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

# D5.7: SmartSDK Roadmap v4

Revision: v.1.0

| | |
|---|---|
| Work package | WP5 |
| Task | Task 5.1 |
| Due date | May 31st, 2018 |
| Submission date | July 26th, 2018 |
| Deliverable lead | INFOTEC |
| Version | 1.0 |
| Authors | Hugo Estrada (INFOTEC), Yolanda Baca (INFOTEC), Samuel Jiménez (INFOTEC), Francisco Monsanto (UBIWHERE), Germán Molina (HOPU), Miguel González (ITESM), Alicia Martínez (CENIDET), Tomás Aliaga (MARTEL), Antonio Macías (CICESE), Jesús Favela (CICESE), L. Enrique Sucar (INAOE). |
| Reviewers | Federico Facca (MARTEL) |

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

**Document Revision History**

| Version | Date | Description of change | List of contributor(s) |
|---|---|---|---|
| V0.1 | 09/07/2018 | Table of Contents and initial version from D5.6. | Tomas Aliaga (MARTEL) |
| V0.2 | 10/07/2018 | Update sections covering Platform tasks | Daniele Pizzolli (FBK) |
| V0.3 | 17/07/2018 | Integrate updates from CICESE and CENIDET. | Blanca Haidee Onofre (CENIDET), Netzahualcóyotl Hernández (CICESE), Tomas Aliaga (MARTEL) |
| V1.0 | 27/07/2018 | Merging pending contributions from partners and final revisions. | Tomas Aliaga (MARTEL) |

**Disclaimer**

The information, documentation and figures available in this deliverable, is written by the SmartSDK (A FIWARE-based Software Development Kit for Smart Applications for the needs of Europe and Mexico) – project consortium under EC grant agreement 723174 and does not necessarily reflect the views of the European Commission. The European Commission is not liable for any use that may be made of the information contained herein.

***Confidential*** *- The information contained in this document and any attachments are confidential. It is governed according to the terms of the project consortium agreement*

**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 | |

** R: Document, report (excluding the periodic and final reports)*

*DEM: Demonstrator, pilot, prototype, plan designs*

*DEC: Websites, patents filing, press & media actions, videos, etc.*

*OTHER: Software, technical diagram, etc.*

# 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 takeup of FIWARE by new developers and its transition from proof-of-concept environments to production ones.

This document updates the third version of the Roadmap (D5.6) discussing covered developments during the period from November 2017 to May 2018, providing also an overview of planned Epics and Features in the period from May 2018 to August 2018. Full documentation of User Stories and tasks is kept in the Agile management tool (JIRA[1]) 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:

- IoT Management Highlights
- Data Management Highlights
- Platform Highlights

This document is the last update of the SmartSDK Roadmap series.

[1] https://jira.fiware.org/projects/SMAR

## TABLE OF CONTENTS

## LIST OF FIGURES

# 1. INTRODUCTION

This document updates the previous version of SmartSDK R&D roadmap (v3). 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 final period, covering from June 2018 to August 2018.

Just like the previously presented versions, this version of the SmartSDK Roadmap is also based on the activities defined through the Agile R&D methodology[2]. 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)[3] following FIWARE Community best practices.

This deliverable focuses on the latest R&D plans of SmartSDK, focusing on priorities for the upcoming months prior to the end of the project:

- ➔ Guide the SmartSDK developers in the R&D activities by keeping in mind main outcomes expected by the end of the project;
- ➔ Provide hints to the wider FIWARE Community of SmartSDK latest plans and what may come afterwards from the results of the project.

## 1.1 Concept

SmartSDK is 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 and 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).

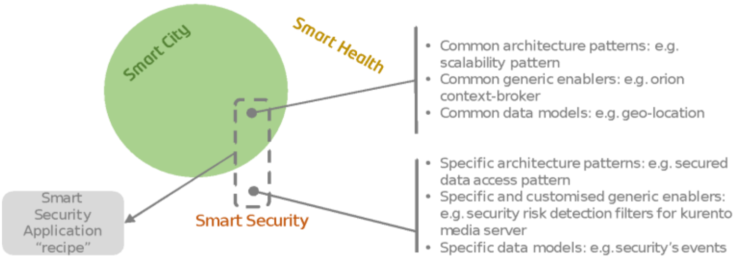
Smart City
Smart Health
Smart Security
Smart Security Application “recipe”
• Common architecture patterns: e.g. scalability pattern
• Common generic enablers: e.g. orion context-broker
• Common data models: e.g. geo-location
• Specific architecture patterns: e.g. secured data access pattern
• Specific and customised generic enablers: e.g. security risk detection filters for kurento media server
• Specific data models: e.g. security’s events

*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.

[2] The methodology and tools are documented in D5.1: Quality Assurance Guidelines [2]. Appendix A of this document includes a summary of the methodology, given that D5.1 is a confidential document.

[3] https://jira.fiware.org/projects/SMAR

## 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.

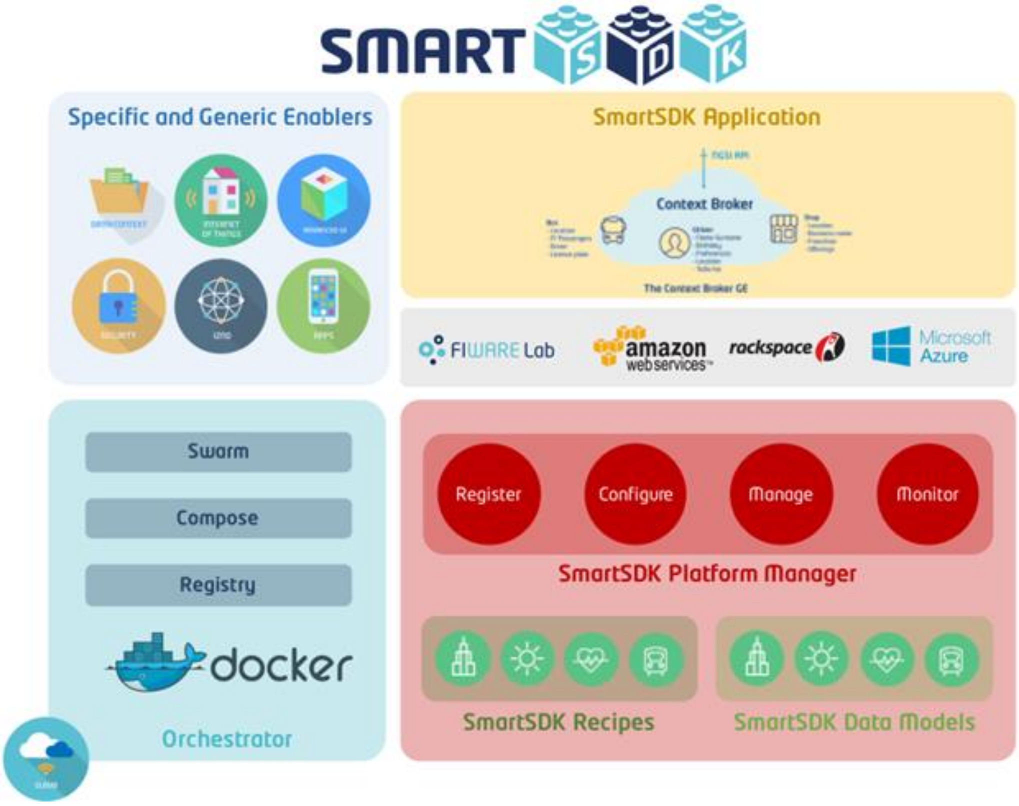

*Figure 2: SmartSDK Architecture.*

The SmartSDK Platform Manager will allow users to register and maintain 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 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 these components integrate 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).

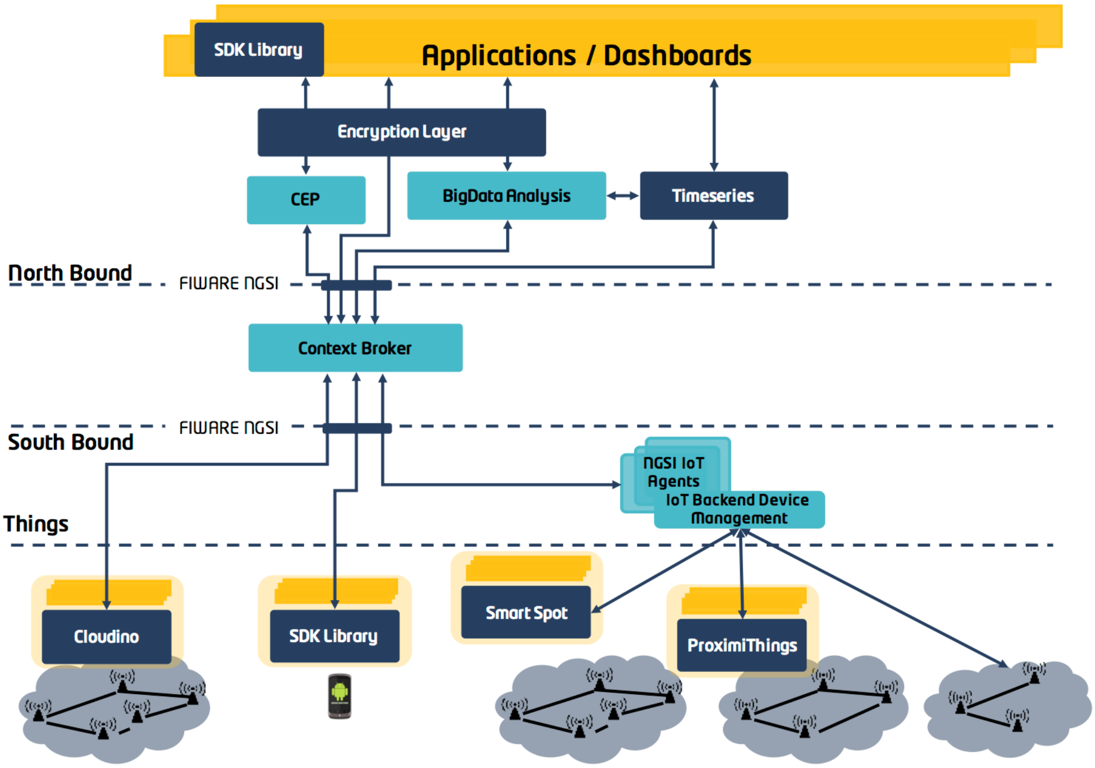

*Figure 3: SmartSDK revised reference architecture for IoT and Data Intensive applications.*

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

# 2. APPLICATIONS

During the period December 2017 – May 2018, 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 June 2018 - August 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 environment-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.

The current version has included data sources such as air quality, traffic, weather, pollen and alerts according to the latest data model definitions.

### 2.1.1 Covered Epics (December 2017 – May 2018)

To support the mentioned Smart City scenario, a set of components have been developed from Sprint 4.2, December 2017 to Sprint 6.1, May 2018. Those components are mentioned in each epic defined in the first version of the roadmap (v1.0).

- **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.
- **Maps management**
  - Pollen maps: display information about pollen concentrations.
  - Show weather conditions on a map. We should have a data source for weather conditions.
  - 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.
- 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. The data is obtained by a crawler from the Mexican aerobiology network page.

➔ **Data models**

- Review and modification of TrafficFlowObserved model, according to the model used by the traffic source.
- Definition of AeroAllergenObserved data model to target pollen information sources.

➔ **IoT Management**

- Develop guidelines to connect sensors.

➔ **Alerts management**

- Alert history: the connection between the ITESM's Orion Context Broker instance and the Alerts Application was established.
- Sending alerts: the connection between the Green Route Back-end and the Alerts Application was established.
- Encrypted alerts: the alert to the NGSI Encryption Layer was sent, as an example of usability of the last-mentioned component, for its encryption. Subsequently, the encrypted alert was stored in the ITESM's Orion Context Broker instance.

- Decrypted alerts: the encrypted alert was retrieved from the ITESM's Orion Context Broker instance and decrypted using the NGSI Encryption Layer.
- Alert data model: the alert schema was updated to incorporate the latest modifications performed to this schema from FIWARE Data Models[4].
- Alerts visualization: a proof of concept (from Insomnia) was carried out between QuantumLeap and Orion Context Broker.

### 2.1.2 Planned Epics (June 2018 – August 2018)

To support the mentioned smart city scenario, a set of components need to be completed and eventually adjusted from June 2018 to August 2018.

➔ **User profile management**

- Add a security certificate to Green Route[5]. This is required to warranty the user account information from users. Also, a security certificate avoid that some browsers emit alerts about insecure and not trusted web pages.
- Attach a "Privacy statement" agreement during user registration in order to notify the user about the safety of its data.
- Show the amount of pollution produced by each transport methods, according to the distances traveled.

➔ **Maps management**

- Visualize heat maps about high levels of pollen, pollution and traffic congestion.
- A user can select a vehicle / public transport from his user profile.
- Public transport location: integrate the real time public transport location in the maps.

➔ **Route management**

- 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.
- Rate and share a route.

➔ **IoT Management**

- Develop guidelines for using data models.
- Develop guidelines for using the IoT agent.
- Develop guidelines to send data from IoT devices to the Orion Context Broker.

➔ **Alerts management**

- Sending alerts: establish the connection between the Green Route's Orion Context Broker instance and the Alerts Application.
- User database: establish the connection between the Green Route's Keyrock instance and the Alerts Application, avoiding double authentication by the user.

[4] http://fiware-datamodels.readthedocs.io/en/latest/Alert/doc/spec/index.html

[5] https://greenroutesdk.com.mx

- Global security certificate for Green Route: integrate the Alerts Application into the Green Route Application with the same security certificate as Green Route.

➔ **Container**

- Deploy the Green Route application on a Docker container.

➔ **Trials**

- Deploy a trial with the INFOTEC community.
- Alerts Application: test and deploy the Alerts Application within the Smart Cities Scenario for its Trial stage.
- ITESM buses: review, test, and deploy the Smart Spot Starter Kit within the Smart Cities Scenario for its Trial stage.

### 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.

## 2.2 Smart Security

The Smart Security application aims to assist a security staff by detecting and analyzing security situations such as access controls, people detection, face recognition, fights, crowd analysis, wrong way driving, etc., through the combination of video cameras and mobile sensors in both, indoor and outdoor scenarios. Examples of such scenarios include parking lots and buildings. The main goal is to provide 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 (December 2017 – May 2018)

To support the mentioned smart security scenario, a set of components have been developed from Sprint 4.2, December 2017 to Sprint 6.1, May 2018. The scope of work covered the following components:

➔ **Data management**

- Main System Update. Because of the requirements of the activity algorithms, the main server has been updated to Ubuntu 16.04, Kurento 6.7.1 and OpenCV 3.4.1.
- User management. A new user management module has been implemented based on Keyrock GE, more specifically Keystone. This module is used by both the mobile and the web app. Also, a data base to save the user information has been implemented.
- Phone Log Query. The component allows to determine if a person is or was inside a specific area at a certain moment based in the information produced by the OnlinePhoneQuery functionality.
- Mobile App. A new android mobile app has been developed. Functionalities include:
    - Log in of users
    - Visualization of the last ten alerts or by day generated by other smartphone users.
    - Generation of alerts by the user.
    - Automatic detection of the user in the campus.
    - Visualization of alerts and devices on map.
    - Notifications about sudden stop event.
- Database implementation (update). A DB module based on CrateDB and MySQL have been implemented to persist the information about users and alerts.
- Data Models:
    - Alert data module. Last version of the Alert data model has been integrated in both web and mobile version of the application.
    - Device: In order to register the smartphone and camera devices the model is used in the system.
- Web GUI update. As more events are delivered, more visual options will be activated in the different views of the system.
    - Visualization on map of the camera locations
        - Visualization of video streaming on map
        - Color representation on map if a visual event is detected
    - View for search mobile users.
    - Alerts generated by mobile users are displayed in the web GUI (Last Activities box).
- Testing: All new management functionalities has been tested.
- Testing on FIWARE node: A basic version of the system has been run in a FIWARE node.

➔ **Pattern recognition**

- Processing module. New or improved of algorithms have been implemented to detect events based on camera and smartphone data. The detection includes:
    - Unauthorized car speeds. Based on smartphone data (accelerometer), an algorithm that is able to detect if the car speed of a smartphone registered user surpasses the maximum permitted has been implemented.
    - Walking, Running and Fighting detection. Based on optical flow a computer vision technique, we have implemented an algorithm able to detect when people walk, run or fight in indoor environments.
    - Bicycle/Motorcycle detection. The Yolo library has been integrated to the system in order to detect vehicles like motorcycles.
    - Face Recognition. The algorithm detects if a person is or not on a video based on visual information produced by input images.
- Video Repositories. Several repositories for algorithm testing have been created and complemented with public data bases.
- Testing and integration of the developed algorithms.

### 2.2.2 Planned Epics (June 2018 – August 2018)

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

➔ **Data management**

- Notifications on Web GUI (Extension): Real time notification (visualization) of the detection result of the visual processing module on the Web GUI.
- DB implementation update in order to support alert and visual events information.
- Design and execution of the trial. We will define the final evaluation and area where the trial will be executed.
- Search. Extension of the search capabilities of the system.
- Dockerization. A basic version of the system will be dockerized.
- Area registration: A new view to register a new surveillance area will be implemented.
- Alert Search view. A new view to search by alerts generated by smartphones will be implemented.
- Trial execution: final version of the system will be tested in the ITESM campus.

➔ **Pattern recognition**

- Forgotten Large Objects. A module that analyzes the visual information and detect if a large object is forgotten in indoor environments.
- Crowd formation. As a signal of risk situation, the module is intended to detect when several persons meet in one point.
- Wrong way driving. This sub-module is intended to detect if a car does not follow the correct way by:
    - Smartphone sensor
    - Visual analysis
- Sudden stop optimization. An improve of the sudden stop algorithm (based on

smartphone sensor) will be implanted.

- 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.
- Face Recognition Optimization. An optimization process will be executed in order improve the response of the algorithm.
- Testing and Integration. Final version of the system considers the testing and integration of the wrong way and forgotten object algorithms based on visual analysis and wrong way and sudden stop based on smartphone sensors.

### 2.2.3 Key used FIWARE Enablers

➔ Keyrock (keystone)

➔ 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.

### Risks

After some tests, we have been able to run a basic and stable version of the system in a node FIWARE. However, we are facing a new issue regarding the integration of activities algorithms into Kurento framework. As more complex situation to detect more complex algorithms (and dependencies) to solve, and unfortunately, we have no support from the Kurento team. Therefore, not all implemented algorithms will be integrated in the final version of the video surveillance system.

## 2.3 Smart Health

Sensors in mobile, wearable, and environmental devices are becoming ubiquitous, allowing the inference of activities and behaviors 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 analyzing 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 (December 2017 – May 2018)

Below, we present a set of updates for each of the components that have been under development from Sprint 4.2 (December 2017) to Sprint 6.1 (May 2018) to support the mentioned Smart Health scenario within the context of two applications (Risk of fall and Rehabilitation).

**Risk of fall App**

**Upload information:**

- ➔ Data integration
  - Integration of heart rate sensor data collected from wearable devices within the data-model.

**Data analysis:**

- ➔ 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.

**Data privacy:**

- ➔ 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**

**Sensing mobility from Smart devices:**

- ➔ 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.

**Data improvement:**

- ➔ Monitorization.

- Constant monitorization and feedback for the patient to take actions based on patient's performance detected by wearable devices.

**Data reduction:**

- ➔ Statistical analysis of patient performance will take place to further be presented to the health specialist

### 2.3.2 Planned Epics (June 2018 - August 2018)

- ➔ Dockerization of MA-Test (mobile application for the risk-fall assessment)
- ➔ Integration of Grafana services as part of the the adaptive Robot application
- ➔ Release and testing final version of the adaptive Robot application.
- ➔ Finishing pilot study, acquisition, processing and data analysis.
- ➔ Update ongoing documentation and reporting

### 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 helps 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 (December 2017 – May 2018)

At the end of 2017, we were worried that rancher v1.6, on which the Platform Manager is based, was not going to be maintained anymore and the support for docker-compose recipes v3 would have to be abandoned because of lack of support in favor of Kubernetes[6]. So, we planned to change our plan for the worst case. Fortunately, it did not turn out to be a real threat, and rancher v1.6 will be fully supported until the end of 2018. Rancher 2.0[7] was launched in May 2018 with frequent minor releases for bug-fixing. Hopefully, in the next month things will stabilize, and the Rancher 2.0 upgrade path may be taken into consideration for further evolution of the project. In the meantime, to conclude our project we focused on:

- ➔ **Support Docker and Rancher working with Docker Swarm**
  - Support the running of the latest version of Docker and Rancher that supports Docker compose v3. Done, our recommendation is to run docker-17.12 on Ubuntu 16.04 LTS.
- ➔ **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. This was not done, instead we ported old code to new Rancher release that will be supported at least until the end of 2018.
  - Development of custom web interface and/or integration of existing web interfaces that allows the deployment of Docker Swarm recipes v3. This was done by adapting the SmartSDK recipes for Portainer.
  - 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. This was done by providing custom templates and script that works out of the box in the FIWARE Lab.
- ➔ **Generic Enablers Recipes**

[6] https://kubernetes.io

[7] https://rancher.com/tags/release/

- Add recipes for enabling the reference architecture patterns for a deployment of CKAN generic enabler. This recipe was dropped due to lower priority, in fact, none of the application scenarios were using CKAN. Instead, we prioritized support for applications seeking to close the gap between WP3 and WP2.
- 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. In the summit of late 2017, the FIWARE community presented API Umbrella as the preferred solution for managing security of APIs in FIWARE deployments. This, we decided to add a recipe for such generic enabler.
- Harmonize (as in checks of consistency) the customization of recipes through environment variables and the documentation of all of them. This task has been done many times and will be continued until the last days of the project, because the evolving nature of the chosen technologies is constantly revealing new and better ways to use and reuse recipes. The most important alignment occurred when we instrumented the automatic adjustments of recipes so that they can be used both manually and through the Smartsdk Platform Manager.

➔ **Investigate and document transition to Kubernetes**

- In the long term, Kubernetes will most likely be the default choice for orchestrating containers, so we wanted to attempt and document the transition to Kubernetes. This was not done mainly because rancher v2.0 was not released on time to evaluate the advantages and disadvantages of the choice.
- Probably we will not able to migrate all the recipes, but we will document the conversion steps for some of them. This also was not done for the same reason as above.
- Finally, in early 2018, Docker announced the support for Kubernetes deployments directly from the Docker tooling. This seems to indicate our docker-based work will live even longer and such translation efforts may not be worth doing as this will be taken care by the same docker tooling in the near future.

### 3.1.2 Planned Epics (June 2018 – August 2018)

There are no planned Epics for this period. Our work on the project stopped at end of May.

### 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) such as “Docker Swarm Secrets for creating great FIWARE platforms” by Federico Facca (MARTEL) and “Quantumleap: a FIWARE Time Series DB” by Tomas Aliaga (MARTEL).

➔ SmartSDK Presentations in FIWARE Tech Summit in Porto (May 2018). The presentation of Federico Facca “CONNECTING TO TIME SERIES DBs USING QuatumLeap” was introduced by showing how to deploy Quantum Leap on the Platform Manager using SmartSDK recipes by Daniele Pizzolli.

## 3.2 Data management

The Data Management activities focus on contributions aimed to enrich the FIWARE Data Management chapter and improve interoperability of NGSI (with respect to devices and open data models). Activities of this chapter have been defined and prioritized considering SmartSDK applications' requirements.

### 3.2.1 Covered Epics (December 2017 – May 2018)

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**

During this period, QuantumLeap gained the first implementation of the methods to retrieve historical records through the methods mentioned in the original proposal of the API. Such planned methods allow retrieval of historical records of one or multiple attribute values of one or multiple NGSI entities. The API is available at https://app.swaggerhub.com/apis/smartsdk/ngsi-tsdb/0.1.

Although we did not integrate a default retention policy in this period as planned, we implemented the methods to delete historical records. Such methods are an essential first step towards retention policies and moreover allows users to clean up historical data in more flexible ways. Users can delete records of one or multiple attributes of one or multiple entities in determined time ranges.

In addition, we implemented and endpoint to control the creation of the Orion Subscription QuantumLeap depends on. This seeks to simplify the workflow of QuantumLeap and to reduce the margins of configuration errors by the less experienced users.

According to the plan, in this period QuantumLeap gained support for Multitenancy, in the same fashion the Orion Context Broker does. This feature enables the support of different tenants sharing the same QL deployment with proper isolation of usage and privacy.

Another feature, not originally planned but added on-demand in this period, was named "geocoding". It permits users of geo-located historical records have an automated translation from address in postal form (number, street, city, country) to geo-json indexes in the backend. This way stored historical records can have an harmonized representation of "locations" and be ready for interesting geo-queries (like filtering data based on proximity).

Finally, as usual and for maintenance purposes, several fixes were implemented for bugs reported by application scenarios using QuantumLeap (e.g. SMAR-708 and SMAR-710). The documentation of the component was partly updated for the needs of the "SmartSDK Guided Tour" launched in this period, specially the items related to deployment. However, a good amount of documentation updates is still missing. The writing of the missing parts is planned for the next period.

**NGSI Encryption Layer**

- Stand-alone application: the first version of the NGSI Encryption Layer integrating a simple encryption algorithm capable of encrypting and decrypting a string of alphanumeric characters was developed.
- Test: the first version of the NGSI Encryption Layer was integrated into the Alerts Application to exemplify its usability.

- Encryption algorithm: the adoption and adaptation of the Rivest, Shamir, and Adleman (RSA) algorithm was selected as the encryption method.
- Key generator: a key generation method that is linked to the user’s ID and specifies the encrypted attributes, chosen by the same user, was developed.

➔ **SDK Library for NGSI**

In this period, the SDK Library for NGSI was developed in Javascript language. This software tool aims at transforming JSON entities to NGSI data models, which can be manipulated or operated by the FIWARE Orion Context Broker. The information about the library is available at https://github.com/smartsdk/ngsi-js-library.

The current version of the NGSI library for JavaScript implements client functionalities to get information of the Orion ContextBroker, and also analysis functions for non-structured JSON objects for them conversion to NGSIv2 entities.

In addition, in this period another feature that was called Data Models Schema Analyzer has been integrated, to enrich the support of the library. In this functionality, the main task is to analyze the structure of NGSI entities in order to verify if the entity matches or not with the structure of a data model defined in the repository “dataModels” of the SmartSDK account in Github. This, taking into consideration the specification of the official schema for each one data models (JSON Schema).

The SDK library for NGSI has been divided in two npm modules. The list of the activities carried out on this period in each module, are the following:

- NGSI- parser npm module:
    - analysis non-structured JSON objects to convert them into entities that follows the NGSI format.
    - generation of queries in string format, through of conditions defined in a JSON object.
    - special function called data Models schemas analyzer. This function verifies if a JSON object fulfill the structure indicated in the official specification of a data model, i.e. the rules specified into JSON Schema of the data model.
- OCB-sender npm module:
    - configurable connection with an Orion ContextBroker instance.
    - execution of custom CRUD operations, of entities, subscriptions, simple queries and geo queries in an Orion Context Broker instance.
    - download function to get the JSON Schemas of data Models hosted in the SmartSDK repository.

Finally, in order to check the functionality of the library, a RESTFul web service has been developed which contains some functions of the library which can be put in practice through API requests. In a first version, this web service was developed as a test case, but then, it became in a key component of the SmartSecurity scenario, and currently it is running for the application of this scenario.

### 3.2.2 Planned Epics (June 2018 - August 2018)

➔ **Time Series for NGSI**

As originally planned, the main focus for the last months of work is mostly documentation and pending bug-fixes.

- Documentation & Maintenance
  - Significant updates to documentation of latest features of QuantumLeap.
  - The repository where QuantumLeap is being developed must be reorganised as explained in https://github.com/smartsdk/ngsi-timeseries-api/issues/22. (Brought from last period)

    There were some tasks planned in previous versions of the roadmap that felt out of scope due to time constraints and or lack of priority such as the following. Of course, if time allows they will be considered for implementation after the aforementioned fixes are done.
  - “Configuration” endpoints to customize QL setup.
  - Allow per-table customization of number of replicas and shards / partitions.

➔ **NGSI Encryption Layer**

- Documentation: documentation must be completed and shared according to the project’s guidelines, as well as simple How-to guides will be created.
- Algorithms integration: adopt and adapt the ElGamal encryption algorithm and two versions of the k-anonymity as anonymization algorithms into the Encryption Layer Application.
- Encryption and anonymization test: test the NGSI Encryption Layer functionality when anonymizing, encrypting, deanonymizing, and decrypting diverse fields of JSON files whose formats correspond to the FIWARE Data Models. Fields that can be anonymized and/or encrypted are leftt to user’s concern according to their requirements

➔ **SDK Library for NGSI**

The activities planned for the last months are focusing on:

- Updates to the technical documentation of the library, to enrich the current documentation in greater detail and complete it with the latest features added in this period.
- Review the functionalities (the code) of both modules of the library, to detect pending bug-fixes and minor improvements.
- Review the data models’ schemas analyzer, in order to add more support in the checkup of the JSON schemas specification.
- Add http headers support to the client functions of ocb-sender.

### 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.

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

## 3.3 IoT Management

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

### 3.3.1 Covered Epics (December 2017 – May 2018)

To support the SmartSDK platform, a set of components have been developed from Sprint 4.2, December 2017 to Sprint 6.1, May 2018. Such components are detailed in each epic defined in the first version of the roadmap (v1.0).

➔ CLOUDINO

- It is possible to configure a SD Card component in the circuit by using Arduino or Cloudino standalone. The Cloudino could be programmed to save the data measured by the sensors in a SD Card; if there is no connection to internet. And when the connection gets established again, the data saved in the SD Card can be sent to an application or to another device.
- Also, if there is no SD Card component, Cloudino can be configured to save the data in the internal memory, until the connection to internet get established, and then, the data can be sent to an application or to another device. The user can choose one or both options, depending the application or the criticality of the data. The next code is an example of how to save data in memory, and then send it to an application.
- In case the connection is lost, Cloudino is configured to reconnect itself to the networks already known.
- Cloudino can be configured to shut down the WiFi connection, while is not necessary to send data. For example, if the data is sent every hour, the WiFi connection can be down all the time, and turn on when the data should be sent. This configuration allows saving a lot of energy. The average consumption of the device with active WiFi is around 80 mA and inactive close to 14 mA.
- The Cloudino GSM Connector was implemented. Once the GSM board is properly connected, it is only necessary to enable the GSM option, through Cloudino Server. The options Active and GSM Active in the Cloudino Cloud Configuration must be True.

➔ SMART SPOT

- Device connection configuration over GPRS and firmware upgrade. This functionality that will allow the update of its firmware remotely which will allow the arrangement of possible bus or security issues.
- Read and Observe FIWARE lwm2m IoT Agent, this functionality allow user to have values of the connected devices without wait until a value update.
- In order to satisfy the project expectations, an Open Hardware Version has been extracted out of the private one. This version has been created with educational, experimental and pilot expectations. This version provides several sensors like temperature, humidity, accelerometer and the hardware connections necessary to use a place of gases.
- Wifi direct capability; this functionality has been discarded since it was going to be developed by the protocol nearby of google which has discarded this functionality.In order to allow users to receive applications and notifications via Wi-Fi, several alternatives are being investigated, which involve providing Wi-Fi connectivity through the device, the application will be transmitted via captive portal or redirecting the connection to the desired URL.
- Crowd Monitoring, this functionality allows the devices to know the crowds of people around them by scanning near mobile devices. This kind of functionalities allow users to create applications that avoid or decongest in these areas.

- Connection Status. This functionality had to be discarded since the use of the current LwM2M does not allow it. In the first instance a modification of the IoTAgent component was proposed but once a deeper study was made it was decided to develop a new one because the current one did not fix security communications and has some misses with the LwM2M standard, the new lwM2M IoTAgent will be published in September 2018, latest.

### 3.3.2 Planned Epics (June 2018 - August 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

- Finalize and validate that the new method of contacting users through wifi is well supported by these, and try that a large number of users come to the application broadcasted by the device in an agile and simple way.
- Continue developing the new IoTAgent that allows the community and users to make use of LwM2M devices in a simple, safe and versatile way using the FIWARE data models.

### 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 at 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.

# 4. CONCLUSIONS

This document presents the last version of the SmartSDK R&D roadmap covering updates from the previous period of six months of work and focusing on the last two from June 2018 to August 2018. It also highlights the planned and achieved contributions to the FIWARE Open Source Community.

In this deliverable, we covered these items 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 final steps until the end of the project.

As regards the Platform (Work Package 3), this document 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 the end of the project.

# REFERENCES

[1] SmartSDK Consortium. Description of Action. July 2016. SmartSDK project is co-funded by the EU's Horizon2020 programme under agreement number 723174 - ©2016 EC and by CONACYT agreement 737373.

[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: SMART SDK 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[8]; this will allow a continuous monitoring of project progress in term of outcomes and exploitable results. The process is depicted in Figure 3.

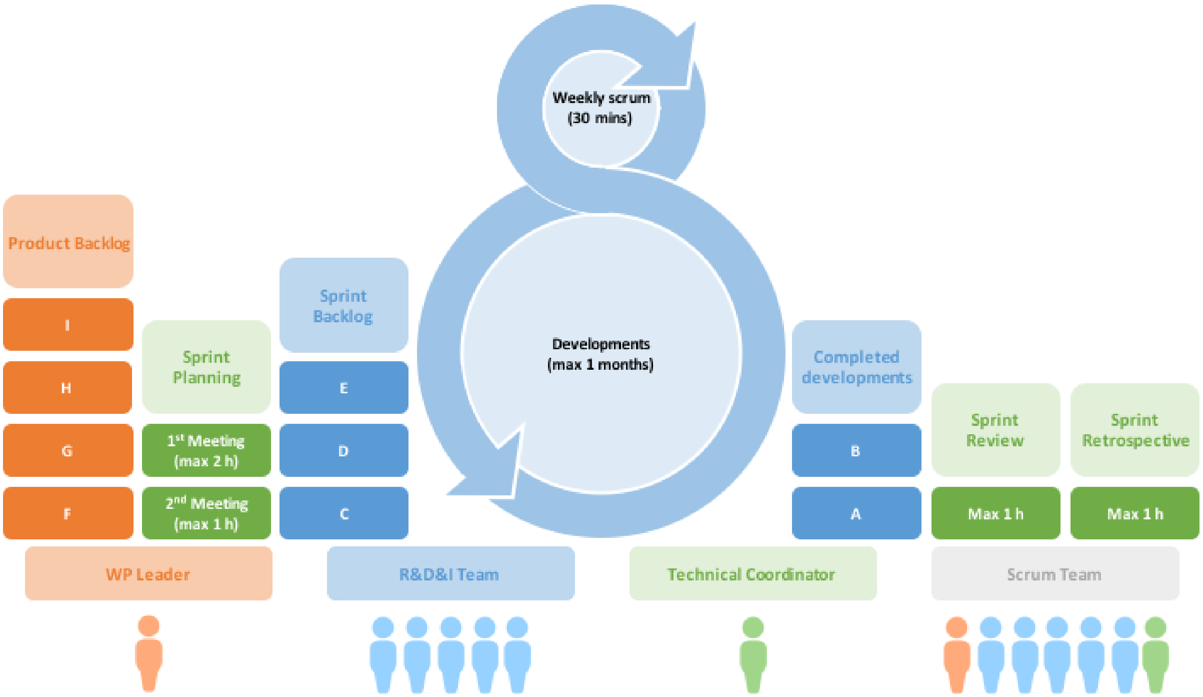

*Figure 4: SmartSDK Agile Development Process.*

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

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

## 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