WP2 Programmable Smart City
D2.5 Self-aware programmable city platform - demonstration

Grant Agreement N°723139
NICT management number: 18301

BIGCLOUT

Big data meeting Cloud and IoT
for empowering the citizen ClouT in smart cities

H2020-EUJ-2016 EU-Japan Joint Call

EU Editor: **CEA** JP Editor**: NTTRD** Nature: Demonstration

Dissemination: PU

Contractual delivery date: 2017-07-01

Submission Date: 2017-07-13

Co-funded by the EU H2020 GA. 723139 and NICT GA. 18301

ABSTRACT

This deliverable describes the second demonstrators of the data collection and redistribution framework of the BigClouT project. Its objective is to show the work of integration of project partners' achievements.

Disclaimer

Revision history

Revision	Date	Description	Author (Organisation)
V0.1	30/04/2018	Initial ToC	Christophe Munilla(CEA)
V0.2	31/05/2018	Integrated Scenario & Functional requirements sections draft	Christophe Munilla(CEA)
V0.3	20/06/2018	D-NR and CityHub second year update and integration sections	Rodger LEA (LANC)
V0.3	21/06/2018	StreamOLAP and JSSpinner second year update and integration sections	Savong BOU (TSU)
V0.4	22/06/2018	Requirements coverage section	Christophe Munilla (CEA)
V0.5	25/06/2018	Add introduction Remove components and tasks coverage section	Christophe Munilla (CEA)
V.06	01/07/2018	Review from CEA and integration of final contributions	Levent Gürgen (CEA)
V0.9	12/07/2018	Final review	Feng Chen (NII)
V1.0	13/07/2018	Submission	Levent Gürgen (CEA)

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This deliverable describes the second demonstrator of the data collection and redistribution framework of the BigClouT platform. After a short overview of the work that was done during the second year on each asset that was part of the first year demonstrators, the document focuses on an integrated view of the data collection and redistribution framework, whose description is guided by the functional requirements. The document presents how a part of the functional requirements are already fulfilled; some others not being covered will be in a future version of the BigClouT platform.

A wide range of components of the BigClouT platform are in interaction in the demonstrator. A data collection and redistribution framework providing also service composition features and benefiting of self-aware mechanisms covers all tasks assigned to the WP2, as it is shown in the Figure 1 below.

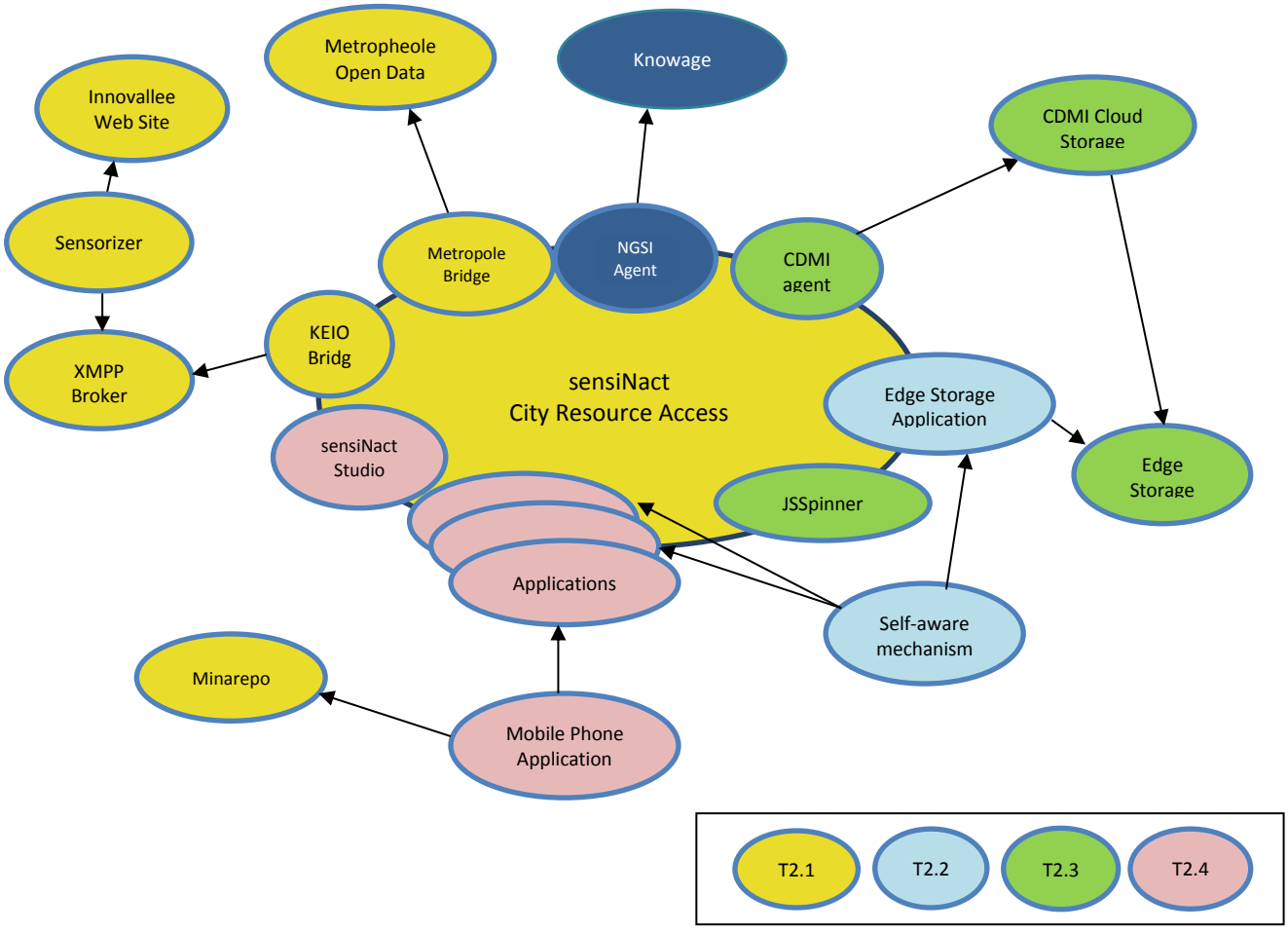

FIGURE 1: COMPONENTS AND TASKS COVERAGE

This deliverable is a "demonstrator" type of deliverable, whose objective is to show the work of integration of project partners' achievements. Technical details of the demonstrated components are given in project deliverables such as D1.1 [1], D1.2 [2], D1.3 [3], D1.4 [4], D2.1 [5], D2.2 [6], D2.3 [7] and D3.1 [8]. The deliverables are available for download at the project website.

2. BIGCLOUT 2ND YEAR DEMONSTRATOR

2.1 First year demonstrator overview

At the end of the first year, a demonstrator showing the first integration of some of the components of the BigClouT platform has been built. Figure 2 shows the components that have been demonstrated, which had been partially integrated. More details can be found in the Deliverable 2.2 [6].

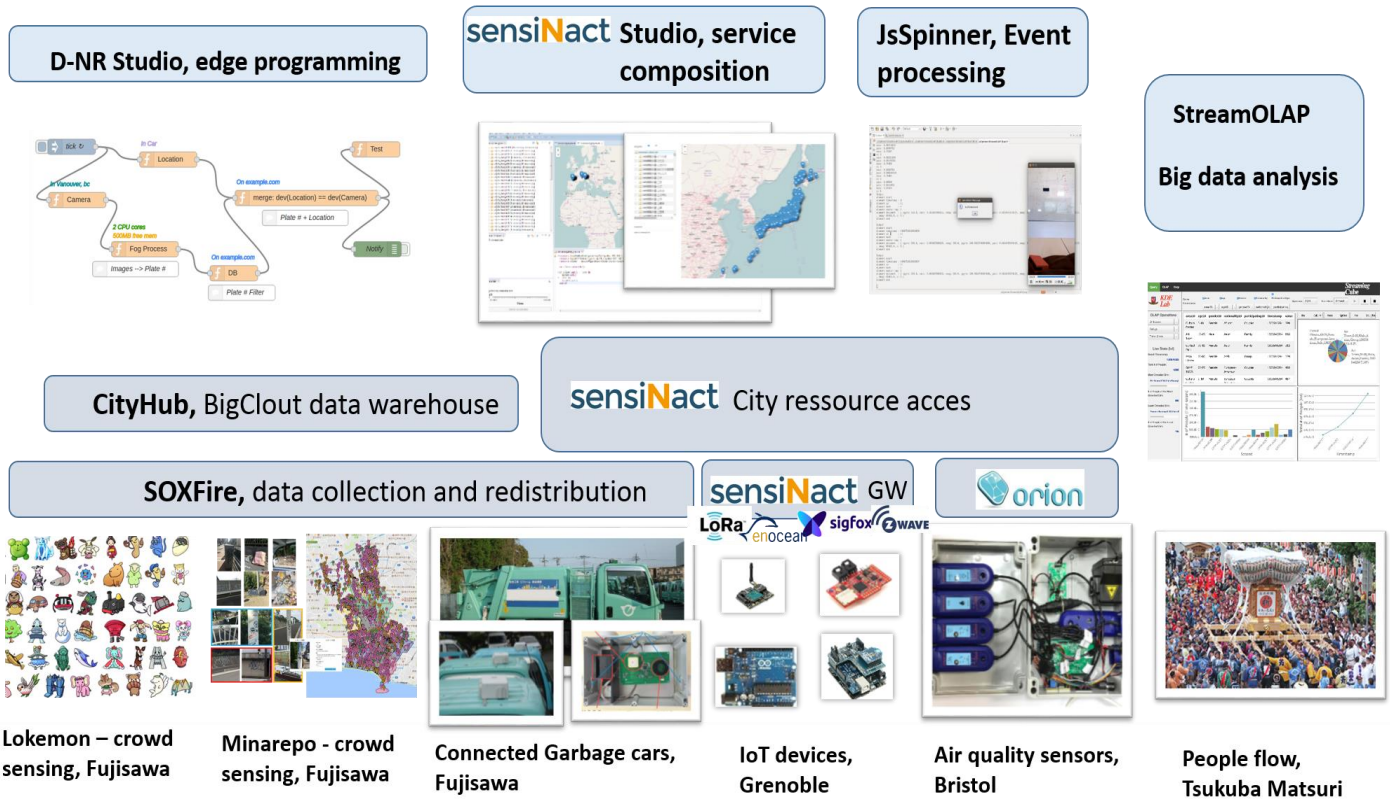

FIGURE 2: FIRST YEAR DEMONSTRATOR ASSETS

The main objective of the second year has been, firstly to update the project assets fulfilling the requirements identified in the project, and secondly to integrate among the components to fulfil the project requirements. In Section 2.2 update on each component is briefly described, while Section 2.3 provides some brief information on the work done in terms of integration with other components.

2.2 Second year assets updates

2.2.1 DN-R

Distributed Node-RED (D-NR) comprises a visual programming tool (D-NR studio) and an Edge Processing framework (D-NR platform). Year 1 was mainly focused on updating and integrating D-NR studio into the BigClouT architecture. In year 2 we switched our focus to the development of an edge processing capability as part of D-NR platform and developed an integration with D-NR studio that allowed developers to use an easy visual annotation tool to

apply constraints to smart city applications that were then used by the underlying D-NR platform to drive edge processing and load balancing.

2.2.2 sensiNact Studio

For the sensiNact Studio, the second year updates were mainly focused on implementation of necessary functions to be compatible with the new version of the sensiNact gateway, in particular the authentication and authorization features, the distribution feature of sensiNact gateway instances, and the sensiNact Application persistence mechanism.

The new version of the sensiNact Studio is available for download at https://projects.eclipse.org/projects/technology.sensinact/developer

2.2.3 JsSpinner

For second year updates, we focused mainly on making the manual and making JsSpinner open source software. It is distributed by Kitagawa & Amagasa Data Engineering Laboratory, University of Tsukuba under the Apache License, Version 2.0. For more information, referring to the detail information following the link: http://www.streamspinner.org/streamingcube/documentation.html

2.2.4 StreamOLAP

For second year updates, we focused mainly on making the manual and making StreamOLAP open source software. It is distributed by Kitagawa & Amagasa Data Engineering Laboratory, University of Tsukuba under the Apache License, Version 2.0. For more information, referring to the detail information following the link: http://www.streamspinner.org/streamingcube/documentation.html

2.2.5 City Hub

City Hub provides a generic data warehouse capability for BigClout based on the CKAN open source platform. In year to we continued to add integration support for BigClout, upgrading the SoxFire integration nodes, adding support for Orion and adding both real-time and historical data assets.

2.2.6 SOXFire

SOXFire provides multi-community sensor data dissemination functionality which allows to set appropriate access model in each PubSub node as virtual sensor with keeping federating different servers. In year two, we updated SOXFire in terms of it's basement software from Openfire version 3 series to version 4. It allows us to manage virtual sensors more intuitively by using web-based interface.

2.2.7 sensiNact gateway

For the sensiNact gateway, the second year updates were mainly focused on:

- the improvements of the security aspects to reduce the number of security rules to be defined for specifying a security policy;

- the improvements of the memory usage of the communication mechanism between multiple sensiNact gateway instances;
- and finally the implementation of the sensiNact applications persistence mechanism.

2.3 Second year assets integration

2.3.1 DN-R

Our focus for D-NR integration in year 2 was on the SoxFire platform from Keio University which is used in the Fujisawa trials. We worked to ensure that D-NR was integrated with the SoxFire protocol and that D-NR could be used as part of the Fijisawa infrastructure monitoring trial. An initial integration has been carried out and D-NR is now used as part of the Fujisawa road condition monitoring service (infrastructure trial) to manage data coming from the WP3 component, Deep-onEdge, running on Garbage trucks in Fujisawa city.

2.3.2 JSSpinner

For second year, we mainly focused on integrating and connecting JsSpinner with sensiNact, which is a data collecting tool provided by CEA. As part of this integration, environmental data collecting from carsensors in Fujisawa city is consumed and analysed by JsSpinner.

2.3.3 Stream OLAP

For second year, we mainly focused on integrating and connecting StreamOLAP with sensiNact, which is a data collecting tool provided by CEA. As part of this integration, environmental data collecting from carsensors in Fujisawa city is consumed and analysed by StreamOLAP.

2.3.4 City Hub

The CityHub data warehouse component was integrated into a number of the BigClout trials with direct support for Fujisawa via Soxfire and indirect connectivity to the Orion broker in the Bristol trial. Additionally, integration with KNOWAGE was tested with a number of data retrieval and analysis tasks using KNOWAGE and the CityHub data warehouse.

2.3.5 SOXFire

SOXFire component was integrated into a number of the BigClouT trials including Fujisawa, Tsukuba in Japan, and also in Grenoble and Bristol via SensiNact technology and MinaRepo application. It is also integrated with D-NR environment with nodered-contrib-sox module (npm install is now available.)

2.3.6 sensiNact gateway

Two integration works relative to the sensiNact gateway were conducted during this second year: the first one consist in extending an existing oneM2M bridge, previously allowing only the use of the MQTT protocol, to permit its use in an HTTP context. The second integration work was also an integration extension one and its purpose was to be able to handle more dynamic data provided by the Grenoble Metropole's open data system.

3. INTEGRATED SCENARIO

In order to demonstrate the effectiveness of the integration and the implementation work that have been realized since the beginning of the project, we created a scenario showing how one Grenoble Innovallee worker can interact with the BigClouT applications and services in the city, implying how the (functional and non-functional) requirements described in Section 4 have been fulfilled. By using the BigClouT app:

- At 7:00am, before his/her working day starts, a Grenoble Innovallee worker checks for traffic status from the app.
- He/She checks also for air pollution status
- He/She decides to not use his/her car and checks for available public transportation. He/She saves the chosen itinerary as favorite to be consulted later in an easier manner
- He/She notifies the system about his/her choice ; the information is stored for a future analysis of the people behavior according to the context and organising future advertisement campaigns for promoting the public transportation, and it is also processed directly to update public transportation usage forecast for the next hours
- He/She asks for being notified of arrivals of buses on a specific station close to his/her home from 8:00am and to be notified of ones on a station close to his/her work place from 6:00pm
- The bus station shelter is damaged; the Grenoble Innovallee worker notices the damage and notifies the municipality about it; the information is stored for future analysis of the degradation of the urban furniture in the city, and to refine the management of the stocks of material necessary to intervene more effectively. The information is also transmitted directly as an event to the application used by municipality to be able to solve the issue in the best times.
- At 11:30am he/she checks for available restaurants and food trucks' menu
- After checking the air quality and weather forecast information, he/she decides to create a proposal to invite his/her contacts to meet at a specific food truck, proposing Japanese food menu for lunch
- At the end of the day, the Grenoble Innovallee worker is notified that his/her bus is arriving at his/her chosen station

3.1 User centric functional scenario

The global scenario specified above can be illustrated graphically by UML Use Case (UC) diagrams, which focus on the users and their interactions with the system, implying the functional requirements. A set of UML use case diagrams is displayed as follows:

- At 7:00am, before his/her working day starts, a Grenoble Innovallee worker checks for traffic status

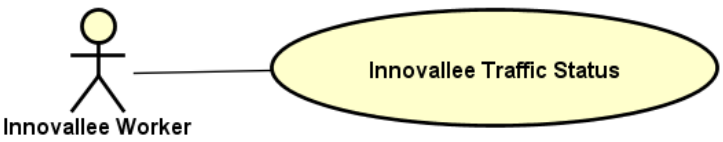

FIGURE 3: TRAFFIC STATUS UC

- A Grenoble Innovallee worker checks for pollution status

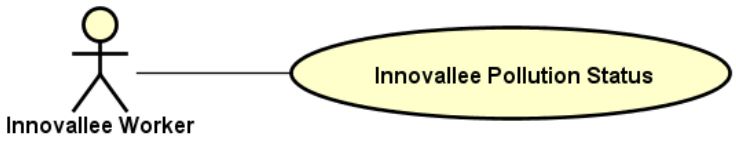

FIGURE 4: POLUTION STATUS UC

- He/She decides to not use his/her car and checks for available public transportation and saves the chosen itinerary as favorite to be consulted later in an easier manner

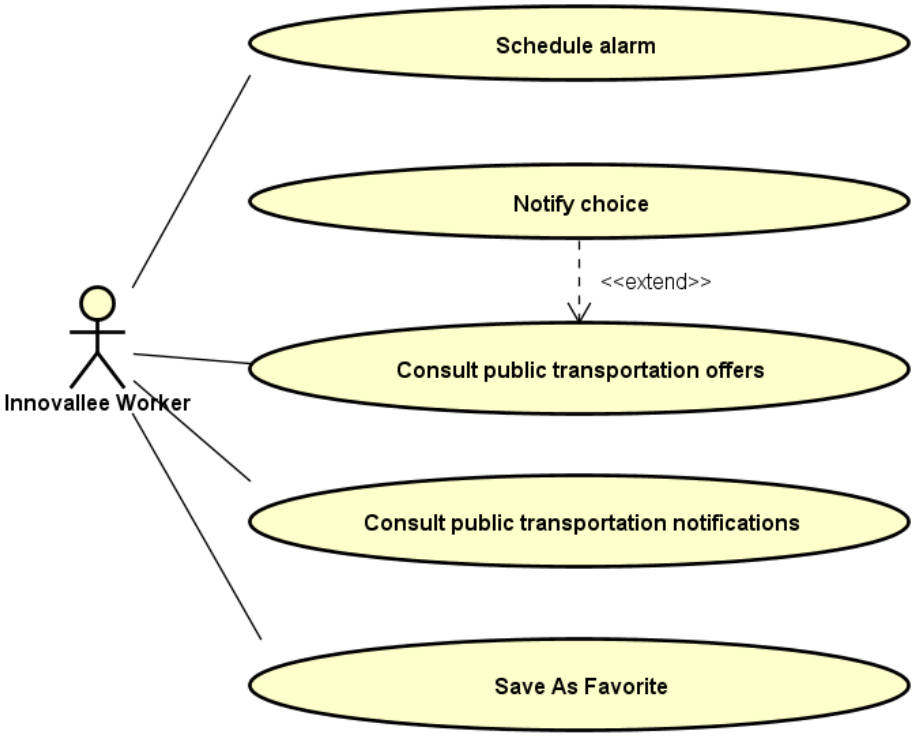

FIGURE 5: PUBLIC TRANSPORTATION UC

- He/She notifies the system about his/her choice (cf.Figure 5)
- He/She asks for being notified of arrivals on a specific station close to his/her home from 8:00am and to be notified of ones on a station close to his/her work place from 6:00pm (cf. Figure 5)
- The bus station shelter is damaged, the Grenoble Innovallee worker notifies the municipality

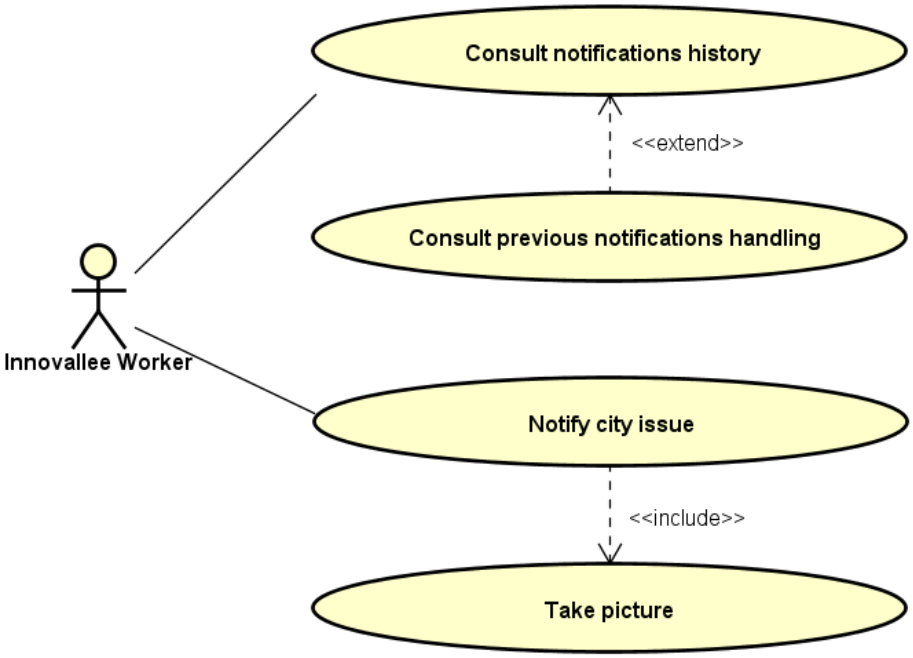

FIGURE 6: CITY PROBLEM ALERT UC

- At 11:00am he/she checks for available restaurants and food trucks' menu.

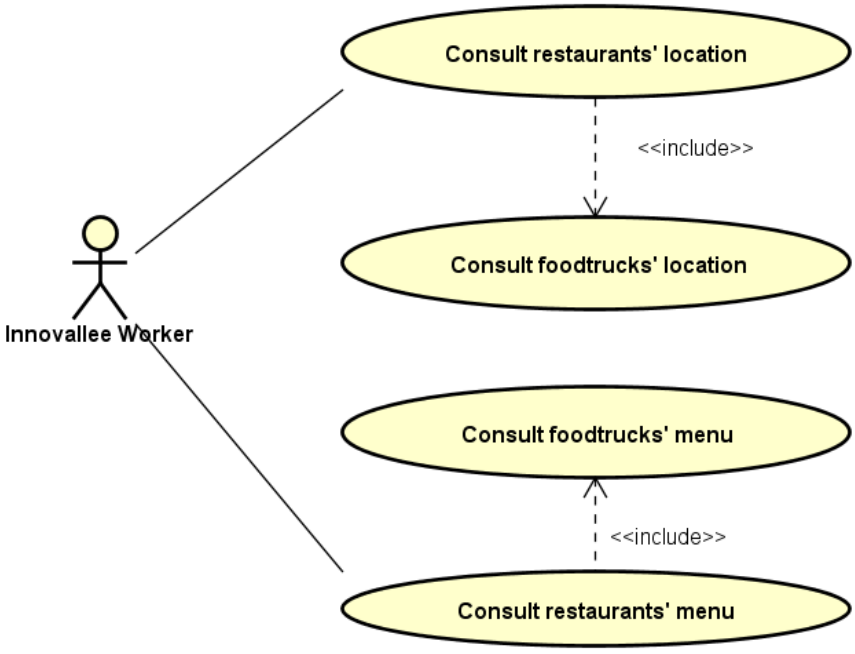

FIGURE 7: RESTAURANT INFORMATION UC

- After the check of the meteorological information he/she decides to create an open proposal to be sent to his/her contacts to meet at a specific food truck, proposing Japanese food menu for lunch

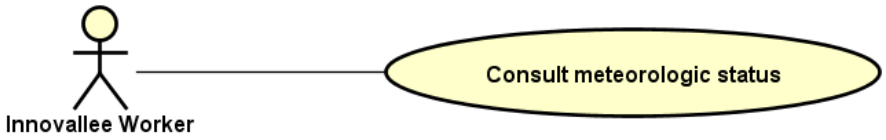

FIGURE 8: METEO STATUS UC

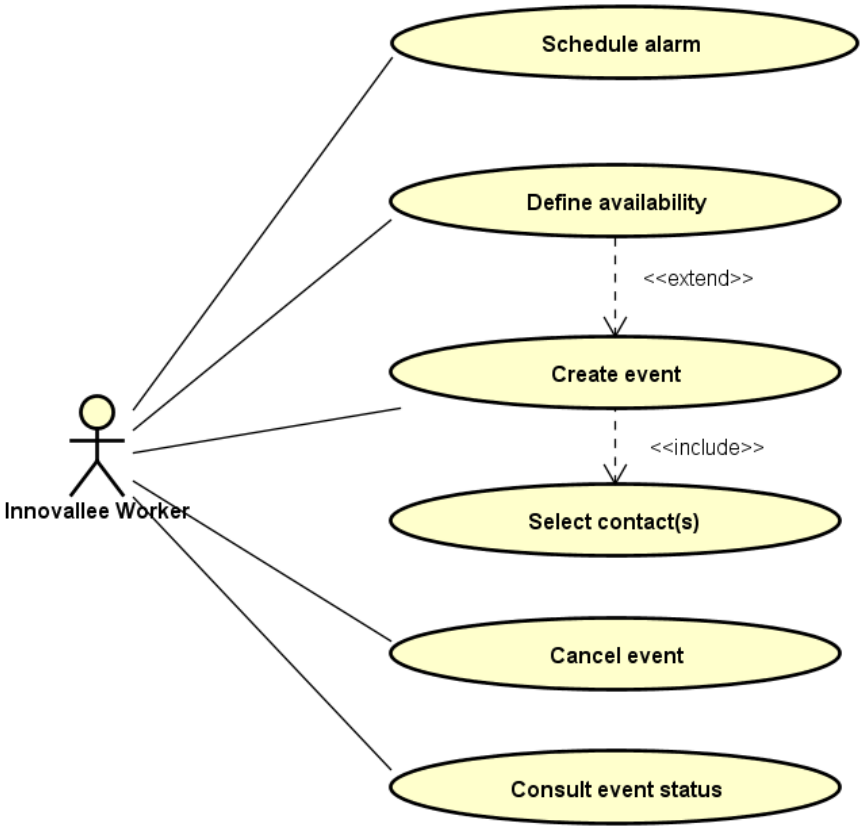

FIGURE 9: EVENT MANAGEMENT UC

- At the end of the day, the Grenoble Innovallee worker is notified that his/her bus is arriving at his/her chosen station (cf.Figure 5)

3.2 Data flow scenario

The global activity diagram presenting the data flow through the entire BigClouT platform is shown below (Figure 10).

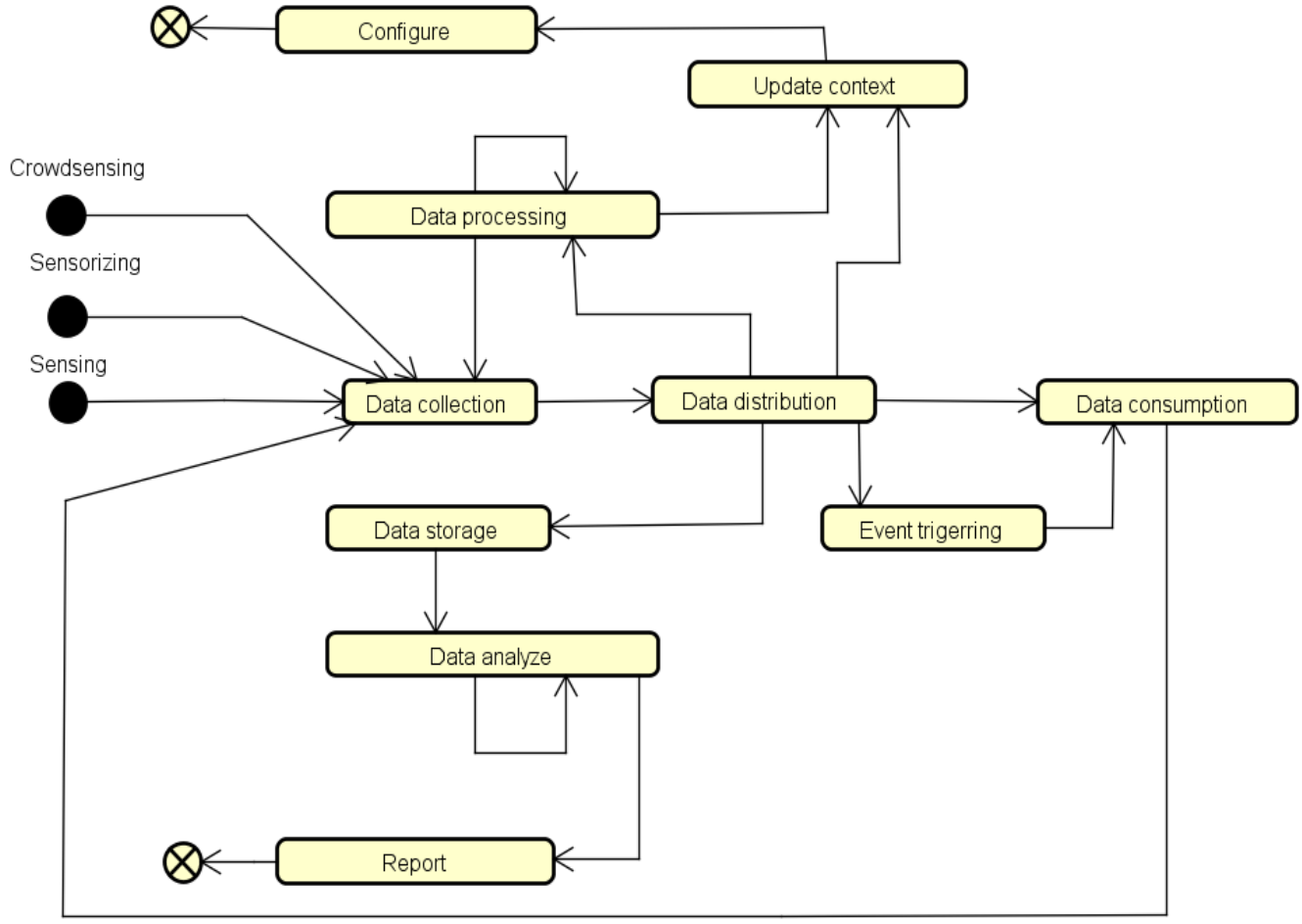

FIGURE 10: BIG CLOUT PLATFORM ACTIVITY DIAGRAM

The data flow of the demonstrator fits this generic data flow defined above for the entire BigClouT platform : Data collected from users by the way of mobile application(s), from stakeholders, from the continuous live data processing, and from IoT devices can be next redistributed as is to the different applicants (like sensiNact applications for example); it can be dispatched as event if ever captured according to registered subscriptions; it is also transmitted to the components in charge of storing them for future analysis, to the ones in charge of continuously processing them in manner of maintaining up-to-date contexts – two distinct contexts exist here: the execution context of the BigClouT platform itself on one hand, and the public transportation usage context on the other hand. The information relative to these two contexts are then treated by the components in charge of self-awareness mechanism and the applications provided to the Grenoble Metropole to monitor the public transportation usage forecast respectively.

4. FUNCTIONAL REQUIREMENTS COVERAGE

An updated list of the functional requirements applying on the BigClouT platform was presented in the last deliverable dedicated to the *architecture and requirements of the platform* [4]. Those requirements were gathered in two distinct lists: one list defines the generic requirements common to all trials, and the other defines the requirements specific to each pilot.

4.1 Generic functional requirements

The generic functional requirements list presents the set of requirements that are shared by all field trials. Those requirements can be mapped to the different elements of the BigClouT platform architecture, as shown in the figure below:

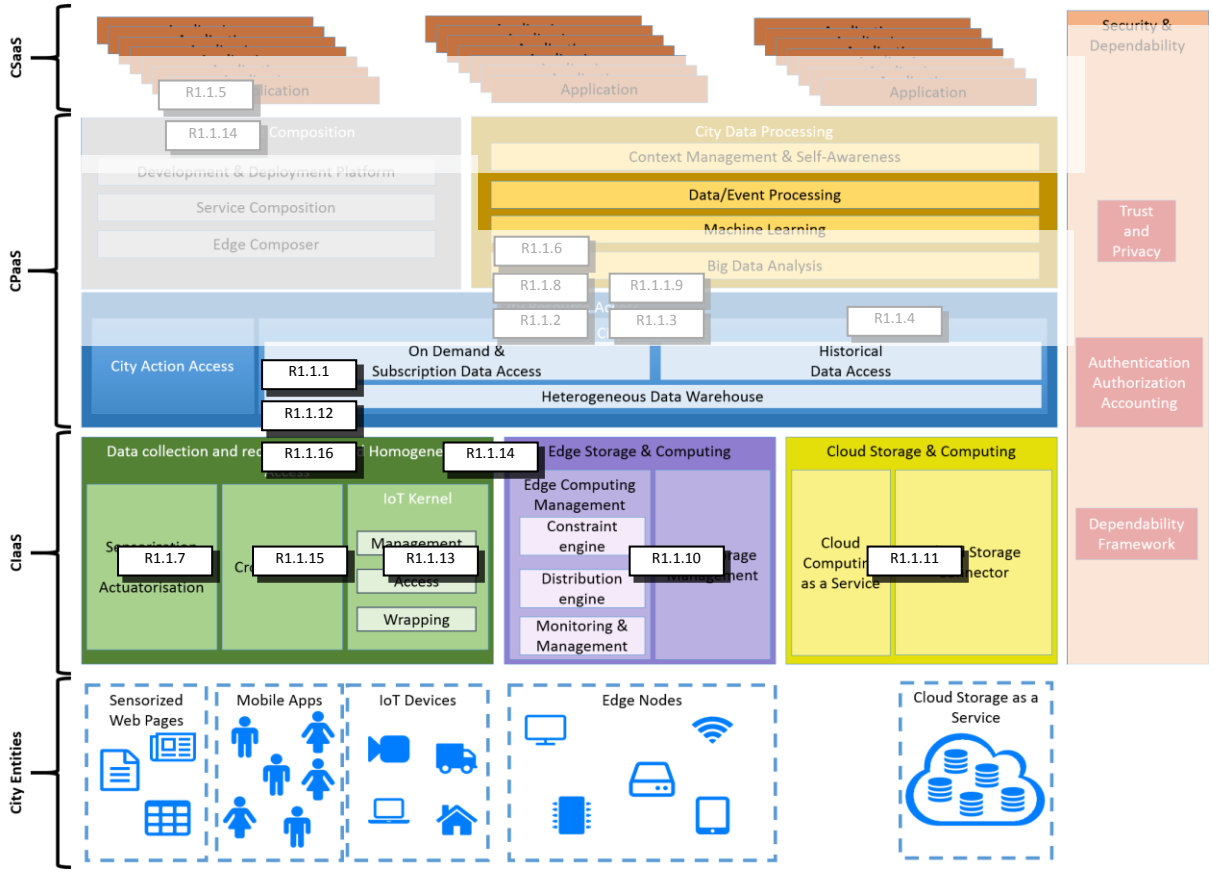

FIGURE 11: REQUIREMENTS TO ARCHITECTURE MAPPING [5] - UPDATED

TABLE 1: BIGCLOUT PLATFORM GENERIC FUNCTIONAL REQUIREMENTS

Code	Description	Coverage
R1.1.1	The platform should be able to access data from sensors on demand and through subscriptions.	The user asks for pollution level and requires to be notified of the departure of public transportation at a scheduled time (taking potential perturbations into account)
R1.1.2	The platform should provide big data analytics functionalities.	The usage is transmitted to the Knowage analytics tool in manner of improving the service offered to city citizens

R1.1.3	The platform should be able to perform predictive analysis.	The user(s) can inform the system of its (their) transportation choice allowing the system to provide public transportation congestion prediction
R1.1.4	The platform should be able to perform recommendations as part of predictive analysis	If users inform the system of their transportation choice, allowing the system to provide public transportation congestion, it is also able to recommend alternative path and/or alternative operational hours
R1.1.5	The platform should provide a dashboard in order to present results of analysis.	Uncovered in this scenario. It is part of WP3 demonstrator
R1.1.6	The platform should provide real-time data processing functionalities.	To provide recommendations, the processed data are the live public transportation usage ones
R1.1.7	The platform should be able to access online data, e.g. from web sites and social networks.	Restaurants menu are retrieved from the Innovallee dedicated web site
R1.1.8	The platform should provide data machine learning functionalities.	Uncovered in this scenario. It is part of WP3 demonstrator
R1.1.9	The platform should provide distributed machine learning functionalities.	Uncovered in this scenario. It is part of WP3 demonstrator
R1.1.10	The platform should provide edge processing functionalities.	Public transportation usage data is processed at the edge (JSSpinner & sensiNact integration)
R1.1.11	The platform should be able to collect and store data.	Notification of urban furniture degradation as well as the all usage of application by the user is stored.
R1.1.12	The platform should be able to provide stored historical data.	A user can view the history of handling of urban furniture degradation notifications.
R1.1.13	The platform should be able to be integrated with existing sensor networks.	Meteorologic and pollution data are collected from already deployed sensor network
R1.1.14	BigClouT should be able to collect data from heterogeneous data sources (open data from the city, real-time traffic information, localisation of users, etc.)	Data relative to public transportation are extracted from the Grenoble Metropole open data system
R1.1.15	BigClouT should provide means to push data from the users and relevant stakeholders into the platform.	A user can create an event into the system, to be proposed to a set of other users
R1.1.16	BigClouT should be able to issue notifications to end users when interesting events occur.	A user can require to be notified of the departure of public transportation at a scheduled time (taking potential perturbations into account)

5. BIBLIOGRAPHY

[1] BigClouT, “D1.1 Data collection tools and architecture”

[2] BigClouT, “D1.2 Citizen centric use cases and requirements”

[3] BigClouT, “D1.3 First BigClouT Architecture”

[4] BigClouT, “D1.4 Updated use cases, requirements and architecture”

[5] BigClouT, “D2.1 Data collection tools and architecture”

[6] BigClouT, “D2.2 Data collection and redistribution framework – demonstration”

[7] BigClouT, “D2.3 Self-aware distributed city data platform”

[8] BigClouT, “D3.1 Big data analytics framework architecture”