Exploiting Aggregated Open Data from Smart Cities for the Future Internet Society

D5.1: Orchestration and adaption methodologies report v1

Authors	Serdar Yumlu, Caner Tosunoglu, Eser Karakaya, Sinan Yucekaya, Ehsan Valizadeh, Omer Ozdemir, Peter Nilsson
Institution lead	SAMPAS
Version	0.8
Reviewers	Javier Cubo
Work package	WP5
Task	T5.1
Due date	31/3/2017
Submission date	28/9/2017
Distribution level (CO, PU):	Public
Abstract	This document describes methodologies for orchestration and adaptation to create an integrated service development, implementation, deployment and management framework.
Keywords	Orchestration, Integration Methodologies, Service Oriented Architecture, Enterprise Service Bus, WSO2

License information	This work is licensed under Creative Commons Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0) http://creativecommons.org/licenses/by-sa/3.0/

Document Description

Document Revision History

Version	*Date*	*Modifications Introduced*	
		Description of change	*Modified by*
v0.1	20/09/16	First draft version and TOC	Serdar Yumlu (SAMPAS)
v0.2	20/03/17	Added Malaga Open Data Formats	Omer Ozdemir (ATOS)
v0.3	31/03/17	Added Karlshamn Open Data Formats	Peter Nilsson (NetPort)
v0.4	12/04/17	Added Malatya Data Services	Ehsan Valizadeh (SAMPAS)
v0.5	19/04/17	Check the status and contributions	Eser Karakaya (SAMPAS)
v0.6	03/05/17	Check the status and contributions	Sinan Yucekaya (SAMPAS)
V0.7	04/05/17	Internal review	Javier Cubo (UMA)
V0.8	25/09/17	Overall revision of document and prepare final version to release	Ehsan Valizadeh (SAMPAS)
V1.0	26/09/17	Final version to release	Malena Donato (ATOS)

Table of Contents

Table of Figures

Table of Tables

Terms and abbreviations

FIWARE	Middleware platform, driven by the European Union, for the development and global deployment of applications for Future Internet
SMART-FI	Provides services for smart city applications, creates business opportunities and improves public's quality of life using Open Data
CC	Creative Commons is an American non-profit organization devoted to expanding the range of creative works available for others to build upon legally and to share
SOA	A service-oriented architecture is a style of software design where services are provided to the other components by application components, through a communication protocol over a network.
ESB	An enterprise service bus implements a communication system between mutually interacting software applications in a service-oriented architecture. It implements a software architecture as depicted on the right.
Mashups	A mash-up (sometimes spelled as one word, mashup) is a Web page or application that integrates

	complementary elements from two or more sources. Mash-ups are often defined by the type of content that they aggregate.
Data Service	Data services refers to service-oriented architecture (SOA) applied to data sourced from the World Wide Web and the Internet as a whole. Web data services enable maximal mashup, reuse, and sharing of structured data (such as relational tables), semi-structured information (such as Extensible Markup Language (XML) documents), and unstructured information (such as RSS feeds, content from Web applications, commercial data from online business sources).
Service Orchestration	Application or service orchestration is the process of integrating two or more applications and/or servicestogether to automate a process, or synchronize data in real-time. Often, point-to-point integration may be used as the path of least resistance.
WSO2	WSO2 is an open source technology provider that increases the agility of digital businesses and enterprises engaging in digital transformation.
SaaS	Software as a service (SaaS) is a software distribution model in which a third-party provider hosts applications and makes them available to customers over the Internet. SaaS is one of three main categories of cloud computing, alongside infrastructure as a service (IaaS) and platform as a service (PaaS).
Microservices	Microservices is a variant of the service-oriented architecture (SOA) architectural style that structures an application as a collection of loosely coupled services. In a microservices architecture, services should be fine-grained and the protocols should be lightweight.
MiDAS	Micro Data Analytic Services
WP	Work Package

Executive Summary

This document aims to provide model-based techniques to facilitate the interoperability among applications and methodologies for orchestration and adaptation. These techniques are used to create an integrated service development, implementation, deployment and management framework to ensure the governance on the creation and operation of services and their testing procedures. Related task of this deliverable also offers support for (i) discovery of added-value combinations of service components and automatically composing and refining these to the needs of the user, and (ii) delivery of these composed services at the right time and right place across different platforms and devices in a qualitative and dependable way.

This deliverable also covers software stacks SMART-FI consortium planned to use within orchestration and adaptation to create an integrated service development, implementation, deployment and management framework.

1 Introduction

This document supposes the first version of Orchestration and adaptation methodologies Report related to the objectives of task 5.1. ESB platforms and methodologies will be investigated and presented initially, such as WSO2 and Mule products and related products. Initial available data services for use cases will be presented at this version.

This deliverable considered as a desk search related to methodologies for orchestration and adaptation to create an integrated service development, implementation, deployment and management framework.

1.1 Document structure

The document is structured as follow. Section 1 is introduction. In section 2, we will introduce concepts and background related to orchestration & adaptation methodologies, SOA and standards, service integration and interoperability, service orchestration & composition techniques, enterprise services bus and mashups. Section 3 presents the Smart-FI data services considered for use cases in detail. With those data services, in section 4 we describe available service based orchestration facilities and resources. Section 5 presents the conclusion. Section 6 contains references.

2 Orchestration & Adaptation Methodologies

New business models have emerged, changing the way the service orchestration is used in the enterprises. Consequently, there is a demand for an increased flexibility as well as improvements in the ways service orchestration systems are designed, enacted and managed. The following two sections introduce these new business models and the importance of runtime adaptability of a service orchestration.

Service brokers play a significant role in the Web services ecosystem. Service brokers position themselves between the traditional service consumers and service providers and generate value by bringing the service providers closer to service consumers. Service brokers may repurpose the provided services and may add additional functionalities such as payment and monitoring mechanisms to existing service offerings. A fundamental requirement for service brokers is to aggregate a number of services to achieve a value addition. Therefore, improved service orchestration capabilities are paramount to ensure the efficient and more agile coordination of these services.

SaaS applications present a personalized client experience while maximizing performance and efficiency. Creating SaaS applications requires detailed knowledge of multi-tenancy, contextual personalization, declarative programming, and infrastructure scaling. Many teams are challenged by the specialized knowledge required to create a SaaS application on a legacy Java platform. SMART-FI consortium is investigating on platforms that provides a SaaS platform that enables SMART-FI platform to rapidly create SaaS applications with a minimal learning curve.

SaaS is gaining popularity as a way of on-demand delivery of software over the Internet. SaaS has become popular due to advantages such as quick return on investment and economies of scale. SaaS vendors can benefit from the advantages provided by service orchestration. SaaS and SOA complement each other SaaS is a software delivery model, whereas SOA is a software construction model. In order to build SaaS applications using SOA and deliver service offerings using service orchestration techniques, it is necessary to provide the support for agility and the support for multi-tenancy in service orchestration management systems.

2.1 Service Oriented Architecture (SOA) and Standards

There are four very important SOA architecture standards from The Open Group that provide the basis for business to create their SOA solutions. They are:

1. The SOA Reference Architecture
2. Open Service Integration Maturity Model (OSIMM)
3. SOA Governance
4. SOA Ontology

The SOA Reference Architecture

The Open Group SOA Reference Architecture (SOA RA) standard provides guidelines and options for making architectural, design, and implementation decisions in the architecture of service oriented solutions, including the architecture of cloud solutions. The goal of the SOA Reference Architecture standard is to provide a blueprint for creating and evaluating architecture. Additionally, it provides insights, patterns and the building blocks for integrating fundamental elements of an SOA into a solution or enterprise architecture.

Open Service Integration Maturity Model

The Open Group Service Integration Maturity Model (OSIMM) standard (and International Standard) gives organizations a method to assess their use of services and develop a roadmap for achieving their business goals with SOA. Adoption scenarios for SOA tend to vary considerably, especially when organizations lack a clear roadmap; the vision for how to proceed on their path to SOA adoption. The SOA journey does not start and end with a single project. As an increasing number of organizations continue to incorporate the use of service orientation as the foundation of their IT strategy, it will become increasingly important for them to assess their current state against several dimensions (from the business down to infrastructure) and identify ways to maximize business benefits from their SOA journey.

SOA Governance

The SOA Governance Framework has been standardized by The Open Group as a result of collaboration with the industry on IBM's SOA governance method. It defines governance to be a means establishing and enforcing how people and solutions work together to achieve organizational objectives. Governance helps to ensure that organizations build the right services, in the right way, at the right time, and then manage and reuse those services effectively. SOA governance does this by overseeing the processes of proactively identifying, assessing, building and managing high-value business services and solutions — those that provide the greatest return on investment. This means creating service reuse and providing agility in the ability to manage the business and IT.

SOA Ontology

The purpose of this standard is to contribute to The Open Group mission of Boundaryless Information Flow, by developing and fostering common understanding of Service-Oriented Architecture (SOA) in order to improve alignment between the business and information technology communities, and facilitate SOA adoption.

It does this in two specific ways.

First, it defines the concepts, terminology, and semantics of SOA in both business and technical terms, in order to:

- Create a foundation for further work in domain-specific areas
- Enable communications between business and technical people
- Enhance the understanding of SOA concepts in the business and technical communities
- Provide a means to state problems and opportunities clearly and unambiguously to promote mutual understanding

Secondly, it potentially contributes to model-driven SOA implementation.

The ontology is designed for use by:

- Business people, to give them a deeper understanding of SOA concepts and how they are used in the enterprise and its environment
- Architects, as metadata for architectural artifacts
- Architecture methodologists, as a component of SOA meta-models
- System and software designers for guidance in terminology and structure

Service Integration and Interoperability

With the advent of Service-Oriented Architecture (SOA) as the blue print for enterprise integration, most organizations tend to move away from monolithic applications and they expose their business functionalities as 'Services'. A 'Service' is defined as a logical representation of a repeatable business activity that has a specified outcome and does not depend on other services, unless it reuses them or it is composed of other services. By grouping various business functionalities as services in a well-defined and self-contained manner, the need for integrating these services to fulfil a business requirement has become a major challenge in the SOA world.

Obviously, there are many ways of integrating services and systems. If an organization really doesn't really care about the maintainability, scalability, troubleshooting, governance etc. then its most likely that the final integration outcome will end up in some sort of a Spaghetti Integration style. This is why the SOA needs a proper infrastructure that can provide a neat way of integrating diverse systems by preserving the ease of scalability, maintainability, troubleshooting and governance.

2.2 Service Orchestration & Composition Techniques

Service Orchestration is the process of accessing multiple fine-grained services from a single coarse-grained service. The service client would only have the visibility to a single coarse-grained service and would be encapsulated from the multiple fine-grained services that are invoked in the process flow.

Service orchestration can have two types of backend services.

1. Blocking backend services (synchronous)

2. Non-blocking backend services (asynchronous)

Blocking backend services would involve multiple services that are invoked as part of a sequence where one service is dependent on the response of the previous service. This type of backend services require the Service Orchestration Layer to manage and handle the service responses and use them as an input parameter for the next service.

2.3 Enterprise Services Bus (ESB)

The standard infrastructure to implement the SOA concept is the Enterprise Service Bus (ESB) and therefore all the communication between the services and the consumers are done via the ESB. Hence, the notion of service integration is a major requirement of any ESB implementation. ESBs should facilitate not just standards based integrations (such as web services) but also other types of integration such as integrating with legacy systems.

2.4 Mashups

With respect to service orchestration is the integration of more than one service to work together. Service composition will create apps generated in form of orchestrators or adaptors with new functionalities to be remotely accessed (SaaS or Mashups), which could be deployed in infrastructures (Clouds). The incremental construction of Future Internet apps through the adaptation and orchestration of reusable services specified with interfaces is an error-prone task.

2.5 Data-as-a-Service Adaptation Methodologies

Cloud-based technology is becoming increasingly complex, and so the as-a-service (aaS) space has, is, and will become increasingly crowded. In fact, it’s getting harder and harder for data professionals to keep track of each Cloud computing model, and how they all differentiate from one another. The reality is that this isn’t as much of a problem as it is an opportunity for data professionals to educate themselves and adapt to new technologies that really make life easier on the Data Management level.

Arguably, Data-as-a-Service (DaaS) is one of the few new kids on the Cloud computing model block to actually deliver on the promise to make life easier. In this article we’ll take a look at the DaaS model, and how it is making an impact.

Data-as-a-Service Explained

The big picture idea behind the DaaS model is all about offloading the risks and burdens of Data Management to a third-party Cloud-based provider. Traditionally, companies housed and managed their own data within a self-contained storage system. The problem with this traditional model is that as data becomes more complex it can be increasingly difficult and expensive to maintain.

With the DaaS Cloud computing model, data is readily accessible through a Cloud-based platform. Simply put, DaaS is a new way of accessing business-critical data within an existing datacenter. Within the DaaS environment information can be delivered to a user regardless of organizational or geographical barriers. Some of the most common business applications powered by DaaS technology includes Customer Resource Management (CRM) and Enterprise Resource Planning (ERP) applications.

Benefits of Data-as-a-Service

The same benefits that come with any major Cloud-computing platform also apply to the Data-as-a-Service space. The main exception for DaaS providers is that their benefits reach for and are deep into the world of Data Management.

- **Agility:** Due to the fact that the majority of today’s DaaS providers are based on Service Oriented Architecture (SOA), there is a great deal of flexibility attached to accessing business-critical data in a DaaS-powered Cloud service. Data can be accessed quickly because the architecture where the data is located is fairly simplistic. This also means that as the data structure needs shift, or geographical needs arise, the changes to data are incredibly easy to implement.
- **High Quality Data:** One major benefit has to do with improved Data Quality. This is largely due to the fact that the bulk of data access is primarily controlled through the data service itself. This adds a robust layer of security and improved data quality.
- **Cost Effectiveness:** Just like any Cloud-based offering, providers in the DaaS space can easily deploy their data delivering applications in a way that off sets many of the costs associated with managing and housing these complex data sets in-house. For instance, one way providers help organizations save money is on the presentation layer of their interfaces and applications. They can build them in such a way that makes it easy to change location-based and organizational assets in a fluid way.

The Future of DaaS: Business Intelligence & Healthcare

According to the popular IT research firm Gartner, the Data-as-a-Service model is expected to serve as a launching pad for the Business Intelligence (BI) and Big Data analytics markets. Right now the BI market is fairly limited to what Gartner refers to as a “build-driven” business model. That is, enterprise organizations merely license software so that they can build analytics on top of that software. The DaaS phenomenon will allow companies to subscribe to data services that bundle BI and analytics applications into the software license.

Beyond the world of basic Business Intelligence, like many other industries, the healthcare industry is rapidly adopting Big Data. As a result, the components needed to effectively manage Big Data greatly benefit from the adoption of Data-as-a-Service architecture. Some of these components include everything from Data Governance to data integrity to data storage innovations to agile information delivery architecture. The next generation of healthcare-centric data architectures will rely on a robust view of the DaaS space.

As with any new Cloud-based solution, there is some convincing that needs to happen before a full-scale DaaS adoption can take place. For starters, every organization from the top down must be convinced of any DaaS provider's inherent value. While the benefits of DaaS adoption are wide and deep, the criticism of Cloud-based data services (privacy, security and data governance) are concerning to say the least.

To look at it from another angle, it's definitely true that most IT processes can and should be measured in ROI. However, in the DaaS space, quantifying ROI can be difficult. Due to the nature of Cloud-based data sharing requires a re-imagining of IT to some degree. This is largely because, in the DaaS environment, Data Management shifts from an IT capability to a collaborative Data Management effort that moves data capability far beyond the supporting applications. This means that attempting to quantify value of DaaS based on money-savings and ROI is incredibly difficult, if not impossible.

The bottom line is that as the need for dynamic Data Management solutions increases, more and more organizations will start to consider DaaS as a viable option for managing mission-critical data in the Cloud. Any solutions that streamlines the Data Management process by synchronizing enterprise data with all internal applications, business processes, and analytical tools positions itself as a viable resource that will improve operational efficiency, while boosting the quality of reporting and data-driven decision making.

Again, the future of DaaS adoption is less dependent on the technical efficiency of the Cloud computing model, and more dependent on organizational alignment. This hinges on whether or not the value of DaaS solutions can be clearly communicated and understood throughout your organization.

3 SMART-FI Data Services

Data Services Server augments service-oriented architecture development efforts by providing an easy-to-use platform for integrating data stores, creating composite data views, and hosting data services. It supports secure and managed data access across federated data stores, data service transactions, and data transformation and validation using a lightweight, developer friendly, agile development approach. It provides federation support, combining data from multiple sources in single response or resource and also supports nested queries across data sources.

There is two ways to use data-as-a-service: by outsourcing your own data or taking advantage of public data managed by a third party.

3.1 MALATYA Data Services

This section introduces Malatya use case data services.

3.1.1 Data Service Description

- POIs
- Business Information
- Request and Complaint

3.1.2 Data Services

Table 1 Malatya Data Services 1

Field	Description
Unique ID	DSMalatya1
Short name	POI List
Data Source Type	RDBMS
Description	**Important Places in Malatya** such as districts, touristic places, museums, hospitals, schools, pharmacies, religious facilities, recreation areas, parks, education zones, cultural zones, government buildings, transportation routes, accommodation, business districts etc. will be presented via services provided by Malatya Metropolitan Municipality.
Expose Type	SOAP
Endpoint (if Available)	http://95.0.57.210:8011/SampasServices/Belediye/CityDynamicsProject/CityDynamicsService?wsdl
Operations (if Available)	queryPOINameByCriteria retrieveBuildingDetailUsing retrieveBuildingDetailUsingClass retrieveBuildingDetailUsingDetail retrieveNightPharmacyByCoordinate

	retrieveNightPharmacyDetail retrievePointDetail retrieveUsingCategory retrieveUsingGroup retrieveUsingType retrieveZoningState

Table 2 Malatya Data Services 2

Field	Description
Unique ID	DSMalatya2
Short name	Business Information
Data Source Type	RDBMS
Description	Business Information: retail industry information for all businesses (restaurants, shops, entertainment places, accommodation, etc.) and enables citizens to share their comments, feedbacks and get recommendations based on time of the day, location, type, ratings, their personalized historical
Expose Type	SOAP
Endpoint (if Available)	-
Operations (if Available)	-

Table 3 Malatya Data Services 3

Field	Description
Unique ID	DSMalatya3
Short name	Request and Complaint
Data Source Type	RDBMS
Description	Request and Complaint: Inputs of users such as scorings, comments, and feedbacks for important places will be recorded via MalatyaInsight application in a way that is accessible to other users. In addition, complaints and demands of citizens will be reported to Municipality directly.

Expose Type	SOAP
Endpoint (if Available)	-
Operations (if Available)	-

3.2 MALAGA Data Services

This section introduces Malaga use case data services. .

3.2.1 Data Service Description

- Bicycle stations real time data
    - Malaga Municipality exposes the real time data of the bicycle stations which contains geo location information, occupied and free slots for each station in the city via the open data portal.
- Bus routes data
    - Malaga Municipality exposes the data of the bus stations which contains geo location information, bus routes and numbers of the buses via the open data portal.
- Vehicle parking areas real time data
    - Malaga Municipality exposes the real time data of the vehicle stations which contains geo location information, occupied and free slots for each station in the city via the open data portal.
- Social events in Malaga
    - We are retrieving data (for social events happening in Malaga) from Eventful API [1] which contains a detailed XML response having all kind of information for each event
- Weather forecast in Malaga
    - We are retrieving daily weather forecasts data from Open Weather Map API [2].

3.2.2 Data Services

Table 4 Malaga Data Services 1

Field	Description
Unique ID	DSMalaga1
Short name	Bicycle stations data
Data Source Type	CSV
Description	List of the bicycle stations with real time parking data
Expose Type	SOAP
Endpoint	http://datosabiertos.malaga.eu/recursos/transporte/EMT/esta

(if Available)	cionamientos/Estacionamientos.csv
Operations (if Available)	

Table 5 Malaga Data Services 2

Field	Description
Unique ID	DSMalaga2
Short name	Bus Routes Data
Data Source Type	CSV
Description	List of static data about bus routes and stations
Expose Type	SOAP
Endpoint (if Available)	http://datosabiertos.malaga.eu/recursos/transporte/EMT/lineasYHorarios/stops.csv
Operations (if Available)	

Table 6 Malaga Data Services 3

Field	Description
Unique ID	DSMalaga3
Short name	Vehicle Parking Stations
Data Source Type	CSV
Description	Real time data for the Vehicle Parking areas
Expose Type	SOAP
Endpoint (if Available)	http://datosabiertos.malaga.eu/recursos/aparcamientos/smassa/smassa_ocupacion.csv
Operations (if Available)	

Table 7 Malaga Data Services 4

Field	Description
Unique ID	DSMalaga4
Short name	Social Events
Data Source Type	XML
Description	List of social events in Malaga
Expose Type	REST
Endpoint (if Available)	http://api.eventful.com/docs/events/search
Operations (if Available)	

Table 8 Malaga Data Services 5

Field	Description
Unique ID	DSMalaga5
Short name	Daily Weather Forecast
Data Source Type	JSON
Description	Daily Weather Forecast data for Malaga
Expose Type	REST
Endpoint (if Available)	http:// api.openweathermap.org/data/2.5/weather
Operations (if Available)	

3.3 KARLSHAMN Data Services

Karlshamn is running two different use cases, based on different data sets.

- Transportation use case – Data from Blekingetrafiken (the bus company)
- Energy use case – Data from Karlshamnsfastigheter (the owner of the building where the energy pilot is implemented)

3.3.1 Transportation Use Cases

3.3.1.1 Data Service Description

For the transportation use case, we are using different open datasets/services to be exposed by Blekingetrafiken, providing for example:

- Bus data
  - Bus number
  - Other static vehicle data
- Real time bus data
  - GPS location of the buses
  - Speed
  - Route
- Bus stations data
  - GPS location of the bus stations
- Bus routes data
  - Route number
  - Departure information
  - Arrival information
- Time tables data
  - Route
  - Direction
  - Time period

3.3.1.2 Data Services

Table 9 KARLSHAMN Data Services 1

Field	Description
Unique ID	DSKarlshamn1
Short name	Bus Data
Data Source Type	JSON
Description	Bus data such as bus number, location, speed, route etc.
Expose Type	REST
Endpoint (if Available)	-
Operations (if Available)	-

Table 10 KARLSHAMN Data Services 2

Field	Description
Unique ID	DSKarlshamn2

Short name	Bus Stations
Data Source Type	JSON
Description	List of data about bus stations
Expose Type	REST
Endpoint (if Available)	-
Operations (if Available)	-

Table 11 KARLSHAMN Data Services 3

Field	Description
Unique ID	DSKarlshamn3
Short name	Bus Routes Data
Data Source Type	JSON
Description	List of data about bus routes
Expose Type	REST
Endpoint (if Available)	-
Operations (if Available)	-

Table 12 KARLSHAMN Data Services 4

Field	Description
Unique ID	DSKarlshamn4
Short name	Bus Time Tables

Data Source Type	JSON
Description	List of bus time tables
Expose Type	REST
Endpoint (if Available)	-
Operations (if Available)	-

3.3.2 Energy Use Cases

3.3.2.1 Data Service Description

For the energy use case, we are using another set of open datasets/services to be exposed by Karlshamnsfastigheter, providing for example:

- Nodes
  - Name
  - Address
  - Status
  - Power consumption
- Sensor measurements, such as:
  - Precence
  - Temperature
  - Illuminance
  - Humidity
- Light monitoring and control
- Power Outlet monitoring and control

3.3.2.2 Data Services

Table 13 KARLSHAMN Data Services 5

Field	Description
Unique ID	DSKarlshamn5
Short name	Node Data
Data Source Type	JSON
Description	Data about nodes in the building automation system
Expose Type	REST

Endpoint (if Available)	
Operations (if Available)	

Table 14 KARLSHAMN Data Services 6

Field	Description
Unique ID	DSKarlshamn6
Short name	Lights Data
Data Source Type	JSON
Description	Data about lights in the building automation system
Expose Type	REST
Endpoint (if Available)	-
Operations (if Available)	-

Table 15 KARLSHAMN Data Services 7

Field	Description
Unique ID	DSKarlshamn7
Short name	Measurements Data
Data Source Type	JSON
Description	Data about measurements from sensors in the building automation system
Expose Type	REST
Endpoint (if Available)	-
Operations (if Available)	-

4 SMART-FI Service Orchestration

SOA, or Service Oriented Architecture, is an approach to developing enterprise systems by loosely coupling interoperable services - small units of software that perform discrete tasks when called upon - from separate systems across different business domains. SOA emerged in the early 2000s, offering IT departments a way to develop new business services by reusing components from existing programs within the enterprise rather than writing functionally redundant code from scratch and developing new infrastructures to support them. With SOA, functionalities are expressed as a collection of services rather than a single application, marking a fundamental shift in how developers approach enterprise architecture design.

A crucial aspect of SOA is service orchestration. Enterprise systems and integration projects designed according to SOA principles depend on successful service orchestration. Finding a platform with enhanced service orchestration capabilities, then, is a high priority for enterprises looking to build their systems according to SOA.

4.1 FIWARE Based Service Orchestration Techniques

Service orchestration is the coordination and arrangement of multiple services exposed as a single aggregate service. Developers utilize service orchestration to support the automation of business processes by loosely coupling services across different applications and enterprises and creating "second-generation," composite applications. In other words, service orchestration is the combination of service interactions to create higher-level business services.

Service orchestration works through the exchange of messages in the domain layer of enterprise applications. Since individual services are not programmed to communicate with other services, messages must be exchanged according to a predetermined business logic and execution order so that the composite service or application can run as it is demanded by the end-user. This is usually accomplished through enterprise application integration (EAI), which enables data integration, and the use of a central messaging engine such as an enterprise service bus (ESB), which routes, transforms and enriches messages.

Related to service orchestration is service choreography. Though both are employed to create composite services and applications in service oriented architectures, it is worth pointing out the differences. A service choreography model works without a central messaging engine or orchestrator while a service orchestration model relies on a central controller to couple services. In the former, the participating services each know the business logic and sequence and timing of message exchanges. In the latter, the participating services don't know that they are being orchestrated as part of a higher-level service; only the central controller knows the business logic and messaging sequence.

Service orchestration is a fundamental aspect of successfully implementing SOA. In a truly service oriented architecture, new applications are created by new orchestrations of existing services - not by writing new code.

4.2 SMART-FI Service Orchestration Methodology

At the SMART-FI platform facilities level, three layers represent the main components, as well as a Governance and Service Market Place. Each facility is capable to perform a set of processes to get its main goal. Heterogenous data sets will be managed in the Data normalization component. It generates data sets with access rules and a normalized schema, based on urban ontologies, which are stored in semantic data store. Next, these normalized data streams are used by the Data analytics microservices and Service orchestration components. The data analytics services enable the development and management of data analytic services in Smart Cities, providing elastic data analytic services to analyse the aggregated data for predictions and recommendations, as well as developing and managing the called Micro Data Analytic Services (MiDAS).

With the Service orchestration facility, mechanisms for deploying and integrating existing or new services and applications will be provided, obtaining more advanced and complex applications (mashups of services) and creating a marketplace that considers the FIWARE Lab and third-party applications. At the application level, our platform will create SMART-FI services to be used for Smart City Applications.

4.3 SMART-FI Service Orchestration in Use Cases

Since it is not possible to predict all the services and application that will emerge in the Smart Cities of the future, cities need an environment that enables innovation and layering of multiple services (data services, analytics services, etc.) on a common infrastructure. This should also allow the introduction of new elements and re-use of existing resources. To ensure the right service is delivered with the right quality and performance to the right users, a mechanism enabling careful planning, orchestration and assurance is required. By facilitating orchestration and integration of different public services, several business and Smart City functionalities, coming from different services, are exposed to the end users as a single service endpoint or as a comprehensive Smart City application. To this end, SMART-FI platform provides methods and tools for deploying and integrating existing or new services and applications, for producing more advanced applications with the composition and orchestration of simpler ones (mashups of services). In order to provide assurance, SMART-FI also aims to create a market- place that will be generated or enriched considering the Store in FIWARE and third- party applications giving value to the public data. For this purpose, SMART-FI platform provides model-based techniques to facilitate the interoperability among applications and methodologies for orchestration and adaptation to create an integrated service development, implementation, deployment and management framework to ensure the governance on the creation and operation of services and their testing procedures. SMART-FI will also support: i) discovery of added-value combinations of service components and automatically composing and refining these to the needs of the user, ii) delivery of these composed services at the right time and right place across different platforms and devices in a qualitative and dependable way, and iii) use of rigorous and lightweight model-based techniques to facilitate interoperability among applications running in an isolated based on orchestration and adaptation methodologies.

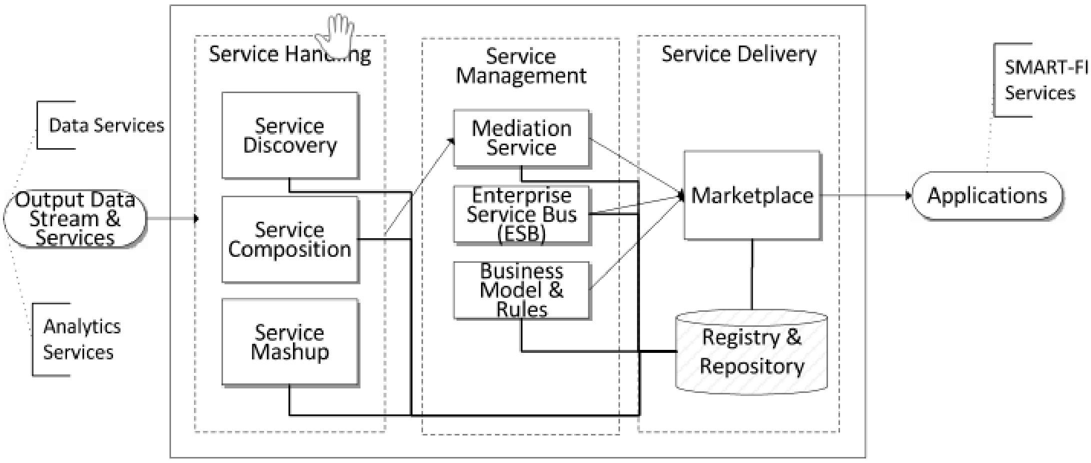

Figure 1 Service Orchestration Architecture

Service orchestration in SMART-FI platform has three different layers when enabling different set of data and analytics services to be provided for the use of smart city based applications. Service Handling is responsible for handling output data stream and services that may come from normalized data services or data analytic services. It comprises different characteristics of Service Discovery, Service Composition and Service Mashup delivery. Service Management is responsible for the mediation of services, business rules and Enterprise Service Bus (ESB) management. The Mediation Service here is a middleware component responsible for providing interoperability among different communication protocols and among different data models. For the effective Service Delivery all services all combined in the Registry and Repository. Composed and integrated services are delivered to the application layer through the Market- place component.

SMART-FI platform offers generative techniques based on integration services to make them pervasive and transparent to the user. Based on intuitive natural-language queries and user profile, mechanisms will infer the requirements and preferences and discover services to provide the desired functionality. Services will be automatically orchestrated on -demand to fulfill functional and QoS requirements. Mashups will be performed by users of the platform using the provided service orchestration and mediation capabilities. Service orchestration and exposure process is going to be achieved with the usage of Enterprise Service Bus (ESB) services. ESB service designers will be used to ease the life of smart city professionals to manage two service endpoints and transform the result into a new data or service. Mediator and sequence based components will used for message mediation and flow construction and holding and transferring the series of mediators respectively.

4.4 Service Orchestration with WSO2 Products

4.4.1 WSO2 ESB

WSO2 ESB is a fast, lightweight, and user-friendly open source ESB with a rich set of features that allows developers to easily implement their service orchestration scenarios.

Service orchestration is a crucial aspect in SOA. WSO2 ESB provides a rich set of features to implement service orchestration use cases. Configuration required to implement service orchestration scenarios has been further simplified with the introduction of 'Call Mediator' in the most recent WSO2 ESB release.

Service chaining and workflows, commonly referred to as orchestrations, are common integration scenarios in enterprise systems development. There are two distinct type of orchestrations we deal with when realizing these systems:

1. **Short-running stateless orchestrations**
   Service chaining messaging patterns that are more synchronous in nature and deals with transient data/sessions.
2. **Long-running stateful orchestrations**
   Service chaining messaging patterns that are asynchronous in nature. These workflows take human input and approvals and are configured to run for longer durations with more persistent sessions.

If we look at these two types a bit closely the short-running ones are more straightforward; it mainly follows a request/response pattern, talks to multiple service endpoints, derives a response from one service call and composes the message to be sent to another service call. Enterprise integration patterns, such as message splitting, transformation, cloning and aggregation, are key building blocks in such orchestrations. The enterprise service bus (ESB) pattern fits this description quite well.

If we look at these two types a bit closely the short-running ones are more straightforward; it mainly follows a request/response pattern, talks to multiple service endpoints, derives a response from one service call and composes the message to be sent to another service call. Enterprise integration patterns, such as message splitting, transformation, cloning and aggregation, are key building blocks in such orchestrations. The enterprise service bus (ESB) pattern fits this description quite well.

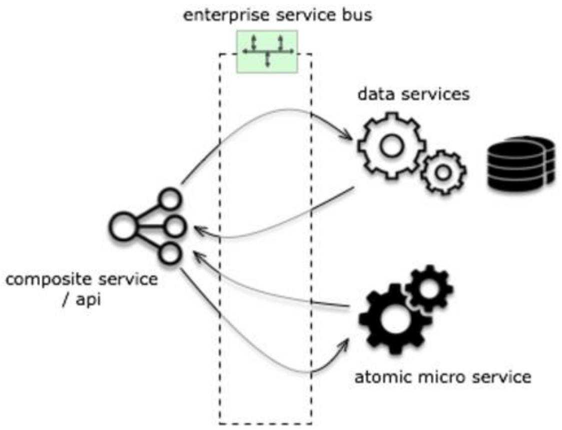

Figure 2 Short-running stateless orchestrations

4.4.2 WSO2 API Manager

Leveraging APIs in a collaborative way introduces new challenges in exercising control, establishing trust, security and regulation. As a result, proper API management is crucial.

WSO2 API Manager overcomes these challenges with a set of features for API creation, publication, lifecycle management, versioning, monetization, governance, security etc. using proven WSO2 products such as WSO2 Enterprise Service Bus, WSO2 Identity Server, and WSO2 Governance Registry. In addition, it is also powered by the WSO2 Data Analytics Server and is immediately ready for massively scalable deployments.

WSO2 API Manager is fully open source and is released under Apache Software License Version 2.0, one of the most business-friendly licenses available today. It provides Web interfaces for development teams to deploy and monitor APIs, and for consumers to subscribe to, discover and consume APIs through a user-friendly storefront. The API Manager also provides complete API governance and shares the same metadata repository as WSO2 Governance Registry. If your setup requires to govern more than APIs, we recommend you to use WSO2 API manager for API governance and WSO2 Governance Registry for the other artifacts. The default communication protocol of the Key Manager is Thrift.

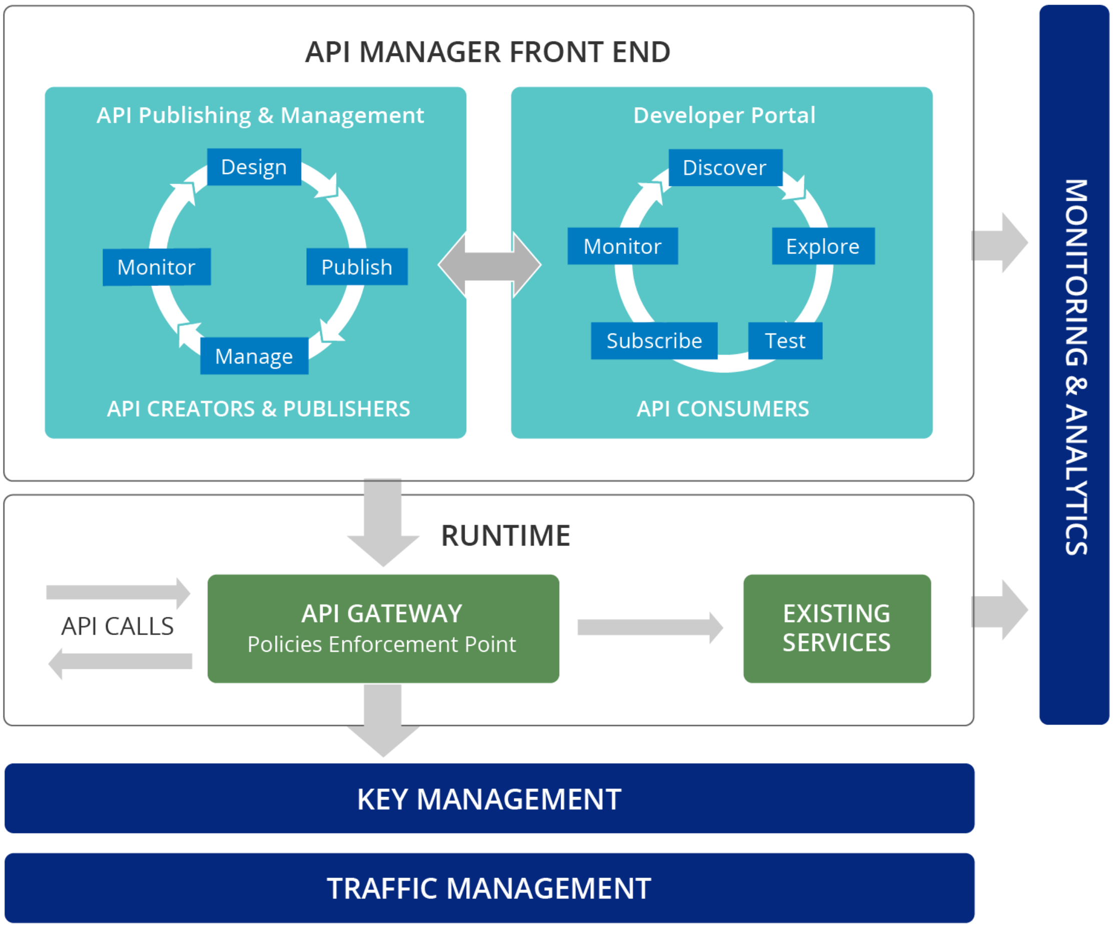

Figure 3 API Manager high-level components

The WSO2 API Manager is an on-going project with continuous improvements and enhancements introduced with each new release to address new business challenges and customer expectations. WSO2 invites users, developers and enthusiasts to get involved or get the assistance of our development teams at many different levels through online forums, mailing lists and support options.

WSO2 API Manager uses Apache Synapse as the mediation engine (which is used by WSO2 ESB too).

5 Conclusion

Most software development efforts today require some sort of application orchestration. Being able to quickly integrate with different applications, manipulate data and mediate message flow is hugely important as organizations are required to do more with less. Application orchestration is a better approach than point-to-point integration because the integration logic is decoupled from the applications themselves and instead is managed in a container that provides security, transaction management, reliability patterns and crucial monitoring and tracking capabilities.

The SMART-FI consortium agrees to examine WSO2 ESB products to cover objectives in task 5.1 but also plans to release second version of this document, mainly with more focus on ESB software stacks are going to act as orchestration and mediation layers in platform.

6 References

- SMART-FI Proposal 2016
- SMART-FI Project Consortium Agreement
- Project website: http://www.smart-fi.eu/
- SMART-FI: Exploiting Open IoT Data from Smart Cities in the Future Internet Society. Stefan Nastic, Javier Cubo, Malena Donato, Schahram Dustdar, Orjan Guthu. Mats Jonsson, Omer Ozdemir, Ernesto Pimentel, M. Serdar Yumlu, 2017. Springer Series, Internet of Things Technology, Communication and Computing
- The Open Group; SOA Reference Architecture http://www.opengroup.org/soa/source-book/soa_refarch/p1.htm
- Service Orchestration as Organization: Building Multi-Tenant Service By Malinda Kapuruge, Jun Han, Alan Colman
- WSO2 website http://wso2.com
- DATAVERSITY website http://www.dataversity.net
- MuleSoft website https://www.mulesoft.com