GRANT AGREEMENT No 609035
FP7-SMARTCITIES-2013

Real-Time IoT Stream Processing and Large-scale Data Analytics for Smart City Applications

Collaborative Project

Smart City Framework

Document Ref.	D2.2
Document Type	Report
Workpackage	WP2
Lead Contractor	ERIC
Author(s)	Vlasios Tsiatsis (Editor, ERIC), Pramod Anantharam (WSU), Payam Barnaghi (UNIS), Marten Fischer (UASO), Frieder Ganz (UNIS), Muhammad Intizar Ali (NUIG), Sefki Kolozali (UNIS), Daniel Kuemper (UASO), Alessandra Mileo(NUIG), Nechifor, Cosmin-Septimiu (SIE), Dan Puiu (SIE), Ralf Tonjes (UASO), Thorben Iggena (UASO)
Contributing Partners	ERIC, NUIG, SIE, UASO, UNIS, WSU
Delivery of Version 1.0	M12 (31st August 2015)
Delivery of Updated Version	11th February 2016
Dissemination Level	Public
Status	Final
Version	V2.3
Reviewed by	Michelle Bak Mikkelsen (AA), Piraba Navaratnam (UNIS) and Nazli Farajidavar (UNIS)

Executive Summary

This report presents the Smart City Framework (SCF), a high level architecture of a platform that describes the scope of the innovations expected from the CityPulse project and the way they are integrated together in a coherent conceptual system. The purpose of this document is to serve as a reference model and architecture (ARM) to be used by smart city stakeholders, project partners and any other interested parties when engaged in technical discussions about smart cities services based on real-time information streams. The Smart City Framework is expected to be an initial architecture to set the main concepts, common language and the boundaries for the whole project while the details are expected to change within the course of the project execution.

The report presents the architecture in different views, functional, interface and information, security and privacy view and thus explains respectively what the framework does, how components interact with each other, the generation and flow of information, and the necessary mechanisms to address security and privacy concerns about city and citizen relevant data. In the presentation of the interface and information view each CityPulse work package (WP) has provided more detailed descriptions of their individual architectures as of the writing of this report.

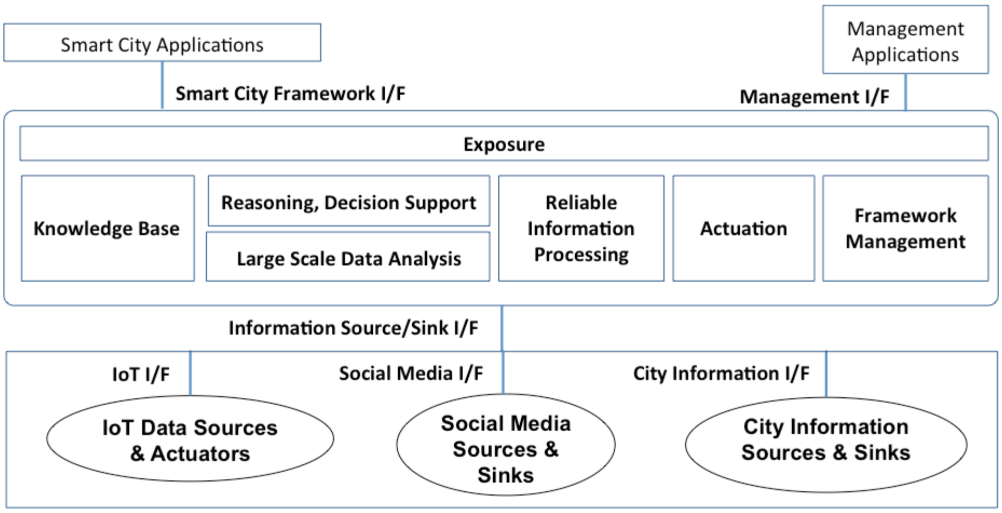

Figure 1 – High-level view of Smart City Framework

On a very high level Smart City Framework (or simply the framework) can be shown in Figure 1. The boundaries of the framework are the different interfaces (I/F in the figure) towards the applications and towards the Information sources/sinks. The framework can be used as a tool to address smart city issues and therefore it should be utilised by smart city related applications. Similarly, the framework needs to use information sources and sinks related to a smart city. The figure shows that Internet of Things (IoT) Sensors deployed in a city environment can be one type of source of real-time information. Moreover, IoT Actuators can be the sinks of information coming from either the framework or the applications. Another type of information source is the legacy information sources provided already by a city e.g. Open Data portals, city GIS (Geographical Information System) data etc. A City Information Sink represents the city infrastructure that the framework can use to store processed information produced by the framework. Apart from information coming from sensors or the City Information Sources, information about the situation in a city can originate from the citizens themselves through the use of social media (e.g. twitter feeds). Therefore these sources are also included as relevant inputs sources for the framework. The difference between these sources and

the IoT or City Information Sources is that the former are potentially sources of unstructured information expressed in natural language (e.g. a tweet is a short text written by a human with potentially a link to a website potentially written by a human) while the IoT or City Information Sources produce structured data although not standardized. The Social information Sinks represent social media channels through which cities could potentially push information to their citizens, e.g. a channel for citizens to receive updates on the traffic situation in certain parts of the city.
The SCF internally consists of a few functional groups (FG) namely the Large Scale Data Analysis, the Reasoning and Decision Support, the Reliable Information Processing, the Knowledge Base, the Actuation, the Framework Management, the Exposure and the Information sources/sinks FG. The Large-Scale Data Analysis addresses issues that are related to the integration of a large scale of heterogeneous sources producing real-time streams and their semantic enrichment. The Reasoning and Decision Support functionality tackles issues that are related to the ability of the SCF to adapt to changing situations based on the real-time information streams. It is responsible for monitoring the semantically enriched streams and adapting the collection of stream information. It is also responsible for providing an API towards the Smart City Applications. The Large Scale Analysis and Reasoning and Decision Support functionalities are supported by a-priori knowledge in the form of the Knowledge Base and reliability and quality of service control mechanisms by the Reliable Information Processing FG. The Actuation FG covers any functionality that allows the SCF to push control commands or information to the IoT actuators, social media sinks and city information sinks. The Framework Management FG includes functionality for the management of the SCF itself such as fault, configuration, security management etc. The Exposure FG covers the mediation of access between CityPulse and Management application and the SCF.

The SCF specification describes more functionality than intended from the description of work and therefore this functionality will not be explored or detailed further within the technical work packages (WP3-WP6). For example a large part of the framework management and management applications are not expected to be developed further in research because they may already be covered by state of the art techniques. Nevertheless, the framework as specified in this report covers the promised description of work and it has already been used by individual work packages other than WP2, for their internal system architecture definitions.

After the individual WPs have performed in depth research on each part of the SCF and after integrating concrete software implementations, they produced an update on the SCF which is included as a separate section in this document. The update presents mainly the functional and interface views as a set of functional components and as a set of high level APIs (Application Programming Interface) methods described in natural language respectively. The update also includes a description of each functional component and some information about the corresponding exposed API as well as implementation details.

1. Contents

Figures

1. Introduction

A city is a form of human habitat extremely diverse in terms of infrastructure and people. Smartness is one of the qualifiers for a city that has mainly emerged from the Information and Communication Technology (ICT) community. The term smart city is not well defined in the literature and subject to interpretation according to the scope of activity that the term is used in. For the purposes of the CityPulse project we choose to scope the term Smart City from [Holler14]. In summary according to [Holler14] a Smart City is a city that utilizes ICT in order to a) efficiently manage existing infrastructure and assist in development of new infrastructure, b) provide efficient services to citizens, c) enable efficient organization of city authorities in order to deliver these efficient services to citizens in a fashion that complies to the environmental mandates and d) enable the private and public sector to develop new and innovative businesses that serve the citizens. A city contains a large number of physical infrastructure and human capital that interacts with each other and motivates the introduction of ICT:

a) A city contains semi-static physical or hard infrastructure for example roads, parks, buildings, trains tracks, etc. This infrastructure needs to be monitored for efficiency and fault detection reasons
b) A city contains energy, water and waste infrastructure in order to support living that potentially need monitoring and control
c) A city involves humans, animals and things involved in the daily human activities which could be supported by a city-wise ICT system

A city-wise ICT system supporting the city authorities, businesses and the citizens need to address a few fundamental challenges:

1) A large number of heterogeneous and distributed information sources such as infrastructure to monitor the environment and human activity (e.g. road traffic)
2) A smart city is a dynamic interaction environment and therefore the large number of information sources produce vast amounts of real-time capturing the current state of the city
3) Humans through social interactions also generate city related information that is typically disseminated through social media or the Internet of People (IoP) as opposed to Internet of Things (IoT)
4) Novel services offered to citizens need to combine all the IoT and IoP sources in order to construct a complete picture of a living city

In order for a smart city to become a reality and address the above challenges the general structure of an ICT system supporting a smart city as well as a number of general guidelines about the smart city ICT infrastructure should be provided. This structure and guidelines is what a “Framework” encompasses in the context of this deliverable. The main reason for providing a framework as opposed to specific architecture and technology recommendations is basically two fold: a) The diversity of the cities in terms of their hard infrastructure and human capital and b) the ever changing nature and capabilities of technology. A framework can be adjusted to a small or a big city as well as to the technologies of today and tomorrow to realise the concrete ICT support platform for a smart city.

The definition of a problem is the single most important step for moving towards the solution. Hence the problem of the specification of the Smart City Framework (SCF) must start with the definition of the framework itself. The definition contains information about the technical objectives of the framework as stated in the CityPulse Description of Work (DoW), the description of the groups of individuals that have a stake on the framework and a set of concerns manifested in requirements, design goals and constraints. In addition, the framework should include the description of the scope and baseline assumptions guiding the formation of the framework as well as the set of components that comprise the SCF together with the relationship between the components. It should be noted that not all parts of the SCF are finalized in this deliverable. Rather this deliverable aims at setting the initial frame in which other work packages (WP) will develop throughout the course of the project.

The SCF reuses concepts from existing platforms, ICT networks and Internet of Things enablers to develop and test a distributed framework for the semantic discovery and processing of large-scale real-time IoT and relevant social data streams. The Smart City Framework aims at bridging the gap between the related technologies and platforms (e.g. iCity platform [iCity]) and the higher-level knowledge that is required from intelligent decision-making by citizens and city operation services. While the existing solutions focus on publishing the data and creating service enablers to interact/access to IoT infrastructures in the cities, the SCF focuses on integration and federation of IoT (and social) data streams and provides processing and analysis mechanisms to aggregate, summarize and create higher-level abstractions from myriad of multimodal streaming data. This provides a mechanism for responding to real-time queries, and event detection and knowledge extraction and discovery processes that are required by end users and city operators to make intelligent and situation-aware decisions in different domains from daily-life tasks (e.g. commuting in the city) to operation planning and maintenance.

The deliverable is structured as follows. Section 2 provides a high level definition of the Framework covering the objectives of framework, the stakeholders and their concerns and the boundary conditions for a smart city framework. Section 3 presents a high level view of the framework originating from the description of work of the CityPulse project, which mainly covers work package responsibilities. Section 4 describes in detail the Smart City Framework from a number of views according to the current presentation style of modern architectures. Section 5 includes a description of the main design concepts, which guided the Smart City Framework. Section 6 outlines the state of the art in existing frameworks and finally Section 7 concludes the report.

2. Definition of the Smart City Framework

The intention behind the SCF is to be used by certain groups of people who act in different roles (also referred to as stakeholders) in order to address their concerns when the framework is to be applied as a tool for addressing smart city use cases such as current traffic conditions in a certain city road, utilization of bus transport, safe event services etc. The reader can refer to the DoW or the

Deliverable D2.1 [CityPulse-D2.1] for more use cases where the SCF is applicable. Therefore the first step into the direction of defining a framework is to define the roles of people interested in such a framework.

2.1 Smart City Framework stakeholders

The SCF stakeholders can be following:

1) City Stakeholders. These can be grouped into three sub-categories
    a. City IT services: Since the project creates the blueprints as well as services for smart cities, these should be typically retrofitted in the existing IT infrastructure of a city. The responsible individuals for smart city information infrastructure are the members of the city IT personnel.
    b. City departments: The city departments that solve specific problems such as transportation, waste management, lighting etc.
    c. City decision makers: The decision makers of a city are the ones who decide how to allocate funds for the development of a new smart city service
2) Third party providers: Enterprises that provide services to the city by, for example, developing phone mobile applications which use open city information sources.
3) Citizens: Citizens are the main users and also end users of the smart city framework services, and requirement providers for applications. The assumption is that the citizens use city authorities as a means to voice their concerns. Therefore, citizen concerns are indirectly expressed as the concerns of the city authorities.

2.2 Stakeholder concerns

Deliverable D2.1 [CityPulse-D2.1] focused on the collection and ranking of smart city scenarios according to the following evaluation criteria:

- User differentiation: This criterion measures the impact (positive or negative of a scenario) to citizen's daily life as well as the degree of acceptance by the society.
- City Relevance: This criterion measures the degree that the scenario is relevant to a city.
- Data Streaming: This criterion measures whether or not the available data are streaming or not
- Decision Support: This criterion indicates the level of complexity of the scenario
- Big Data: This criterion measures the existence of big data volumes and large number of sources

Deliverable D2.1 [CityPulse-D2.1] also produced a list of requirements relating the above criteria and use cases. The relevant technical requirements for the Smart City Framework are the Data Streaming, Decision Support and Big Data. Here we provide a consolidated list of the relevant requirements written in technical language. Please note that the words “SHALL”, “SHOULD”, “SHOULD NOT” have similar semantics as in the Internet Engineering task Force (IETF) Request For Comments (RFC) documents [RFC2119]:

1) Requirements on Data Streaming

a. The SCF SHOULD expose the data by API to applications
b. The SCF SHOULD support (near) real-time streaming of data
c. The SCF processing power SHOULD be sufficient to deliver the data to the applications
d. The SCF SHOULD support secure mechanisms of collecting, processing and disseminating input and processed information
e. THE SCF SHOULD support reliability of data and information transport

2) Decision Support
    a. The SCF SHOULD support multiple and different types of input data
    b. The SCF SHOULD support actuation and control loops and automation
    c. The SCF SHOULD support automation
3) Big Data
    a. The SCF SHOULD be scalable in terms of users and data streams
    b. The SCF SHOULD have privacy preserving mechanisms

2.2.1 Discussion

The above technical requirement list as well as the selected scenarios from deliverable D2.1 were used as a basis for the development of the SCF and to ignite further study for the different technical work packages. Here we summarise some of the motivation of the choices of the SCF components based on these individual WP studies.

With some of the scenarios requiring complex inferences to be performed on computationally intensive reasoning tasks, on the technical side the decision support components needs to be expressive enough to deal with missing knowledge, conflicts and optimization problems. The complexity of the reasoning is not the only technical requirement. In fact Decision Support capabilities need to cater for adaptability and continuous evaluation through incremental reasoning, so that only the most recent and pertinent knowledge about the real world is taken into account. The need to reason about missing information via common sense and default knowledge led us towards the use of non-monotonic reasoning frameworks that can also cater for conflict resolution and constraints, and that are extended with native streaming operators for incremental evaluation of the inference rules over sliding windows.

Some of the selected scenarios deal with data and information created by people using mobile applications or web interfaces. Some scenarios make an extensive use of simple physical sensors, e.g. temperature or pollution sensors. Due to faulty sensors or intentionally provided misinformation the data sets have to be evaluated before usage. To achieve this, all data sets are annotated with quality values describing the grade of the provided information. The expected number of data streams on the one side and limited memory and computation time on the other side require the use of incremental algorithms to annotate data sets. To disregard incorrect information sources an additional reputation has to be integrated into the framework that rates the trustworthiness of data sources over time and allow applications to select the most proper and trusty data stream. Aggregation mechanisms have to be applied to efficiently monitor a sensor's past behaviour. Additionally, the integration of test possibilities for new applications ensures that they are working correctly before they are used in the real world environment.

2.3 Baseline assumptions, scope and boundary conditions

On a very high level the realization of the framework can be presented as in the Figure 2, which provides a first order approximation for the scope of the framework. The framework can be used as a tool to address smart city issues and therefore it should be utilised by smart city related applications. Similarly, the framework needs to use information sources and sinks related to a smart city. The figure shows that Internet of Things (IoT) Sensors deployed in a city environment can be one type of source of real-time information. Moreover IoT Actuators can be the sinks of information coming from either the framework or the applications. Another type of information source is the legacy information sources provided already by a city e.g. Open Data portals, city GIS (Geographical Information System) data etc. A City Information Sink represents the city infrastructure that the framework can use to store processed information produced by the framework. Apart from information coming from sensors or the City Information Sources, information about the situation in a city can originate from the citizens themselves through the use of social media (e.g. twitter feeds). Therefore these sources are also included as relevant inputs sources for the framework. The difference between these sources and the IoT or City Information Sources is that the former are potentially sources of unstructured information expressed in natural language (e.g. a tweet is a short text written by a human with potentially a link to a website potentially written by a human) while the IoT or City Information Sources produce structured data although not standardized. The Social Information Sinks represent social media channels through which cities could potentially push information to their citizens, e.g. a channel for citizens to receive updates on the traffic situation in certain parts of the city.

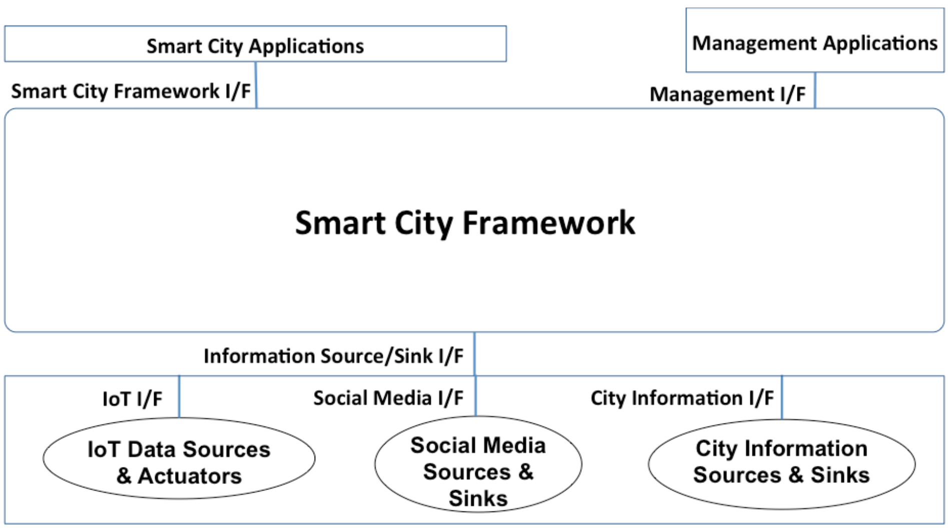

Figure 2 – Smart City Framework scope

From the high level view of the framework we can make the following observations. Please note that the words “SHALL”, “SHOULD”, “SHOULD NOT” have similar semantics as in the Internet Engineering task Force (IETF) Request For Comments (RFC) documents [RFC2119]. For completeness we reiterate here the main terms used below. "SHALL", means that the definition is an absolute requirement of the specification. “SHOULD” means that mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course. “SHOULD NOT” means that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.

1. The SCF SHOULD integrate online IoT Data Sources, online Social Media Sources as well as City Information Sources, which are typically semi-static sources (for the definition of the semi-static as opposed to real-time please see below). The interfaces (IoT, Social Media, City Information interfaces) supported by the SCF for accessing these data sources SHOULD be specified or recommendations SHOULD be provided by the project
2. The SCF SHALL support access to Smart City Information Sinks and SHALL specify the interfaces (IoT Actuator, Social Media Sink, City Information Sink interfaces) towards these sinks.
3. The SCF SHOULD support Smart City Applications accessing the value added services that the SCF provides by using the above-mentioned data sources. The interfaces for application developers (Smart City Application interfaces) SHOULD be specified or recommendations SHOULD be provided by the project
4. The SCF SHALL support Smart Framework management portals as a special type of application accessing a special type of interfaces (Management Interfaces). The interfaces for management application developers SHOULD be specified or recommendations SHOULD be provided by the project
5. The SCF SHOULD provide recommendations or SHALL specify the internal SCF architecture
6. The SCF SHOULD provide recommendations or SHALL specify the internal SCF components
7. The SCF SHOULD NOT stretch in the domain of IoT Sources/Sinks, Social Media Sources/Sinks or the City Information Sources/Sinks in other words, these technical areas are not concerns of any SCF stakeholder. In other words, the SCF does not provide guidelines for which City information sources should exist.
8. The SCF SHOULD NOT stretch to the domain of application frameworks for creating SCF applications. In other words, the SCF does not provide guidelines for which City application should exist.

The three different types of information sources (IoT , Social Media and City Information) are typical representations of both real-time and semi-static information. By real-time information sources we mean the sources that report relatively fast, i.e. from a few samples per day to a few samples per second while by semi-static information sources we mean sources that report relatively slow or contain non-changing information (e.g. city real estate property coordinates that may change on the order of once per a few tens of years).

2.4 Contribution of Work Packages and stakeholder concerns

The SCF definition and specification as captured in this document aims at providing recommendations about certain parts of the framework. Since the framework contains functional components, interfaces etc. the specification of which is subject to further research from other work packages (WP3, WP4, WP5), the framework details shall be finalised after the delivery of this document in month 12 (M12). This means that this specification is inherently a first attempt to capture an integrated system architecture and certain details will change throughout the course of the project. The list below describes the types of results either in paper specification and/or software components that CityPulse aims at delivering throughout its duration. A work package (WP) name inside the parentheses indicates that the particular WP has provided input for this deliverable until the delivery date and that the specific WP is responsible for the further specification of the corresponding part of the framework even after the delivery of this document.

1) Overall architecture (WP2/D2.1, D2.2)
2) Information models (WP3, WP4, WP5)
3) Software components (WP3, WP4, WP5, WP6)
4) Interfaces (WP2/D2.2, WP3, WP4, WP5)
5) Tools (WP3, WP4, WP5, WP6)
6) Applications (WP2/D2.1, WP3, WP4, WP5, WP6)

An initial mapping between these parts of the framework specification and the stakeholders, is given in the table below. An "X" indicates that a stakeholder may be interested in the specific part of the SCF because this part potentially addresses the specific stakeholder concerns. The City IT services are interested in every facet of the framework and specific components since the need to know how the system works in order to support other city departments, the city authorities or the citizens. The City Departments as mainly users of the framework are mainly interested in the tools and applications using the framework and the information model that these tools and application use. The City Decision Makers as well as the citizens need to understand the framework mainly from the services it provides to the citizens and therefore they are interested only the in tools and applications that the citizens could use. The Third Party providers that could potentially contribute to the city with software in the framework, tools and applications need to know the respective details as well as the information models, interfaces and overall architecture in order to do the appropriate system integration.

Deliverable/Stakeholder	City Stakeholders			Third party providers
	City IT services	City Departments	City Decision makers	
Recommendation documents				
Overall architecture	X			X
Information models	X	X		X
Software components	X			
Interfaces	X			X

Tools	X	X	X	X
Applications	X	X	X	X

The evaluation of the SCF as well as individual results from each WP is a responsibility of WP6 and the results will be recorded to WP6 deliverables.

3. Description of Work High Level architecture

The high level architecture presented in the DoW is shown in Figure 3. The figure presents an "architecture" picture that combines functional groups/components and interfaces with WP responsibilities. This high level architecture was used as a starting point for the specification of the SCF. The Large-Scale Data Analysis functionality addresses issues that are related to the integration of a large scale of heterogeneous sources producing real-time streams and their semantic enrichment. The Real-Time Intelligence functionality tackles issues that are related to the ability of the SCF to adapt to changing situations based on the real-time information streams. It is responsible for monitoring the semantically enriched streams and adapting the collection of stream information. It is also responsible for providing an API towards the Smart City Applications. The Large Scale Analysis and Real-time Intelligence functionalities are supported by a-priori knowledge in the form of a knowledge base and reliability control mechanisms for the incoming raw as well as semantic information.

One observation about the high level architecture from the DoW is that it does not address the issue of actuation, information dissemination or storage. The DoW presents a CityPulse system as mainly a data analytics system while a Smart City Framework is both an analytics and action framework at least from a conceptual point of view. Therefore, the presentation of the SCF also includes the actuation, information dissemination and storage functional components for the sake of completeness. However the description of these components is rather high level as at the time of the writing of this report there is no corresponding work plan for the project to address these components.

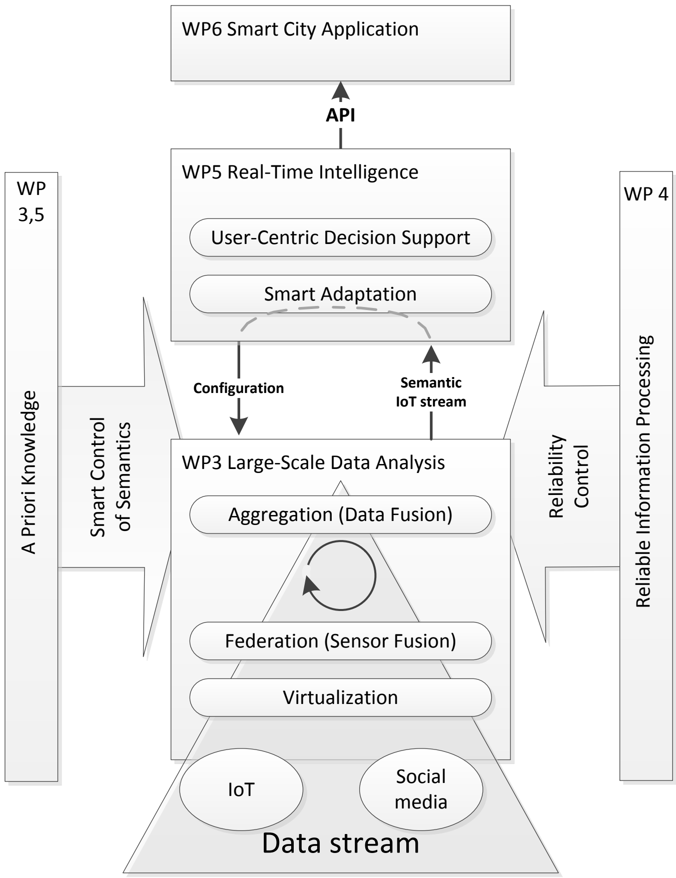

Figure 3 – Smart City Framework high level architecture

4. Smart City Framework Specification

The specification of the SCF follows the best practises with respect to the specification of a system architecture. We follow the approach of Rozanski & Woods [Rozanski11] of presenting an architecture using multiple views and perspectives with the exception that we limit the presented views for SCF since it is mainly a framework and not a concrete system architecture. The difference between a framework and a concrete system architecture is the following: A framework is described with only a limited number of views (e.g. functional, interface etc) since it provides recommendations about how the system should function should it be implemented; a concrete system architecture is a description of a functioning system which in turn means that more details could be described such as which machines host which software that implements certain functionality. For the description of the SCF we have chosen to include the following views:

- **Functional View**: This is a description of what the system does without providing details on how and where the functions are realized.
- **Interface and Information view**: This is a description of how the different functional components communicate with each other and how information is generated, how it flows and how it is transformed in the framework. Typically in a system architecture description each of these views is covered in a separate section but these views are quite similar for the

CityPulse SCF since the SCF is mainly a data flow framework. In other words the CityPulse SCF describes streams of data flowing through the SCF components and therefore most of the interfaces between components are data flow interfaces.

- **Security, Privacy View**: This special view describes the security and privacy functions for the Smart City Framework.

4.1 Functional View

Starting from Figure 3 and elaborating the functionality of the SCF based on input from WP3, WP4, WP5, Figure 4 presents a more detailed architecture with blue boxes representing functional groups (FG) and black boxes/ellipses/cylinders showing functional components (FC) that comprise the functional groups. The blue lines represent the interface between the SCF and external systems (Sources, Sinks, Applications). The functional view elaborates on the functional groups and components.

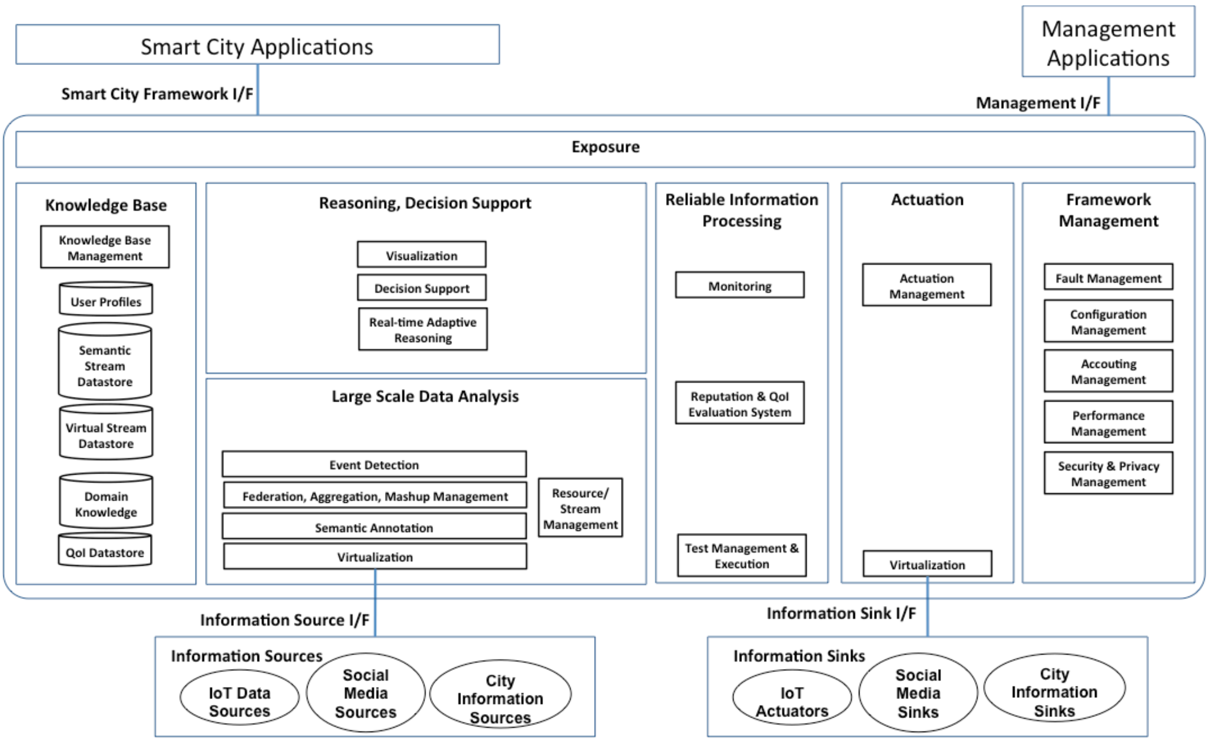

Figure 4 – Smart City Framework functional view

4.1.1 Information Sources Functional Group

The Information Sources group contains the sources of city related information. These are of three kinds:

a) Internet of Things deployments which represent sensor and identification tag infrastructure deployed in the city environment; these sources can be streaming asynchronously or they can be polled

b) Social media sources that represent streaming text messages from social media such as Twitter[Twitter].

c) City information sources that represent both static and dynamic information repositories about the city such as city plans, periodic traffic reports from city entry points etc.

4.1.2 Information Sinks Functional Group

The Information Sinks group contains the possible sinks for information or control. By “sinks” we mean endpoints that can accept information and cause a change in the physical or digital world. These are of three kinds:

d) Internet of Things deployments which represent actuator and identification tag infrastructure deployed in the city environment to which control commands can be dispatched (actuators) or identification information could be written (tags). An example of an actuator is a city sign that the city authorities use to push public announcements to the citizens.
e) Social media sinks that represent e.g. social media channels to which information could be send (e.g. “main street is heavily congested”).
f) City information sinks that represent both static and dynamic information repositories about the city that could potentially be updated by the SCF. An example of a city information sink is a traffic congestion repository operated by the city authorities.

4.1.3 Large Scale Data Analysis Functional Group

Virtualization Function

The virtualisation function (not necessarily a single software component) provides open APIs and common services to publish real world data streams from IoT, social media and city information sources, and create virtual representations of them. It uses the selected interfaces towards the data sources and converts them into streams if applicable and if they are not already in this form. An example of a source that could be converted into a stream but not originally offered as such is a sensor that can only be polled. For the sources which cannot be converted into streams such as static demographic data this function allows access to the related data with a request-response type of interface. This function is responsible for the connectivity aspects of on-boarding new sources for example reachability information (e.g. IP address), interface description, access control credentials, etc. This function is also responsible for adaptation between a proprietary source interface and the internal SCF interface that other components use.

Semantic Annotation Function

This function performs semantic annotation to the virtualized streams or the responses of semi-static information sources. In order to perform the semantic annotation this function utilises a knowledge base of static and inferred facts about the information source (e.g. location), hence this function is related to the Knowledge Base functional group. The output of the semantic annotation function is a semantically annotated stream or semantically annotated response from the virtualized semi-static information source.

Federation, Aggregation and Mash up Management Function

This functional component is responsible for the integration and abstraction of the federated data streams. Integration of relevant streams is performed by automatic discovery and binding of streaming sources by the discovery component (not shown in the picture to avoid cluttering). The Discovery component utilises a stream metadata store to automatically discover relevant data streams and filters the discovered streams based on QoS/QoI constraints and preferences defined within the application request. This component is also responsible for the integration of heterogeneous data streams and (semi) static data sources to generate mashed up streams according to the application requirement.

Event Detection

The output of the data aggregation model can vary over time. Especially in sensor networks or IoT deployments the state of an observation is constantly changing and most likely following some patterns. To perceive a concept or phenomena both the current and past states are required. To model and include this time-dependent aspect of higher-level concept creation, we combine the previous static model with machine learning techniques. Thus, it enables to use the abstractions obtained from the data aggregation component to acquire events and processes that occur over a certain amount of time. For instance, the congestion level of traffic changes during a day from normal over busy and back to normal that represents a daily traffic pattern. Each of these patterns can be modelled as a new state that eventually can be perceived as outliers. Finally, the event detection component transforms observation and measurement data (originated from sensory devices) and relations into higher-level abstractions to formalise concepts and knowledge from the underlying raw data [Ganz13].

Resource/Stream Management Function

This function interfaces the requests from other functional groups, performs run-time management of semantic annotation, aggregation and mash up and event filtering. It includes a Resource/Stream Directory which manages metadata information about the information sources currently feeding information to the framework, the virtualized streams or the virtualized semi-static information sources, the aggregate and mashed up streams and their configuration (e.g. stream #5 is an aggregate of streams #1 and #3 with the aggregation function #56), the event specifications and their configurations etc.

4.1.4 Knowledge Base Functional Group

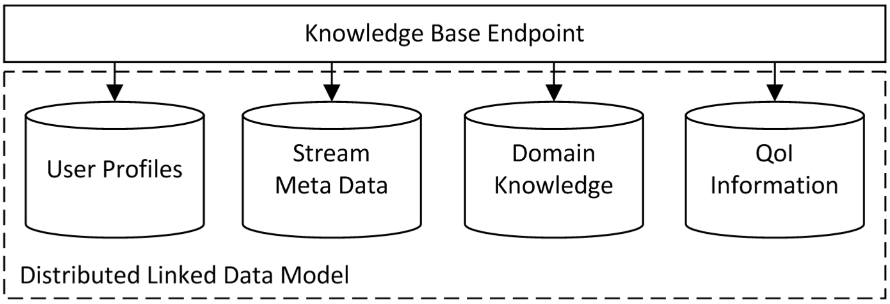

Figure 5 – SCF Knowledge Base

Knowledge Management Function

This function interfaces all the other components and performs access to the stores in the Knowledge base FG. The management component provides a unified access (end)-point to the information and offers the means to update and alter resources during the workflow to allow components to update and rectify parameters throughout their evaluation. The knowledge base (KB) contains all information that is acquired through the SCF, which is deemed worthy to be stored. The knowledge base represents the Meta information about the particular streams including provenance, QoI parameters and the streaming data. The KB is represented as a semantic model that follows the linked data approach to model and trace relations between parameters, stream sources and the data payload. The data payload is partly stored in the Stream Datastore upon request. The Knowledge management component serves as a single entry endpoint for all information related queries to avoid redundancies and keep an integrated data model.

User Profiles Function

This is a store of the user profiles used for the user-centric decision support described later in the text. The user storage models the relations between created and maintained streams and provides an access control mechanism to provide rights management. The user-centric decision support refers to decision support mechanisms that take into account the types of users (e.g. elderly people in a city environment) making the requests to the SCF.

Semantic Stream Datastore Function

This is a store of selected semantically annotated streams generated by the Large Scale Data Analysis FG i.e. semantically annotated virtualized streams, aggregated/mashed up streams or event streams. Please note that not all the possible semantically annotated streams are stored here, only a selected set. The configurations about which streams are stored exist in the Resource/Stream Management function of the Large Scale Data Analysis FG.

Virtual Stream Datastore Function

This is a store of raw virtualized streams generated by the Large Scale Data Analysis FG. Please note that not all the possible raw streams are stored here, only a selected set. The configurations about which streams are stored exist in the Resource/Stream Management function of the Large Scale Data Analysis FG. For efficiency purposes the storage of semantically annotated streams described above may be split into two parts: a) this component/function which stores the raw streams and b) another component which only stores the semantic annotation of the stored raw streams with references to the corresponding raw streams. The component which stores only the semantic annotations with the references could be the Semantic Stream Datastore described above.

Domain Knowledge Function

This contains static/factual as well as inferred knowledge objects and rules. For example, it may contain city map information annotated with road addresses as well as rules operating on this knowledge to group certain parts of a city bounded certain roads into one city region with a specific name.

QoI Datastore

This store maintains the Quality of Information (QoI) and reputation results produced by the Reputation and QoI Evaluation system Function described below.

4.1.5 Reliable Information Processing Functional Group

Reputation & QoI Evaluation System Function

This function is triggered by new events detected in the Large Scale Data Analysis FG. It updates the QoI data for one or more corresponding data streams and also informs the Reasoning and Decision Support FG about significant changes of the QoI in the data streams. The resulting QoI and reputation describing a data stream is stored in the QoI Datastore. Other components are able to query the QoI at the QoI Datastore. Also, new data streams can be registered with initial QoI values.

Test Management and Execution Function

This function is responsible to test the reliability, robustness and performance of the Smart City applications. It utilises (a subset of) the CityPulse Reference Data Set (will be described in D2.3) injected as test data into the Virtualization component through the SCF interface. The Test Management Function controls the Test Execution and also evaluates the availability and reliability of information provided for applications as well as the application's output.

Monitoring Function

This function observes data streams and detects outliers, violations against the QoI and other inconsistencies, e.g. the breakdown of a data source. If necessary, e.g. because of mismatching QoI requirements, the fault recovery function is informed. Unlike the testing this is done during runtime and hence has to cope with large amounts of data. To avoid performance issues the monitoring function therefore directly interacts with the stream virtualisation.

4.1.6 Reasoning and Decision Support Functional Group

This FG caters for the adaptive capability of the framework. This adaptability is realized in terms of flexible control for configuration of relevant data sources and context-driven filtering of relevant events for decision support considering usage patterns and profiles of classes of users to suggest optimal configuration of smart city applications.

Real-time Adaptive Reasoning Function

The adaptive functionality of this component lays in the ability of identifying and reacting to changes in two different ways:

1) monitoring and detecting changes in Non-Functional Properties (NFP) at the service stream level, to suggest changes in the discovery and binding process if due to such change, some application-specific constraints on those properties are violated
2) monitoring events that might be relevant for the reasoning task to be performed, matching those events against contextual requirements and user profiles to determine whether they refer to a relevant change in the real world or they need to be filtered out (this operation might need feedback from the user)

Decision Support Function

This functional block is in charge of reasoning about events that are relevant for a particular task. These events are detected by the adaptive reasoning functional block. The Decision Support functional block has different reasoning modules that are application dependent (such as a module that computes the optimal path from one location to another, according to some constraints and preferences specified by the user explicitly or implicitly derived by users' profiles), and will also provide a generic module to incorporate (implicit and explicit) usage patterns, contextual information and configuration preferences to provide support to decisions. The Decision Support functional block is also including the functionalities that allow deriving and updating user profiles.

Visualization Function

The visualization function will have two main capabilities:

- an application-dependent user interface that would enable easy configuration of preferences and requirements to be used in the decision support phase
- a set of modalities for generating diagrams, intensity graphs and map overlay to provide the user with the results of Decision Support and Adaptive Reasoning

User feedback will be considered and specific modalities to personalize users' experience will be designed.

4.1.7 Actuation Functional Group

The actuation FG includes the functions that keep track of the potential actuators that could be controlled or the information sinks that could be updated as a result of the decisions from the framework e.g. street lights to be turned off when no people are around.

Virtualization Function

The virtualisation function (not necessarily a single software component) provides open APIs and common services to allow access to heterogeneous control commands or sinks of information. This function is responsible for the connectivity aspects of on-boarding new sinks for example reachability information (e.g. IP address), interface description, access control credentials, etc. This function is also responsible for adaptation between a proprietary sink interface and the internal SCF interface that other components use.

Actuation Management Function

The actuation management function provides the execution environment of the decomposition of complex actuation tasks to simple actuation commands for the actuators or the information sinks. For example a complex actuation task could be to turn off the streetlights for a specific street. This complex actuation task is decomposed into single commands to individual streetlights on the specific street. This function also includes concurrency control of multiple actuation requests to the same actuator/sink. This function includes metadata information/descriptions about the actuators or information sinks as well as description of complex actuation tasks and their decomposition into simpler actions.

4.1.8 Exposure Functional Group

The exposure functional group includes one single function to mediate external requests performed by Applications (Smart City Applications or Management Applications). The mediation includes a) request routing functions such as redirecting Smart City Application interface calls to the e.g. the Reasoning, Decision Support component while redirecting Management Application interface calls to Framework Management functions, b) request load balancing in the case of a function being distributed in multiple physical machines, c) enforcing access control rules (e.g. allowing only anonymous and aggregated data access to applications or allowing administrator user to access management functions) and d) format adaptation of responses to requests e.g. the Reasoning and Decision Support returns a response in format A and the application expects the response in format B with A and B different.

4.1.9 Framework Management Functional Group

The Framework management FG covers operational aspects of the framework ensuring that the framework delivers the services to the City stakeholders through the CityPulse Applications. For this purpose the management FG includes the typical FCAPS (Fault, Configuration, Accounting Performance and Security) functions. Typically all these functions expose special management interfaces for other framework functions to send or retrieve respective information as well as interfaces for management application, which are typically run by human operators.

Fault Management Function

The fault management function is responsible for the detection and resolution of faults in the framework. It is assumed that this function handles the escalation of faults and exceptions coming from other framework functions and it typically involves a human operator for the fault resolution.

Configuration Management Function

The configuration management function maintains the configuration of the framework and allows other functions to read and write this configuration. This function also contains a software repository for the framework software components.

Accounting Management Function

The accounting management maintains statistics and accounting figures for the operation of each framework component as well the interactions between framework components and the interactions across the smart city framework. The purpose of this function is to assist any charging and billing system based on usage statistics. This function may be redundant when no charging of the framework usage is expected.

Performance Management

The performance management function is responsible for the collection and analysis of performance statistics for the operation of the framework. For example a smart city application request may take too long to be processed in the framework and therefore its end-to-end delay performance will be poor. Performance management is typically used along with configuration management in order to improve poor performing systems.

Security and Privacy Management Function

The Security and Privacy management function is responsible for the secure operation of the whole framework as well as the necessary privacy preservation functions. This function includes components such as an Authentication (which in turn includes Identity Management), Authorization (access control rules and enforcement thereof), Key management (for all the authentication/encryption keys involved in the framework), Trust and Reputation, Privacy Preservation, etc. This function is presented in more detail in the Security and Privacy view below.

4.2 Interface and Information View

This view describes the interfaces between the functional components in the architecture as well as the information flow between the functional components. The choice to merge these two views was based on the fact that the SCF developed in CityPulse is mainly a data flow framework with functional components operating in data flows or real-time streams. In certain views the descriptions also include implementation details (e.g. a Publish-Subscribe engine), which should be part of the deployment view but we chose to leave these details in this view for keeping a balanced presentation of the framework. Otherwise the deployment view would be very limited compared to the other two views since real-life implementation of the SCF functions are in initial stage in this phase of the project. The interface and information view takes each of the main functional groups for the SCF and provides a detailed explanation of the interfaces and information flow within the FGs.

4.2.1 Large Scale Data Analysis FG internal interfaces

The Large Scale Data Analysis FG internal interfaces aim at semantic annotation and analysis of IoT and social network data streams by taking into account dimensionality reduction and reliability processing. The framework involves six main components: a) virtualisation, b) middleware, c) semantic annotation c) data federation and e) data aggregation f) event detection. Figure 6 depicts the main components and basic workflow of the framework. Initially, the sensor nodes transmit either raw data or aggregated data to the virtualisation component. The virtualisation component includes different APIs and interfaces and is able to collect and store data from heterogeneous nodes. After collecting the data, it forwards the data to three components of the system, namely, semantic annotation, data federation, and reliable information processing. The data federation component discovers and mashes up to obtain enriched information regarding data stream. In case of the data was not aggregated and discretised by any module, the pattern creation and discretisation process is applied in the data aggregation component. Afterwards, the event detection component performs abstraction process using machine learning techniques. The abstracted data is finally accessible through the middleware where different layers such as real-time intelligence layer or graphical user interface can be used to access the data.

Virtualisation

The virtualisation component is introduced to facilitate seamless access and management of sensor observation and measurement data based on semantic web technologies. It defines an interface in which sensor services can collaborate and cooperate to support data access and integration from different sensor networks and application services that provides the real world information for intelligent decision making. With the semantic descriptions and service interfaces, it enables unconstrained access to large-scale and distributed sensor services across heterogeneous sensor networks

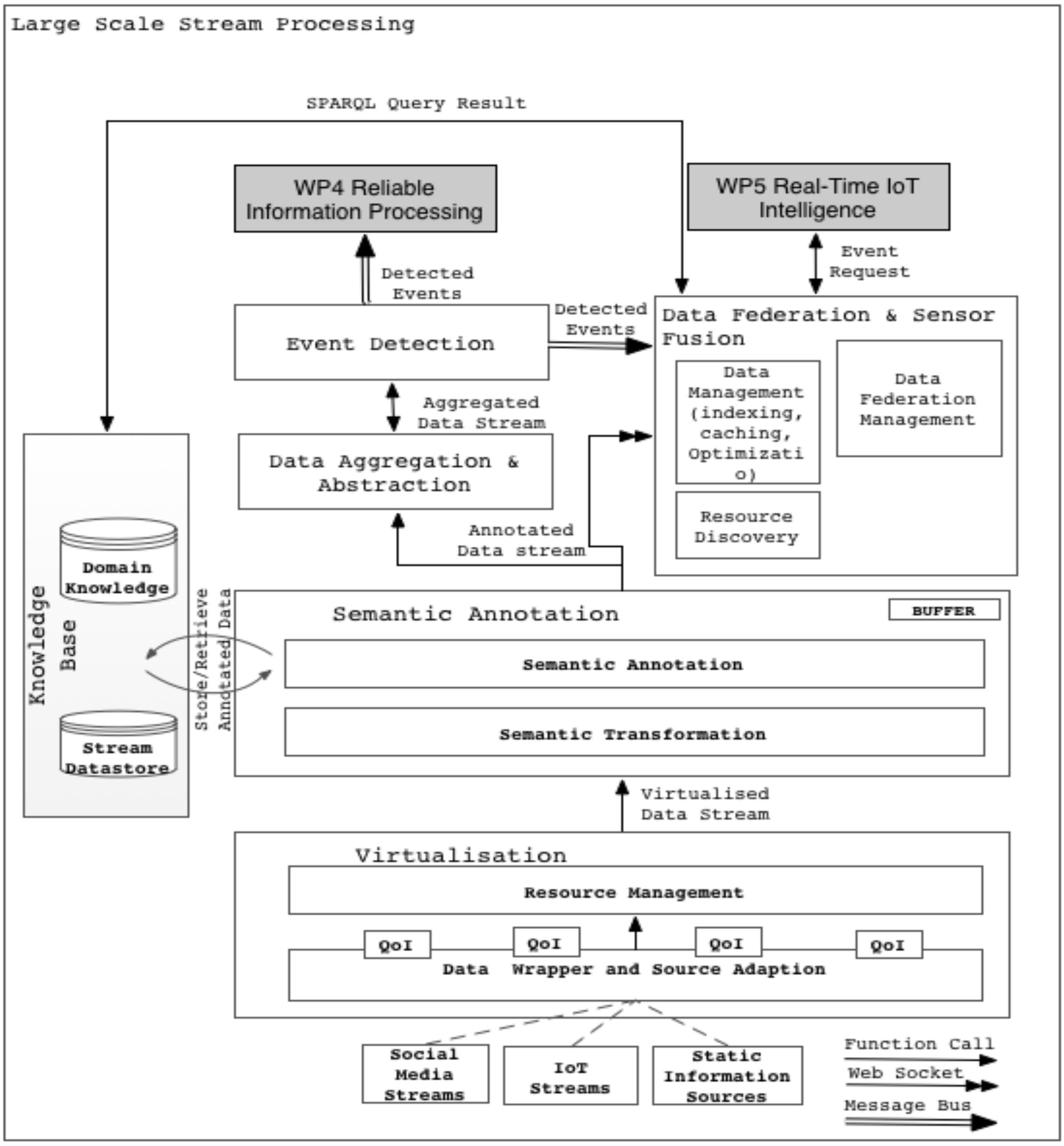

Figure 6 – Information flow of the Large Scale Data Analysis FG

Middleware

The middleware component is represented with double line arrows in the information flow view, which enables delivery of large volume of data that can influence the performance of the smart city systems that use IoT data. The data wrappers manage the communication protocol with the sensors and provide sensor observations into the system in an effort to annotate them using ontologies. However, while IoT applications are distributed systems, there's two crucial points that needs to be accomplished for the communication between the components of the framework. Firstly, the data coming from the data wrappers has to be passed through to the data processing components such as the Reliable Information Processing and the Semantic Annotation. Since the components can be developed on different platforms there is a need for a unified, platform neutral format for the messages to enable the communication. Secondly, the components need a possibility to pass signalling messages to each other e.g. to initiate processes or keep the other components informed about recognised events. In order to make sure that these possibly critical signalling messages are delivered in a timely fashion and are not blocked by the continuously published data from the wrappers, different channels are used for the tasks.

Semantic Annotation

Describing the obtained data stream for interoperability or facilitated search is the core objective of semantic annotation component, as many information management tools. However, the amount of traffic generated by Smart City applications can be voluminous, particularly for real time applications in environments with resource constrains devices, for example sensors with limited bandwidth, memory or power. Therefore, the information model that is being used by the system not only needs to explicitly represent the meaning and relationships of terms in vocabularies but also should be lightweight in order to reduce the traffic and processing time. In this component, we use a lightweight information model to annotate sensory data, which is based on well-known models such as SSN[SSN] and IoT.est[Wang12].

Data Federation and Sensor Fusion

This component enables automated discovery and federation of the heterogeneous data by determining relevant sensor data sources and their processing techniques. An event request containing functional requirements (e.g. type of sensors) and non-functional requirements (e.g. QoI/QoS constraints and preferences) is received by this component. Discovery component processes the event request and discovers the relevant data source by utilising knowledge base containing stream description and their QoI/QoS values. Once the relevant data sources are determined, the federation component integrates the semantically annotated data streams according to the user requests. Various optimisation techniques (e.g. indexing and caching) are also applied to efficiently integrate relevant stream as well as static data sources. Federation component generates mashed up data streams as an output.

Data Aggregation

Data aggregation and compression are the most important remedies utilized to decrease communication traffic in IoT. Data aggregation or data fusion can help to extract meaningful features from collected sensor streams. Another method to reduce the communication traffic for the sensory data is to reduce the size of the messages. This can be applied using data compression algorithms. However, compressing the data itself could lead to a loss of information (in lossy compression) and the compression techniques can require higher power consumption as it might require data processing before transmission, and in long term observations (e.g. environmental monitoring applications) compression techniques can still produce large amounts of data [Kimura05]. Chen et al. [Chen09] give an overview about approaches that create a summarised data stream of a set of sensory data streams and use the aggregated data for transmission. The aggregation of the data usually relies on the mathematical sum, max, min, average and count aggregate functions [Jesus11]. In large distributed WSN this usually happens via clustering algorithms; however this can lead to loss of important data that has been masked due to the aggregation in lower layers.

In this component, we perform local data processing and create higher-level data abstractions that can be communicated globally. We process the stored raw sensory data using the Knowledge Acquisition Toolkit ([KAT]), which is designed to extract and represent human understandable information from raw data. The toolkit includes a collection of algorithms ranging from data and signal pre-processing algorithms such as Frequency Filters, dimensionality reduction techniques such as PCA, Wavelets, FFT, SAX, and Feature Extraction and Abstraction and Inference methods such as Clustering, Classification and Logical Reasoning.

Event detection

While the state of sensor observations are continually changing and most likely following some patterns, we use the event detection component in order to perceive a concept or phenomena using both the current and past states. To model and include this time-dependent aspect of higher-level concept creation, we use machine learning methods to acquire events and processes that occur over a certain amount of time. For instance, the state of a road traffic or congestion (e.g. busy, normal, low), that occurs in various regions, can be modelled in the event detection component based on several abstractions inferred during the day, and derivations from this pattern can lead to a new event observation that eventually can be provided to the real-time intelligence layer.

4.2.2 Reliable Information Processing FG internal interfaces

The reliable information processing aims at the annotation of information sources with QoI and reputation. This annotation and a conflict resolution component should support other components of the CityPulse framework with handling different data sources. An additionally added test component is responsible for testing new application at the time they are designed.

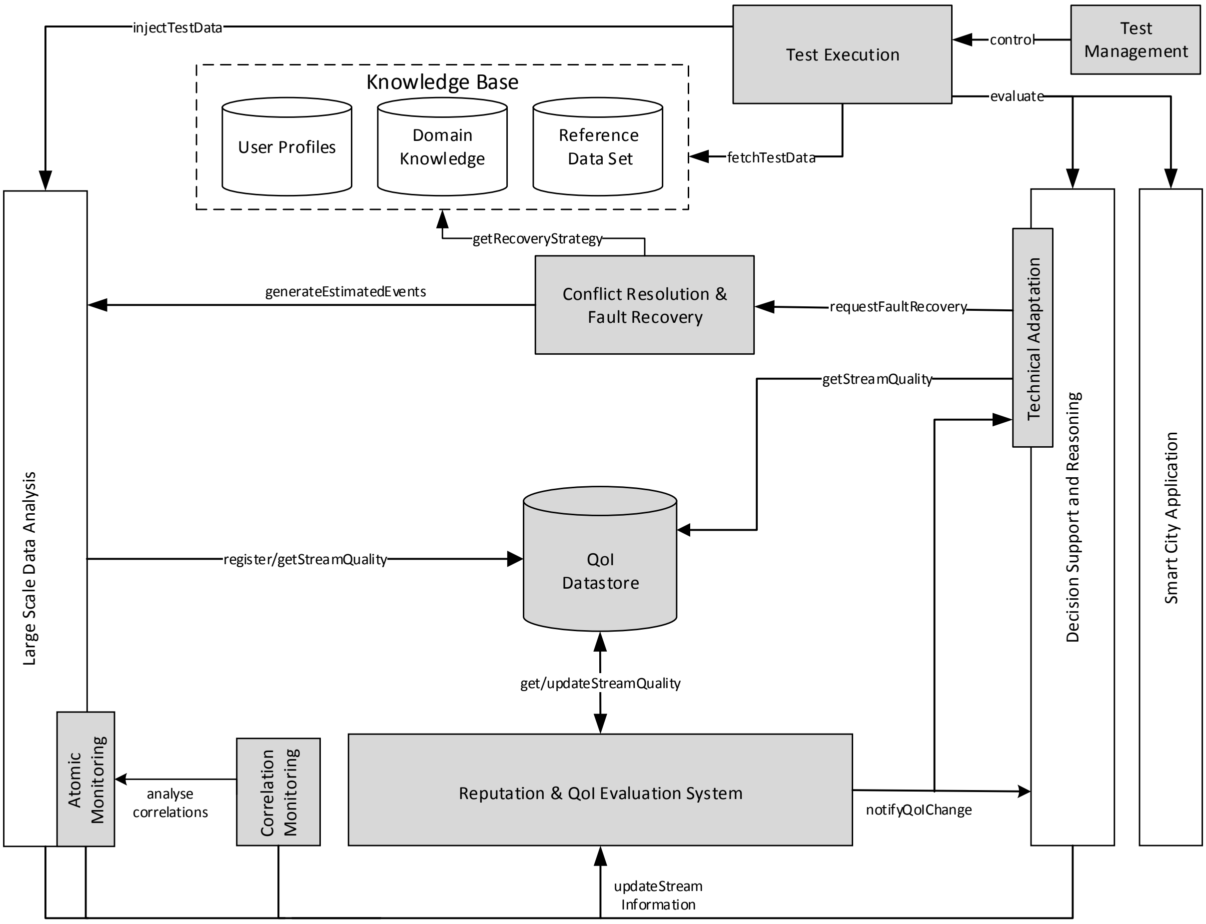

Figure 7 – Reliable Information Processing architecture

The reliable information processing consists of the following components which are shown in Figure 7 and described in detail in the following:

- Reputation & QoI Evaluation System and QoI Datastore
- Atomic & Correlation Monitoring
- Conflict Resolution & Fault Recovery and Technical Adaption
- Test Execution and Test Management

Reputation & QoI Evaluation System and QoI Datastore

The Reputation & QoI Evaluation System is the main component of the reliable information processing. It is responsible for the active evaluation of QoI for data sources and their steady adaption triggered by events that could be sent by the monitoring components and the event management components of other CityPulse framework modules. In contrast to the simpler monitoring components the Evaluation System has a complete view about all data sources and can find additional relationships between the data sources. With additional historical data is possible to adapt the QoI and save them to the QoI Datastore. In addition the reputation of data sources is calculated to maintain the trustworthiness of different data sources. Both the QoI and the reputation of a source are saved in the QoI Datastore. This store delivers this additional information to other components that need them to select the best stream for a specific use case. On the fly translation to semantic annotation formats ensures compatibility with various representation systems.

Atomic & Correlation Monitoring

The monitoring module consists of two components. The Atomic Monitoring is responsible for watching one single data stream for inconsistencies. As an example a data stream for an indoor temperature sensor should only deliver temperatures between 10 and 40 degrees. Otherwise there might be something strange happen or the sensor has a defect. Additionally the Atomic Monitoring could apply an incremental time series analysis to the data streams to model an expected behaviour of the delivered information. Because of the fact that the monitoring needs direct access to the data streams it is placed near the virtualisation of the stream.

In contrast to the Atomic Monitoring that watches the RAW data of one stream the Correlation Monitoring combines some Atomic Monitoring components and watches the abstract values that are delivered. Bases on entity-, time- and geospatial- relationships the different Atomic Monitoring components are aggregated and checked for plausibility. With the Correlation Monitoring it is possible to detect faulty information sources within a group of data sources by comparison with other group members.

Conflict Resolution & Fault Recovery and Technical Adaption

The conflict resolution and fault recovery component is triggered when one or several streams, used by an application, are not able to ensure the same QoI parameters (as were specified when the application was instantiated). Its role is to temporary generated estimated events for the above mentioned streams. Different interpolation methods and prediction models will be used for generating the estimated events. The interpolation methods are used when there are similar data sources in the surrounding area of the “faulty” data stream. If there are no other streams in the surrounding area a prediction model will be used instead. In this case fault recovery component will cache the events generated by the data source in the previous period (defined by the domain expert, e.g. one hour) and using this data will generate the predictions.

The fault recovery component generates the estimated events only for a short period of time. If the problem persists (the QoI of the stream was not restored to the initial value) the adaptation component triggers the resource discovery component to find an alternative data source.

Test Execution and Test Management

The additional test module adds the possibility to test new CityPulse applications. Therefore a reference data set is used. Driven by the use of the TestManagement component the data set is injected via the TestExecution into the Virtualisation layer of the CityPulse framework. The correct function of the new application is then evaluated through some evaluation interfaces. The whole testing component is designed to get used within the design time of a new application prior to deployment for public usage.

4.2.3 Reasoning and Decision Support FG internal interfaces

Figure 8 illustrates the communication and information flow between the different components in the Reasoning and Decision Support FG.

The flow is initiated by a user request, coming from the smart city application via the visualization component of the Reasoning and Decision Support FG.

The implicit user profile and the explicit application request and user requirements as well as background knowledge are mapped to a machine-readable format (XML-like) and passed onto the Decision Support component. The Decision Support component triggers a request for relevant up-to-date data to the stream discovery service, which comply with user preferences and initial application requirements based on NFP (e.g. accuracy, time of response, and other QoI parameters). The response to such request is passed back to the Decision Support component that provides an initial response to the visualization component for the user.

At this point two adaptive mechanisms are triggered till the user completes the task (e.g. gets to a destination along a path):

1. Technical Adaptation: this component receives notifications from the QoI interface in WP4 whenever there are some changes in the QoI of the streaming sources that have been selected by the Data Federation and Mash-up component to provide data; when such notified changes are in contrast with the QoI of sources selected for the composition plan that is providing answers to the application request, a request for adaptation and re-discovery of new more suitable sources is requested to the Federation component.
2. Unexpected event detection (and contextual filtering): as soon as the initial answer is provided to the user, the Decision Support component subscribes to a set of events that are related to the reasoning task (e.g. for travel planner task, accidents or road works might be relevant events). When any of such event is detected, the Event Manager sends a notification to the Contextual Filtering component; the Contextual Filtering component considers the user-dependent query response generated initially to determine if the relevant event is potentially critical and, if so, generates a trigger for a new data request to the Data Retriever, and communicates a list of critical events to the Decision Support component to perform a new computation and provide updated results to the user.

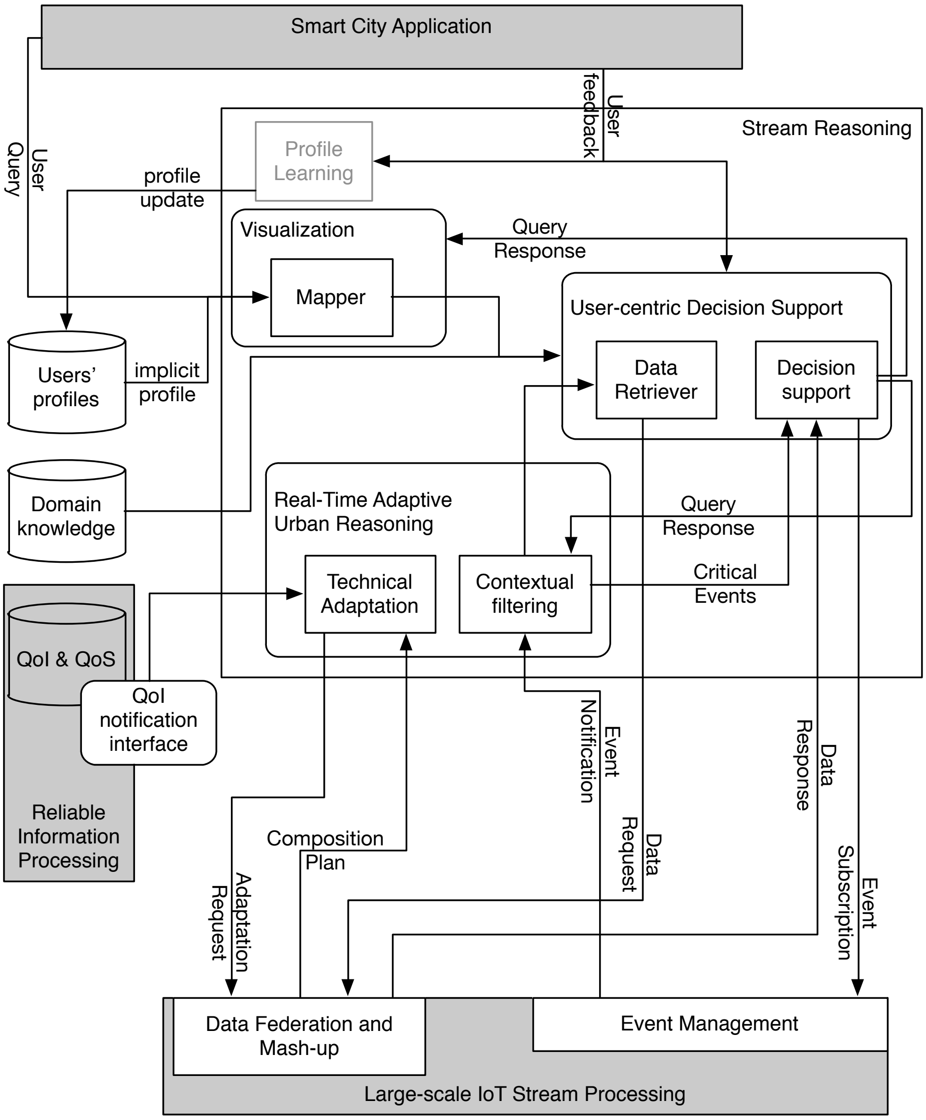

Figure 8 – Reasoning and Decision Support FG interfaces and Information Flow

4.2.4 Actuation FG interfaces and information flow

Figure 9 shows the information flow for the actuation functional group. The information flow is typically unidirectional from a CityPulse application down to an information sink. A CityPulse application issues a complex actuation task request and the request is checked in the Exposure function for application authentication and authorization rights to access the Actuation Management function. The Actuation Management function decomposes the complex actuation task into simple actuation tasks and discovers the simple information sinks, which can receive the simple actuation tasks. The simple actuation tasks are also checked for the proper authorization rights (e.g. if the CityPulse application has the right to send an actuation task to a specific actuator) and the simple tasks are dispatched to the Virtualization function. The Virtualization function adapts the simple actuation task language to the specific formats/language that the specific information sink accepts and dispatches the actuation request.

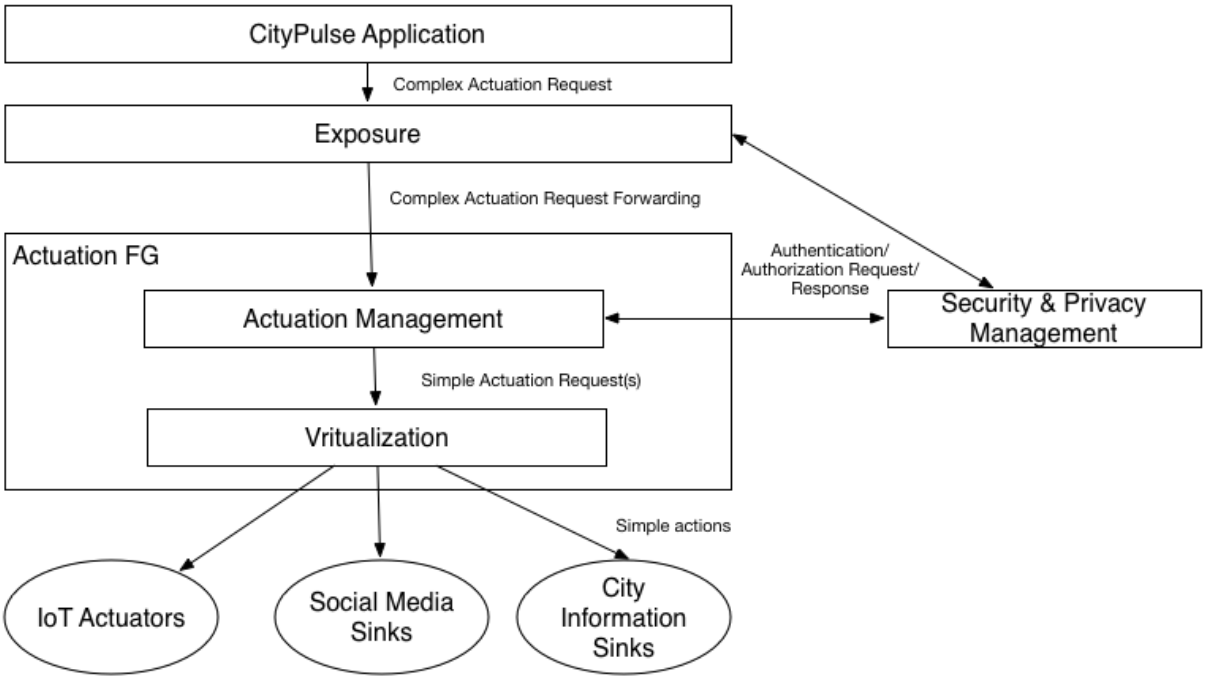

Figure 9 – Actuation FG interface and information view

4.3 Security and Privacy View

The CityPulse SCF integrates real-time information emanating from different types of information sources (IoT, social media and city sources) and either exposes processed information to users through the CityPulse applications or consumes information via the actuation tasks. Therefore the SCF security issues touch upon IoT, social media as well as streaming & static information security concerns. Moreover the SCF processing of physical world information may trigger privacy concerns that need to be addressed.

Since the focus of the CityPulse project is on the analytics functionality rather than the security and privacy concerns, we choose to rely on the state of the art techniques for addressing these concerns rather than re-invent existing work.

4.3.1 General Security and Privacy considerations

The Security Management function in the Framework Management FG is expected to address potential security issues with the SCF. It consists of Authentication (which in turn includes Identity Management), Authorization (access control rules and enforcement thereof), Key management (for all the authentication/encryption keys involved in the framework) and Trust & Reputation. One part of the Trust and Reputation function is implemented within the Reliable Information Processing FG and it covers the reliability and trust of information sources. With respect to trust of the SCF users it is assumed that there is a trust relationship between the users and the SCF in the sense that the SCF includes in the Authentication function the list of users allowed to access the SCF and in the Authorization function a list of access rights for each user. With respect to Authentication, Authorization state of the art solutions such as OpenID [OpenID], WebID [WebID], XACML[XACML] could also be used in the realization of the SCF.

With respect to privacy there are only a few potential entry and exposure points of private user-related information that the SCF needs to address. Private user-related information may enter the SCF through: a) though user registrations that result in the user profiles being stored in the Knowledge Base, b) the information sources (IoT, social media, city information sources). Moreover potentially private information (e.g. user profile information or stream data) may be exposed to CityPulse applications though the CityPulse API. User profiles as already described in deliverable D5.1 [CityPulse-D5.1] are stored for the sole purpose to assist the decision support mechanisms and through access control mechanisms (Authorization function) they are shared only to their owners. Aggregation techniques are used to create the profiles of groups of users (aggregate user profiles), which are shared to any CityPulse application through the CityPulse API if the application needs this information (e.g. "users between the ages of 20-40 prefer specific kinds of information").

With respect to the information sources the city information sources are assumed and expected to be void of personal user information and thus the IoT and social media sources might be the source for any privacy concern. Below we present a discussion on the specific security and privacy aspects of the IoT and social media streams and how the state of the art can be used to address these concerns.

In general we can assume that the SCF does not expose (via the Exposure FG) individual raw stream data (either IoT or social media data) since the Large Scale Data Analysis FG and Reasoning & Decision Support FG are expected to aggregate and fuse the streams thus removing any personal information. Examples of enabling technologies for aggregation are summarization techniques (via machine learning) are data cubes [DataCubes].

4.3.2 IoT Stream Security and Privacy

For the IoT security and privacy issues the SCF could reuse the relevant concepts from the state of the art coming from the IoT-A Trust, Security, Privacy model [IoT-A-ARM][ioT-A-D4.2] in addition to the trust and reputation concepts covered in the Reliable Information Processing FG, which are still work in progress. The IoT-A privacy mechanisms are based on Authentication, Identity Management and Authorization in the sense that they allow for derivation of pseudo-identifiers for users (Identity management, Authentication) and access control for stream information and SCF services enforced by the Authorization function.

Moreover semantically annotated IoT streams (in the Large Scale Data Analysis FG) can re-use the standardisation efforts by both the W3C [W3C] and OASIS [OASIS] with respect to privacy for web data since annotated IoT streams do not differ radically from annotated web data. The relevant standards (including WebID [WebID] as well as Access control languages like XACML [XACML]) have been enhanced or adapted through semantic technologies and they fulfil the requirement for a secure and privacy-preserving SCF.

4.3.3 Social Media Stream Privacy

This section addresses only the privacy issues from the social media sources while the security aspects (e.g. encryption of social media streams) is out of scope since by definition social media streams are public.

The social data within the context of the CityPulse project are expected to be collected from social media sources via open APIs (e.g. Twitter API) and/or user submitted information via smart devices (e.g. an app running on users mobile devices). Typically this kind of social information collected from sources such as Twitter is often in natural text form as shown in Figure 10. The motivation behind the use of social media streams in the context of Smart Cities is to provide a comprehensive view of events in a city (complementing other modalities such as observations provided by the IoT resources). Twitter (a microblogging platform) has developed into a near real-time source of information spanning heterogeneous topics of varying importance. With over 500 million users world-wide, Twitter generates 500 million tweets a day.

(a) Power shutdown (b) Poor water quality (c) Traffic jam due to procession (d) Delay in public transit system

Figure 10 – Tweets reporting various concerns about a city spanning power supply, water quality, traffic jams, and public transport delays

Increasingly, tweets do provide interesting and vital information such as status of public transport, traffic and environmental conditions, public safety, and general events in a city. The CityPulse project attempts to address the following research questions: a) how city infrastructure related events can be extracted from tweets, b) how event and location knowledge bases can be exploited for event extraction and c) how accurate the extraction of the city events is from tweets.

However users who have published Twitter data also have profiles on the SCF and sometimes with their real name and even with their location. Even though the Twitter data is public this public data can still be personal. Privacy in this context is an important issue. The project will consider different approaches to address this issue. We will not develop new methods and do research in privacy preservation but will use the state-of-the-art solutions to address the privacy issues.

A few key steps are considered in using social data from in CityPulse. For city event analysis it is expected that tweets will be stripped off user identification information resulting in a stream of data which includes only the text, time, location and other relevant metadata. Moreover the resulting stream of text is tagged via an automated process. An example of the result of the removal or personal information and the tagging of a tweet is shown in Figure 11.

Accident B-EVENT on O the O Golden B-LOCATION Highway I-LOCATION at O the O Viking O robots O in O Devland O JHB O , O ambo O truck O , O injured B-EVENT treating O themselves O

Figure 11 – A sample twitter text with annotation regarding the topic and location

However, tweets can still contain information about persons, location and other data that can make a person identifiable or refer to an event that has some privacy concerns. To address this issue, a social data analysis method is expected to use a pre-defined taxonomy of the events (i.e. an event ontology) in which only the events defined in this taxonomy will be identified and reported. These events will be generic for example accident, fire, traffic jam, air pollution, slow moving traffic and the event report will only contain the event "concept", start and end time and the location of the event. Figure 12 shows how location of an event is expected to be reported without any personal identification information. For internal trust and reliability issues there might be a pre-analysis of the collected data before submitting it to our event analysis but this analysis will take place close to the social media source and will not store the personal information.

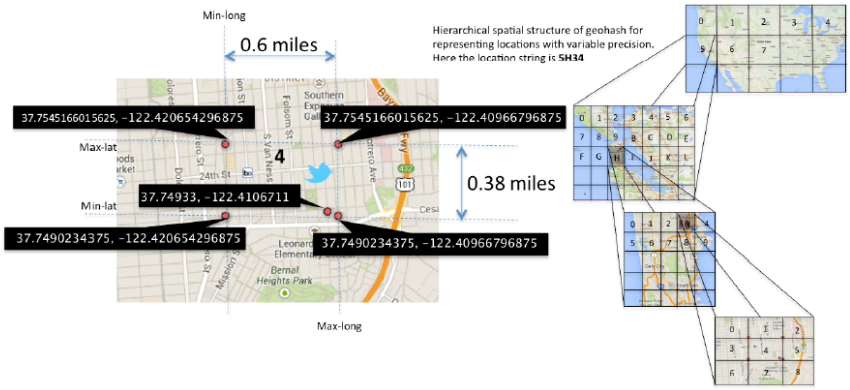

Figure 12 – Reporting location and using Geo-hashing for defining boundary of an event

Figure 13 shows a set of sample event concepts, which we are using for social data analysis. The results of the latter work will be reported in WP3 and here we only show the event concepts to clarify the privacy issues. Figure 14 shows a set of events that will be reported after the social media analysis.

incident;truck-fire
special-events;festival
incident;emergency-maintenance
incident;accident-involving-a-motorcycle
obstructions;downed-power-lines
traffic-conditions;residual-delays
sporting-events;race-event
incident;disabled-semi-trailer
incident;accident
disasters;grass-fire
incident-response-status;police-department-activity

Figure 13 – Event concepts in social data analysis

1	⟨incident;road-construction, [37.628892, -122.41652], 2013-07-31T09:19:46.0000, 1800⟩$_a$
2	⟨fair, [38.433036, -122.703], 2013-08-04T10:00:00.0000, 2013-08-04T23:00:00.0000⟩$_s$
3	⟨football-game, [37.715272, -122.387296], 2013-08-25T13:00:00.0000, 2013-08-25T21:00:00.0000⟩$_s$
4	⟨baseball-game, [37.778752, -122.390288], 2013-09-03T18:15:00.0000, 2013-09-09T23:00:00.0000⟩$_s$
5	⟨concert, [37.423516, -122.07812], 2013-09-14T09:00:00.0000, 2013-09-14T23:00:00.0000⟩$_s$
6	⟨football-game, [37.87112, -122.251824], 2013-09-14T11:59:00.0000, 2013-09-14T20:00:00.0000⟩$_s$
7	⟨concert, [37.423516, -122.07812], 2013-09-15T10:00:00.0000, 2013-09-15T23:00:00.0000⟩$_s$
8	⟨incident;accident, [37.768712, -122.407712], 2013-09-17T17:53:53.0000, 900⟩$_a$
9	⟨concert, [37.332192, -121.900544], 2013-10-07T18:30:00.0000, 2013-10-07T23:00:00.0000⟩$_s$
10	⟨concert, [37.423516, -122.07812], 2013-10-12T18:00:00.0000, 2013-10-12T23:00:00.0000⟩$_s$

Figure 14 – A set of events in a social data analysis scenario

In the above we discussed an approach that we have taken in the project for privacy preservation in reporting the city incidents using the social data; however there still could be other security and privacy issues and concern depending on the use-case and how the data is going to be collected and used.

4.3.4 Discussion

Overall for both IoT and social data source, there are several state-of-the-art techniques for aggregation, anonymisation and privacy preservation that can be used in the CityPulse SCF.

During implementation of the use-cases we will analyse the privacy and security requirements (the privacy issues in non-technical terms are already included in the use-case scenarios reported in D2.1 [CityPulse-D2.1]) and we will provide an application level privacy and security solution for the use-cases and will report on the best practices.

4.4 Smart City Framework and the relationship to IoT-A

The EU FP7 IoT-A project [IoT-A] provides an Architecture Reference Model (ARM) [IoT-A-ARM] for IoT concrete architectures. Since the CityPulse Smart City Framework includes some IoT components, a subset of its functional architecture could be mapped to the IoT-A reference architecture. One way to visually relate the two reference architectures is show in the Figure 15 below.

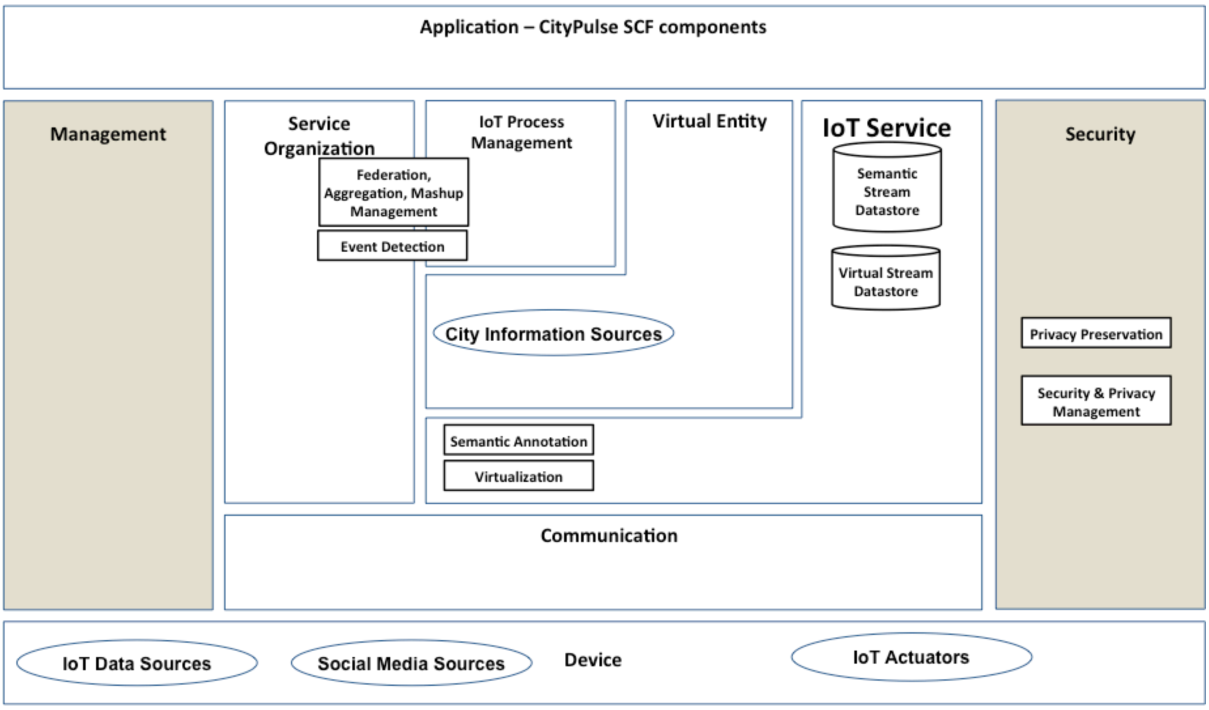

Figure 15 – Mapping a subset of the CityPulse functional components to the IoT-A reference architecture

The implications of mapping the Smart City Framework components to the IoT-A are:

1) IoT-A only covers IoT related functional components. Other CityPulse SCF components that are not IoT-specific are mapped to the Application Functional Group of the IoT-A architecture.
2) The SCF components mapped to IoT-A functional groups IoT-Service, Virtual Entity (VE), Service Organization and IoT Process Management are assumed to follow a Service Oriented Architecture principle. In turn this means that these components expose service interfaces that can be invoked by any other service in the SCF and the requirement that the service interfaces are registered into a service registry so that they are discoverable.
3) The SCF components mapped to the IoT-A functional groups IoT-Service, Virtual Entity (VE), Service Organization and IoT Process Management are assumed to expose their interfaces to any other SCF components in the SCF. The motivation is that the FGs IoT-Service VE Service, Service Organization and IoT Process Management are assumed in the IoT-A architecture to expose their services to the applications i.e. high level components.

The CityPulse SCF components that relate to the IoT-A functional architecture view are the following:

1) IoT Sources and IoT Actuators map to the IoT-A Device functional group since these are typically the sensors and actuators deployed in a city environment. Moreover Social Media Sources that report IoT-related information can also be conceptually mapped in the IoT-A Device FG in the sense that e.g the mobile phone used for reporting e.g. a fire can be considered an IoT “sensor” for fires.

2) The Virtualization and Semantic Annotation functional components produce either virtual raw streams or semantically annotated streams of real-time city information. These streams from the point of view of the IoT-A functional view are IoT Services that expose the real-time streams through standardized interfaces. Moreover repeating the first implication stated above, these functional components are assumed to produce IoT-related information. In contrast a social media stream source that goes through the Virtualization and Semantic Annotation functions may not always produce IoT related information.
3) The Virtual and Semantic Stream Datastores can also be viewed as IoT Services exposing access to the historical IoT-related information collected in a city environment.
4) The City Information Sources that are IoT- related (e.g. maps, locations of public places, parking spot availability, etc) can be considered a Virtual Entity Services from an IoT-A point of view. The reason is that typically the City Information Services provide the information annotated with a Virtual Entity information attached to it, e.g. "the parking space on Main Street has 5 free spots". Here "Main Street" is the Virtual Entity and "5 free spots" is the IoT service that provides the number of free parking spots on Main Street. The City Information Sources also expose a Virtual Entity type of access interface i.e. an interface that allows queries based on Virtual Entity identifiers as basic parts of the query, e.g. "What are the free parking spots on Main Street?".
5) The Event Detection FC and the Federation, Aggregation & Mashup Management FC can be mapped to the IoT-A Service Organization and IoT Process Management since the operations that these FCs represent are typically compositions of semantically annotated streams or workflows that involve semantically annotated streams of city information.
6) The Security & Privacy FC from the SCF can be mapped to the Security FG of the IoT-A since the latter covers Trust, Reputation and Privacy apart from the typical security functions such as confidentiality protection.

4.5 Smart City Framework Update

While the rest of the document includes the Smart City Framework design at the end of the first year of the CityPulse project, this section describes an update of the framework at the end of the second year. This update is not required according to the DoW, nevertheless the project decided to publish it as a courtesy to the research community. The update reflects the impact of the results of the individual work packages on the system architecture and the update should be considered as the CityPulse final architecture. Figure 16 below shows the updated functional architecture annotated with the Application Programming Interfaces (APIs) for each functional component. The description of the functions and the APIs is provided below. Please note that the APIs are described in natural language and not in a specific programming language although some information may exist about the implementation details of selected functions. A subset of these APIs are either already implemented or will be implemented in in the CityPulse prototype platform.

Resource Management

The Resource management component is responsible for managing all Data Wrappers. During runtime an application developer or the CityPulse framework operator can deploy new Data Wrappers to include data from new data streams. Furthermore deployed Data Wrappers can be deactivated or removed from the system. For a definition of the Data Wrapper please see below.

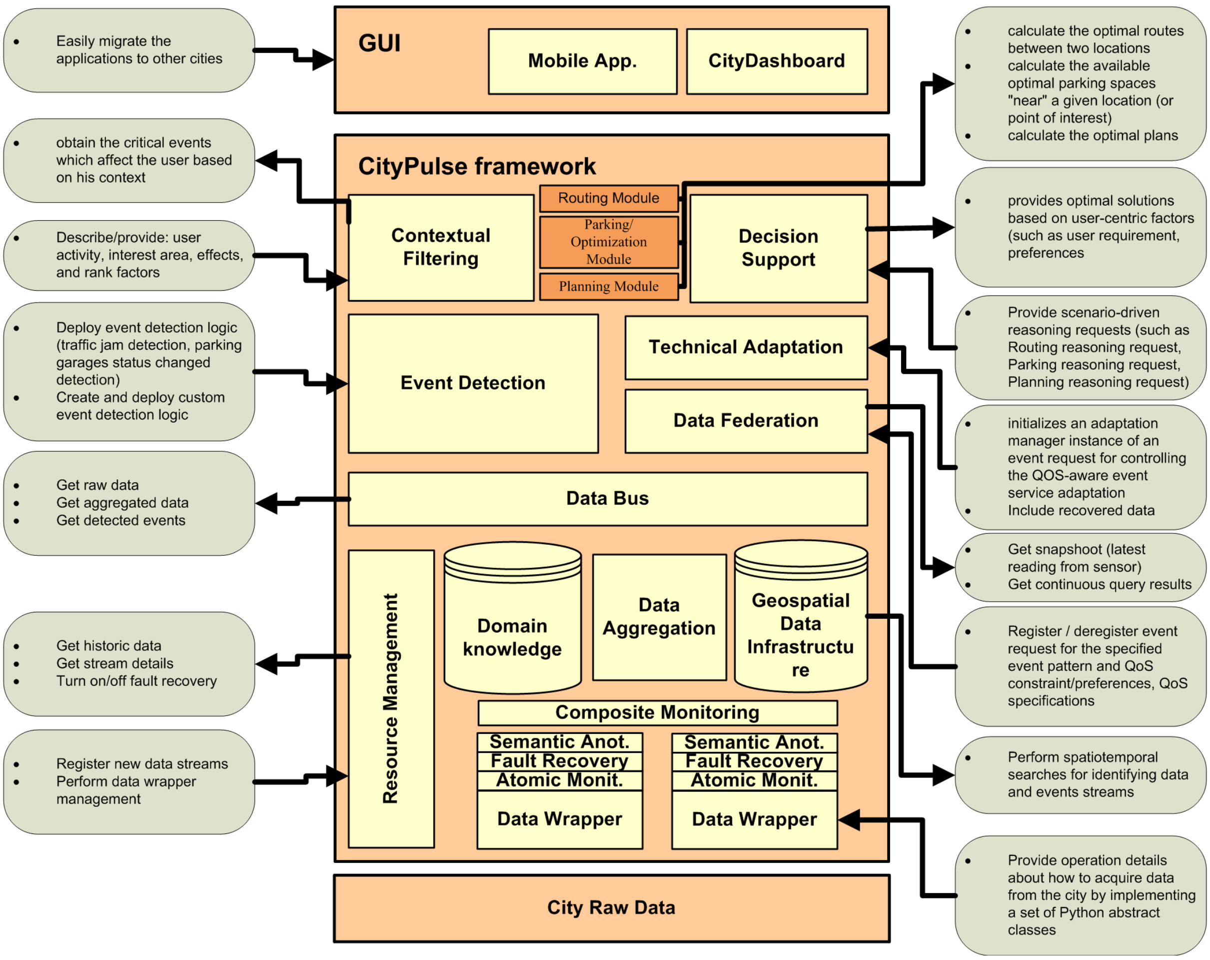

Figure 16 – Update of the Smart City Framework at the end of year two (2) of the project

Data Wrappers and Semantic Annotation

A Data Wrapper is the gateway of a data stream into the CityPulse framework. Depending on the type of stream, transport technology or the stream data format many different Data Wrappers may exist. Basic information about the stream is provided via a Sensor Description. A Sensor Description also holds all required information for the semantic annotation of the stream and observations. Operational details are implemented extending a set of abstract classes in the programming language Python. Often required features, such as periodic pulling of data from a HTTP resource, have been developed within the CityPulse project and can be used directly. The implementation along with a deployment descriptor can be bundled and transferred to the Resource Management via the Resource Management API.

Data Aggregation

The Data Aggregation component deals with large volumes of data using time series analysis and data compression techniques to reduce the size of raw sensor observations that are delivered by data wrappers. This allows reducing the communication overhead in the CityPulse framework and helps performing more advanced tasks in large scale, such as clustering, outlier detection or event detection. To effectively access and use sensor data, the semantic representation of the aggregations and abstractions are crucial to provide machine-interpretable observations for higher-level interpretations of the real world context. To date, most of the smart city frameworks transmit raw sensor data and lack energy efficient time series analysis as well as granular semantic representation of the temporal and spatial information for the aggregated data.

Data Federation

The main objective of the Data Federation component is to compose sensor streams and respond to explicit user/application queries over the streams. The sensor streams are modelled as primitive event services and the query is modelled as a complex event pattern. Users/Applications can also specify their non-functional requirements as QoS constraints and preferences.

Event Detection

The Event Detection component is generic and can be used to detect relevant events for the city by processing the sensor streams. The component has a Java API, which allows the user to define custom made event detection patterns.

Contextual Filtering

The Contextual Filtering component aims at providing adaptive feedback to the users by triggering new configurations that take into account the potential effect of detected events and user's context. The application developers can trigger the filtering mechanism in order to obtain the critical events, which affect a user based on her/his context. They can also rank these critical events by providing the logic of ranking factors (eg. the closer (temporal) events are, the more critical they are).In detail, developers need to provide arguments; UserActivity, InterestArea, Effects, and RankFactors with their total order as input of the function startCF(UserActivity, InterestArea, Effects, RankFactors) in order to obtain the ContextualEvent with the criticality.

Decision Support

The Decision Support component supports decision-making that takes into accounts the user-centric factors and usage patterns. User-centric factors such as handling user requirements, preferences and previous application usage patterns will enable goal-driven customisation of smart city applications and provide dedicated decision-making support to users.

Fault Recovery

The Fault Recovery component is strongly integrated in the Data Wrapper component. When the Fault Recovery is turned on, an instance of Fault Recovery component is triggered for estimating the observations when the stream quality is low.

Atomic Monitoring

The Atomic Monitoring component is strongly integrated into the Data Wrapper. After some initial processing steps on the received data by the wrapper, an instance of the Atomic Monitoring is called to calculate the quality.

Composite Monitoring

The composite monitoring provides the validation of events by comparing them to correlated streams that could affect or be affected by the event. This data evaluation will be triggered automatically if resources are available but can also be triggered manually via an API, which allows evaluating a certain event. The API can also be used to update the models of potentially affected service categories.

Geo-spatial Data Infrastructure

This component allows spatiotemporal searches for related events and streams. It allows access to the geo data infrastructure, which is based on Openstreetmap and gives access to the multimodal routing tool.

Technical Adaptation

The Technical Adaptation component monitors the QoS performance of the queries registered at the Data Federation component and make adjustments when necessary. The recovery and adjustments can be transparent to the user/application. Currently the technical adaptation module is tightly coupled with the ACEIS engine and provides an API to specify the mode of the adaptation as follows:

- createTechnicalAdaptationManager(EventRequest,AdaptationMode). This method initializes an adaptation manager instance for an event request, which controls the qos-aware event service adaptation. The AdaptationMode can be local, global or incremental, indicating different adaptation strategies to be used.

5. Design concepts for technical work

In this section we present the main initial design concepts which guided the design of the main functional groups of the smart city framework: a) Large Scale Data Analysis, b) Reasoning and Decision Support and c) Reliable Information Processing

5.1 Large Scale Data Analysis

The design of the Large Scale Data Analysis functional group is guided by the following design concepts:

i. Unified access to heterogeneous data sources. While smart city data is multi-source, multi-modal and varies in data format, having a unified access allows modelling the resources in a manner to access these resources systematically in interoperable way.
ii. Semantic Interoperability. Using semantic descriptions for sensor data enables representation, formalisation and enhanced interoperability of sensor data. We use information models as a design concept to store semantic concepts that represent a phenomena and attributes from the real world that are understandable for the human user and also interpretable for due machines to the standardised data representation.
iii. Efficient data aggregation and summarisation. To cope with the large amount of data that has to be processed and stored, dimensionality reduction techniques can be applied to reduce the size and length of the data by applying different methods on the data while keeping the key features and patterns. The aim is to reduce the amount of data by having a high granular representation of the sensor measurements at times when there is high activity and a lower granular representation in times of low activity.
iv. Real-time event detection. To perceive a concept or phenomena both the current and past states are required. To model and include this time-dependent aspect of higher-level concept creation, we aim to use data mining techniques, which will enable to predict events and processes that occur over a certain amount of time, based on available IoT data or extracting knowledge from human sensory data (on social media).

5.2 Reasoning, Decision Support

Knowledge-based automation for stream discovery and configuration relies on the following concepts that will be developed and documented in future deliverables by WP5 and that help scalability:

i. Uniform representation of stream metadata. This concept is related to the interoperability requirement and is achieved by the technical work in WP3 through virtualization and semantic annotation
ii. Re-usability of event patterns. In event detection, reusability of event pattern can make it more efficient to detect complex events. With the reuse (partial) results of similar event queries where partial patterns have been already evaluated, there is no need to re-evaluate the whole query, saving in time and resources that might get critical in real-time applications; This is why efficient discovery and composition of streaming event services in CityPulse have a big advantage when reusability is considered
iii. Real-time monitoring. The key for configuration control is the ability to monitor the quality of the streaming information that is going to be processed in our Smart City Framework: Real-time monitoring enables to continuously verify the quality metrics of the IoT streams involved in stream federation and discovery and to take action when the quality drops (adaptive control loop)
iv. Critical assessment. In order to determine whether any particular update in the quality of incoming streams is critical requires to identify application-driven threshold for acceptable data quality (at the level of data streams) and quantify the impact of high-level events on the particular decision process that is using them (at the level of events and decision support). The first aspect is specified at design time by the application and can be adapted by the user before the application is run, while the second aspect requires to assess the contextual

relevance of an event and it is done by building a context graph as documented in deliverable D5.1 [CityPulse-D5.1].

v. Adaptability: Once the criticality level (be it for quality updates or for high-level events) is defined and assessed, the Smart city Framework is expected to have the ability to trigger actions that change the way configuration and processing are performed. For stream discovery, this means switching one streaming source for another when the quality drops, while for event detection and decision support, it means changing the way reasoning is performed to take into account changes in the real world that might affect the reasoning outcome (such as an accident might invalidate a previously computed navigation path that might no longer be optimal).

5.3 Reliable Information Processing

The Reliable Information Processing relies on the following concepts that will be defined in future deliverables of WP4 to ensure efficient data processing and scalability:

i. Uniform quality information representation: To achieve full interoperability a clearly structured information representation for the distribution of stream quality is defined. It is based on semantic annotation and extensively described interfaces.

ii. Event Based Notification: Decreasing quality of smart city data streams can lead to immediate need of changing information sources for applications. Therefore the reliability processing uses event based publish/subscribe mechanisms to inform the Real-Time Adaptive Urban Reasoning and the Data Federation and Mash-up.

iii. Event Based Information Processing and Real-Time Atomic Monitoring: The utilisation of event detection allows an efficient analysis of aggregated data without an enormous increase of needed network infrastructure. It is used to enable virtualisation modules to inform the Reputation & QoI Evaluation System about events that could lead to quality annotation adaptation.

iv. Composite Monitoring of Aggregated Data: Correlations of various data streams can lead to validation or disproof of the correctness of a data stream. Therefore adapted knowledge is compared to information provided by similar streams without comparing the raw data. E.g. the average speed of traffic in an area can be compared to areas and dates of traffic incidents and thereby validate the expected behaviour.

v. Incremental Time Series Analysis: To prevent recurrent time series analysis of old datasets, incremental approaches that use knowledge gathered by previous algorithmic operations enable an efficient data processing.

vi. Reusing Stream and Application Descriptions to Generate Tests: A model-based testing approach utilises information of interfaces and streams to semi-automatically generate test cases for novel smart city applications and conflict resolution algorithms. A Reference Data Set is used to as test data input as well as to compute a test verdict.

6. State of the Art

This section presents the state of the art on smart city frameworks as defined for this deliverable. The existing work does not have exactly the same aims as the CityPulse project and therefore covers parts of the targeted functionality by CityPulse.

The Smart City Framework aims at bridging the gap between the related technologies and platforms (e.g. iCity platform [iCity]) and the higher-level knowledge that is required from intelligent decision-making by citizens and city operation services. While the existing solutions focus on publishing the data and creating service enablers to interact/access to IoT infrastructures in the cities, the SCF focuses on integration and federation of IoT (and social) data streams and provides processing and analysis mechanisms to aggregate, summarize and create higher-level abstractions from myriad of multimodal streaming data. This provides a mechanism for responding to real-time queries, and event detection and knowledge extraction and discovery processes that are required by end users and city operators to make intelligent and situation-aware decisions in different domains from daily-life tasks (e.g. commuting in the city) to operation planning and maintenance.

6.1 KAT

The Knowledge Acquisition Toolkit [KAT] is designed to support the process of knowledge acquisition from numerical sensory data. It aims to provide a toolkit that is able to extract and represent human understandable and/or machine interpretable information from raw data.

The toolkit includes a collection of algorithms on each step of the acquisition workflow ranging from data and signal pre-processing algorithms such as Frequency Filters, dimensionality reduction techniques such as PCA, Wavelets, FFT, SAX, and Feature Extraction and Abstraction and Inference methods such as Clustering, Classification and Logical Reasoning. KAT can be used to design and evaluate algorithms for sensor data that aim to extract and find new insights from the data.

6.2 GSN

Global Sensor Networks (GSN)[GSN] provides a generic platform for deploying sensor networks and processing data produced by the sensor network in a distributed fashion.

GSN achieves this goal by supporting rapid and simple deployment of a wide range of sensor network technologies, facilitating the flexible integration and discovery of sensor networks and sensor data, enabling fast deployment and addition of new platforms, providing distributed querying, filtering, and combination of sensor data, and supporting the dynamic adaption of the system configuration during operation.

The development of GSN is based on the observation that most of the requirements between the software platforms developed for sensor networks are similar, and it offers virtual sensors as a simple and powerful abstraction which enables the user to declaratively specify XML-based deployment descriptors in combination with the possibility to integrate sensor network data through plain SQL queries over local and remote sensor data sources.

6.3 iCity

The European project iCity [iCity] provides a framework for smart cities to create, deploy and operate services that utilise publically available information from digital assets and infrastructures. iCity extends the idea of Open Data and approaches to develop Open Infrastructures where public users can access and use the information from ICT networks that are deployed in the cities. The project provides tools and mechanisms for the third party service developers to enable sharing and ac-cessing the infrastructure and the data from the smart cities to the end-users and city authorities.

In overall iCity provides a platform for smart cities to share their information and provide access to the city infrastructures. The platform will enable third parties to develop services that access and use this information. The project is expected to create a set of services for public interest (e.g. mobile apps, web services) created by third parties or end-users that will be offered through the iCity Platform. iCity also aims to make the users involved in creation and utilisation of the services on the iCity platform.

6.4 iCore

The European project iCore [iCore] aims to empower the IoT through cognitive technologies so as to address two key issues: (i) how to abstract the technological heterogeneity of objects/devices in the IoT, while enhancing reliability and (ii) how to consider the views of different stakeholders for ensuring proper application provision, business integrity and, thus maximize exploitation opportunities. The cognitive framework [iCoreArhRef] comprises 3 levels of functionality, reusable for diverse applications: Virtual Objects (VOs), Composite Virtual Objects (CVOs) and Service level. VOs are virtual representations of devices (e.g. sensors, actuators, smartphones, etc.) associated to everyday objects or people (e.g. a table, a room, an elderly, etc.) that hide the underlying technological heterogeneity. CVOs are cognitive mash-ups of semantically interoperable VOs, delivering services in accordance with user/stakeholder requirements. The latter are derived at the Service level, which also includes capabilities for building, updating and exploiting Real World Knowledge. Cognitive entities at all levels provide the means for self-management (configuration, healing, optimization).

6.5 IoT.est

The European project IoT.est [IoT.est] aims at developing an IoT service creation environment whilst bridging the gap between various business services and the heterogeneity of networked sensors, actuators and objects. The approach employs semantic service descriptions to compose IoT services and derive corresponding functional conformance tests, semi-automatically. To enable IoT integration, a consistent service concept is specified. The implemented platform enables re-usage of atomic IoT services by composing them with the use of BPMN (Business Process Model and Notation).

6.6 OpenIoT

The European project OpenIoT [OpenIoT] aims at providing an Open Source Blueprint for large scale self-organizing cloud environments for IoT Applications. Besides enabling the concept of "Sensing-as-a-service" and providing efficient ways to manage cloud environments in IoT context via utility-based IoT services, OpenIoT also includes a middleware platform and tools for development of cloud-based IoT applications. Such platform blends IoT with semantic technologies (based on SSN ontology) for better interoperability, and supports virtually any sensor type available, including physical and virtual sensors. Furthermore, the platform provides access to an enhanced version of GSN (X-GSN), includes several visual tools and supports the implementation of on-demand IoT services. The OpenIoT architecture is a practical instantiation of the IOT-A [IoT-A] / IERC [IERC] ARM.

6.7 S-Cube

The European Network of Excellence S-Cube [S-Cube] exploited the synergy and learning effects across traditional research boundaries, and devised an integrated set of principles, techniques and methods for engineering, adapting and monitoring hybrid service based applications, while guaranteeing end-to-end quality provision and SLA conformance.

6.8 PLAY

The European project PLAY [PLAY] aimed at revolutionising the way that People, Things and Services cooperate in the Future Internet by enabling ubiquitous exchange of information between heterogeneous services in large scale distributed environments. It proposed the idea of situational-driven process adaptability and strives to develop an elastic and reliable architecture for dynamic and complex event handling.

6.9 Smart Santander

The European project SmartSantander [SmartSantander] is a unique in the world city-scale experimental research facility in support of typical applications and services for a smart city. The testbed that has been deployed has a dual purpose. On the one hand it allows real-world experimentation on Internet-of-Things related technologies (protocols, middleware, applications, etc.). On the other hand it is currently supporting the provision of smart city services aimed at enhancing the quality of life in the city of Santander.

The framework provides interconnected IoT infrastructure and physical deployment of IoT in the following cities: Santander, Guildford, Lubeck and Belgrade. It supports secure IoT communication providing support for various experiments. The SmartSantander testbed consists of four subsystems, which operate across a set of different devices (IoT, Gateway and Testbed server nodes) providing different characteristics and capabilities, as follows: Authentication, Authorization and Accounting; Testbed management; Experimental support; Application support.

6.10 SPITFIRE

The European project SPITFIRE [SPITFIRE] project aims to facilitate the efficient development of robust, interoperable and scalable applications in the Internet of Things. The SPITFIRE key objectives include: enabling search, interpretation and transformation of low level data on the basis of its explicit semantics, and developing a common semantic abstraction of the real world entities. The Service model and event-handling in SmartCity may benefit from the approaches developed in SPITFIRE project. However, the focus of the CityPulse project goes beyond providing IoT abstraction layer. CityPulse plans to provide an open service market matchmaking platform making both IoT and various data streams discoverable taking into the account environmental dynamics. CityPulse will provide support for reactive, smart applications allowing building smarter and real-world driven applications.

7. Conclusion

This report presented the CityPulse Smart City Framework, a high level architecture of the CityPulse technical components, how they are related and how they interact with each other. The SCF is an initial step for other CityPulse work packages to set a common language and common boundaries of the individual innovations as well as city stakeholders as a general map of the innovations that the project will contribute with. However the SCF specification describes more functionality than intended from the description of work and therefore this functionality will not be explored or detailed further within the technical work packages (WP3-WP6). For example a large part of the framework management and management applications are not expected to be developed further in research because they may already be covered by state of the art techniques. Nevertheless the framework as specified in this report covers the promised description of work and it has already been used by individual work packages other than WP2, for their internal system architecture definitions.

8. Abbreviations

ARM	Architecture Reference Model
BPMN	Business Process Model and Notation
CVO	Composite Virtual Object
DoW	Description of Work
FC	Functional Component
FCAPS	Fault, Configuration, Accounting, Performance, Security
FFT	Fast Fourier Transform
FG	Functional Group
FP	Functional Properties
FP7	Seventh Framework Programme (European research project funding framework)
GIS	Geographical Information System
GSN	Global Sensor Networks
I/F	Interface
ICT	Information and Communication Technology
IERC	European Research Cluster on the Internet of Things
IETF	Internet Engineering Task Force
IoP	Internet of People
IoT	Internet of Things

IoT-A	Internet of Things Architecture (EC FP7 project)
KAT	Knowledge Acquisition Toolkit
KB	Knowledge Base
NFP	Non-Functional Properties
OASIS	Organization for the Advancement of Structured Information Standards
PCA	Principal Component Analysis
QoI	Quality of Information
RFC	Request For Comments
SAX	Symbolic Aggregate Approximation
SCF	Smart City Framework
SLA	Service Layer Agreement
SSN	Semantic Sensor Networks
VE	Virtual Entity
VO	Virtual Object
W3C	World Wide Web Consortium
WP	Work package
WSN	Wireless Sensor Networks
XACML	OASIS eXtensible Access Control Markup Language
XML	eXtensible Markup Language

9. References

[Chen09] Chen, Y., Shu, J., Zhang, S., Liu, L., and Sun, L., "Data fusion in wireless sensor networks," in Proc. 2nd ISECS, vol. 2. May 2009, pp. 504–509.

[CityPulse-D2.1] Presser, M. (editor), Vestergaard, L., Ganea S., "Smart City Use Cases and Requirements", EU FP7 CityPulse Deliverable D2.1, April 2014.

[CityPulse-D5.1] Mileo, A. (editor), et.al., "Real-time Adaptive Urban Reasoning", EU FP7 CityPulse Deliverable D5.1, July 2014.

[DataCubes] Han, J., Kamber, M., Pei, J., "Data Mining: Concepts and Techniques", Morgan Kaufmann Publishers, July 2011, 3rd ed., chapter 5.

[Ganz13] Ganz, F., Barnaghi, P., Carrez, F., "Information Abstraction for Heterogeneous Real World Internet Data", IEEE Sensors Journal, vol. 13, no. 10, pp. 3793-3805, 2013.

[GSN] Global Sensor Networks, http://info.iet.unipi.it/~marco/teaching/AmI/GSN.pdf

[Holler14] Holler, J., Tsiatsis, V., Mulligan C., Karnouskos, S., Avesand S., Boyle, D., "From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence", Elsevier, 2014, DOI: http://dx.doi.org/10.1016/B978-0-12-407684-6.00001-2

[iCity] Linked Open Apps Ecosystem To Open Up Innovation In Smart Cities (iCity), http://www.icityproject.com/

[iCore] Cognitive Virtual Objects (iCore) project http://www.iot-icore.eu/

[iCoreArchRef] Minerva R., (editor) et. al, "D2.3 iCore Architecture Reference Model", EU FP7 project, iCore Deliverable, http://www.iot-icore.eu/public-deliverables.

[IERC] European Research Cluster on the Internet of Things: http://www.internet-of-things-research.eu/

[IoT-A] Internet of Things Architecture project, http://www.iot-a.eu

[IoT-A-ARM] Carrez, F., Bauer, M., Boussard, M., Bui, N., Jardak, C., Loof, J.D., Magerkurth C., , Meissner, S., Nettsträter, A., Olivereau, A., Thoma, M., Walewski, J.W., Stefan, J., Salinas, A. (2013). 'Deliverable D1.5 – Final architectural reference model for the IoT v3.0', Internet of Things – Architecture IoT-A EC Project deliverable D1.5. [online]. Available at: http://www.iot-a.eu/public/public-documents/d1.5/at_download/file

[IoT-A-D4.2] Gruschka, N., Gessner, D.(editors), et. al., "Concepts and Solutions for the Privacy and Security in the Resolution Architecture", IoT-A Deliverable D4.2, Feb, 2012, http://www.iot-a.eu/public/public-documents/d4.2/at_download/file

[IoT.est] Internet of Things Environment for Service Creation and Testing project, http://ict-iotest.eu/

[Jesus11] Jesus, P, Baquero, C., and Almeida, P. S., "A Survey of Distributed Data Aggregation Algorithms", Cambridge, U.K.: Cambridge Univ. Press, Oct. 2011.

[Kimura05] Kimura N. and S. Latifi S., "A survey on data compression in wireless sensor networks," in Proc. ITCC, vol. 2. Apr. 2005, pp. 8–13.

[KAT] Knowledge Acquisition Toolkit: http://info.ee.surrey.ac.uk/CCSR/KAT/

[OASIS] Organization for the Advancement of Structured Information Standards, https://www.oasis-open.org/

[OpenID] OpenID, The Internet Identify Layer, http://openid.net/

[OpenIoT] OpenIoT project: http://openiot.eu/

[PLAY] Pushing dynamic and ubiquitous interaction between services Leveraged in the Future Internet by ApplYing complex event processing (PLAY), http://www.play-project.eu/

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", IETF RFC 2119, http://www.ietf.org/rfc/rfc2119.txt

[Rozanski11] Rozanski, N., Woods, E. (2011). Software systems architecture: Working with stakeholders using viewpoints and perspectives (2nd ed.) Addison-Wesley

[S-Cube] The Software, Services and Systems Network (S-Cube), http://www.s-cube-network.eu/

[SmartSantander] SmartSantander project, http://www.smartsantander.eu/

[SPITFIRE] Semantic Service Provisioning for the Internet of Things using Future Internet Research by Experiment (SPITFIRE), http://www.spitfire-project.eu/

[SSN] Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., et al. "The SSN ontology of the W3C semantic sensor network incubator group", Web Semantics: Science, Services and Agents on the World Wide Web, 17:25–32, 2012.

[Twitter] http://twitter.com

[W3C] World Wide Wed Consortium, http://www.w3.org/

[Wang12] Wang, W., De, S., Toenjes, R., Reetz, E., and Moessner, K., "A comprehensive ontology for knowledge representation in the internet of things", in 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pages 1793–1798. IEEE, 2012.

[WebID] WebID - Universal Login and Identity for the Web, http://webid.info

[XACML] OASIS eXtensible Access Control Markup Language (XACML) TC https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml