Grant Agreement No.: 723174
Call: H2020-ICT-2016-2017
Topic: ICT-38-2016 - MEXICO: Collaboration on ICT
Type of action: RIA

D5.4: SmartSDK Roadmap v3

Revision: v.1.0

Work package	WP 5
Task	Task 5.1
Due date	30/11/2017
Submission date	30/11/2017
Deliverable lead	MARTEL
Version	1.0
Authors	Hugo Estrada (INFOTEC), Blanca Vázquez (INFOTEC), Francisco Monsanto (UBIWHERE), Germán Molina (HOPU), Miguel González (ITESM), Alicia Martínez (CENIDET), Hugo Estrada (INFOTEC), Blanca Vázquez (INFOTEC), Tomas Aliaga (MARTEL) Antonio Macías (CICESE), Jesús Favela (CICESE), Enrique Succar (INAOE).
Reviewers	Federico M. Facca (MARTEL)

Abstract	This document provides an overview of the planned R&D achievements focusing on the period from December 2017 to May 2018
Keywords	Technical Roadmap, Agile, Epics, FIWARE

Document Revision History

Version	Date	Description of change	List of contributor(s)
V1.0	27/11/2017	Integration of contributions from all partners, checking spelling and typos.	Tomas Aliaga (MARTEL)
V0.3	20/11/2017	Updating roadmap of Application Scenarios	Jesus Favela (CICESE), Netzahualcoyotl Hernandez (CICESE), Miguel Palacios (INAOE), Enrique Succar (INAOE), German Molina (HOPU), Rachel B. (INFOTEC), Miguel Gonzalez, Nestor Velasco-Bermeo (ITESM) (CENIDET), Alicia Martínez
V0.2	03/11/2017	Updating roadmap of Platform Manager.	Daniele Pizzolli (FBK), Tomas Aliaga (MARTEL)
V0.1	31/10/2017	Table of Contents and initial version.	Tomas Aliaga (MARTEL)

Disclaimer

Copyright notice

Project co-funded by the European Commission in the H2020 Programme		
Nature of the deliverable*:	R	
Dissemination Level		
PU	Public, fully open, e.g. web	✓
CI	Classified, information as referred to in Commission Decision 2001/844/EC	
CO	Confidential to SmartSDK project and Commission Services	

EXECUTIVE SUMMARY

SmartSDK is the FIWARE's "cookbook" for developing smart applications in the Smart City, Smart Healthcare, and Smart Security domains. Concretely this means that SmartSDK refines, combines and develops new FIWARE Generic Enablers (GEs) and FIWARE Data Models into a set of well-codified and ready-to-use solutions. This is very important to facilitate the take up of FIWARE by new developers and its transition from proof-of-concept environments to production ones.

This document updates the second version of the Roadmap (D5.4) discussing covered developments during the period from June 2017 to November 2017, and providing an overview of planned Epics and Features in the period from December 2017 to May 2018. Full documentation of User Stories and tasks is kept in the Agile management tool (JIRA) for the interested readers to review upon request.

This document describes an overview of the components defined and developed in the following three domains:

- Smart city for monitoring pollution and traffic;
- Smart security for intelligent video surveillance;
- Smart healthcare for mobile sensing.

In addition, activities for the SmartSDK platform and its contribution to the FIWARE Community will be focused on:

- Curating the data models for inclusion in FIWARE catalogue;
- Launching the SmartSDK Platform as part of FIWARE Lab;
- Releasing guided tour of SmartSDK combining the different outcomes in a easy to follow guide and simple demo application;

The next release of this document (May 2018) will refine the components of the smart scenarios according to the evolution on the project and requirements in the applications.

TABLE OF CONTENTS

TABLE OF FIGURES

1. INTRODUCTION

This document updates the previous version of SmartSDK R&D roadmap v2. It highlights what was accomplished and what changed in the original work plan of the previous six months and brings an overview of the planned work and contributions to the FIWARE Open Source Community for the next roadmap period, covering from December 2017 to May 2018.

Just like the previously presented versions, this roadmap update will also be subject to eventual minor changes according to the different technical and non-technical aspects that influence the project: e.g. needs of the application scenarios, community relevance of the outcomes, available resources, etc.

Likewise, this version of the SmartSDK Roadmap is also based on the activities defined through the Agile R&D methodology . However, this document is not meant to be a detailed discussion of the all the identified epics, features and user stories. Those details are left in the issue tracking platform (JIRA) following FIWARE Community best practices. Their definition evolves at each sprint, particularly the user stories.

This deliverable focuses instead on providing a high-level picture of R&D plans of SmartSDK, focusing on priorities for the upcoming months, in order to:

- Guide the SmartSDK developers in the R&D activities by keeping in mind main outcomes expected by May 2018;
- Provide hints to the wider FIWARE Community of SmartSDK plans, so as to favour the establishment of collaborations.

1.1. Concept

SmartSDK is the FIWARE's "cookbook" for developing smart applications in the Smart City, Smart Healthcare, and Smart Security domains. SmartSDK is looking into applications developed so far within Europe and Mexico (using FIWARE or alternative Open Source technologies), analysing them and making a cookbook for developing applications in the Smart City, Smart Healthcare, and Smart Security domains.

The "cookbook" is based on: a set of architecture patterns (i.e. the basic cooking processes), a set of Generic Enablers (i.e. the basic ingredients) and a set of data models (i.e. the spices and flavours binding the ingredients through the cooking process).

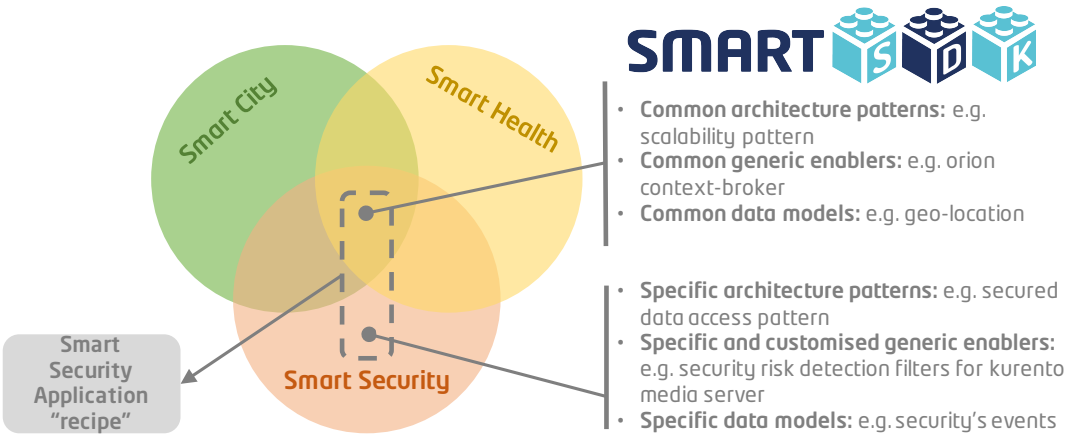

Figure 1. SmartSDK's cookbook concept.

Cookbooks often include a set of basic recipes that act as a common basis for different dishes. Similarly, SmartSDK will include as well basic patterns, Enablers and data models that can be used for

the development of data-intensive and IoT-based Smart applications and extend and complement them with specific patterns, Enablers and data models that are useful in the context of Smart City, Smart Healthcare, and Smart Security domains.

1.2. Overall architecture

SmartSDK will base its architecture on the FIWARE Reference Architecture and extend it to its wider scope. The core of the system supporting the deployment, management and monitoring of Smart applications is the SmartSDK Platform Manager.

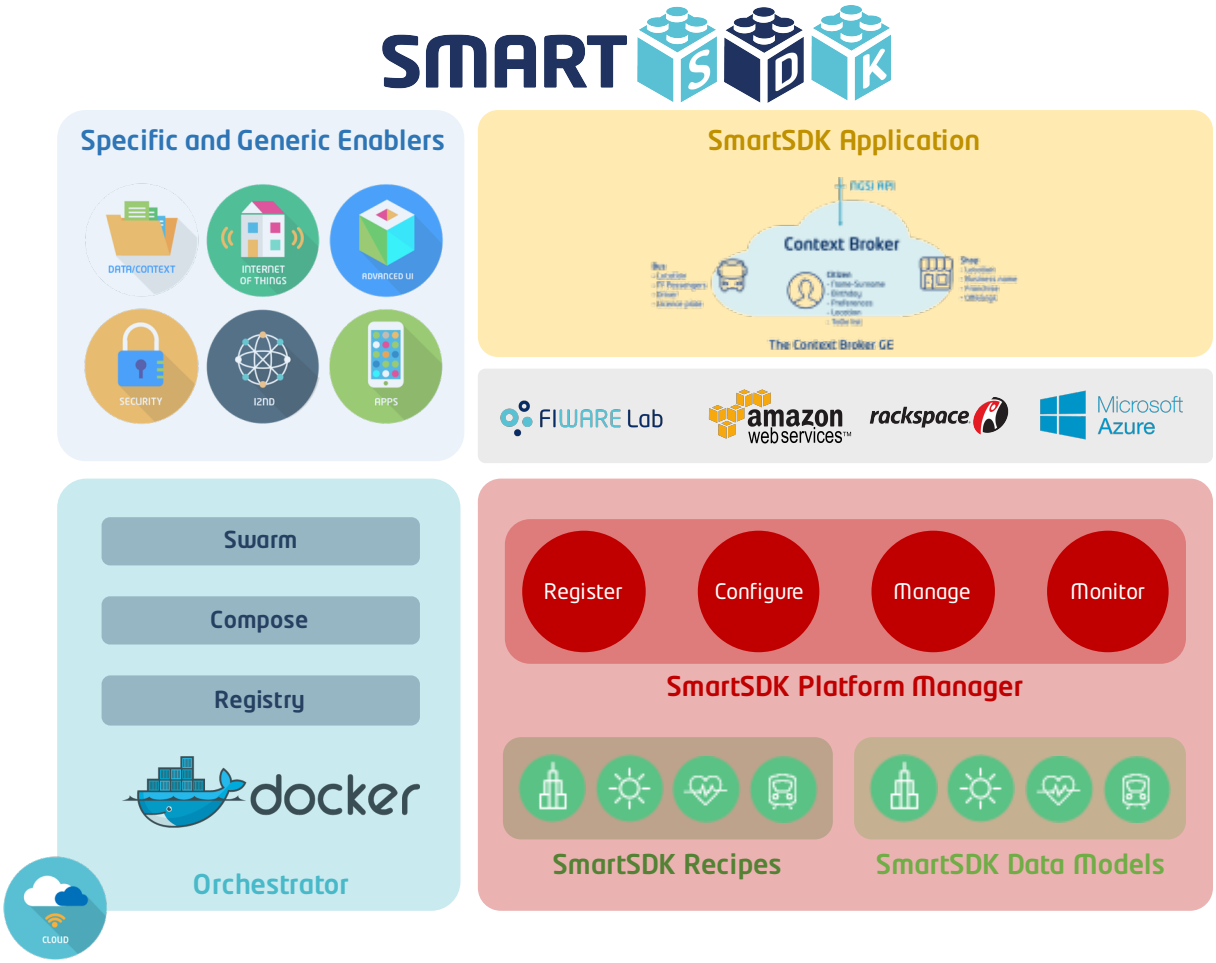

Figure 2. SmartSDK Architecture.

The SmartSDK Platform Manager will allow registering and maintaining the SmartSDK Recipes and SmartSDK Data Models:

- SmartSDK Recipes describe reference architectures for the application domains covered in the project and link them with Data Models. The recipes include relations among the Enablers and leverage on cloud patterns and typical FIWARE patterns for data-intensive and IoT-based applications.
- SmartSDK Data Models are NGSI-based information models defined to support the exchange of data in data-intensive and IoT-based applications in the domains covered in the project.

Recipes will be described using Docker Compose file descriptors or similar solutions (e.g. Kubernetes PODs descriptions) following the orientation that the FIWARE Cloud Hosting Chapter will take on containers orchestration.

Developers will be able to select a SmartSDK Recipe from a dashboard and provide configuration

parameters for the deployment and dynamic self-configuration of their Smart application.

The SmartSDK Platform Manager will deploy and manage configured Smart applications through the FIWARE Cloud Hosting Generic Enablers, for example, leveraging on the Docker GE.

Through the SmartSDK Platform Manager users will be able to control the status of their Smart applications, and re-configure them and eventually re-deploy them (e.g. on different hosting solution from FIWARE Lab, including commercial FIWARE providers or alternative commercial cloud providers).

Beyond the core of the system, SmartSDK Recipes and SmartSDK Data Models, SmartSDK will provide new Enablers and enhanced ones. These contributions will be derived also from the experience developing the trial applications, as presented in the following section.

1.3. Enhanced IoT and Data Management FIWARE Architecture

As mentioned above and detailed in the following sections, SmartSDK contributes to the FIWARE Community and number of new components dealing with IoT enablement and Data Management. In this section, we shortly present how this component integrates with the current FIWARE Reference Architecture for IoT and Data Management. Figure 3 shows the interaction between the different components (dark blue for the novel ones introduced by SmartSDK and light blue for the original FIWARE components).

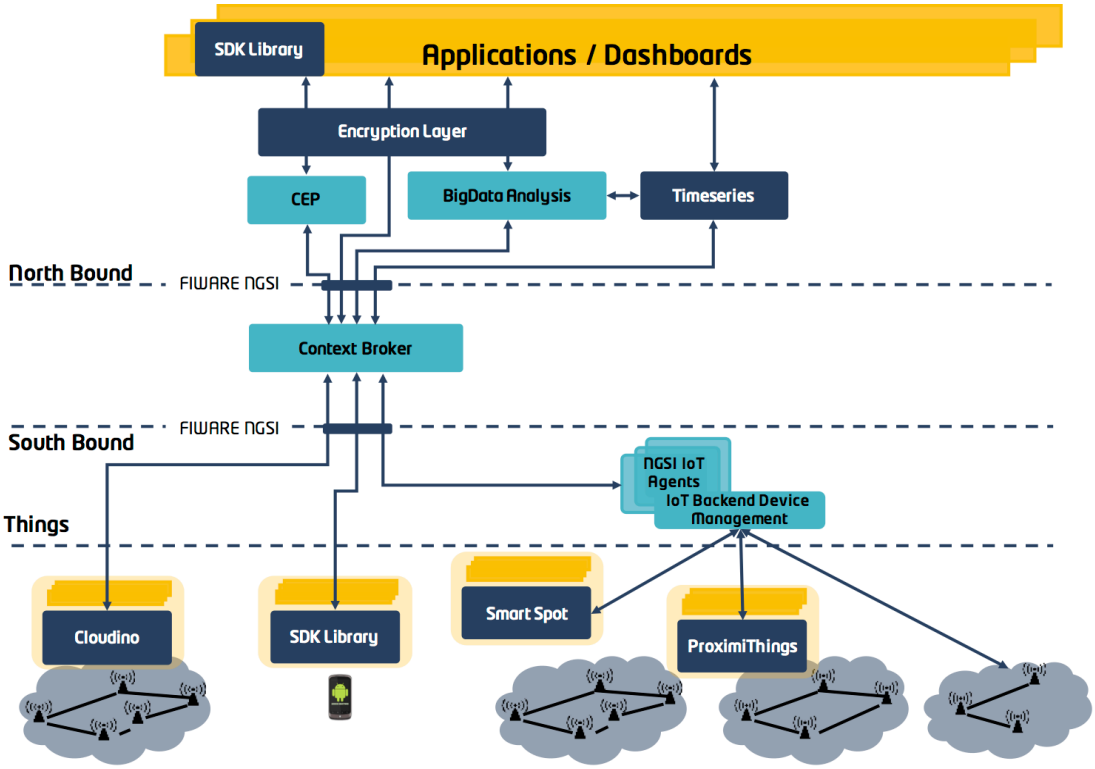

Figure 3: SmartSDK revised reference architecture for IoT and Data Intensive applications.

At the Internet of Things enablement layer, SmartSDK introduces:

- Cloudino a NGSI native hardware module that integrates with Arduino-like boards and allows to manage sensor data collection and publication to Context Broker, without the need of intermediary IoT agents.
- Smart Spot, a Physical Web solution integrated into FIWARE through the LwM2M IoT Agent.
- ProximiThings, a FIWARE-enabled framework for the incorporation of proxemic interaction

capabilities in IoT systems, integrated into FIWARE through the MQTT IoT Agent.

At the Data Management layer, SmartSDK introduces:

- A set of NGSI SDK Libraries that allows (mobile) applications to consume and produce NGSI data (and accordingly the SDK Library is presented in the figure in both roles).
- A Timeseries database implementing NGSIv2 APIs, that replaces STH Comet, providing native support for Timeseries database in FIWARE and native integration with timeseries data visualization tools such as Grafana.
- An Encryption Layer that support the fine-grained encryption and decryption of NGSI attributes, thus ensuring data protection and preservation for NGSI data sets.

2. APPLICATIONS

During the period June 2017 - November 2017, different software components of the three scenarios: Smart City, Smart Security and Smart Health which started in the previous period continued their development activities in parallel. This section presents the progress of those applications domains and their planned achievements for the period December 2017- May 2018.

2.1. Smart city

The application developed in the Smart City domain (called Green Route) will focus on supporting the citizens' mobility in high-polluted cities, like Mexico City, with the aim of improving the life quality and fostering environmental friendly behaviours by citizens. The end-user perspective is shortly summarised below.

The objective of the application is to help users find the best route to reach a destination, considering the user profile (such as health conditions) and the user preferences, such as transport type. The application proposes the ideal route for the user, avoiding streets with high levels of pollution, traffic jams or pollens, etc., allowing for instance, to obtain the preferred routes for people with respiratory diseases.

A particular version of Green Route was developed to pin damaged constructions in some cities in Mexico (i.e. Jojutla and Cuernavaca) during the earthquake held on September 19th 2017 in Mexico. The specific points in the city with damaged houses were geo-located and these locations were used to generate alerts that could be seen for Green Route users. The gathering points with provisions for affected people were also located in the map of Green Route. This version was used during the first days after the earthquake by Mexican public authorities and people living in these cities.

2.1.1. Covered Epics (June 2017 – November 2017)

To support the mentioned Smart City scenario, a set of components have been developed from Sprint 2.2, June 2017 to Sprint 4.1, November 2017. Those components are mentioned in each epic defined in the roadmap 1.0.

➔ **User profile management**

- A light application has been developed in order to send information related to Green Route. This application was developed using HTML, Bootstrap and CSS files, also the application is sent to the users through the Smart Spot.
- A module has been designed and developed to let the users register their events, including the location of the event, date, hour, severity level, and so on. Only for pollen alerts, the user can add additional information related with his health. In the asthma case, the additional information is omitted because is a sensitive alert that should be sent urgently.
- A module has been developed that allows the user to update their profile, for example: age, gender, disabilities, and so on. Thus, the user can keep their personal information up-to-date.
- A user can subscribe to interest groups. According with the groups in which the user is subscribed, the user could receive notifications of pollen, asthma attacks, weather, traffic, and so on.
- A user can generate an alert which can be visualized for all the users of Green Route.
- Only for users, in the section "Show all alerts", the health events are visualized per day. In order to avoid showing many information to the user. Only the Green Route

manager can visualize all the historical data of health events, traffic, pollution, and so on. This historical data will be useful to obtain statistics.

- Minor bugs were fixed in manager view. In the section for create groups, some bugs were fixed and that section works in a correct manner now.

➔ Vehicle profile management

- A schedule public transport module has been developed. This component allows to register the information of public transport of the Mexico City. For example, the departure time and arrival time, the frequency between of the departures, the days that the transport is available, and also the type of the public transport.
- We have developed a module focused in the public transport managers. In this module, we have added a map or the manager indicates on the map the location of the buses stations. In this way, each point will be saved in GTFS format. We are working in the first version of this module.

➔ **Maps management**

- The data from the fixed data acquisition unit was integrated in the map. On the map, the air quality data from Mexico City is visualized.
- Minor issues were fixed: data sources for some pollutants that seemed to be displaying incorrect values were validated.

➔ **Route management**

- A module has been developed to show the public buses from ITESM on map. The module process KML files to obtain the coordinates on the road.

➔ **Data context management**

- A time series visualization of Mexico City's Air Quality was implemented using the QuantumLeap component, CrateDB timeseries database and Grafana in order to persist and visualize the data sent to the Orion Context Broker. The Quantumleap component has been added to the Green Route.
- Green Route is subscribed to the Smart Spot and Cloudino each minute. The data obtained are: temperature, humidity, position and address. Green Route has a set of filters, and according to the values collected, the type of alert is defined, for example: heat alert, cold alert, and so on.
- We consider it is important to add information of air quality from other cities. Thus, we have subscribed Green Route to different endpoints, using the air quality data available in the FIWARE Lab. Currently, Green Route, shows the air quality alerts of France, India and China.
- We have built a smart plant. This plant sends data to the Orion Context Broker, related with her soil moisture, temperature and relative humidity. This data is visualized in the Green Route alerts.

➔ **IoT Management**

- LwM2M protocol minor bugs have been fixed: this enabler contained minor protocol bug that has been fixed through pull request to the official repository and accepted.
- Communications between the IoT agent and the Smart Spot has been improved, in order to save data and communication cost.

➔ **Data Models**

- A user data model has been defined with the fields requested for smart city scenario,

also an example was defined and shared on github. The model takes into account the following data: name, family name, gender, birth date, home address and work place.

➔ **Alerts management**

- The design of the alert application was made. This activity is related to the design of mockups for the alert application. Including the design of the view to select the alert type, the view to generate an alert about traffic jam, the view to generate an alert about accident, the view to generate an alert about a weather hazard, the view to generate an alert about the high levels of pollutants, the view to generate an alert about the high levels of pollen, the view to generate an alert about the asthma attacks.
- The alert application has been developed according with the mockups defined. The information of the alerts is sent according to the alert data model.
- A module was developed on Angular 2.0 in order to allow a user to select the alert type to be sent, and send any of the different alert types available, like accident alert, pollution alert, weather alert, pollen alert, asthma alert and traffic jam alert.
- The "observed event" catalog was defined, including the types of events incorporated into each type of alerts for the alert application.
- Defined and implemented the alerts database schema: built the schema that will allow developers to save a log about the alerts generated by users registered in the alert application

2.1.2. Planned Epics (December 2017 – May 2018)

To support the mentioned smart city scenario, a set of components need to be developed from December 2017 to May 2018.

➔ **User profile management**

- Configure health notifications: this component allows to user configure health notifications according his groups and the user could decide the periods to save his notifications.
- Add a security certificate to Green Route. This is required to guarantee the user account information to be safe. In addition, a security certificate avoids that some browsers emit alerts about insecure and not trusted web pages.
- Show the amount of pollution produced by each transport methods, according to the distances travelled.

➔ **Maps management**

- Public transport location: integrate the real time public transport location in the maps.
- Pollens maps: display information about pollen concentrations.
- Show weather conditions on a map. We should have a data source for weather conditions.
- Visualize heat maps about high levels of pollen, pollution and traffic congestion.
- Show buildings represented in 3D.
- Visualize traffic information in real time on a map.
- A graphical component will be developed to provide users information regarding the ETA (Estimated Time of Arrival) of a specific route within a route at a bus stop previously defined by the user. Such component will incentivize the usage of the app among users towards the planned tests with the ITESM bus routes. This component

will be an evolution of a component developed by CENIDET to estimate arriving times of public transport buses. We will reuse the algorithms to determine the estimated time for the next bus arrival.

- **Route management**
  - Congestion and traffic information: display real-time traffic information on a map.
  - Best routes: integrate traffic and pollution data in the best routes calculation.
  - The user could share their favourite routes, and the rate of the route.
  - The notifications from the Alerts Mobile Application will be shown on the map. The notifications could be about traffic jam, weather conditions, high level of pollutant, asthma attacks, pollen, car accidents, and so on.
  - Integrate the user profile data like health and vehicle with the route engine.
  - Obtain alternative bike paths.
  - A first version of the Bus position component will be developed for smartphones. By tracking the user's GPS sensor data, the route management engine will be able to not only track but also report the real location of the bus in real time. The developed app will be installed on the ITESM bus drivers' phones as a first trial of this component.
- **Data context management**
  - Public transport management: this component will allow to obtain the position from a public transport vehicle and this information could be available in the smart city application for all the users.
  - Traffic information: this component will allow to obtain real traffic data from public transport. This information could be shown for all the users.
  - Obtain data from pollen using the concept human-as-sensor.
- **Data models**
  - Review the traffic flow model, according to the model used by the traffic source.
- **IoT Management**
  - Develop guidelines to connect sensors.
  - Develop guidelines for using data models.
  - Develop guidelines to send data from IoT devices to the Orion Context Broker.
  - Develop guidelines for using the IoT agent.
- **Alerts management**
  - Define and establish connections with external information sources.
  - Define and implement the alerts database schema, including the event observed schema.
  - Establish the connection between the Orion Context Broker and the alerts application.
  - Establish the connection between the user database and the alerts application.
- **Container**
  - Deploy the Green Route application on a Docker container.
- **Trials**

- Deploy a trial with the INFOTEC community

2.1.3. Key used FIWARE Enablers

- Identity manager - KeyRock
- Orion Context Broker
- Backend Device Management – IDAS
- Cygnus
- Cosmos Big-Data Analytics

2.1.4. Contributions to the FIWARE Community

The main contributions of the FIWARE Community from these activities are:

- Creation of a new data model for alerts generated by a user in a given location. This model could be useful to send alerts to the Orion Context Broker related many scenarios: traffic, pollution, suspicious activities, foods, and others. A web service could be subscribed to OCB to obtain data generated by users and generated notifications or trigger other actions. The alert data model can be consulted through this link:

  https://github.com/smartsdk/dataModels/tree/master/Alert

- Creation of new data models for trips. These models include all properties which can be used to find the user's favourite public transport trip and all data related to it. The data models are based on the General Transit Feed Specification (GTFS) that defines a common format for public transportation schedules and associated geographic information.
- Publication of real air quality data from Mexico City through Orion Context Broker. The data contain the measures of pollutants of carbon monoxide, ozone, nitrogen dioxide, sulfur dioxide and particles pollution. The measures are updated each hour. Any SMEs, start-ups and entrepreneurs could obtain the air quality data to develop highly innovative services and applications.
- SmartSDK Presentations in FIWARE Tech Summit in Málaga (November 2017) related to SmartSpot.

2.2. Smart Security

The Smart Security application will focus on detecting and analyzing security risk such as theft, access controls, people detection, fights, crowd analysis, etc., through the combination of video cameras and mobile sensors, in both, indoor and outdoor scenarios, for instance, parking lots and buildings. The main goal is providing support to the end-user (i.e. security guard) in order to prevent risky situations and consequently improve the citizens' quality of life.

2.2.1. Covered Epics (June 2017 - November 2017)

To support the mentioned smart security scenario, a set of components have been developed[1] from Sprint 2.2, June 2017 to Sprint 4.1, November 2017. Components developed are:

- **Data management**

[1] It is worth mentioning that the Smart Security team is undergoing a re-planning process. Mainly, new integration, testing and "dockerisation" tasks have been added. Because of this, the dates of some tasks have changed.

- Database: Database Design and Implementation to store video alerts and smartphone data, users, etc. generated by the system.
- Storage based on movement. Design and implementation of a storage module based on movement detection. This module will help the user retrieve the relevant videos to consult the events detected. The module is an extension of the basic storage capabilities of the system. The main objective is to optimize the storage space.
- Online Smartphone Query. The module will let us query if a person registered with a smartphone is in one of the registered areas of interest. The smartphone GPS is used to verify the user position.
- Review of the System architecture and Data models. The main objective was to evaluate the first version of the architecture and proposed data models. Currently, we have a more clear and specific version for the security scenario.
- Integration: Several tests to evaluate the communication between the modules, the data transmission to the Context Broker as well as the visualization of the algorithms results.
- Interface: A GUI based on Web2Py with more visual functionalities: camera management, markers and basic user management. An historical box is activated catching the events produced by the movement detector algorithm and the PersonInArea event detected by the smartphone functionality.
- A more stable version is available in smartsdksecuirty.com.mx. Certificates have been installed.

➔ **Pattern recognition**

- Design and implementation of a component to determine when a registered smartphone is inside a specific area.
- Development of a software module, based on the data from smartphone, capable of detecting vehicles traveling at high speeds.
- Development of a software module (Kurento filter) capable of detecting vehicles on a video stream. The objective of this implementation was to extend the people detection algorithm previously developed in order to be used as a complementary vehicle detector into the pipeline of the system.
- Development of a software module (Kurento filter) capable of detecting persons in a video stream in outdoor environments. Extension of the indoor people detector algorithm for outdoor environments.
- Video Database of faces in a video stream in indoor environments. We recorded a small video database in order to test our face recognition algorithm for verification.
- Design of a software module to make online queries about whether a registered smartphone is in the campus or not.
- Analysis of the camera views more convenient for the detection and recognition capabilities of the system.

2.2.2. Planned Epics (December 2017 - May 2018)

The following components will be developed in the next stages of the project:

➔ **Data management**

- Phone Log Query. The PhoneLogQuery functionality will allow to determine if a

person was inside a specific area at a certain moment. PhoneLogQuery is a functionality that complements the OnlinePhoneQuery functionality. In addition, the OnlinePhoneQuery and Phone functionalities will allow to identify the place and the moment in which an event detected by the smartphone has occurred.

- Database implementation (update). The update and implementation of the new version of the database is directly related to the revision and the adjustments that are being made to the data models and the architecture used in the SmartSDKSecurity scenario. For this new version, the possibility of using a hybrid database that exploits the characteristics of relational and non-relational databases is being studied. The database engines considered are MongoDB and CrateDB since these database engines are those used by the Orion Context Broker and Quantum Leap respectively.
- Encryption layer service that will improve the privacy on the front-end side, will consider, in the first version, the encryption of the users' personal information, as well as location. Concerning the backend side, the use of the encryption layer is not considered, as far as the solution will be containerised and running on a local server.
- Alerts module. It will be integrated and improved, following the recommendations of the FIWARE Foundation and the actual version of the Alerts Data model, already available in the SmartSDK Github account. The evolution will take care in maintaining the generalization of the Alerts Data Model, and the final functionalities of the Smart Security Pattern recognition algorithms.
- GUI update. As more events are delivered, more visual options will be activated in the different views of the system.
- Integration and Dockerisation. We are planning incremental integration and "dockerisation" processes.

➔ **Pattern recognition**

- Processing module. New or improved of algorithms will be developed to detect events based on camera and smartphone data. The detection includes:
  - Unauthorized car speeds. Based on smartphone data (accelerometer) the module is intended to detect if the car speed of a smartphone registered user surpasses the maximum permitted speed.
  - Improved people detection in outdoor and indoor environments. Previous version of people detection in outdoor and indoor environments will be improved in order to obtain a better performance.
  - Sudden car stops. The module is intended to detect sudden car stops based on the information gathered by the smartphone of registered users.
  - People Fighting. Based on computer vision algorithms the module is intended to detect when people are fighting in the scene of one of the camera system.
  - People Running. As a basic activity and based on computer vision algorithms the module is intended to detect when a person runs in front of one of the cameras of the system.
  - Bicycle/Motorcycle. This sub-module is based on a classification algorithm to detect two wheel vehicles as bicycle and motorcycle.
  - Crowd formation. As a signal of risk situation, the module is intended to detect when several persons meet in one point.
  - Cars Wrongly Parked. The objective of the module is to detect if a car is wrongly parked based on basic information about the parking area. Algorithms to detect this

kind of events will be designed and implemented.

- Wrong way driving. This sub-module is intended to detect if a car does not follow the correct way.
- Forgotten Large Objects. One of the main events reported to security guards is related to forgotten objects. This module is intended to detect this type of events.
- Long Time Staying. From the security point of view, one suspicious activity is when people stay for a long time in some specific area. This sub-module will be able to detect this type of activities.
- Car collisions. This sub-module is intended to detect when two cars make physical contact.

- Video Repositories. In order to be able to test the new version of the algorithms we will create several video repositories with several camera views from the two study cases: INAOE and ITESM.

2.2.3. Key used FIWARE Enablers

- Orion Context Broker
- Kurento
- QuantumLeap

2.2.4. Contributions to the FIWARE Community

The main contributions to the FIWARE Community are:

- An architecture for video surveillance systems based on FIWARE which could be used as a guide to develop this kind of systems and exploit the Generic Enablers given by FIWARE.
- Definition of a new data model for video surveillance. This model would be useful to send information (i.e. alerts) to the Orion Context Broker related with security.
- Visualization and notification of events in areas of interest to detect vandalism or even to prevent incidents that may trigger tragic events. Main sources of information are smartphones and video cameras.

2.2.5. Blocking Issues

Currently, we are facing a problem with the implementation than can impact the final version of the system. Although we have a stable version of the surveillance system running on a local server, it has not been possible to run the system on a FIWARE node, some issues with the video streaming have inhibited the correct functionality on a FIWARE node. Nonetheless, we are currently working on this issue.

2.3. Smart Health

Sensors in mobile, wearable, and environmental devices are becoming ubiquitous, allowing the inference of activities and behaviours associated with health by facilitating the collection of daily-life data. The Smart Health task focuses on creating a specific set of components to support the development of healthcare mobile applications based on analysing data gathered from sensors in mobile and wearable devices. In particularly, we will be presenting an application developed in the context of mobility assessment to address the harmonization and management of mobile sensing data using and extending the FIWARE platform.

2.3.1. Covered Epics (June 2017 - November 2017)

A set of components have been under development from Sprint 2.2 (June 2017) and will continue until Sprint 4.1 (November 2017). Next, we present a brief description of the components developed and planned for this period:

- First data analyses to cope with the variable of interest related to fall risk.
- Subject study. A first subject study was conducted to test both: FIWARE's integrated component and application development.
- Data integration from volatile data and persistent services such as Cosmos GE.
- The mechanisms for analysing data were integrated on the application architecture, thus, results can be retrieved from the client-side.
- Variants analysis and implications for the integration of wearable devices integration.
- First advances for the development of a project's proof of concept for empirically testing the flexibility of the data-model over wearable sensing devices such as a smartwatch.

2.3.2.Planned Epics (December 2017 - May 2018)

To support the mentioned Smart Health scenario, a set of components need to be developed from December 2017 to May 2018 within the context of two applications (Risk of fall and Rehabilitation).

- Risk of fall App
  - Data integration
    - Integration of heart rate sensor data collected from wearable devices within the data-model.
  - Trigger action
    - Actuator setup to performs an action based on gathered heart rate data. Information is to be addressed by the notification and condition services provided by FIWARE.
  - Notifications
    - Reception of a reminder and the respective interface functions to address the actions towards removing collected data.
  - Data deletion
    - Ensure mechanism to guarantee that data is deleted after a specific period of time and the person in change's authorization.

- Rehabilitation App
  - Wearable devices integration.
    - Data model testing by integrating the use of wearable devices such as smartwatches. As established on respective architecture, data is to be send towards the Orion Context Broker in first instance.
  - Monitorization.
    - Data would be monitorized and accessed as record reference. Data is to be retrieved directly from the QuantumLeap component.
  - Synchronization.

- Devices (i.e., smartphone and smartwatch) used during the collection of data are to be synchronized to facilitate port-processing analysis.

- Cloud.
  - Collected data from wearable devices should be sheltered on the Cloud for further analysis.
- Monitorization.
  - Constant monitorization and feedback for the patient to take actions based on patient's performance detected by wearable devices.
- Statistical analysis of patient performance will take place to further be presented to the health specialist

2.3.3. Key used FIWARE Enablers

- Cygnus
- QuantumLeap
- Orion Context Broker

2.3.4. Contributions to the FIWARE Community

The contributions focus on three components:

- A data model to address sensor data in the context of healthcare, as well as the extension of current FIWARE's models towards the collection of sensor data for mobile devices such as smartphones and smart-watches.
- Proof of concept of two software implementations, build upon FIWARE components such as Cygnus, Orion Context Broker, and QuantumLeap.
  - An application to collect and monitor sensor data from mobile devices (i.e., smartphones) focuses on measuring the risk of fall in elderly population.
  - An application to monitor patients under rehabilitation treatment, based on wearable devices (i.e., smart-watches) and statistical analysis to be presented to the end-user

3. PLATFORM

With the aim of contributing to FIWARE OS Community and supporting the development of the scenarios depicted in the previous section, SmartSDK will work on the development of new functionalities for existing GEs and novel ones. The functionalities are grouped around three main groups:

- The SmartSDK Platform: the glue that help the provisioning and automation of smart applications.
- The Data Management: the enhanced set of functionalities that increase the capacity of FIWARE Data Management platform.
- The IoT Management: the enhanced set of functionalities that increase the capacity of FIWARE IoT platform.

3.1. Platform

3.1.1. Covered Epics (June 2017 - November 2017)

- Generic Enablers Recipes
  - Development of recipes (in the form of shell scripts and Docker Compose version 3 files) to enable the applicability of the Scalability, High Availability and Co-location patterns to the deployment of the CKAN and IDAS Generic Enablers. The recipe for the CKAN components has been postponed and instead a new one for QuantumLeap was anticipated, considering this component was starting to get used by the application scenarios. Regarding IDAS, the repository included recipes for the three most used versions of the IoT Agents; namely "json", "ul" and "lwm2m".
- User Interface for Docker Compose v3
  - The current user interface does not support the advanced features offered by Docker Compose format v3. It only supports a basic view of the running Docker instances. The Portainer project[2] should support this in the near future.
- User management integrated with FIWARE Lab for SmartSDK platform
  - We plan to automate the build and the documentation of the custom Docker image used for the FIWARE Lab Oauth integration.
- Overlay network
  - To overcome the limitation of rancher networking we plan to explore and document other VPN alternatives: for example, Freelan, VDE2 and GVPE.
- Persistent storage configuration
  - There are also few discrepancies regarding the support of shared storage in Docker Compose v2 and v3 to be further investigated. The optional persistent storage configuration can be enhanced, for example by using a separate storage network.
- SmartSDK Platform Manager in HA
  - Good production practices for cloud distributed application encourage the setup of highly available services. Currently our testing environment do not have the resources to offer HA, but we plan to overcome this issue in the near future.

[2] *http://portainer.io/*

3.1.2. Planned Epics (December 2017 – May 2018)

During the last few months, some interesting shifts took place in the software landscape commonly known as "container orchestration managers". Kubernetes had a boost in interest, contribution and deployment, leaving Docker Swarm (SmartSDK main orchestration choice) far behind in community support. There are still use cases where Docker Swarm is a reasonable choice, and we think that the scope of this project still falls into that. Docker Swarm is still less resource intensive, easier to setup and is much easier to grasp for newcomers compared to Kubernetes, which is more enterprise-oriented.

Docker Inc. the company that drives most of the work on Docker Swarm, realizing the industry swift towards Kubernetes, has decided to focus on a slow integration of their tooling with the Kubernetes orchestrator. Right now, with the enterprise edition of Docker, it is possible (at least for some operating systems) to deploy containers in Kubernetes using Docker Swarm recipes. The status and the plan for the integration for the community edition is still unclear, as it is unclear for how long the current version of Docker and Docker Swarm will be maintained. However, there are other tools that may help the transition from Docker Swarm to Kubernetes, for example kompose. Probably kompose will not solve all of our use cases (for example some recipes need to talk directly with the API of Docker Swarm), but for sure it could help in the transition.

Rancher Inc., the company that still drives most of the development of Rancher, the software we use as the base of our Platform Manager, realized the same and in a similar line starved the support for Docker Swarm in Rancher focusing in a tight integration with Kubernetes.

Within the SmartSDK project, we followed closely those events, because of the direct impact on our project. In mid-October 2017, right after the Dockercon Europe, where the rumors were confirmed, we had a cross project discussion and meeting with FI-NEXT and with the FIWARE Lab in order to redefine our strategy.

We decided to initially focus on the development of internal tools or glue tools to allow the use of Docker Swarm with our current set of recipes. Staring from command line tools and then, if feasible, adding some graphical user interface based on a website like platform. In parallel, we plan to support the latest version of rancher that supports Docker Swarm and to document the transition steps.

Rebasing all our previous work on Kubernetes in a single step is not feasible for our project, but we will follow closely the evolution of the tools that will ease the migration to Kubernetes in order to give a comprehensive set of choice to our users. It is not unlikely that such tools will be developed and maintained by the community, offering an alternative to the Docker enterprise edition that we may not be able to use for example for reasons related to licensing term or licensing fees.

Given this lengthy introduction we plan to work on the following epics.

- ➔ Support Docker and Rancher working with Docker Swarm
  - Support the running of the latest version of Docker and Rancher that supports Docker compose v3.
- ➔ Custom tools and glue tools to replace Rancher
  - Development of custom tools and/or integration of existing tools like for example docker-machine in order to replace Rancher for deploying the Docker Swarm hosts.
  - Development of custom web interface and/or integration of existing web interfaces that allows the deployment of Docker Swarm recipes v3.
  - Take care of preserving existing work on the application catalog, networking configuration (e.g. VPN, overlay network, MTU workaround) and if possible FIWARE Lab integration.
- ➔ Generic Enablers Recipes

- Add recipes for enabling the reference architecture patterns for a deployment of CKAN generic enabler.
- Add recipes or update existing ones to facilitate usage of the security-enabling Generic Enablers for Authentication and Authorization such as PEP and/or AuthZForce, whichever happens to be more relevant during the next months.
- Harmonize (as in checks of consistency) the customization of recipes through environment variables and the documentation of all of them.

➔ Investigate and document transition to Kubernetes

- In the long term, Kubernetes will be the default choice for orchestrating container, so we will try to ease and document the transition to Kubernetes.
- Probably we will not able to migrate all the recipes, but we will document the conversion steps for some of them.

3.1.3. Contributions to the FIWARE Community

➔ A container management solution integrated into FIWARE Lab.

➔ A set of recipes allowing production grade deployment of FIWARE GE using Docker available at http://github.com/smartsdk/smartsdk-recipes.

➔ Presentations in 2nd FIWARE Summit in Utrecth (May 2017).

➔ SmartSDK Presentations in FIWARE Tech Summit in Málaga (November 2017): “Docker Swarm Secrets for creating great FIWARE platforms” by Federico Facca (MARTEL).

3.2. Data management

The Data Management activities will focus on contributions that will enrich the FIWARE Data Management chapter and improve interoperability of NGSI (with respect to devices and open data models). Activities have been defined and prioritized taking into account SmartSDK applications’ requirements.

3.2.1. Covered Epics (June 2017 - November 2017)

For each of the main Data Management tasks, we here contrast the previously planned tasks with the actual outcomes until the time of writing.

➔ Time Series for NGSI

- As originally planned, we developed and validated the custom translators from NGSI to TSDB specifics such as CrateDB, RethinkDB and InfluxDB.
- The CrateDB translator was further developed and enriched to support different data types (dynamically incoming data) and started supporting some FIWARE official Data Models.
- We developed the first version of QuantumLeap (QL), demoing a complete end-to-end solution persisting historical NGSI data coming from southbound IoT Layer and supporting visualization.
- As planned, support for visualization of data using Grafana was included and showcased in the first project review in Brussels on July 2017.

➔ NGSI Encryption Layer

- As previously planned, a manuscript is undergoing the final revisions by the authors.

A survey will describe the current and latest approaches focusing on data encryption for cloud based applications covering aspects such as performance, attribute encryption algorithms mainly. The survey is planned to be submitted to a Q1 or Q2 Journal by the last week of November

- A first version of the NGSI Encryption Layer (defined in deliverable 3.2 as NGSI attribute encryption NGSI-AE.) was created. The application provides the users the possibility to select from all the available FIWARE Data models the attributes that will be encrypted before sending the data to the OCB. The application consists of an RESTful API that allows developers automate the process without using a GUI as for the former application. Tests were conducted to verify the integrity of the encrypted content as so the resulting JSON is compliant with the data model's standards.

➔ SDK Library for NGSI

- Develop a first version of the JavaScripy Library based on current Android mobile application.
- The first NGSI library version for JavaScript was developed based on the NGSI library for Android previously developed. The NGSI library for JavaScript is being used in the Online Phone Query functionality to facilitate the transfer of data between the Online Phone Query functionality and the Context Broker. This library is being used in the smart security scenario to provide contexts information to the Orion Context Broker.

3.2.2. Planned Epics (December 2017 - May 2018)

➔ Time Series for NGSI

- NGSI API
  - Add simple query endpoints like those of Comet STH to retrieve historical data
  - Add "Configuration" endpoints to customize QL setup.
  - Optional: allow direct inserts (by-passing Orion when it is not needed).
- Advanced Topics
  - Support retention policy
  - Allow per-table customization of number of replicas and shards / partitions.
- Multitenancy
  - To support different tenants sharing the same QL deployment with proper isolation of usage and privacy.
- Documentation & refactoring
  - Documentation must be reviewed and extended with the new features.
  - The repository where QuantumLeap is being developed must be reorganised as explained in https://github.com/smartsdk/ngsi-timeseries-api/issues/22.

➔ NGSI Encryption Layer

- Integrate a token generation method that will connect to the user's ID and will specify which attributes will be kept private (encrypted) while stored at the OCB.
- Analyze the viability of a whitelisting approach to provide developers a simpler and practical approach to consume user's data.
- Test runs will be conducted with other scenarios to analyze the application's

capabilities regarding petitions and processing times.

- Documentation
  - Documentation must be completed and shared according to the project's guidelines.
  - Simple How-to guides will be created.

➔ SDK Library for NGSI

- The current version of the NGSI library for JavaScript allows the user to create, browse, update and delete entities. The following activities are:
  - Review and analyze the current structure of the NGSI library for JavaScript
  - Develop and implement the functionalities that allow managing subscriptions and batch operations.
  - create and apply the test cases to check the functionality of the library
- Documentation
  - The existing library documentation should be supplemented and examples of how to use it should be increased

3.2.3. Contributions to the FIWARE Community

The main contributions of the FIWARE Community from these activities are:

➔ A native NGSIv2 Timeseries database, including support for data visualization.

➔ A facility to support data encryption on top of NGSI providers.

➔ A SDK to support mobile devices interaction with NGSIv2 providers and consumers.

➔ SmartSDK Presentations in FIWARE Tech Summit in Málaga (November 2017): "Quantumleap: a FIWARE Time Series DB" by Tomas Aliaga (MARTEL).

3.3. IoT Management

The IoT Management activities will focus on contributions that will enrich the FIWARE-ready IoT ecosystem, in particular with the aim of improving Cloudino framework. Activities have been defined and prioritised taking into account SmartSDK applications' requirements.

3.3.1. Covered Epics (June 2017 - November 2017)

To support the SmartSDK platform, a set of components have been developed from Sprint 2.1, June 2017 to Sprint 4.1, November 2017. Those components are mentioned in each epic defined in the roadmap 1.0.

➔ CLOUDINO

- Cloudino Cloud Service. We have been developing a specific cloud for Cloudino (Cloudino.io Cloud Service) to enable developers to connect with different open/private clouds. In this context, the Cloudino Cloud is an intermediate among the data capture by the electronic device in the specific cloud platforms.

➔ SMART SPOT

- Sensor integration. For this project will be integrated several sensors, like temperature, humidity, air quality and accelerometer, some of these sensors have to be defined in the OMA protocol and conveniently parsed to NGSI entities.
- GPS Integration in order to track devices installed over vehicles and sensors

measurements.

- Deploy the required Generic Enablers to connect Smart Spot devices with the Orion Context Broker of Green Route.
- Smart Spot Gas sensors calibration to greenroute use case.

3.3.2. Planned Epics (December 2017 - May 2018)

➔ CLOUDINO

- Create a version of Cloudino Cloud as a Generic Enabler. The current version of Cloudino needs to be properly packaged as Docker container to be aligned with Generic Enabler best practices

➔ SMART SPOT

- Device connection configuration over GPRS and firmware upgrade.
- Update to lwm2m IoT Agent to the latest version in order to adopt the NGSI FIWARE Data Models.
- Device full integration in green route use cases.

3.3.3. Contributions to the FIWARE Community

➔ Smart Spot Data Model. For the correct integration of the Smart Spot in the FIWARE ecosystem, we have been working in a NGSI data models approved and certificated by FIWARE. The work is available in https://github.com/fiware/dataModels/tree/master/PointOfInteraction.

➔ IoT Agent Fixes. For the correct behavior of the device, some corrections were made in this FIWARE enabler, this correction were about the way of managing the LwM2M protocol. Corrections were merged in the master branch by the Generic Enabler owner.

➔ SmartSDK Presentations in FIWARE Tech Summit in Málaga (November 2017) related to SmartSpot.

4. CONCLUSIONS

This document presents a new version of the SmartSDK R&D roadmap covering updates from the previous period, and focusing on the period from December 2017 to May 2018. It also aims to highlight the planned contributions to the FIWARE Open Source Community. The next release of this document (May 2018) will cover the final roadmap steps until the end of the project.

In this deliverable, we quickly overviewed the following items for each of the application scenarios covered in Work Package 2:

- The epics and features covered in the R&D phase to support such a use-case scenario.
- The usage of FIWARE Enablers.
- The contributions to the FIWARE community.
- The priorities until May 2018.

As regards the Platform (Work Package 3), the deliverable highlights:

- The epics and features covered in the R&D phase to implement new Enablers and simplify usage of existing ones.
- The planned contributions to the FIWARE community.
- The priorities until May 2018.

REFERENCES

[1] SmartSDK Consortium. Description of Action. July 2016.

[2] SmartSDK Consortium. D5.1: Quality Assurance Guidelines. November 2016.

[3] Dean Leffingwell. 2011. Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise (1st ed.). Addison-Wesley Professional

APPENDIX A SMARTSDK AGILE METHODOLOGY

To drive the creation of tangible project outcomes, SmartSDK adopts an agile iterative methodology. The methodology is driven by the applications in the Smart City, Smart Healthcare and Smart Security domains and the applications' validation into trials.

The analysis of the state-of-the-art and of the applications steers the Research and Development of planned outcomes (SmartSDK Platform and the Applications to be trialled). The R&D outcomes are then validated through the trialling of the Applications with real-users. Validated outcomes are then adopted into Technology transfer activities (i.e. trainings and contributions to standardization bodies). Feedbacks and outputs of the process feed the next cycle of activities.

The whole project foresees 2 major cycles, each of which will include two minor iterations. The first set of outcomes will be trialled in Mexico at the end of the first cycle (M12), while second stage trials will leverage on the European FIWARE ecosystem (M24).

The methodology will be implemented using the Lean Agile approach by Leffingwell [3] adopted by FIWARE Agile Development Methodology[3]; this will allow a continuous monitoring of project progress in term of outcomes and exploitable results. The process is depicted in Figure 4.

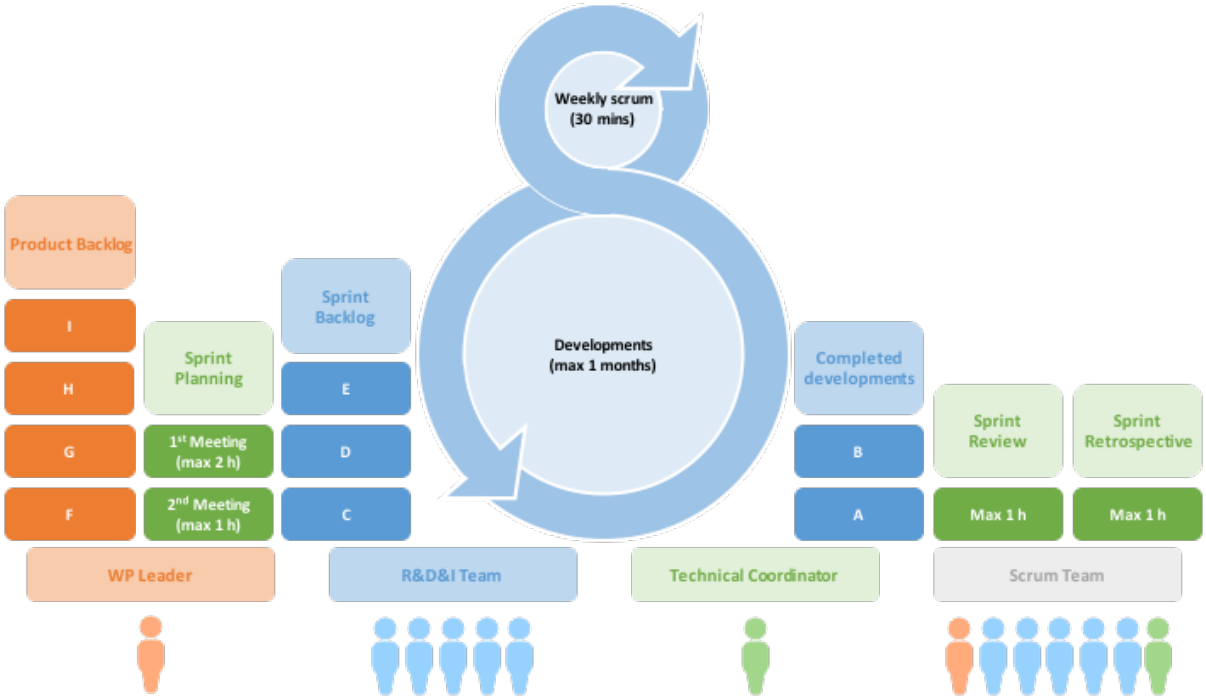

Figure 4. SmartSDK Agile Development Process.

The Agile meeting are planned as follows:

- Sprint planning 1st Monday of the month 16:30 – 17:30
- If needed also 1st Tuesday of the month 16:30 – 17:30
- Weekly scrum all Mondays 17:00-17:30
- Sprint review: last Thursday of the Month 16:30 – 17:30

[3] http://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE_Agile_Development_Methodology

The first Sprint is only for design purposes and last longer (from the beginning of September to the end of October), then the Sprint are monthly. Every third Sprint includes a Release.

A.1 The Process

1. The application scenarios' requirements will be described in a plain-text document. From this description, Application owners will derive main Epics.
2. Epics will be assigned to a Component (an Application, a Data Model, or an Enabler; part of WP2 or WP3).
3. Component Owners will define a set of Features for each Epic.
4. Features will be decomposed in User Stories.
5. User stories will be then planned for specific Sprints during the Sprint Planning, and validated within Sprint Reviews that will include demonstration of achieved results.
6. Each User Story should be refined and detailed before being assigned to a Sprint.

The methodology will be implemented leveraging on the FIWARE JIRA management tool.

A.1.1 Roles and responsibilities

- Product Owner
  - Who: All the task leaders
  - What:
    - Define user stories
    - Define features
    - Define epics
    - Prioritize the backlog
- Developer
  - Who: All the people active in the "Development" / "Delivery"
  - What:
    - Refine and Implement user stories
    - Document progresses
- Scrum Master
  - Who: Federico/Tomas (Martel)
  - What:
    - Help to prepare the sprints planning and sprints review sessions
    - Check status of activities
    - Facilitate collaboration
    - Remove obstacles
- Scrum Team
  - Who: Developers, Product Owners and Scrum Master
  - What:

- Assign/Re-Assign stories to components (in collaboration with other product owners)
- Sprint planning
- Sprint review