference.

In most cases, the effectiveness of data service system for Smart City is enabled by the availability of private data owned and managed by *City Operators* addressing specific domains: mobility operator, energy providers, business services (health, water), telecom operators, tourist operators, university, etc. They are the city stakeholders providing data and services with different granularities and size. For example, in the city, we can have few energy operators with capillary house distribution, many public transport operators with thousands of vehicles/busses, some telecom operators deploying in the city from tens to hundred thousands or millions of sensors. Different granularity implies different methods for collecting and for providing access to data such as publication of open data files and/or statistics, publication of real time data with

consumptions, mobility flow, energy, weather, etc. Real Time data are provided by city operators through some APIs as Web Services or REST calls. The APIs for providing to the data aggregator of the city may be compliant with multiple standards (such as DATEX II for mobility, intelligent transport system [18] for public services, parking, IETF [19] for Internet of Things, Green Button Connect [20] for energy data collection). On the other hand, some of the peripheral data kind collected are not supported by any standard, thus custom solutions are adopted, such as the status of hospitals' emergency units, the status of heart quakes in the regional area, etc. In the E015 digital ecosystem, in order to cope with a large amount of APIs in the city, an effort was made, for setting up an environment for collecting documentation about available data via services/APIs and interfaces in the city, by using a web portal [21]. But this is not solving the data and service/API interoperability problems since one App that need multiple data has to access them using multiple protocols and then the data are still to be aggregated and made semantically uniform.

The effective deploy of smart services for city users is very frequently viable only by exploiting the semantic integration of open data, private data and real time data coming from administrations and different operators. This implies specific processes of reconciliation and the adoption of unifying data models and ontologies as in Km4City [22]. The semantic aggregation of data coming from several domains is unfeasible without a common ontology, since data are produced by different institutions/companies, by using different formats and aims, different references to geographical elements, and different standards in different moments. Thus, data sets are rarely semantically interoperable each other. In addition, they may present different licensing models: some of them can be open, while other may be private of some city operator that would not be interested to lose the ownership by releasing them into an unregulated environment, or could simply provide some restrictions (e.g., no commercial); see for example the data of car sharing companies that are typically private of the company. For open data, as well as for private data, several different licensing models can be adopted [23], [24] enabling or preventing some business models, or simply their usage.

Therefore, well aggregated and re-conciliated data (open and private) can be exploited by reasoning algorithms for enabling sophisticated service delivering [22]. For example, routing in the city, inter-modality routing, easy parking can be delivered by some personal assistants on the basis of the user profile. Moreover, in order to provide information about what is around a current GPS position, the integration of geographic information and services is needed; while the integration of geolocalized services and the assessment of typical people flows may help the city in improving public services and transport, providing suggestions to the city users, and planning changes in the city. Thus, aggregated data can be exploited to implement a large number of services and applications by structuring the Smart City Architecture and the corresponding Smart City APIs.

This paper presents the work performed in the context of the Sii-Mobility smart city project on mobility and transport features and integrated with city services in general (http://www.sii-mobility.org). Sii-Mobility project includes 24 partners from industry and research, and it is partially founded by the Italian Ministry of Research; it aims to provide innovative services for mobility operators and city users moving in the city. Among the most interesting features, it includes smart algorithms for personal assistance, solutions for dynamically shaping restricted traffic zones, the production of personalized suggestions for moving city users. These requirements necessitated a deep analysis of the state of the art solution, to identify and then develop a solution that allowed making reasoning and deduction on city data collected from city operators, as open data and private data, as static and real time data, as multiple domain data for producing suggestions.

The paper is organized as follows. Section II presents the comparison of the state of the art of data aggregation solutions. Section III presents the Sii-Mobility solution, scenarios and available data. Section IV provides a description of Smart City APIs included in Sii-Mobility, and a comparison among other APIs to those of Sii-Mobility and some examples of the most interesting API calls and constructs. In Section V reports the usage of Sii-Mobility Smart City APIs based on Km4city. Conclusions are drawn in Section VI.

II. SMART CITY API ARCHITECTURES AND COMPARISON

According to the introduction, different kinds of Smart City APIs can be set up with the aim of enabling Smart City Services and Applications, and their corresponding architectural solutions. They mainly differ each other from the strategy to transform data to services for the city (from data to business), as represented in **Figure 1**, in which three main approaches are presented. Even if some hybrid solutions may be setup they represent the most important cases. It is worth noting that, when we talk about Web and Mobile App, we mean applications that exploit data and service accesses, put in the hands of final users (city users), and/or for the City Operators. Moreover, in the following examples, the Decision Makers are the key personnel of public administrations or city operators interested in extracting from the aggregated and integrated data eventual deductions, data and/or alerts.
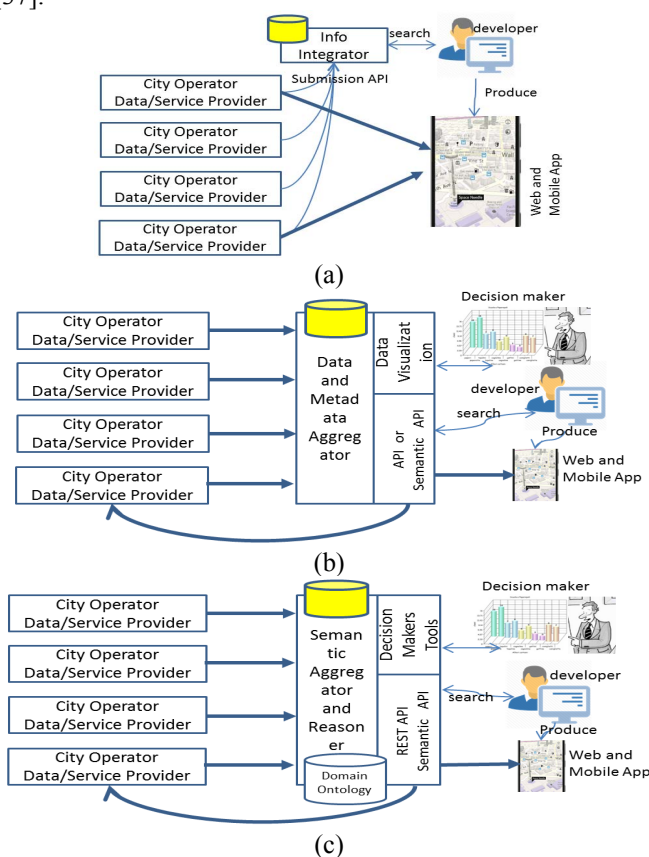
Case (a): the Info Integrator of Figure 1 collects information about APIs provided by different data and/or service providers (including their authentication and licensing), and provides a common place for developers and other city operators to browse and learn how to access at the exposed API services and data. Data/service providers can be city operators such as: mobility operators, energy operators, waste and water operators. They may provide some open and/or private data, static and/or real time data. In Case (a), the data/services are not integrated each other, each API set allows to access the specific data/service of a single operator. Thus, the API and the data are not semantically interoperable, and the problems in managing the semantic integration of data and services are left in the hands of the developers, that have to cope with

different, not harmonized APIs and providers, different authentications, and so on. The developers have to select the data, get them and integrate them every time they change. This also implies that they may have to arrange different contracts and agreement with each data and service provider. An example of this Case (a) is the E015 solution for Milan [21], where structured information is requested to the API providers via Excel files, and published in their original formats on the E015 Web portal. This approach aims to share and may be to regularize the APIs of the city operators, rather than facilitating the production of integrated services for city users. The approach enabling the passage from "*data to services*" is not well activated, since the data is not interoperable, and the licensing agreement is in multiple relationships arranged between each developer and several data providers. The solutions belonging to this category does not satisfy most of the sub-goals of the EIP document about the Urban Platforms [37].

(a)

(b)

(c)

Figure 1: From Data to Services of Smart City Architectures: comparison of different solutions and generations. (description in the text). The thicker lines carry data streams and data services.

Case (b): the Data and Metadata Aggregator of Figure 1 collects data and metadata information (mainly open data) to index and aggregate them in a common model according to the structure of the open data files and tables in input. The resulting aggregated data are made accessible to web and mobile App via some automatically generated APIs disregarding their semantics and thus the match among entities

collected from different data sets and representing the same elements. In some cases, graphic rendering is provided via some data visualization tool, presenting the similar problems of semantic. The automatically produced model does not lead to a satisfactory semantic interoperable data service [22]. The data are not re-conciliated each other, and maintain the same quality of the original, missing data are still missing. The data integration is mainly performed on the basis of syntactic and lexical forms of data names and content vales. The resulting database is typically a set of tables with traditional MySQL capabilities Thus, the provided APIs are not supported by a domain-oriented ontology. In some cases, a semantic model for data access is provided as marketing strategy for the 5 stars; and thus they provide SPARQL based API and service. The generated ontological model is a mere representation of tables and does not provide significant inferential support. This also means to have limited reasoning on time and space. Belong to this category of solutions CKAN [11], ArcGIS OpenData [13], OpenDataSoft [12] based on ArcGIS, SOCRATA [25] also based on ArcGIS. The solutions based on ArcGIS provide more capabilities on geospatial queries. Case (b) approaches can be regarded as first level solutions for data interoperability, and can be viable when mainly open data are integrated, without real-time and/or private data. The solutions compliant to Case (b) do not cover all sub-goals of the Urban Platforms [37] since data are not harmonized.

In fact, when open data have to be integrated with private data and real time data, as in the cases of urban mobility services, the solution has to cope with a number of special cases that have to be separately and manually treated, to be integrated with the rest of data, destroying the advantages of integration. A solution to these problems is the adoption of a Case (c) solution.

Case (c): the Semantic Aggregator and Reasoner (see Figure 1) collects data and services from the City Operators, to aggregate and integrate them in a unified and semantically interoperable model based on a multi-domain ontology. This approach allows re-conciliating data and exploiting a coherent model to reduce the errors, integrating data representing the same concept and coming from different structures, operators, and sources. The usage of a multi-domain ontology allows the adoption of a model representing relationships of specialization among classes and relationships, aggregation, association, and similarity, that enable the inferential processes in the RDF Graph Database [26], [22]. Thus, the obtained knowledge base can be used for creating strategies for data quality improvement and for setting up algorithms and reasoning about the several aspects and services belonging to multiple domains. This advantage is also evident on the provided API and tools for Decision Makers. For the same reason, the obtained Knowledge Base, by populating the ontology with data and inference, can be profitably and easily used for producing smart services such as routing, multimodal routing, suggestions on demand, personal assistants, connected drive, etc. As described in the following sections, fit to this latter case, partially covering all features: CitySDK [27], which provides some limitations and has been developed in an

EC project involving major cities and providing specific REST API; and OASC (Open & Agile Smart Cities) adopted the FIWARE NGSI API agnostic model [28] for producing smart city API based on CitySDK with the corresponding limitations; and more widely covering features Km4City [22] exploited by Sii-Mobility Smart City project, RESOLUTE H2020 Project and REPLICATE H2020 Project of the European Commission, providing Smarty City API of Km4City. [36], [22].

Solutions of Case (c) mainly differ from those of (b) for the presence of a real ontological model among city entity and not on data structure of the tables. The solutions of Case (c) are better ranked with respect to the sub-goals of the Urban Platforms [37] covering aspects connected to the harmonization of data, and production of intelligent services. Moreover, the implementation of user experience for value added services (subgoal 5) is only accessible in a few of them as analyzed in the following. The ontology can model city domains entities and their relationships and not only metadata of data sets and tables as in Case (b). An effective integration at semantic level of the data domain enables the creation of Smart Decision Support Systems that exploit the possibility of making semantic queries on multiple domains, to make probabilistic reasoning on Bayesian decision support [29], and to enable the production of algorithms for implementing personalized routing and Personal Assistants in the city. Case (c) solutions have to cope with Graph Database collecting huge amount of data, thus resulting in Big Data cases and scenarios presenting relevant data such as variety, velocity, veracity, volume, etc. [30].

Table I summarizes the above presented comparative analysis of the different architectures and Smart City API solutions, to enable the city to pass from data to services, thus from data to applications for the City Users and for the Decision Makers. As a final consideration, the main difference that can be perceived by the City Users and by the Decision Makers is the number of smart and cross domain services that Case (c) can provide with respect to the others.

The Case (c) category of Smart City approaches represents the new generation of solutions, and it is the most interesting area for the study reported in this paper about the resulting APIs, which are more powerful with respect to those provided by the other cases.

III. SII-MOBILITY SCENARIOS AND SOLUTION

This section presents the Sii-Mobility solution (http://www.sii-mobility.org), scenarios and available data. Terrestrial transportation systems are more often affected by congestion situations due to the fact that specific situations cannot be easily foreseen by using traditional Intelligent Transport Systems (ITS), that only cope with transport data and mobility aspects, disregarding events, energy, weather, people flow, etc. The local transportation system presents very high social costs related to the citizens' uneasiness with respect to the available mobility solutions. This is mainly due to the scarce interoperability and intelligence among management and monitoring transport systems, services for mobility, services

and systems for goods transportation, ordinances and public services (such as hospital, centers, museums), events, private transport, rail transport, car parks, and moving people, because of the limited capacity of the system to incorporate and react to changes in the city and citizens.

	Case (a) Info Integrator	Case (b) Data and Metadata Aggregator	Case (c) Semantic Aggregator and Reasoner
Addressing Open Data	Y	Y	Y
Addressing Private Data	Y	Y	Y
Addressing Real Time Data	Y	Y	Y
Addressing Services	Y	N	Y
Providing Data Search	N	Y	Y
Providing Metadata Search	Y	Y	Y
Providing Space Reasoning	N	(Y)	Y
Providing Time Reasoning	N	(Y)	Y
Providing Integrated Authenticated Access to data	Only metadata	Y	Y
Providing Syntactic Interoperable Data/Services	N	Y	Y
Providing Semantic Interoperable data/Services	N	N	Y
Independent from the Data model changes	N	N	Y
Providing REST API on data	N	Y	Y
Providing SPARQL API on data	N	(Y)	Y
Providing inference support on Data	N	N	Y
Providing Data Visualization	N	Y	Y
Providing Decision Maker Support	N	(Y)	Y

Table I: Comparison of different smart city solutions to pass from Data to Services. (Y) means yes with limited capabilities, due to the limited model of the ontology in representing aspects of data entities of the city in: time, space, multiple domains, structures, service and relationships.

With the aim of producing a smart city infrastructure for solving these problems, the Sii-Mobility project is working on enabling technologies for smart city and mobility, to integrate and produce the Support of Integrated Interoperability (SII). SII aims to create a smart city big data framework for providing integrated data, supporting services as mobile app, decision support, integrated ticketing, personal assistants, participative portals, crowd sourcing, etc. It also aims to support Data Analytics and Intelligence based on integrated data collected from Public Administrations related open data, private data from operators, and personal data coming from social media and city users.

The main objectives of Sii-Mobility are to: (i) reduce social costs of mobility; (ii) simplify the use of mobility systems; (iii) provide solutions for assisting connected drivers/people exploiting inter-modality (connect drive, smart drive or walk); (iv) define solutions for the interoperability with other smart cities' management systems; (v) define solution for city user engagement and participation and awareness, personalized management of access policies; (vi) study and propose solutions for dynamic management of RTZ boundaries; (vii)

real-time monitoring of supply and demand of public transport.

As described in the previous section, Case (c) solution is the most suitable for providing smart services as those of Sii-Mobility, provided by mobility operators as routing, personal moving assistant, geolocalized suggestions, etc.

Therefore, Sii-Mobility adopted a Case (c) solution for data aggregation and service production by exploiting and improving the Km4City Ontology (http://www.disit.org/km4city) [22], [26], as the main ontological model. According to READY4 SmartCities FP7 CSA project of the European Commission [38], Km4City is the most comprehensive smart city ontology at the state of the art (in terms of coverage), among the analyzed ontologies for smart cities. The Km4City ontology has been improved in Sii-Mobility by adding more features and details for mobility and transport. Thus, the resulting model was used as a basis of the deployed Smart City APIs by defining and developing: (i) data analytics algorithms and tools for suggestion and geolocalization, routing, etc. [31]; (ii) decision support system [29]; (iii) smart city API and producing some demonstration applications, as the Dashboard solution (http://www.disit.org/dashboard).

Km4City, via its ServiceMap [36] allows to visuallygenerate queries and Smart City API calls, which are sent via email as the programming code for the production of web and mobile applications. Km4City is modeling multiple domain aspects related to mobility, services, Wi-Fi, cultural services, energy, structure (streets, civic numbers, green areas, sensors, busses, etc.) and much more. On the basis of the integrated knowledge model, a number of services can be developed. Some examples are accessible from mobile and web applications such as the mobile App "*Florence what where*", available on all the mobile platforms. Data in the Florence area and region are coming from the Observatory for Mobility and Transport (MIIC) of the Tuscany Region, from the LAMMA consortium of the Florence Municipality, etc. These data are about mobility and transport, cultural heritage, hospitals, environment, services, emergencies, shops, tourism, wine and food services, education, wellness, etc.

IV. SMART CITY API OF SEMANTIC AGGREGATORS AND REASONERS

This section presents the Sii-Mobility Smart City APIs and compares them with other APIs provided by other solutions belonging to the Case (c) category of data aggregators. The main functionalities of the Smart City APIs for Semantic Aggregators and Reasoners, Case (c), can be classified in few groups that include general functionalities and specific functionalities strictly connected with the domain (search, mobility, environment, user participation and awareness, smart city interoperability, etc).

CitySDK, is a *service development kit* for cities and developers that aims to harmonize APIs across cities, [27]. The CitySDK SOA architecture is typically structured on SQL DB. The project makes transformations on each datasets in order to obtain and manage uniform data. There are no semantic relationships among the data collected from data providers; only some of the links between an event and the Point of Interest to which it refers are established. The work on interoperability is limited at the API level, and in the drafting of guidelines and standards for the data providers and the API users.

ECIM has been derived from EPIC [32], [33], and it has been developed to cover needs of public administrations and small businesses for citizens. The ECIM approach aggregates public and private services to combine them creating and validating new ones. It has been piloted on Brussels, Paris, Barcelona, in the context of an EC project. In this case, there are no semantic descriptors related to the offered services, neither to the datasets. The information coming from data providers are typically converted from their native formats to JSON, and put in a common ECIM MySQL database. Therefore, the ECIM solution differs from the E015 solution [21], since ECIM operates a partial API integration. The offered services can be either open or at payment. The ECIM project focuses on mobility, and the participatory actions are planned with the purpose of creating new services meeting citizens' needs.

Transport.API [34] is a new service for providing aggregated open data in the UK. This startup makes available, via a Rest API system a relevant number of datasets integrating both static and real-time data, mainly regarding mobility aspects. For these reasons, it can only be partially classified as a Case (c) solution. Its main limitations are related to the low integration with other domain data, and with the provided licensing model, being a profit service unsuitable for the smart cities.

Navitia.io is an open source project exploiting OpenDataSoft for open data aggregation in France [35]. The Navitia.io interoperability is at the API level; the datasets are collected from various domains, aggregated into a single database, classified by broad categories (a taxonomy applied on the metadata describing each dataset). There is no uniform semantic model describing and interconnecting the datasets themselves: reconciliation or quality improvement mechanisms are completely missing. For these reasons, it can only be partially classified as a Case (c) solution. Moreover, the service is not hosted on a powerful server, or it is not scalable, thus the users are asked to limit their requests to one per second.

Km4City ontological model and data aggregation [22], with its Smart City API, has been recently augmented with results of the Sii-Mobility project on mobility and transport. Thus, it enables the development and deploy of personal assistant services, for example to implement Mobile Apps for city users with multi-domain information, and at the same time collects data from the users about their preferences and behavior. Km4City can be adopted by city operators and industries to create web and mobile applications, and to exploit published/integrated open and private data, static and real time data. On the other hand, it can be easily extended and applied to different smart cities.

In **Table II**, the comparison of the above commented Smart City API approaches is presented, based on the main API

functionalities and services exposed. From the comparison, it is evident that the major difference has been registered for Km4City/Sii-Mobility that provides support to (i) perform semantic queries with inference on aggregated data, (ii) realize personal assistant, and (iii) implement dashboards.

Front end Smart City API domains to provide services to management smart city applications, and to web and mobile applications.	CitySDK	ECIM	Transport.API	Navitia.io	Km4City
API: Service Search					
Search Full Text	x2	x2		X	X
Search around a GPS point	X	X	X	X	X
Search along a line, polyline	X				X
Search in an area, set of points	X			X	X
Search for region, municipality, etc.				X	X
API: Mobility					
Get Real time delay of Public Busses	X		X	X	X
Get Traffic Flows Status	X	X	X	X	X
Get Parking Status	X	X		X	X
Get a Routing (multi stop planning)	X	X	X		(x)
Get an Intermodal Routing	X	X	X	X	
Get an Integrated Ticketing		X	X	X	(x)
Get a Routing for Good Delivering	X		X		
API: Environment, Sensors and Actuators					
Get Weather Forecast	X			X	X
Get Sensor/Actuator Value/Status	X	X		X	X
API: User Participation and Awareness					
Get Social Media Monitoring Info			X		X
Save Crowd Sourcing Comments	x1		X		X
Save Crowd Sourcing Votes	X	X	X		X
Save Crowd Sourcing Media	X	X	X		X
Get Events in the city/area	X				X
API: Personal Assistant					
Save User Profile	(x)	(x)		(x)	X
Get Suggestions on Demand					X
Get Civil Protection in Push					X
Save Mobile Sensors Status	X	X		X	X
API: Smart City Interoperability and Dash Board					
Save Indicator Values (GPS, CompanyID)					X
Get Indicator Values (Company, ID)					X
API: Domains of Geo Located Services					
Culture and Tourism	X	X			X
Point of Interest	X	X	X	X	X
Mobility and transport, parking, flow	X	X	X	X	X
Education and training	X	X			X
Government and Pub Services	X	X			X
Commerce and Industry	X	X			X
Health and personal	X	X			X
Public Energy, Energy and home	X	X			X
Energy and Mobility	X	X	X		X
API kind of Call					
SPARQL Query					X
SPARQL Query with Inference					X
REST	X	X	X	X	X
Query ID					X
Non Functional					
Direct API Authentication	X	X	X	X	X
API Authentication via Social Media		X			
Data Licensing Control	X		X	X	X

Table II: Comparison of Smart City API solutions: (x) limited, "x1" Open311 Interface, "x2" full text search only on data set metadata and not on data.

The proposed approach enabled Sii-Mobility to produce suggestions on demand, as depicted by the workload of the ServiceMap service [36], presented in the next section.

A. Structure of Smart City API

A strongly relevant difference from the reviewed smart city APIs and those provided by Km4City consists in the possibility of posing requests, by using different modalities:

- **SPARQL Query**: calls are directly performed on the RDF Store endpoint using the standard SPARQL query protocol (based on REST) using GET or POST requests with the *query* parameter containing the SPARQL query;
- **SPARQL Query with Inference**: calls are directly performed on the RDF Store endpoint http://servicemap.disit.org/WebAppGrafo/sparql, using the standard SPARQL query protocol (based on REST) using GET or POST requests with the *query* parameter containing the SPARQL query, including inference aspects in the case of Virtuoso, or automatically exploiting the inference in the case of OWLIM [26];
- **REST**: calls are performed by using APIs using full text, keywords, service ID (URI) to get info, geolocation, service ID (URI) to get closer services, time, etc.;
- **Query ID**: calls are performed by using a QueryID (identification) assigned by the ServiceMap tool manager, after having performed a query by using the graphic user interface, as a visual query.

The following example consists in requesting all services in around 0.1 km with respect to the location of service identified by the URI http://www.disit.org/km4city/resource/45e458d0c9e5bf53e34e7e2fed73a6b4 (which is the Cupola of the Santa Maria del Fiore, Florence's Cathedral). The SPARQL query is:

```
SELECT ?name ?dist WHERE {
  ?s a km4c:Service.
  ?s geo:geometry ?g1.
  ?s schema:name ?name.
  <http://www.disit.org/km4city/resource/45e45...>
    geo:geometry ?g2.
  BIND(bif:st_distance(?g1,?g2) AS ?dist)
  FILTER(?dist<=0.1)
} ORDER BY ?dist
```

A similar request can be performed using the REST API http://servicemap.disit.org/WebAppGrafo/api/v1/?selection=http://www.disit.org/km4city/resource/45e458d0c9e5bf53e34e7e2fed73a6b4&categories=Service&maxDist=100&format=json, where *selection* indicates the service URI around which services should be located, *categories* indicates the kind of service to be located, *maxDist* indicates the max distance in meters and *format* indicates the return data format (JSON or HTML). The same query presented above can be performed as Smart City API based on Query ID. This technique assigns a QueryID to a complex query, allowing the developers to copy/paste the REST call, to perform the query identified by the QueryID, disregarding the complexity of the query. This approach simplifies the work of programmers, and at the same

time makes the implementation of web and mobile Apps more independent from the data of the smart city. For example, traditionally a certain API call adopted in mobile App can request a list of services' types in an area. If the developer decides to change the list of requested service type, the mobile App has to be rebuilt. On the contrary, adopting an API call based on QueryID, the developer can simply change the query associated with the QueryID without rebuilding the application. For example, to get the HTML, the URL represents the same query stored with a QueryID http://servicemap.disit.org/WebAppGrafo/api/v1/?queryId=9e5662a352d90ad4bc77690277a371ab&format=html (see Figure 2). In Sii-Mobility, the developers may use the ServiceMap tool to compose visually some geographical and textual queries, and from them to request the sending of an email containing the calls that can be performed for obtaining the same resulting data in JSON and/or web page in HTML. In Figure 2, the ServiceMap at work is presented.

Figure 2: ServiceMap results performing the visual query on Km4City knowledge base.

V. SOME DATA RESULTS ABOUT THE USAGE OF SII-MOBILITY KM4CITY SMART CITY API.

This section reports some data regarding the usage of Sii-Mobility/Km4City Smart City APIs. The Smart City API are exploited by a number of applications: ServiceMap [36], a number of mobile Apps on Google Play, Apple Store and Windows Store (called *Firenze where what. Km4city*, *Toscana where what.Km4city*), Smart Decision System [29], Florence Dashboard (http://www.disit.org/dashboard), Web app (http://www.km4city.org).
The numbers about the received calls from the applications to Km4City API are reported in Table III, and the trend in the last months is reported in Figure 4. The Km4City API service has been opened to a larger audience at the end of February by promoting Mobile Apps. This explains the growing number of requests in March with respect to the testing phases.

API mode	Jan.	Feb.	Mar.
api-service-info	21.118	36.826	173.303
api-services-by-gps	5.573	4.789	5.997
api-location	1.418	3.738	18.444
api-events-day	263	267	17,678
api-text-search	533	440	729
api-services	157	131	401
api-events-week	21	26	36
api-events-month	20	25	25
api-services-by-queryid	13	20	15
api-service-photo	10	5	11
api-services-by-municipality	2	13	3
api-service-stars	1	-	5
api-service-comment	1	-	4
total	**29.130**	**46.280**	**216.651**
format	**Jan.**	**Feb.**	**Mar.**
HTML (calls on servicesmap interface for developers)	179	157	444
JSON	28.939	46.118	216.187
total	**29.118**	**46.275**	**216.651**

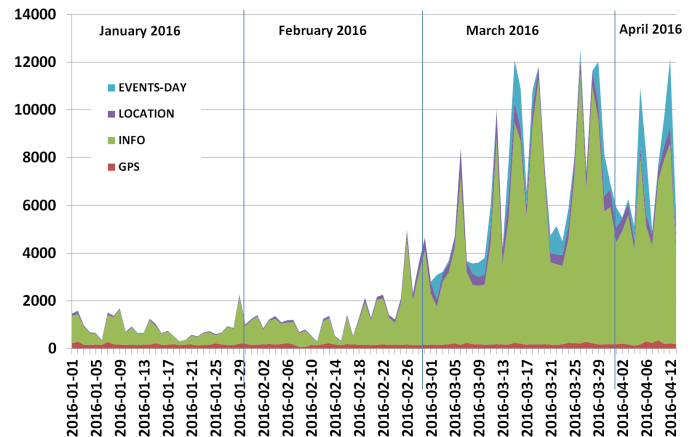**Table III: Km4City Smart City API Calls, number of calls in the period.**

Figure 4: Trend of the Km4City Smart City API calls.

According to Figure 4, most of the calls are due to the *api-service-info*, which is the call for requesting the data about specific information associated to a given URI of a service. This kind of API call is mainly provoked by the computer suggestions by the Personal Assistant, which are delivered to the mobile App, for the suggestion of on demand service of Sii-Mobility. Another relevant number of calls are due to the API call for requesting the *api-location*, by giving GPS coordinates that provide the address (street, civic number, municipality name and URI, etc.) This service is exploited as a first query for requesting weather forecast, services, mobility services, etc. Also, the events (mainly entertainment) are frequently requested, by the Personal Assistant and directly by the city users. Note that in practical all the most requested calls are exploiting inference on the Km4City ontology and the Knowledge Base, thus demonstrating the demanding of Case (c) Smart City APIs.

VI. Conclusions

Most of the smart city solutions are transforming data to services for city users and operators. Services integrate open and private data, static and real time data coming from the administrations and from private operators. The possibility of exploiting aggregated and re-conciliated data by reasoning algorithms enable the production of sophisticated services, such as those for implementing personal assistants, connected drive, smart services, etc. Different kinds of Smart City APIs depend on architectural solutions to pass from data to services. This paper presented a comparison among the data aggregation solutions, by putting in evidence the needs of semantic interoperable aggregated data, in order to provide smart city services. The work has been performed in the context of the Sii-Mobility national smart city project, on mobility and transport integrated with services. Sii-Mobility is grounded on the Km4City ontology and tools, for Smart City data aggregation and service production. Thus, the Km4City APIs have been compared with respect to the state of the art solutions, to show that Km4City has a larger coverage and presents new approaches for Smart City API. Finally, the data about the proposed API consumption have been presented. Km4City, with the Sii-Mobility related improvements, is presently adopted in RESOLUTE H2020 and REPLICATE, H2020 European Commission research and development projects.

References

[1] IBM Smart City http://www-935.ibm.com/services/us/gbs/bus/html/smarter-cities.html

[2] IBM Institute for Business Value, "How Smart is your city? Helping cities measure progress", [online]. Available: http://www.ibm.com/smarterplanet/global/files/uk__en_uk__cities__ibm_sp_pov_smartcity.pdf , Oct. 2013

[3] Alcatel-Lucent Market and Consumer Insight team, "Getting Smart about Smart Cities Understanding the market opportunity in the cities of tomorrow", Oct. 2013

[4] CISCOhttp://www.cisco.com/c/dam/en_us/about/ac79/docs/ps/motm/Smart-City-Framework.pdf http://www.cisco.com, http://www.cisco.com/c/dam/en_us/solutions/industries/docs/gov/everything-for-cities.pdf

[5] Anthopoulos, Leonidas, and Panos Fitsilis. "Exploring architectural and organizational features in smart cities." *Advanced Communication Technology (ICACT), 2014 16th Int. Conference on*. IEEE, 2014.

[6] Filipponi, L.; Vitaletti, A.; Landi, G.; Memeo, V.; Laura, G.; Pucci, P., "Smart City: An Event Driven Architecture for Monitoring Public Spaces with Heterogeneous Sensors," in Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on , vol., no., pp.281-286, 18-25 July 2010

[7] Domingo, A.; Bellalta, B.; Palacin, M.; Oliver, M.; Almirall, E., "Public Open Sensor Data: Revolutionizing Smart Cities," in Technology and Society Magazine, IEEE , vol.32, no.4, pp.50-56, winter 2013, doi: 10.1109/MTS.2013.2286421

[8] Chourabi, Hafedh, et al. "Understanding smart cities: An integrative framework." *System Science (HICSS), 2012 45th Hawaii International Conference on*. IEEE, 2012.

[9] *Smart cities: Ranking of European medium-sized cities*. Centre of Regional Science, Vienna University of Technology, 2007.

[10] Shapiro, Jesse M. "Smart cities: quality of life, productivity, and the growth effects of human capital." *The review of economics and statistics* 88.2 (2006): 324-335.

[11] CKAN: http://ckan.org

[12] OpenDataSoft: https://www.opendatasoft.com/

[13] ArcGIS OpenData: http://opendata.arcgis.com/

[14] RDF https://www.w3.org/RDF/

[15] SPARQL: https://www.w3.org/TR/rdf-sparql-query/

[16] P. Bellini, P. Nesi, A. Venturi, "Linked Open Graph: browsing multiple SPARQL entry points to build your own LOD views", http://log.disit.org International Journal of Visual Language and Computing, Elsevier, 2014, DOI information: http://dx.doi.org/10.1016/j.jvlc.2014.10.003

[17] 5 Stars Open Data from Tim Barneers Lee, http://www.slideshare.net/TheODINC/tim-bernerslees-5star-open-data-scheme

[18] DATEX II: http://www.datex2.eu/sites/www.datex2.eu/files/Datex_Brochure_2011.pdf

[19] IETF: https://www.ietf.org

[20] Green Button Connect: http://www.greenbuttonconnect.com/

[21] E015 digital ecosystem, http://www.e015.expo2015.org/

[22] P. Bellini, M. Benigni, R. Billero, P. Nesi and N. Rauch, "Km4City Ontology Bulding vs Data Harvesting and Cleaning for Smart-city Services", International Journal of Visual Language and Computing, Elsevier, 2014, http://dx.doi.org/10.1016/j.jvlc.2014.10.023

[23] Korn, N., Oppenheim, C.. "Licensing Open Data: A Practical Guide". In: Discovery [online]. June 2011 [cit. 2012-02-20]. Retrieved from http://discovery.ac.uk/files/pdf/Licensing_Open_Data_A_Practical_Guide.pdf

[24] Villata S., Delaforge N., Gandon F., Gyrard A.. "An Access Control Model for Linked Data". OTM Workshops, Oct 2011, Heraklion, Greece. Springer, 7046, pp.454-463, 2011, LNCS.

[25] SOCRATA: https://www.socrata.com/

[26] P. Bellini, I. Bruno, P. Nesi, N. Rauch, "Graph Databases Methodology and Tool Supporting Index/Store Versioning", publication on JVLC, Journal of Visual Languages and Computing, Elsevier, 2015 http://www.sciencedirect.com/science/article/pii/S1045926X15000750

[27] CitySDK: http://www.citysdk.eu

[28] OASC: http://oascities.org/ NGSI OASC: http://wiki.fiware.org/FI-WARE_NGSI_Open_RESTful_API_Specification

[29] M. Bartolozzi, P. Bellini, P. Nesi, G. Pantaleo and L. Santi, "A Smart Decision Support System for Smart City", IEEE International Conference on Smart City/SocialCom/SustainCom together with DataCom 2015 and SC2, December 2015, Cina, IEEE press. http://smartds.disit.org

[30] 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

[31] P. Nesi, G. Pantaleo, M. Tenti, "Geographical Localization of Web-Visible Human Activities by employing Natural Language Processing, Pattern Matching and Clustering Based Solutions", Journal: Engineering Applications of Artificial Intelligence, Elsevier. 10.1016/j.engappai.2016.01.011 http://dx.doi.org/10.1016/j.engappai.2016.01.011

[32] ECIM, European Cloud Marketplace for Intelligent mobility, http://ecim-iminds.rhcloud.com, (FP7 CIP-ICT-PSP-2013-2017).

[33] EPIC, European Platform for Intelligent Cities, http://www.epic-cities.eu, ICT PSP (2011-2013)

[34] Transport.API, http://www.transportapi.com

[35] Navitia.io, https://navitia.opendatasoft.com

[36] ServiceMap: http://servicemap.disit.org , http://www.disit.org/6597

[37] Requirements Specification For Urban Platforms, EIP Project, version 2.2, 2016, European Innovation Partnership for Smart Cities & Communities (EIP_SCC).

[38] READY4SmartCities http://smartcity.linkeddata.es/index.html