

SYNCHRONICITY

GA no:	732240
Action full title:	SynchroniCity: Delivering an IoT enabled Digital Single Market for Europe and Beyond
Call/Topic:	Large Scale Pilots
Type of action:	Innovation Action (IA)
Starting date of action:	01.01.2017
Project duration:	36 months
Project end date:	31.12. 2019
Deliverable number:	D2.6
Deliverable title:	Guidelines for the integration of IoT devices in OASC compliant platforms
Document version:	1.0
WP number:	WP2
Lead beneficiary:	28-UDG
Main author(s):	Eunah Kim (UDG), Arturo Medela (TST), Bart Braem (IMEC), Rubén Molina Moreno (HOPU), Salud Ruiz Fernández (HOPU), Cédric Crettez (MI), Francesco Arigliano (ENG), Martino Maggio (ENG), Paolo Fosci (MIL), Kaisa Silelius (FVH), Koen Cornelissen (Antwerpen), Omer Faruk Ozdemir (ATOS)
Internal reviewers:	Filipe Aranda de Sá (POR), Andrea Gaglione (DigiCat), SeungMyeong Jeong (KETI), Jaime Ventura (POR), Martin Brynskov (AU)
Type of deliverable:	Report
Dissemination level:	PU
Delivery date from Annex 1:	M14
Actual delivery date:	M20 (30.08.2018)



Co-funded by the
European Union

This deliverable is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732240.

Executive Summary

IoT technologies are being developed and deployed in diverse industry domains and naturally they embrace heterogeneity on devices, data characters, data communication technologies, data models, platforms, etc. As more services are converged and break out the barriers of the industry border, technical challenges on providing interoperable solutions are growing. However, the development of a new standard protocol to fit the needs takes pretty long effort and procedure reinventing and anyway it will be hard to find new homogeneous technologies that fit for all. Rather, the studies are performing to provide context integration minimizing the complexity from the reinvention of the existing work.

SynchroniCity aims to establish the technical foundations of a digital single market for IoT-enabled smart cities in Europe and beyond, where IoT device manufacturers, system integrators and solution providers can innovate and compete in an open environment. In order to realize the objectives, it is important to design minimal interoperability points in the smart city platforms. Each city should be allowed to use its own smart city platforms and chosen technologies on its smart city service development. In order to build a synchronized framework, a minimal set of APIs for open data gathering and sharing should be agreed and implemented.

With this principle, SynchroniCity has defined a minimal set of interoperability points in the southbound and northbound of the platform. This deliverable shows the collected data of the current southbound information from the 8 cities called Reference Zones (RZs) that participate in the SynchroniCity project. For the platform integration model, the data from Seongnam Korea has been also collected in order to include a city using the oneM2M platform. The collected data has been depicted in a table for each city. In order to see commonality and differences, the collected data has been categorized per application and showed used systems, networking (link) technologies and data/messaging protocols. After the analysis, integration models have been discussed with the focus on context data sharing. Diverse smart city platforms are used in SynchroniCity's RZs and the integration model has been discussed. Examples on how to build an NGSI adapter for communicating with context data management module have been included for guiding cities that need to build one.

We expect that the analysed statistical data from the SynchroniCity's RZs will be useful for other cities that search for a solution in the same application domain, and the integration models and guidelines will help to measure the required effort in the integration with the SynchroniCity framework, participating the foundation building of the Digital Single Market (DSM).

Abbreviations

AE	Application Entity
API	Application Programming Interface
BLE	Bluetooth® Low Energy (BLE)
CoAP	Constrained Application Protocol
CKAN	Comprehensive Knowledge Archive Network
CSE	Common Services Entity
D	Deliverable
DALI	Digital Addressable Lighting Interface
DASH7	Developers' Alliance for Standards Harmonization of ISO 18000-7
DSM	Digital Single Market
DTLS	Datagram Transport Layer Security
EC	European Commission
GPS	Global Positioning System
GPRS	General Packet Radio Service
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technologies
IoT	Internet of Things
ITS	Intelligent Transportation Systems
JSON	JavaScript Object Notation
LEZ	Low Emission Zone
LoRaWAN	Long Range Wide-Area Network
LTE	Long Term Evolution
M2M	Machine to Machine
Mca	A reference point for M2M communication between CSE and AE
Mcc	A reference point for M2M communication between CSE and CSE
Mcn	A reference point for M2M communication between CSE and NSE
MQTT	Message Queue Telemetry Transport
NFC	Near-field communication
NGSI	Next Generation Service Interfaces
NSE	Network Service Entity
OASC	Open & Agile Smart Cities
REST	Representational state transfer
RFID	Radio Frequency Identification
RZ	Reference Zone
SDH	Synchronous Digital Hierarchy
SSL	Secure Sockets Layer
TETRA	Terrestrial Trunked Radio
TLS	Transport Layer Security
UDG	Universal Device Gateway

UL2.0	FIWARE UltraLight 2.0
WP	Work Package
WS	Web Socket
WT	Work Task

Contents

1	Introduction	8
1.1	WP & Task objectives	8
1.2	Deliverable objectives and methodology	8
2	Overview of IoT solutions deployed	12
2.1	ANTWERP	12
2.2	CAROUGE	13
2.3	EINDHOVEN	14
2.4	HELSINKI	14
2.5	MANCHESTER	15
2.6	MILAN	15
2.7	PORTO	17
2.8	SANTANDER	18
3	Comparative analysis of IoT integration models	20
3.1	Smart Parking	20
3.2	Smart Noise control	22
3.3	Smart Street Light	24
3.4	Smart Mobility and Traffic	26
3.5	Culture	28
3.6	Citizen information	29
3.7	Environment data	30
3.8	Energy management	31
3.9	Safety Alarms	31
3.10	Security	31
3.11	Observation	32
4	Generic integration models and Guidelines	33
4.1	Integration models	33
4.2	Integration guidelines	41
4.2.1	Generic description of an Ad-hoc NGSI Adapter	41
4.2.2	NGSI Adapter Example 1: FIWARE IDAS Agent	43
4.2.3	NGSI Adapter Example 2: UDG NSGI Adapter	47
5	Conclusion	50
	References	51
	Annex	53

List of Figures

Figure 1. T2.4 work methodology.....	9
Figure 2. SynchroniCity Reference Architecture and its key interoperability points (red arrows)....	10
Figure 3. Simplified view of the Southbound and Northbound interfaces.....	11
Figure 4. Parking - Connection Info.....	20
Figure 5. Parking – Data Communication Protocols	21
Figure 6. Parking – Data Format.....	21
Figure 7. Parking – IoT Device Management	21
Figure 8. Parking – Context Data Management	22
Figure 9. Noise – Connection Info.....	22
Figure 10. Noise – Data Format.....	22
Figure 11. Noise – Data Communication Protocol.....	23
Figure 12. Noise – IoT Device Management	23
Figure 13. Noise – Southbound Context Data Management	23
Figure 14. Street Light – Connection Info.....	24
Figure 15. Street Light – Data Format.....	24
Figure 16. Street Light – Data Communication Protocol.....	25
Figure 17. Street Light – IoT Device Management	25
Figure 18. Street Light – Context Data Management	25
Figure 19. Mobility – Connection Info	26
Figure 20. Mobility – Data Format	26
Figure 21. Mobility – Data Communication Protocol.....	27
Figure 22. Mobility – IoT Device Management	27
Figure 23. Mobility – Context Data Management.....	27
Figure 24. Culture – Data Communication Protocol	28
Figure 25. Culture – Data Format.....	28
Figure 26. Culture – IoT Device Management.....	28
Figure 27. Culture – Context Data Management	29
Figure 28. Citizen – IoT Device Version	29
Figure 29. Citizen – Connection Info	29
Figure 30. Environment – Data Communication Protocol.....	30
Figure 31. Environment – Data Format	30
Figure 32. Environment – IoT device management	31
Figure 33. Environment – Context Data Management.....	31
Figure 34. Southbound interoperability requirements	33
Figure 35. Support of diverse IoT protocols.....	34

Figure 36. DYAMAND - IoT Gateway integration model in Antwerp.	37
Figure 37. UDG integration model in Carouge	38
Figure 38. Helsinki Urban Platform integration model	38
Figure 39. WSO2 Integration model in Milan.	39
Figure 40. FIWARE Integration model.....	40
Figure 41. oneM2M Integration model.....	41
Figure 42. Ad-hoc NGSI adapter.....	42
Figure 43. Data polling	43
Figure 44. Subscription Scenario	43
Figure 45. Agent Manager integration	44
Figure 46. UDG as Carouge integrator.....	47
Figure 47. Creation of a NGSI v2 entity.....	48
Figure 48. Subscription to a NGSI v2 entity.....	48
Figure 49. Update of a NGSI v2 entity.....	49

List of Tables

Table 1. IoT information of Antwerp Smart City applications	13
Table 2. IoT information of Carouge Smart City applications.....	13
Table 3. IoT information of Eindhoven Smart City applications.....	14
Table 4. IoT information of Helsinki Smart City applications.....	14
Table 5. IoT information of Milan Smart City applications.....	16
Table 6. IoT information of Porto Smart City applications.....	17
Table 7. IoT information of Santander Smart City applications	19
Table 8. Integration models.....	36
Table A1. Reference Zones Analysis: Parking	53
Table A2. Reference Zones Analysis: Noise	54
Table A3. Reference Zones Analysis: Street Light	55
Table A4. Reference Zones Analysis: Mobility and Traffic.....	55
Table A 5. Reference Zones Analysis: Environment	58
Table A6. Reference Zones Analysis: Culture.....	60
Table A7. Reference Zones Analysis: Citizen	61
Table A8. Reference Zones Analysis: Energy Management	62
Table A9. Reference Zones Analysis: Safety Alarms	62
Table A10. Reference Zones Analysis: Security	63

1 Introduction

SynchroniCity represents the first attempt to establish the technical foundations of a digital single market (DSM) for IoT-enabled smart cities in Europe and beyond, where IoT device manufacturers, system integrators and solution providers can innovate and compete in an open environment. The big challenge is to provide an interoperable vendor ecosystem without vendor lock-in and city lock-in with provision of portability of IoT-enabled city services. In order to realize this vision, it is important to design minimal interoperability points in the city platform, while allowing individual choices on technology selection from the cities and avoiding reinvention of the wheel or addition of complexity on the existing platform.

In the architecture decision, a minimal set of interoperability points has been defined at the project consortium meeting based on the SynchroniCity architecture defined in D2.1 [1]. This deliverable D2.6 handles the southbound integration where heterogeneous IoT devices and systems are connected to and communicate with the platforms.

1.1 WP & Task objectives

The main WP2 (IoT-Enabled Framework Component Provisioning and Integration) objectives are to set up the roots for a real digital single market for IoT-enabled urban services through building a shared framework which defines the minimum set of APIs, a set of common data models for city domains, an environment for easy integration of IoT devices in a city platform, and a framework for data access management including licensing, privacy and market mechanisms.

Task 2.4 (Enablement of a DSM for IoT devices manufacturing for Smart Cities) is focused on developing a marketplace for IoT products and solutions that are interoperable with the SynchroniCity framework. This task addresses southbound extension of IoT-based smart cities, enabling interested cities to access a list of IoT products and solutions easily deployable and fully interoperable with the SynchroniCity architecture and deployment. The southbound will be designed to integrate multiple platforms to be used in smart city domains, with respect of OASC [3] compliant platforms, such as FIWARE [4] and oneM2M [5] platforms. The design of interworking will be carried out from protocol level as well as data model level, which make it possible for any data from the platforms to be gathered and shared within the DSM. At the end of this process, a catalogue of devices ready for the integration will be published.

1.2 Deliverable objectives and methodology

The aim of D2.6 (Guidelines for the integration of IoT devices in OASC compliant platforms) is to provide guidelines for southbound integration, so that IoT products and solutions can easily be deployed into the SynchroniCity framework defined in WP2.

It should be noted that the goal of this Deliverable is not to redesign a protocol, API, Agent or platform modules, but to analyse the existing solutions in RZs and to provide guidelines in order to meet the minimum requirements on the integration of the IoT devices in OASC compliant platforms. The integration models are designed as generic as possible in order to include all standard IoT communication protocols used in vertical smart city domains, and to focus on cross-domain interoperability.

Figure 1 illustrates the work methodology that was adopted within the context of T2.4. The information on IoT devices deployed and corresponding IoT protocols in the RZs has been collected and analysed, in order to understand the current situation of each RZ and the future needs from the cities. As different applications may need different requirements on communications and data exchanges with IoT devices, the analysis has been made per application deployed in the RZs. The direction of analysis is to see whether there are common requirements in the choice of IoT protocols

and to provide statistical data on each application built in the RZs, so that other cities may use it as a reference. According to the analysis results, the integration models and guidelines are provided. The integration models are built based on the analysis results to cover the most common protocols and data formats connecting heterogeneous things and to provide easily adaptable steps for diverse situations in the RZs. The following sections describe details of the RZs' input and corresponding guidelines. The final step in the diagram is not the scope of this Deliverable and will be performed as a next step of this Task (T2.4).

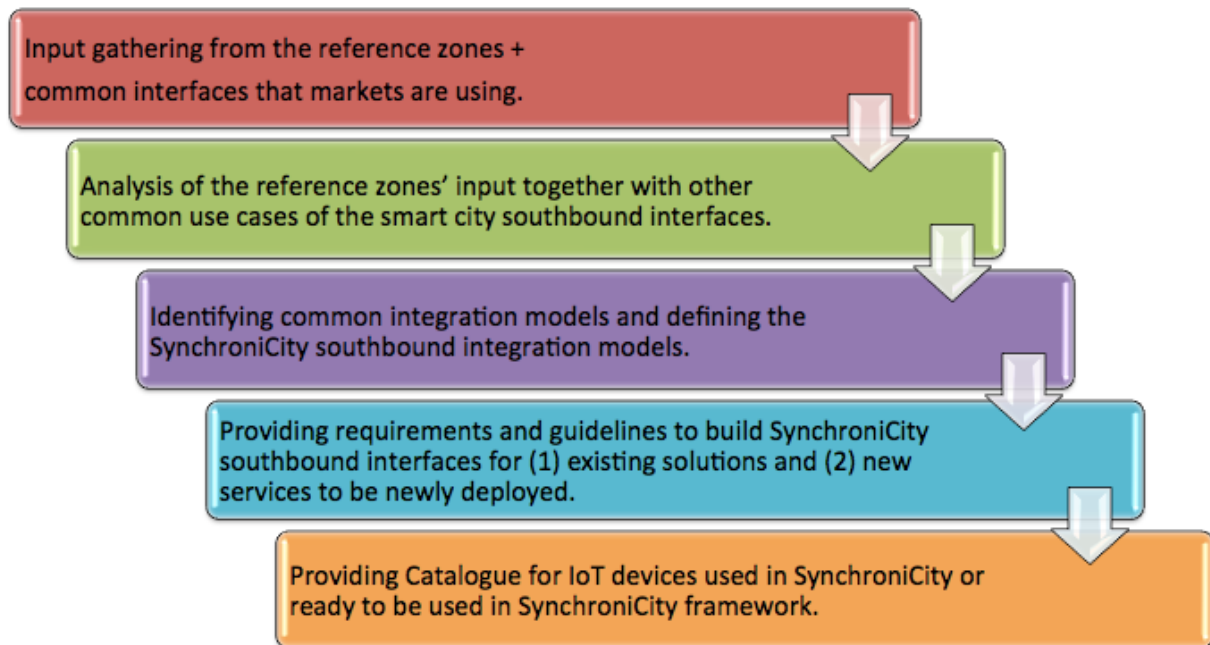


Figure 1. T2.4 work methodology

Figure 2 shows the key interoperability points on the SynchroniCity Reference Architecture (details on the architecture are provided in D2.1 [1]). The red arrows indicate the minimum interoperability points where the integration with physical things, data and services are made. In this architecture, the layer with integration points where physical things are connected is called “southbound”, while the “northbound” refers to the layer with the integration points where processed and stored data are used by smart city services and applications.

The scope of this Deliverable is where the physical things are integrated into the platform highlighted with the blue dotted box in the figure (southbound). As this is the first step to collect data from heterogeneous sources, it is important to support diverse communication protocols to be used in smart city services.

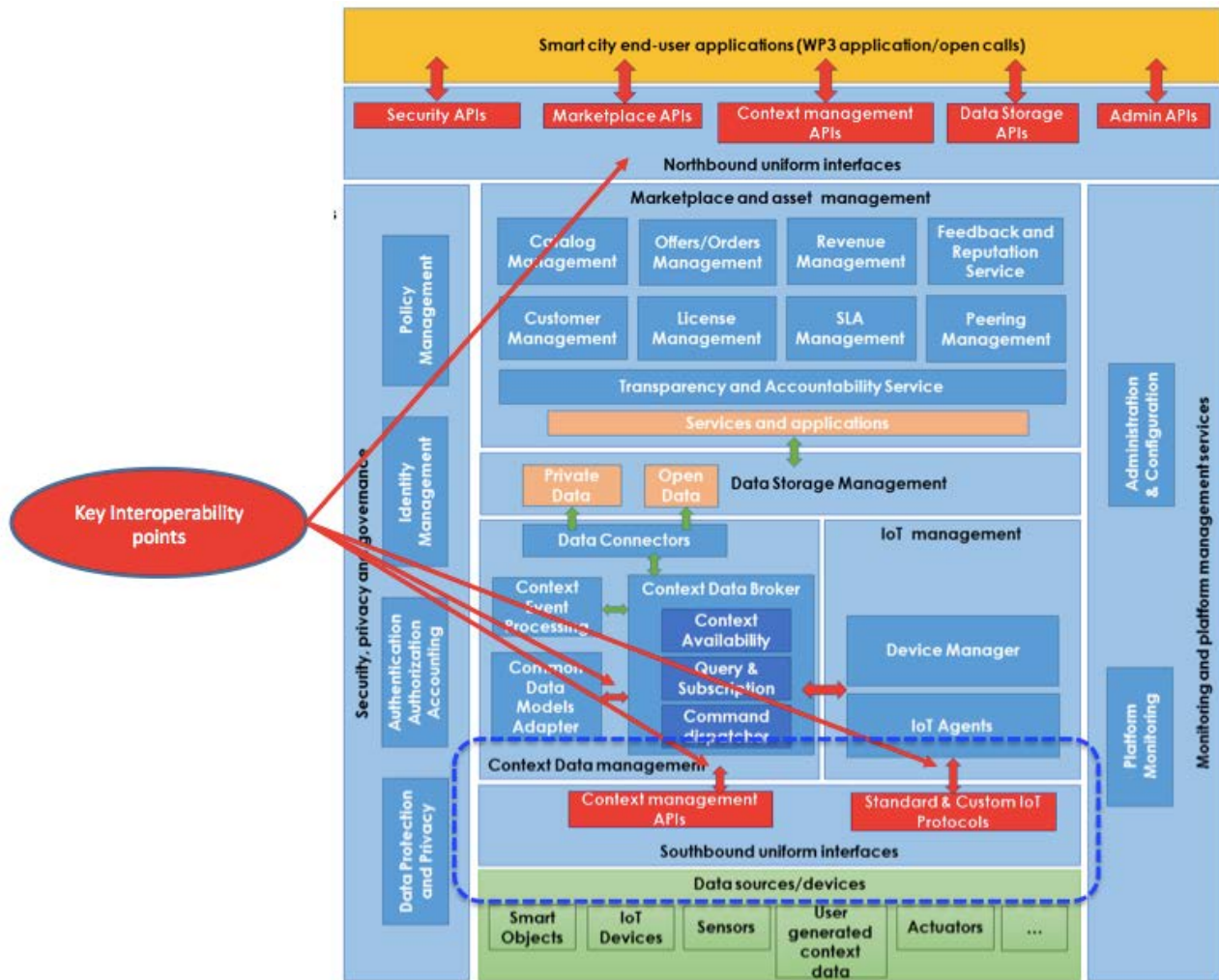


Figure 2. SynchroniCity Reference Architecture and its key interoperability points (red arrows)

As the focus of this project is not to define a single protocol or set of protocols, but to provide a framework which enables diverse existing solutions to be easily integrated, it is important to find common requirements of the southbound integration.

The simplified view on the interoperability points in the architecture is shown in Figure 3, which illustrates the role of the southbound and northbound interfaces. It must support heterogeneous network connectivity of things and IoT communication protocols used (and to be used) in the RZs, and yet provide seamless interoperability in data sharing. Thus, it is expected no redesign or reimplementation on the currently used technologies in IoT device management and communication protocols, which are selected by the needs of the city services in the RZs.

The focus is on the reusability of the collected data from the data sources of each RZ in order to transform them into new services. The following sections describe the details of the collected data and its analysis results, followed by the integration models and guidelines collected.

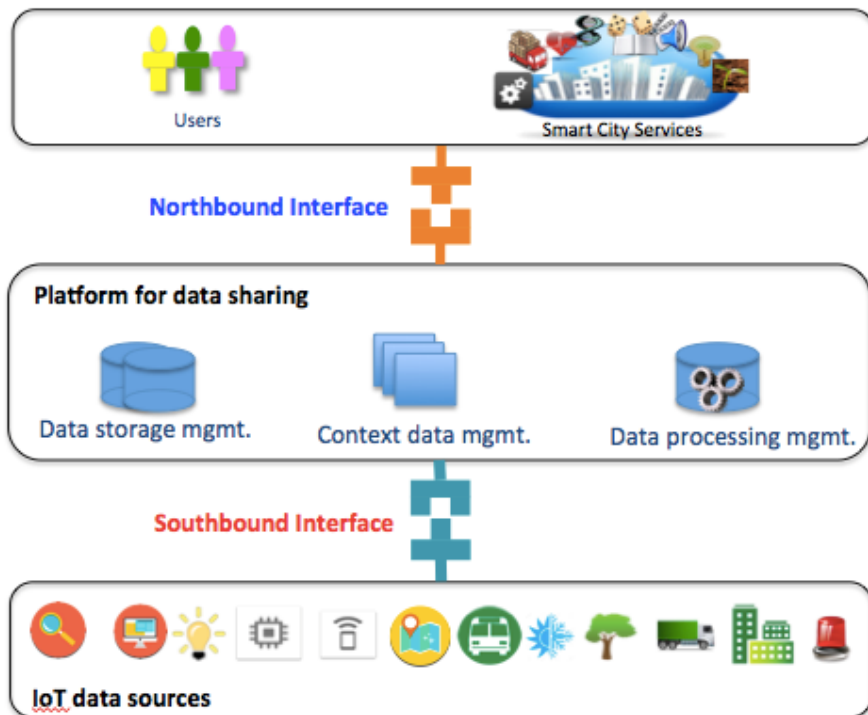


Figure 3. Simplified view of the Southbound and Northbound interfaces

2 Overview of IoT solutions deployed

Nowadays, the society develops their activities in a transformation of the context based on the exploitation of the available resources by the use of the Information and Communication Technologies (ICT). According to a publication from Ovance [2], whose study is based on a data collected by HIS Technology (business with presence in more than 31 countries), the number of smart cities will be most likely the quadruple during a period of approximately 12 years from 2013. Also, the majority of the world population is expected to live in urban areas and thus increasing the number of city inhabitants. Thereby, it will be necessary to manage a higher traffic density, different luminosity and connectivity resources, supply of water, gas, electricity and a whole list of public services.

This section intends to report the collection of the southbound information installed and used in the diverse smart city services running in the 8 RZs: Antwerp, Carouge, Eindhoven, Helsinki, Manchester, Milan, Porto and Santander. The collected data on installed sensors and used IoT technologies for their communication and integration in the cities is used to understand the current situation of the RZs and to find out commonalities. The following section describes the data collected, where the cities are listed by alphabetical order.

The collected data is as followings:

- IoT devices (e.g., sensors, back-end devices, etc.) deployed in the RZs for smart city services.
- Networking technologies providing connectivity to the IoT devices
- IoT data communication protocols being used by IoT devices for exchanging signals and messages including sensed data.
- Data format used in the data communication protocols.
- Southbound modules and/or APIs handling device management, context management, communication with its platform, data storage management if applicable.

It is noted that some information is not provided because the cities do not have detailed technical information on the installed system, while the cities receive processed data of the applications from the other organizations.

2.1 ANTWERP

Antwerp is a Flemish city in Belgium, and is the capital of the Antwerp province in the community of Flanders. With a population over 520,000, it is the most populous city proper in Belgium. Its metropolitan area houses around 1.2 million people.

This city implemented four different smart city applications, as shown in Table 1. The parking solution has been implemented with two different IoT devices supported via Sigfox and LoRaWAN communications, which use the REST API with JSON [8] data format.

In the Mobility application, the city cycling service is implemented by comprising with three device types: cycling push sensor, bike sharing stations and camera, relying on LoRaWAN and wired connectivity, respectively. Also, the data format is based on JSON.

As for Environment application, two models are used: fixed/mobile air quality sensors from IMEC [7] and mobile air quality sensor from Gypsy [41]. A Proximus LoRaWAN base station [9] is used for both models as a back-end proxy, providing LoRaWAN connectivity. Overall, LoRaWAN (+Sigfox & DASH7[42]) is used with JSON data format.

The Smart Waste solution accounts with two models: Bigbelly [10] and Enevo [11]. Bigbelly and TWS (Total Waste System) [12] are used as back-end support of IoT devices.

Table 1. IoT information of Antwerp Smart City applications

Application	Connectivity	Data comm. protocol	Data format
Parking	Sigfox	HTTP/REST	JSON
	Sigfox/LoRaWAN	HTTP/REST	JSON
Mobility	LoRaWAN	HTTP/REST	JSON
	GPS	-	n/a
	Ethernet	TCP/IP	n/a
Environment	LoRaWAN (+Sigfox & DASH7)	-	JSON
	LoRaWAN (+Sigfox & DASH7)	-	JSON
Smart Waste	GPRS	HTTP/REST	JSON
	GPRS	HTTP/REST	JSON

2.2 CAROUGE

Carouge is a municipality and a city in Switzerland. The city has a population around 22,000 in an area of 2.7 km². Currently, there are two different smart city applications deployed in the city.

On the one hand, there is a smart parking solution implemented in different parts of the city based on a device IEM Pres. The information collected from the devices is transferred to the platform through a LoRaWAN connection, where HTTPS is used as a data communication protocol and JSON as a data format. The sensing data is collected in a UDG (Universal Device Gateway) server [13], which provides intermediate services for the southbound integration, such as a device management and communication and also the connection to the Orion Context Broker that provides context data management. CKAN [25] and MongoDB [45] have been installed and provide data storage and direct data communication for the open data.

On the other hand, there is a noise monitoring solution in different points of the city with OrbiWise [14] sensors. LoRaWAN provides sensor connectivity and the sensing data collected by different sensors is transferred via HTTPS. UDG is also used in this application for providing interoperability with SynchroniCity platform, providing protocol adapting, device management and context management. Orion Context Broker [24] has been installed and CKAN and MongoDB provide data storage and communication services.

Table 2. IoT information of Carouge Smart City applications

Application	Connectivity	Data comm. protocol	Data format
Parking	LoRaWAN	HTTPS	JSON
Noise	LoRaWAN	HTTPS	JSON

2.3 EINDHOVEN

Eindhoven is a municipality and a city in the south of the Netherlands. The city's population reached 223,220 in January 2015.

In the city, there are three smart city solutions: parking, mobility and environment. The details on each service are describing below.

In the parking service, there is a solution supported by a proprietary device produced by Q-Park [43]. In fact, both backend IoT device management and device connectivity are based on Q-Park technologies.

In the mobility service, there are connected smart traffic lights and street lights. For the first one, connectivity is supported by ITS-G5 [15], whereas the second one is supported by 2G connectivity. Regarding the data communication protocols, CoAP, DTLS [47], DALI [48] and 1-10V [49] are used with Philips proprietary relational data models for data representation. The device management is performed through Philips City Touch [16].

Finally, in the environment solution, there is a smart air device (AIREAS [17] Airbox) with two versions: the first one is used for intelligent air quality sensors and the second one is measurement system. The connectivity is provided by GSM technology, and AIREAS API v2 is used as a data communication protocol with JSON data format.

Table 3. IoT information of Eindhoven Smart City applications

Application	Connectivity	Data comm. protocol	Data format
Parking	Proprietary via Q-Park	HTTPS	JSON
Mobility	ITS-G5	-	n/a
	2G	COAP, DTLS, DALI, 1-10V	Philips CT proprietary relational data models
Environment	GSM	AIREAS API v2	JSON

2.4 HELSINKI

The city of Helsinki is the capital and the largest city of Finland. Helsinki is the seat of the region of Uusimaa in southern Finland, on the shore of the Gulf of Finland. Helsinki has a population of 616,690 in an area of 184.5 km², making it the most populous city in Finland.

Currently, this city has deployed two IoT solutions for noise and environment monitoring. The noise detection service is based on Cesva TA120 [18], supported by 3G/4G connectivity using AIREAS API [17]. The data is sent in JSON format and the sensor communication is made by sending posts to a customized backend system, where data is transformed and proxied to Orion context broker [24] through NGSI API.

The environment application is based on a Vaisala AQ sensor, AQT420 [20]. It is noted that the Finnish Meteorological Institute manages and collects the sensing data and the city will receive the data through an open API from the Finish Meteorological Institute.

Table 4. IoT information of Helsinki Smart City applications

Application	Connectivity	Data comm. protocol	Data format
Noise	3G / 4G	HTTP	Sentilo [19] JSON

Environment	Sensor->gateway: serial, gateway->platform: wifi	Modbus (ASCII/CSV)	Proprietary
-------------	---	--------------------	-------------

2.5 MANCHESTER

The smart city solutions deployed in Manchester are Street Light, Transport, Environment, Culture, Health and Social Care. Unlike from the other RZs, Manchester is in a special program in which the city itself does not have any control in the southbound information such as which types of devices are installed with which networking connectivity technologies and which data communication protocols are used in the IoT infrastructure. The city only receives data from different IoT infrastructure owners and provides smart city services with the gathered data.

2.6 MILAN

Milan is the capital of Lombardy and the second most populous city in Italy after Rome, with the city having a population of 1.3 million people while its province-level municipality has a population of 3.2 million. Its continuously built-up urban area has a population estimated to be about 5,270,000 over 1,891 Km².

Milan has a large number of parking applications. Devices were distributed throughout this city, reporting a large number of parameters such as smart parking sensors for disabled drivers, smart parking sensors for electric vehicles and smart parking sensors to monitor no-parking areas. The sensors are connected with LoRaWAN / NB-IoT and the messages are exchanged using MQTT over TLS / SSL and CoAP over TLS / SSL. The data format to this extension is JSON and the communication is achieved through a REST API.

According to the mobility and traffic extension in Milan, there are a large number of solutions around the city. Some of them are metro status, metro station (parking status), Around me (point of interest), BikeMe (bike sharing station-bases service), bus/tram waiting time, local train circulation status, flight time table and parking availability of Linate [21] and Malpensa [22] Airports and, Bergamo airport [23] flights timetables, airport and train network status. The majority of them use SOAP Web services, except Bergamo airport flight timetable, Linate and Malpensa airport flights timetable and parking availability, whose communication interface are supported by a REST API.

Regarding the traffic solution, there are four different solutions implemented around the city. One of them is the implementation of loops through cable / GSM / Wi-Fi connection; the second solution is based on the deployment of fixed cameras with SDH [50] / Wi-Fi; the third one is the deployment of mobile cameras around the city transmitting data over 3G / 4G links; the last solution is based on local police agents. Various types of loops exist, to measure traffic flow, speed and car dimensions.

Regarding the environment solution, three different solutions are deployed around the city: one of them is the lamp post weather, another one is the cameras Low Emission Zone (LEZ) with 3G / 4G connection, and the last solution is based on the deployment of devices with capacity to detect the temperature, humidity, pressure, air pollution and UV radiation.

There is an IoT solution for a cultural service that is available in Android and iOS operating system using 3G / 4G networks as well, and in energy management heating system which is made of temperature sensors with GSM / GPRS connectivity.

Milan has a safety alarm system by deploying volumetric sensors, cameras, motion detectors, door alarms, fire detectors and smoke detectors. All of them are using Wi-Fi connectivity and proprietary control units as backend IoT proxies.

Finally, the security solution comprises security cameras, mobile GPS and radio, and vehicles' GPS and radio. LoRaWAN provides connectivity to the security cameras, while the communication of the stations to stations is supported by wired networks.

Table 5. IoT information of Milan Smart City applications

Application	Connectivity	Data comm. protocol	Data format
Parking	LoRaWAN / NB-IoT	LoRaWAN: MQTT+TLS/SSL NB-IoT: CoAP+TLS/SSL	JSON
	LoRaWAN / NB-IoT	LoRaWAN: MQTT+TLS/SSL NB-IoT: CoAP+TLS/SSL	JSON
	LoRaWAN / NB-IoT	LoRaWAN: MQTT+TLS/SSL NB-IoT: CoAP+TLS/SSL	JSON
Mobility	-	-	XML
Traffic	Cable/GSM/SDH	proprietary protocol	n/a
	SDH/Wi-Fi	proprietary protocol	n/a
	3G/4G	proprietary protocol	n/a
	-	proprietary protocol	Italian standard incidents report protocol
Environment	-	-	JSON /XML
	3G / 4G	proprietary protocol	n/a
Energy Management	from control unit to central server: GSM/GPRS	proprietary protocol	n/a
	from control unit to central server: GSM/GPRS	proprietary protocol	n/a
Safety Alarms	Wi-Fi	proprietary protocol	n/a
Security	LoRaWAN	proprietary protocol	n/a
	communication station-to-station wired (SDS-SCN); AVL; GRI Protocol	proprietary protocol (Tetra [38]- GRI protocol[37])	proprietary protocol (Tetra - GRI protocol)
	communication station-to-station wired (SDS-SCN); AVL; GRI Protocol	proprietary protocol (Tetra - GRI reporting protocol)	proprietary protocol (Tetra - GRI protocol)
Culture	3G/4G	-	JSON

2.7 PORTO

The city of Porto is the second largest city in Portugal, and the centre of the economic development of the Northern region of the country. Porto has more than 210,000 inhabitants and it's the centre of a large metropolitan area with around 1.8 million inhabitants and an area of 2,040 km², which makes it one of the major urban areas in Southern Europe.

Nowadays, the city of Porto has five different smart solutions, in the areas of mobility, environment, tourism and public transportation, traffic and water metering. The details about each solution are described below.

The environmental monitoring sensors are supplied by Citibrain/Monitar, and comprise three different stations: SmartNOISESense (noise), SmartMETEOSense (meteorological parameters) and SmartAIRSense (air pollution). They are prepared to collect information in a data format of JSON about the levels of noise, particles (PM2.5, PM10), gas concentration (O_x, NO₂, CO), air (temperature, relative humidity), atmospheric pressure, wind (direction, speed), rainfall, UV and solar radiation. According to connectivity, all of them are integrated via HTTP, CoAP and MQTT.

Regarding the tourism and public transportation solution, the models are Accent Systems iBKS Plus [26] (active) and SMARTRAC BullsEye NFC [27] (passive). These solutions are implemented to provide contextualized information in the areas of public transportation and tourism to inhabitants, visitors and tourists. The communication media are Bluetooth® Low Energy (BLE) for the active beacons and NFC for the passive beacons, which use the iBeacon™ [28] and Eddystone™ [29] protocols (active beacons), and the ISO/IEC14443 [30] type A and NFC Forum type 2 protocols [31] (passive beacons).

The third solution comprises wireless water meters and wireless data concentrators, which are used to measure the amount of water supply consumption at the client's premises and to aggregate data from several meters, respectively. The models are Sappel (Diehl Metering) IZAR CP R3.5 [32] and Sappel (Diehl Metering) Izar Receiver GPRS/LAN [33].

Another smart solution is related with mobility and traffic. In this case, the solution comprises vehicle counters and video cameras implemented around the city, which collect information in real time about the general traffic status through a TXT data format and video format, respectively. Also, the communication protocols to this mobility solution are proprietary protocols.

Table 6. IoT information of Porto Smart City applications

Application	Connectivity	Data comm. protocol	Data format
Mobility	Copper wires	-	TXT
	Ethernet	ONVIF [35]	n/a
Environment	Wi-Fi	HTTP	JSON
	Wi-Fi	CoAP	JSON
	Wi-Fi	MQTT	JSON
Tourism and transportation	Bluetooth® Low Energy (BLE)	iBeacon™ and Eddystone™	TXT
	NFC	ISO/IEC14443 type A, NFC Forum type 2	TXT

Water Metering	Wireless M-Bus @ 868,95 MHz	Proprietary (PRIOS Protocol)	PRIOS
	Wireless M-Bus @ 868,95 MHz (downstream) GSM/GPRS, TCP/IP (upstream)	Proprietary (PRIOS Protocol)	XML-Enc (encrypted)

2.8 SANTANDER

The city of Santander is the capital of the autonomous community and historical region of Cantabria situated on the north coast of Spain. Located east of Gijon and west of Bilbao, the city has a population about 178,000 (2013) in an area of 35 km².

The city of Santander has seven different applications, and the details about each solution are described below.

The parking extension provides real-time parking information through the sensors deployed around the city. Meshlium (Libelium [34]) using 802.15.4/3g/Wi-Fi/Ethernet is used as backend IoT proxy. REST API, MQTT and WebSocket are used for data communication protocols, which are transmitted in a UL2.0 format.

The noise sensors are responsible for the measurement of noise levels in the city through the implementation of Libelium [34] devices with similar communication protocols of the parking extension.

In the street light application, the models for this application are use the Meshlium (Libelium) presence and luminosity nodes. Meshlium (Libelium) is chosen as a back-end proxy with 802.15.4 / 3G / Wi-Fi / Ethernet connectivity. Sensor data is transmitted through REST API, MQTT and WebSocket, which transmit the data in a UL2.0 format to communicate with context data management module.

For mobility and traffic, Meshlium (Libelium) vehicle counter sensors and Meshlium (Libelium) vehicle speed sensors are deployed. Meshlium (Libelium), whose connectivity is based on 802.15.4 / 3G / Wi-Fi / Ethernet, is used as a backend IoT proxy system. REST API, MQTT and WebSocket are used as data communication protocols, which transmit the data in a UL2.0 format to communicate with context data management module.

In the environment application, there are Meshlium Noise (Libelium) sensors, Meshlium Luminosity (Libelium) sensors, Meshlium CO (Libelium) sensors, Meshlium Temperature (Libelium) sensors, Smart Spot (environmental devices), GPRS module and MOBILE node. 802.15.4 / 3G / Wi-Fi / Ethernet are providing the connectivity to Libelium devices. For the smart spot air quality service, Wi-Fi and M2M connectivity is provided, and for the mobile nodes, GPRS is used. According to the data communication protocol, all the Libelium devices are supported via REST API, MQTT and WebSocket. The Smart Spot communications are supported by LwM2M via FIWARE and connected to the Orion Context Broker.

Another application in Santander is smart gardens, where Libelium devices (irrigation sensor and agriculture sensor) are deployed, connected via 802.15.4 / 3G / Wi-Fi / Ethernet. REST API, MQTT and WebSocket are used as data communication protocols.

Finally, for a citizen service, the city has a mobile application, with a version for Android and iOS, using the typical available connectivity technologies on these devices (Wi-Fi, 3G, 4G and Bluetooth Low Energy).

In this city, the Libelium (Spanish company) is dominating in the deployment of IoT solutions due to the fact that it participated in the Smart Santander project [46]. Santander is considered as one of the most successful cases in the deployment of IoT smart city solutions. It has implemented different solutions in different contexts.

Table 7. IoT information of Santander Smart City applications

Application	Connectivity	Data comm. protocol	Data format
Parking	802.15.4 / 3G / Wi-Fi / Ethernet	REST API, MQTT, WebSocket	UL2.0
Noise	-	-	-
Street Light	802.15.4 / 3G / Wi-Fi / Ethernet	REST API, MQTT, WebSocket	UL2.0
Mobility	802.15.4 / 3G / Wi-Fi / Ethernet	REST API, MQTT, WebSocket	UL2.0
	802.15.4 / 3G / Wi-Fi / Ethernet	REST API, MQTT, WebSocket	UL2.0
Environment	Wi-Fi, M2M	LwM2M	JSON, TEXT, TLV, Packet
	802.15.4 / 3G / Wi-Fi / Ethernet	REST API, MQTT, WebSocket	UL2.0
	802.15.4 / 3G / Wi-Fi / Ethernet	REST API, MQTT, WebSocket	UL2.0
	802.15.4 / 3G / Wi-Fi / Ethernet	REST API, MQTT, WebSocket	UL2.0
	802.15.4 / 3G / Wi-Fi / Ethernet	REST API, MQTT, WebSocket	UL2.0
	GPRS	REST API, MQTT, WebSocket	UL2.0
Smart Gardens	GPRS	REST API, MQTT, WebSocket	UL2.0
	802.15.4 / 3G / Wi-Fi / Ethernet	REST API, MQTT, WebSocket	UL2.0
	802.15.4 / 3G / Wi-Fi / Ethernet	REST API, MQTT, WebSocket	UL2.0
Citizen	Wi-Fi, 3G, 4G, Bluetooth LE	-	n/a

3 Comparative analysis of IoT integration models

The SynchroniCity project fosters and welcomes diversity among the RZs, not only cultural but also technological, given that for similar problems each one of them could have and apply different solutions. One of the project goals consists in harmonizing these diverse approaches and look for a way to converge.

Having this aim in mind, each RZ performed a thorough exercise to characterize their own IoT infrastructure to better understand which IoT devices are employed in each application. Some points considered as critical were accounted for, and as a result, the collected data is presented in this section.

As a first noticeable factor, it is worth mentioning that up to 13 applications were distinguished with a staggering number of 92 IoT devices, covering their needs. The analysis has been made by digging into the different solutions given in each RZ to each one of those applications, which could be considered as city challenges.

3.1 Smart Parking

One of the most pressing subjects, addressed in up to 6 RZs involved in the project, is the one related to the availability and management of parking spots in their urban environment. So, various parking sensors are being currently employed, with a third part of them employing a proprietary server or gateway for IoT devices. Most of them rely on a LoRa / LoRaWAN connectivity (Figure 4) and a REST or MQTT data communication protocol (Figure 5). It is worth highlighting Milan's case, where the parking issue receives a particular treatment depending whether the area to control is a restricted-parking place, some electric vehicles parking spots are available or even there is a zone for disabled people to park their cars. As for the data formats employed, the trend implies using JSON as the preferred one, as depicted in Figure 6, and each RZ uses different IoT device management systems shown in Figure 7. However, the majority of context data management solutions in the southbound interface are not clear as for today, although some RZs are already employing Orion Context Broker and REST API (Figure 8).

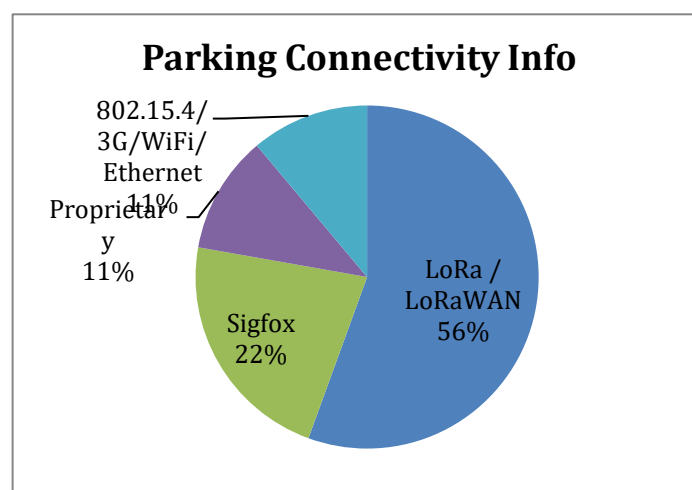


Figure 4. Parking - Connection Info

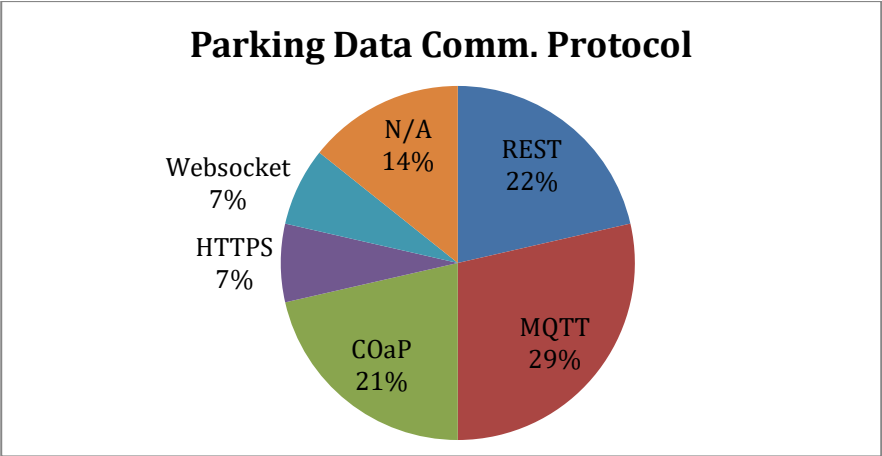


Figure 5. Parking – Data Communication Protocols

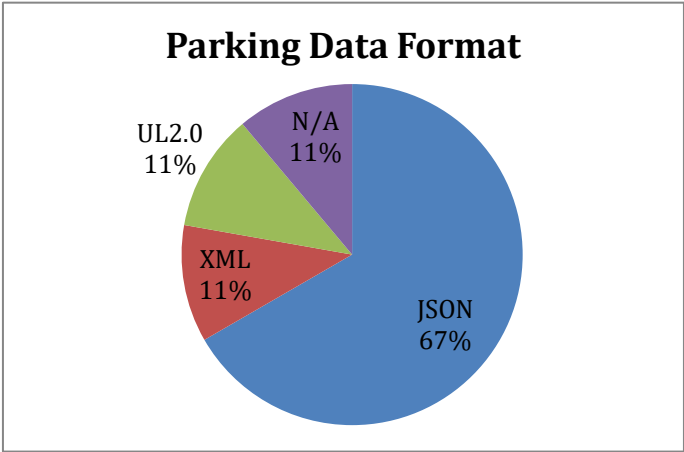


Figure 6. Parking – Data Format

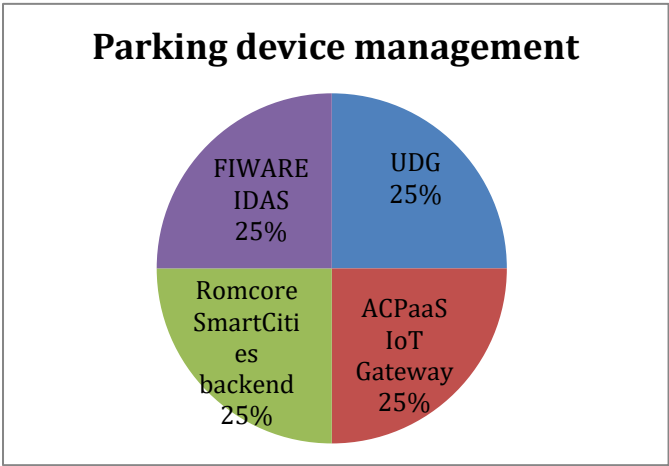


Figure 7. Parking – IoT Device Management

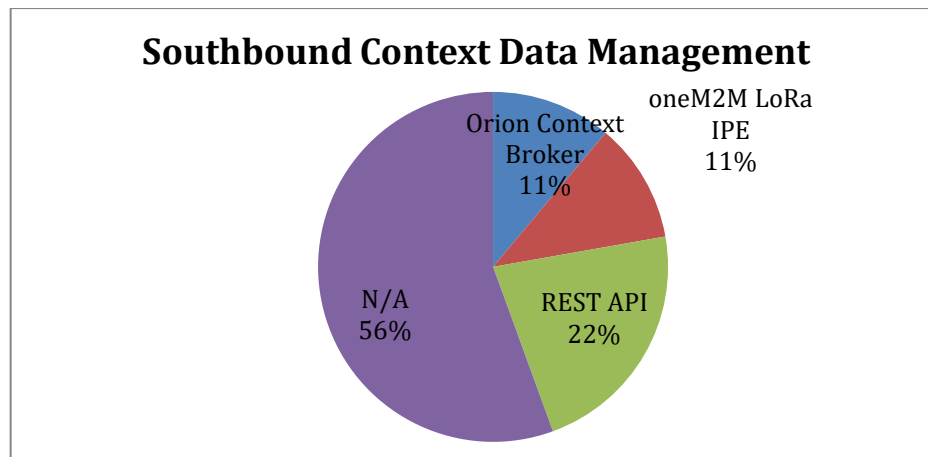


Figure 8. Parking – Context Data Management

3.2 Smart Noise control

Next in line the problem of noise appears, being considered as a hot topic by up to three RZs. These RZs provide different inputs on the networking protocols (Figure 9), while the data communication protocols and formats are similar (Figure 10 and Figure 11), with different security measures applied. One RZ describes its device management system but two don't (Figure 12). What is somehow achieved is a common context data management, where Orion Context Broker can be employed by the three cases if needed (Figure 13).

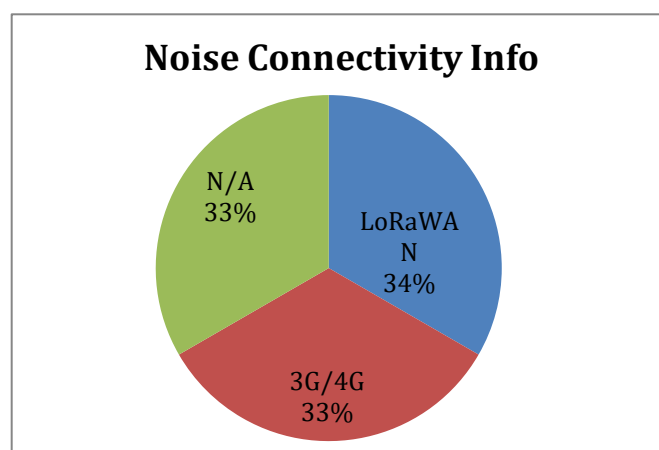


Figure 9. Noise – Connection Info

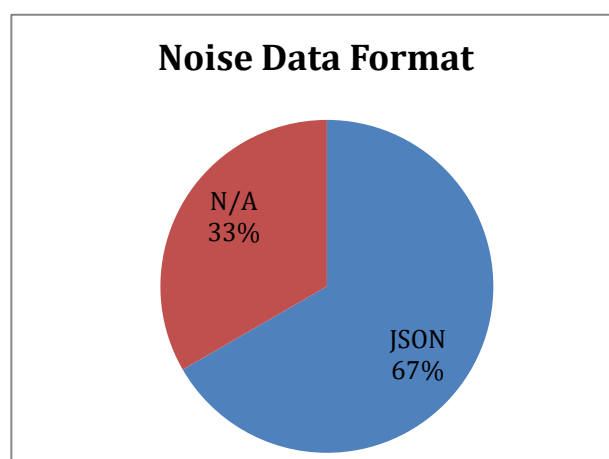


Figure 10. Noise – Data Format

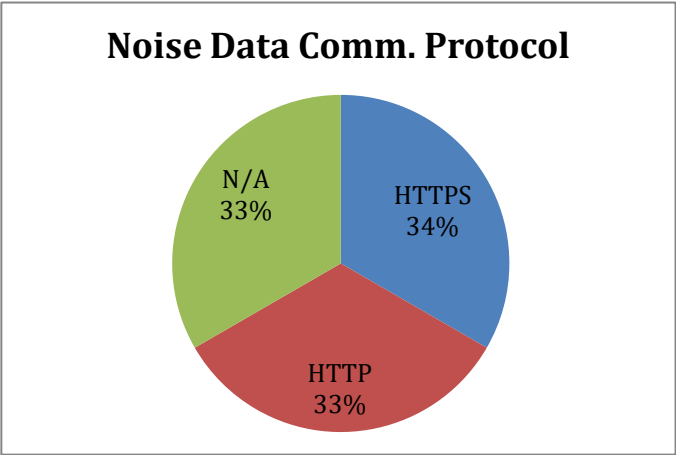


Figure 11. Noise – Data Communication Protocol

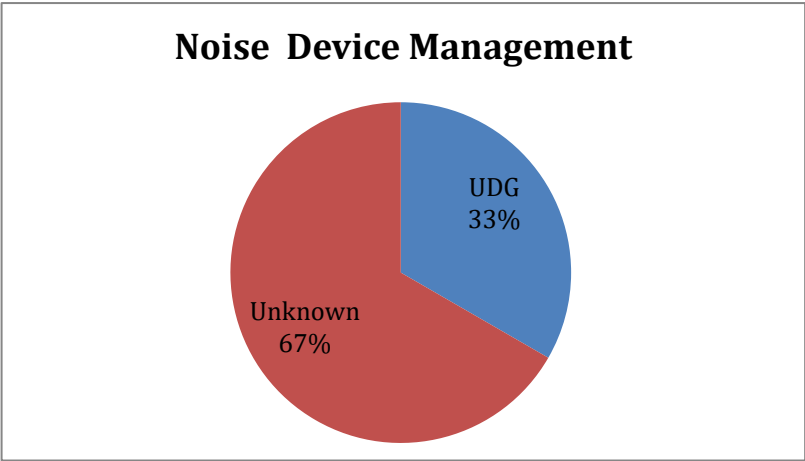


Figure 12. Noise – IoT Device Management

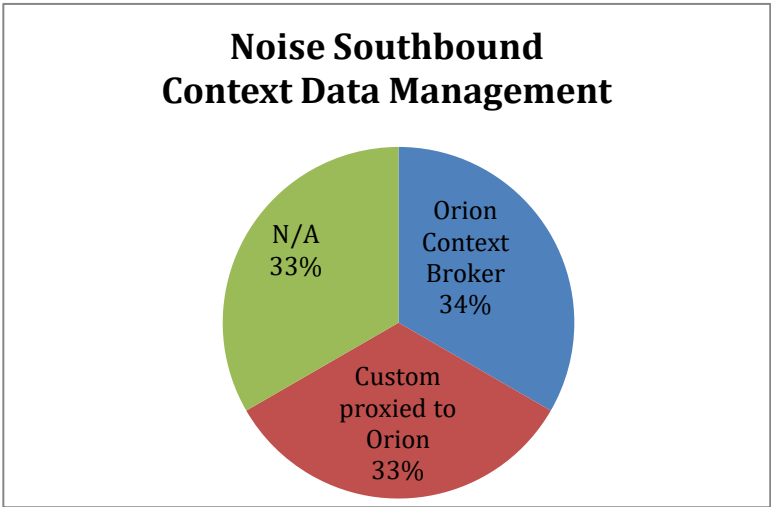


Figure 13. Noise – Southbound Context Data Management

3.3 Smart Street Light

On the other hand, the situation concerning the street light concept can be found. Not much data is available with regard to this topic, but in any case truth is the actuation when presence is detected and luminosity dimming is getting more and more important in the smart city context.

Three RZs run this application, and there are diverse network connecting technologies used in this application as shown in Figure 14, but not so much information has been collected on the data format as shown in Figure 15.

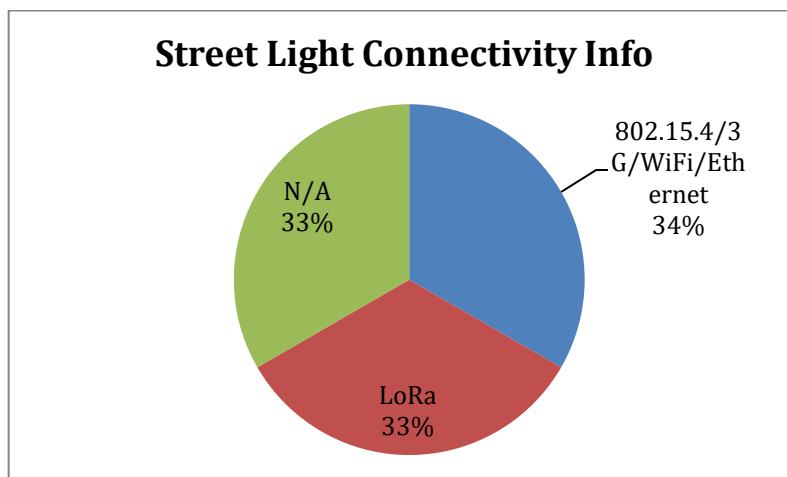


Figure 14. Street Light – Connection Info

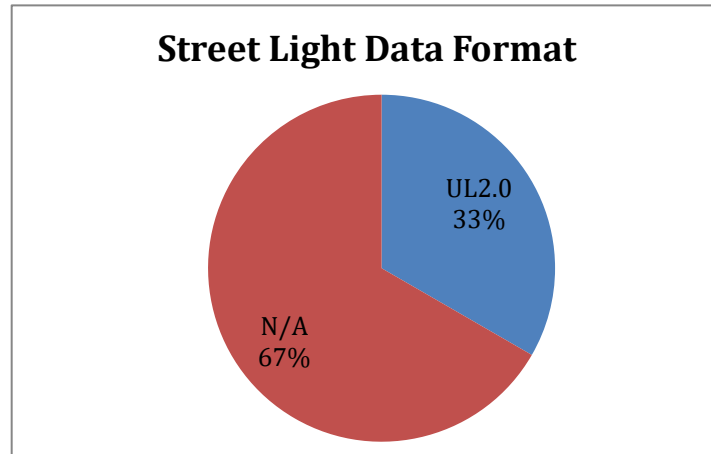


Figure 15. Street Light – Data Format

The same situation applies to the case of the data communication protocols (Figure 16), IoT device management (Figure 17) and the context data management, where no clear option is preferred (Figure 18).

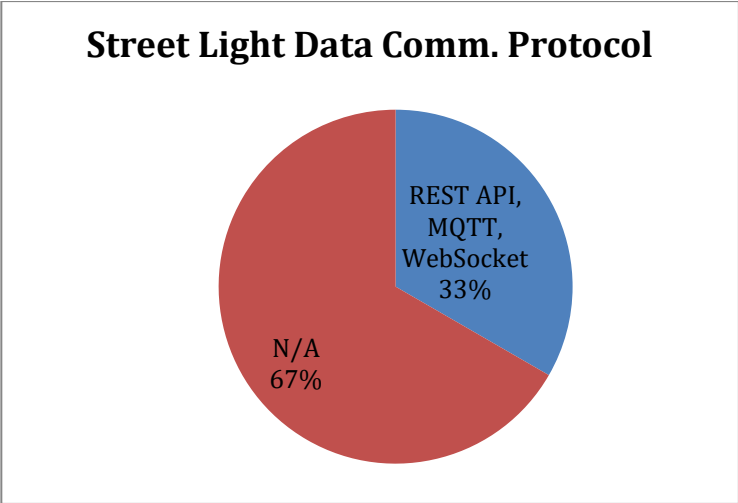


Figure 16. Street Light – Data Communication Protocol

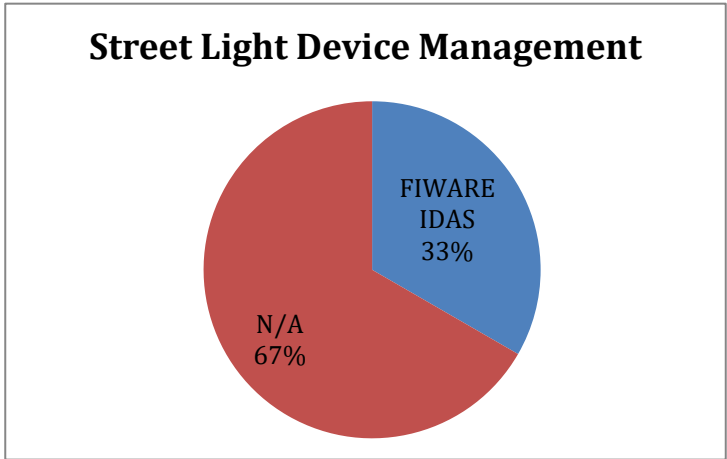


Figure 17. Street Light – IoT Device Management

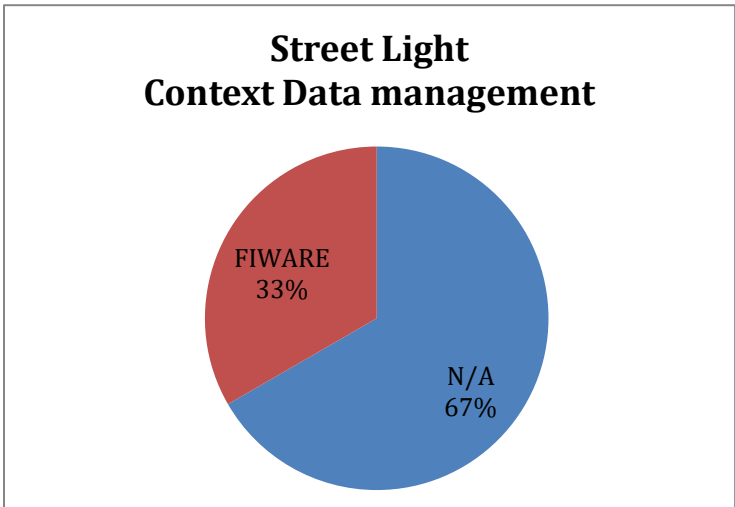


Figure 18. Street Light – Context Data Management

3.4 Smart Mobility and Traffic

Another hot topic is the one related to what has been considered as a mix of mobility and traffic. Almost every RZ involved in the project shows its desire to improve the traffic conditions, especially in their downtowns in order to avoid the trouble derived from congestion and traffic jams. At the current status, six applications are provided across five RZs. The approaches vary from the strict road controls, checking for the number of cars passing by and/or their speeds, to the encouragement to use another means of transportation apart from the car, such as bikes, buses or metro. Regarding this application, it is not easy to find a common ground, since several solutions rely on proprietary solutions that employ different connections and data formats.

The details are shown below in a scope of the connectivity info (Figure 19), the preferred data formats (Figure 20), the data communication protocol (Figure 21), IoT device management (Figure 22) and the context data management (Figure 23).

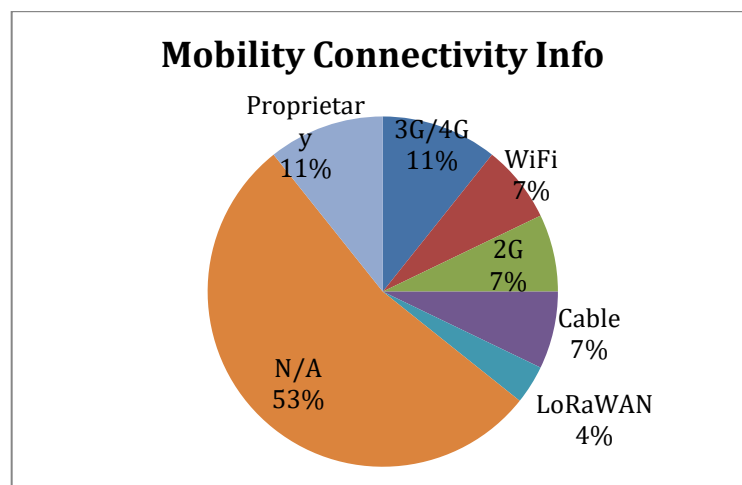


Figure 19. Mobility – Connection Info

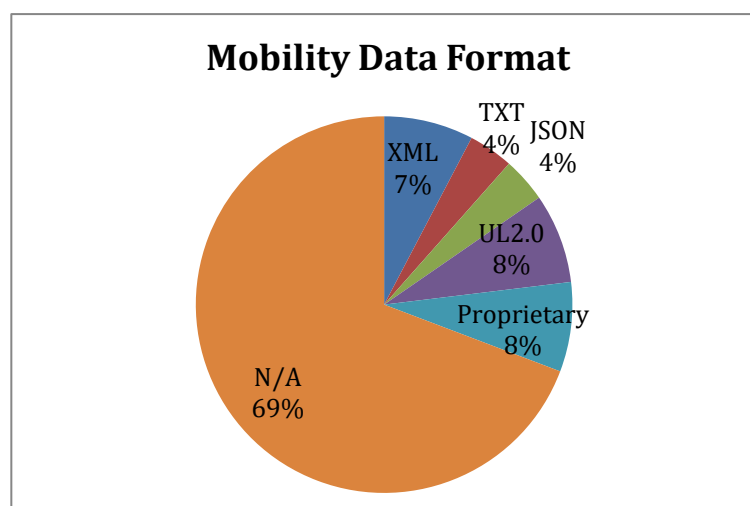


Figure 20. Mobility – Data Format

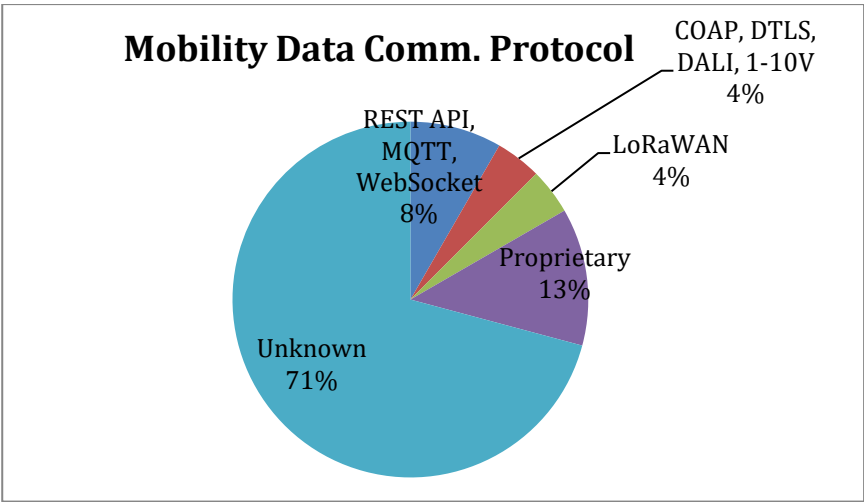


Figure 21. Mobility – Data Communication Protocol

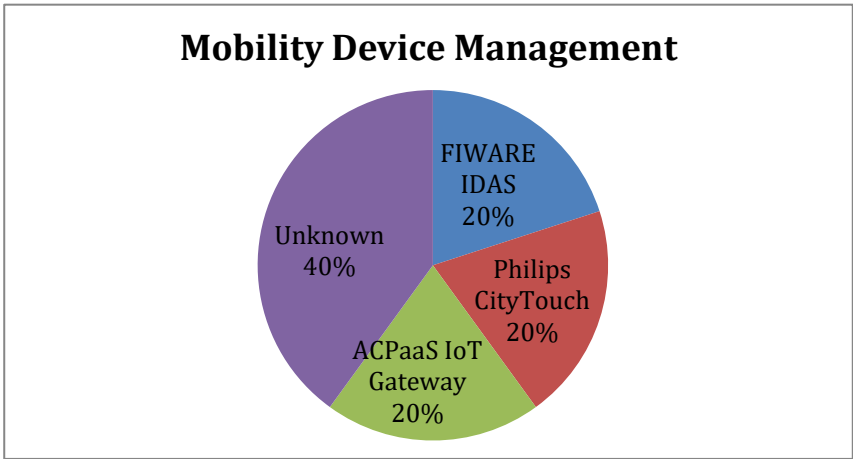


Figure 22. Mobility – IoT Device Management

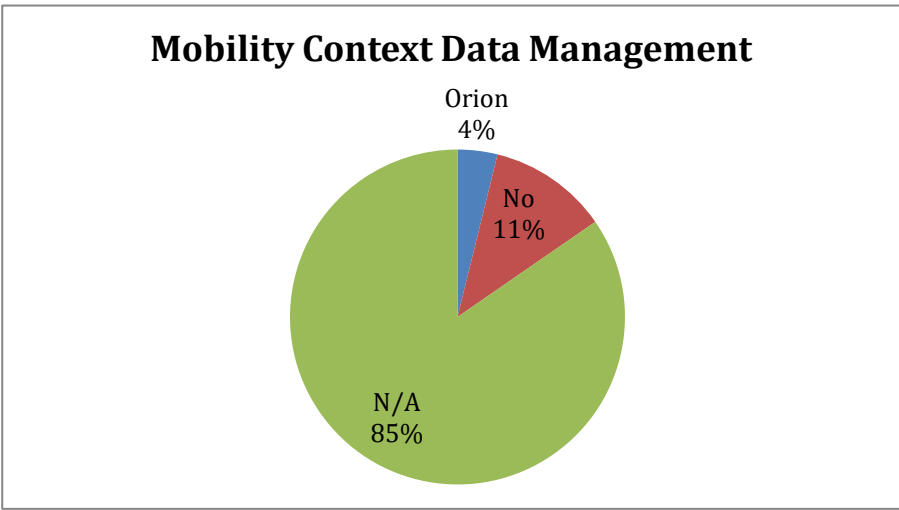


Figure 23. Mobility – Context Data Management

3.5 Culture

To assist not only locals but also tourists in getting to know the city, a cultural approach is followed by two RZs, using mainly beacons to that extent, but not enough valid information is available with regard to the data communication protocols among devices (Figure 24) data format (Figure 25) and REST API for the southbound communication are used, allowing for the interaction with a context broker (Figure 27). IoT device management is provided directly from the platform (Figure 28).

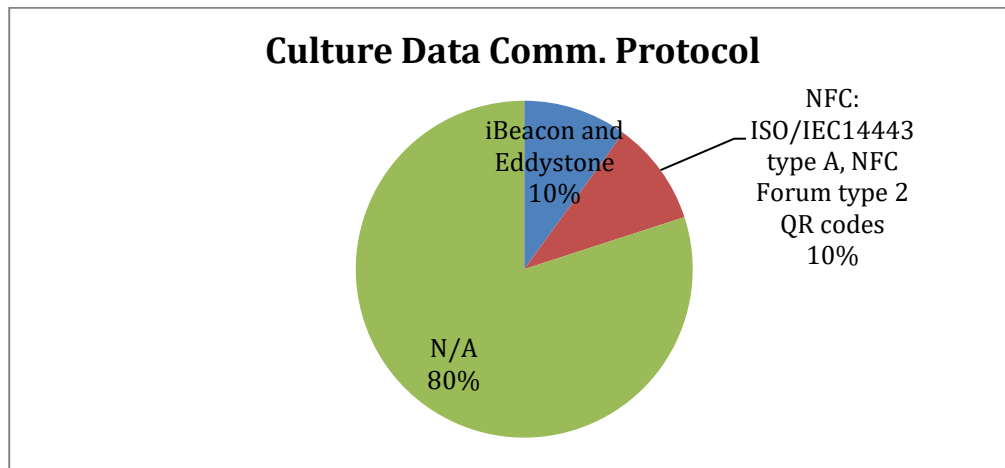


Figure 24. Culture – Data Communication Protocol

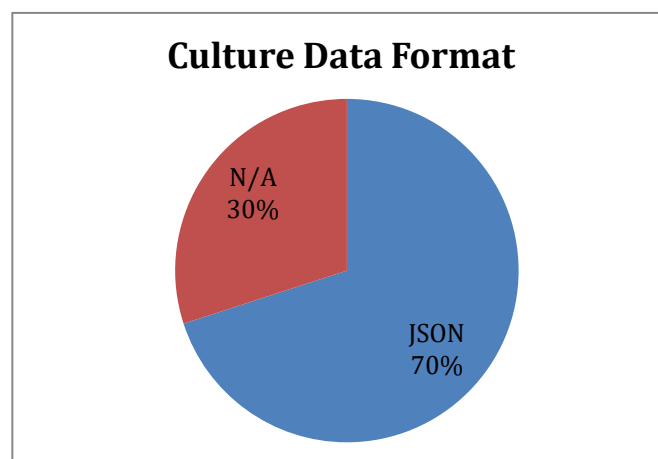


Figure 25. Culture – Data Format

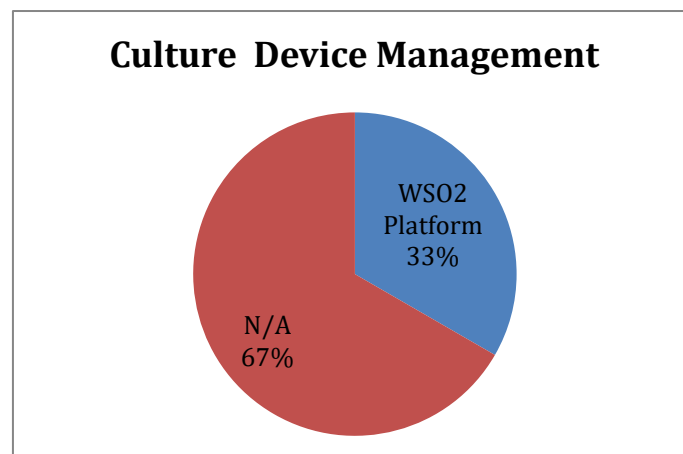


Figure 26. Culture – IoT Device Management

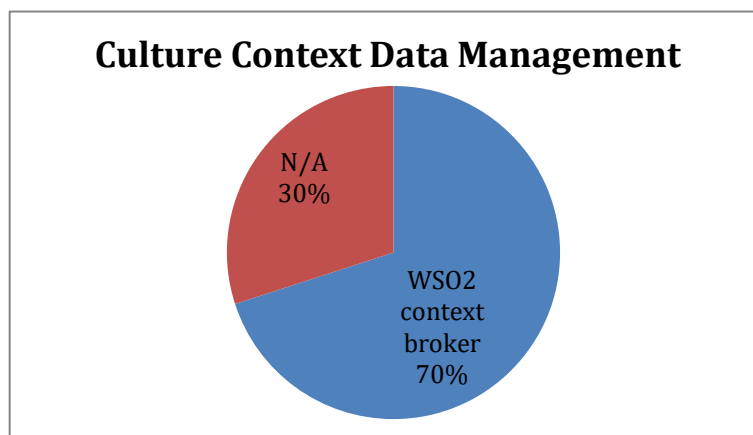


Figure 27. Culture – Context Data Management

3.6 Citizen information

Two different RZs put the focus on their citizens, counting on their own mobile phones to trigger actions through a mobile app available in mainly both Android and iOS, as shown in Figure 28. Connectivity is provided by cellular networks (3G/4G) that typically are available at those devices, as depicted in Figure 29.

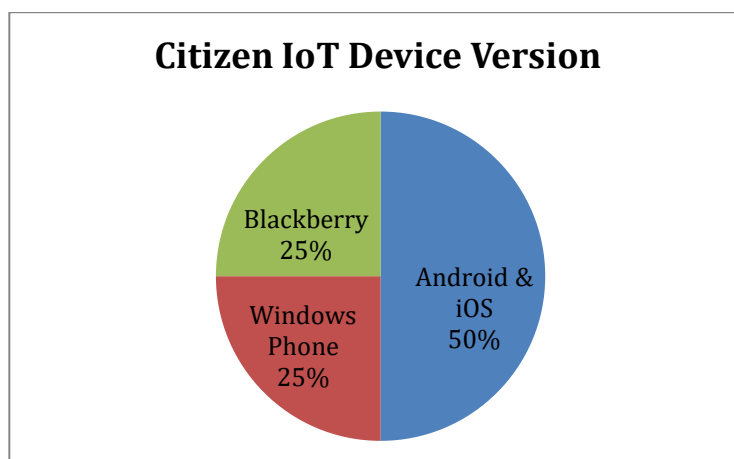


Figure 28. Citizen – IoT Device Version

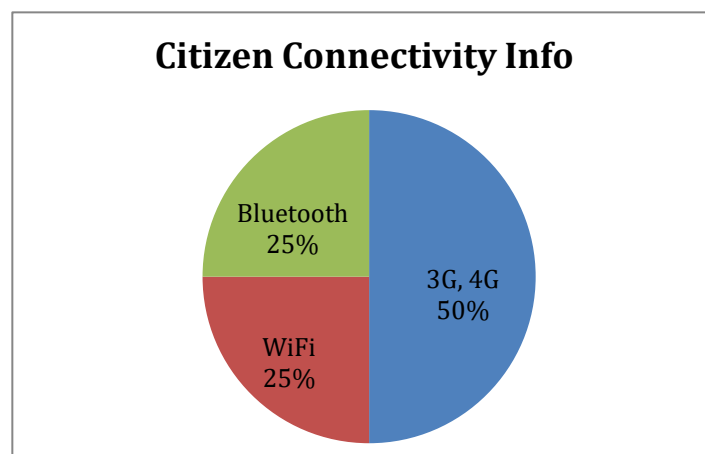


Figure 29. Citizen – Connection Info

3.7 Environment data

Seven RZs collect environment data through their smart city application. For the environment data capturing, as summarized in Figure 30, the data communication protocols vary, with REST / MQTT / Websocket as popular choices but also missing information and proprietary protocols for almost half of the available data. Contrary to the diverse choice of IoT device, the data formats are much more open with JSON and UL 2.0 accounting for almost 70%, as summarized in Figure 31.

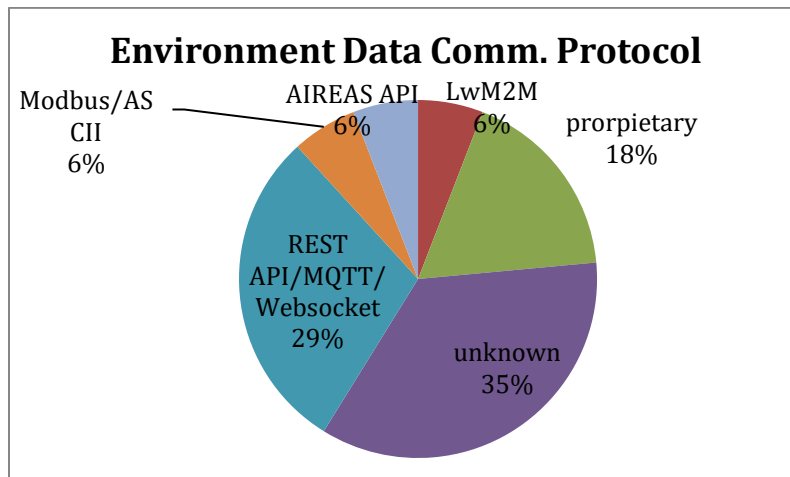


Figure 30. Environment – Data Communication Protocol

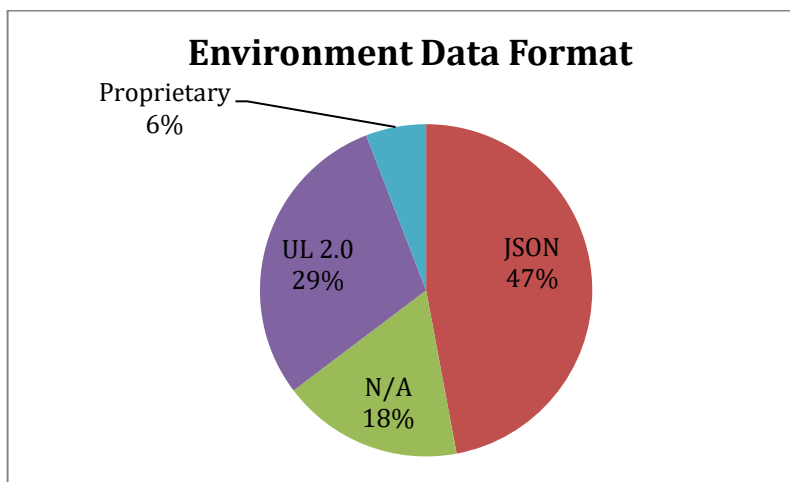


Figure 31. Environment – Data Format

RZ's platforms provide customized IoT device management (Figure 32), and for environment context data management, only one RZ managed the WSO2 context broker (Figure 33), the missing information can be indicated the need for further harmonization within the project.

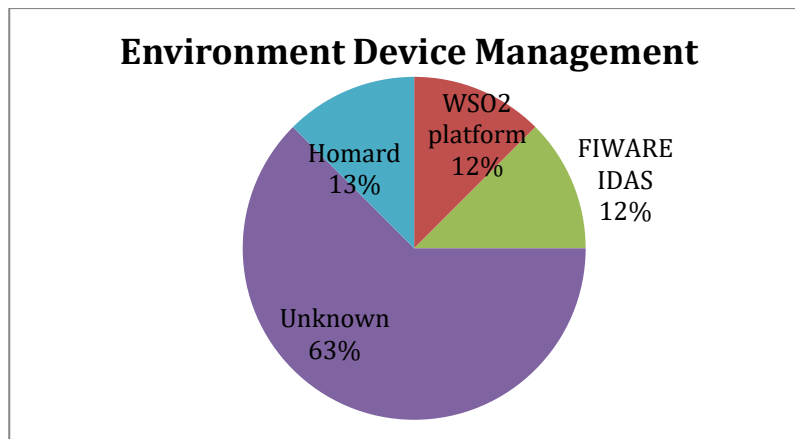


Figure 32. Environment – IoT device management

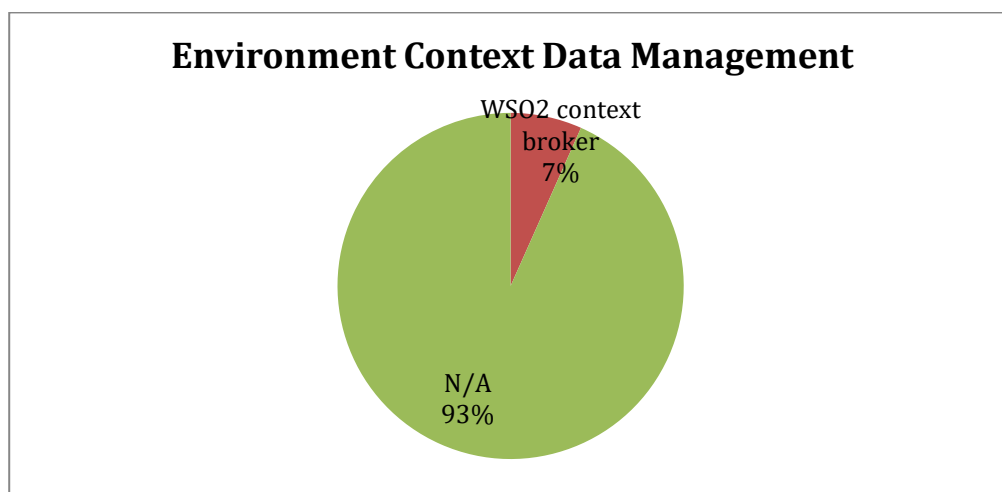


Figure 33. Environment – Context Data Management

3.8 Energy management

This application is only implemented in Milan. The energy management situation is being handled through a dedicated Control Units as Backend proxy for the IoT devices, relying on a proprietary data communication protocol and using cellular connections (GSM / GPRS) from their control units to their central servers.

3.9 Safety Alarms

Safety Alarms is also implemented only in Milan managed by local polices. Again the solution is depending on proprietary data communication protocols and Wi-Fi is used for network connectivity. An Italian solution, Axitea [6] is used for IoT device management.

3.10 Security

Milan conducts some activities in the field of security as well. To this end, three different IoT devices are being employed, relying in two kinds of backend proxy for IoT devices to provide network connection between sensors and control units (one with radio communication server, and one with switch). GSM / GPRS are used in network connection and all of them use a proprietary data

communication protocol (Tetra [38] - GRI Protocol [37]) and two of them using Tetra as their data format.

3.11 Observation

Some RZs are in the midst of deploying modern wireless water meters but scarce data is available regarding their particular characteristics. Often only the device model name and a proprietary communication protocol are described (e.g. PRIOS protocol [36]).

Even though the smart waste initiative reaches every major city around the world, only one of the RZs present a solid application, using a GPRS connection and a REST API, as well as a REST API southbound communication and a promise to employ an Orion Context Broker instance as its context data management tool. The same goes for the smart gardens case, where one of the RZs presents interesting initiatives, relying on a 3G / GPRS communication and the chance of employing REST API, MQTT or WebSockets as data communication protocols.

Two different RZs put the focus on their citizens, counting on their own mobile phones to trigger actions through a mobile app available in both Android and iOS operating systems, and employing as connection one of the options typically at hand at those devices (cellular and Wi-Fi communications). The latest stretch of applications relates to initiatives taking place in Milan. First, the energy management situation is being handled through two different models of IoT devices, relying on a proprietary data communication protocol and using cellular connections from their control units to their central servers. The safety alarms solution also depends on proprietary data communication protocols and this time using Wi-Fi as the connection technology. Last, Milan conducts some activity in the field of security. To this end, three different IoT devices are being employed. All of them use a proprietary data communication protocol and at least two of them use Tetra.

4 Generic integration models and Guidelines

As SynchroniCity aims to be platform-agnostic, it is important to make the framework to be open to all platforms that RZs are using. At the same time, it is also important to have common APIs for the integration. Figure 34 illustrates the principle requirements on the southbound interoperability: support of diverse networking connectivity, support of heterogeneous IoT data communication protocols, seamless device management, provision of context management and common data formats to the platform.

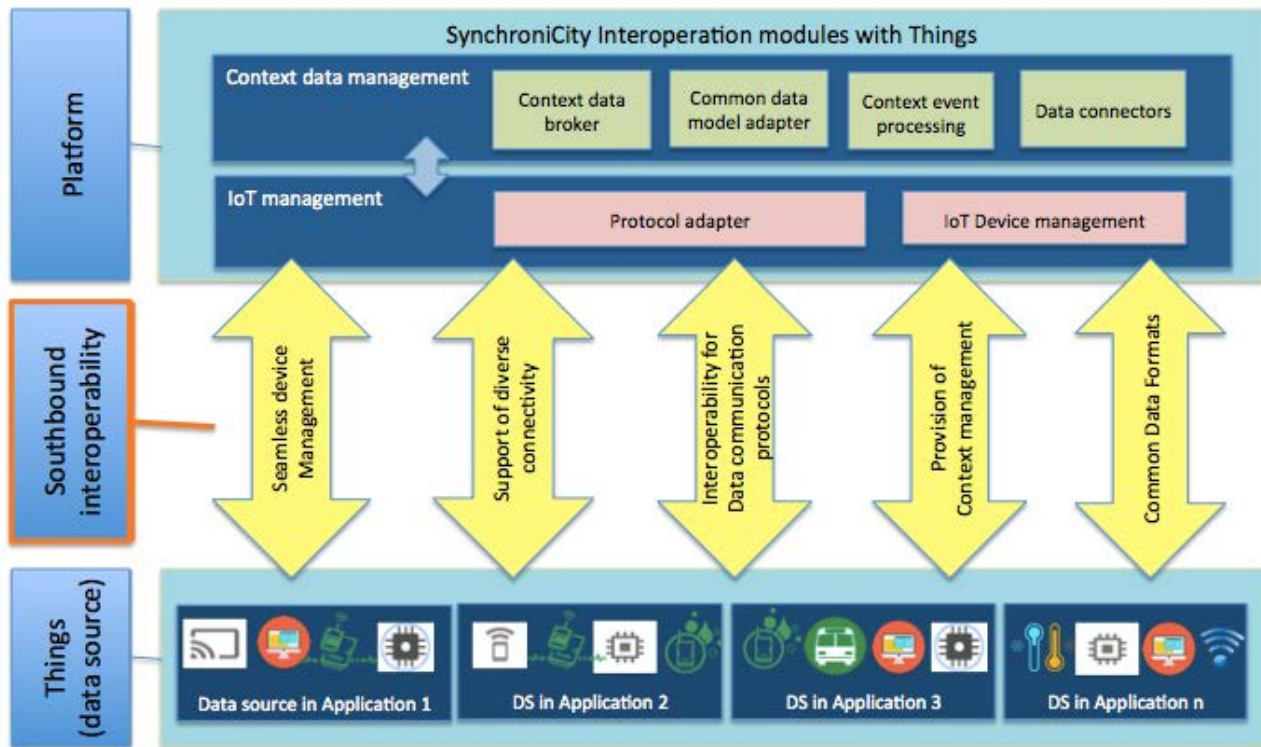


Figure 34. Southbound interoperability requirements

As shown in the survey results explained in the Section 2 and 3, RZs use various network connectivity technologies depending on the needs and available device solutions in the markets. Also the choice of data communication protocols is diverse depending on the device and platform choices. These two points are not to be modified in the scope of SynchroniCity, as there should be no limits to the use of the technologies that are deployed and working.

This section describes integration models for each RZ that use different platforms taking into account the current situation described in the Section 2 and 3. It also provides implementation guidelines for the common API to be used to communicate with the Context Data Management Module.

4.1 Integration models

Eventually, SynchroniCity should support the needs of the broader cities than the needs of the current RZs, but as a starting point, the integration models are designed taking into account of the IoT platforms that the RZs are using as following:

- DYAMAND [39] (Antwerp)
- UDG [13] (Carouge)
- FIWARE [4] (Eindhoven, Porto and Santander)

- Helsinki Urban platform (Helsinki)
- WSO2 [40] (Milan)
- oneM2M [5] (Seongnam)

It is noted that Manchester is not included here as the city does not have any possession and control of IoT devices and corresponding southbound functionalities as it is mentioned in the Section 2. The city only controls data that has been transferred to the city's platform from the diverse sources deployed in the city via diverse national projects.

As we have seen in the Section 3, there are diverse physical devices installed in RZs using different networking technologies and data communication protocols in different applications. Some includes legacy systems and protocols. Diverse networking (connectivity) technologies among devices are being used (e.g., LoRaWAN, IEEE 802.15.4, 3G, Wi-Fi, etc.) for different applications. The device manufacturer provides proxies to connect sensing devices to the network infrastructure or a separate gateway is used to handle both the network connectivity and data communication. The survey results also show diverse data communication protocols are used such as MQTT, REST API, CoAP, HTTP / HTTPS, and Websocket. The interoperability of the heterogeneous IoT protocols is often solved by intermediate systems such as proxies or gateways. Although the widely used smart city platforms such as FIWARE or oneM2M provide heterogeneous IoT device integration, the set of supporting protocols directly from the platforms are limited and the users should build customized proxies, gateways or adapters for industry specified protocols which are not supported by the platform. In SynchroniCity, most of the RZs built gateways or proxies for IoT integration, and Antwerp uses imec's DYAMAND and Carouge uses UDG (Universal Device Gateway), which are dedicated for heterogeneous protocol integrations covering multiple application domains with a minimal set of platform services.

Considering the different system configuration in the RZs, the southbound integration model must consider no reinvention and provide the integration models and guidelines to minimize the overhead created by the effort to be compliant with the SynchroniCity framework.

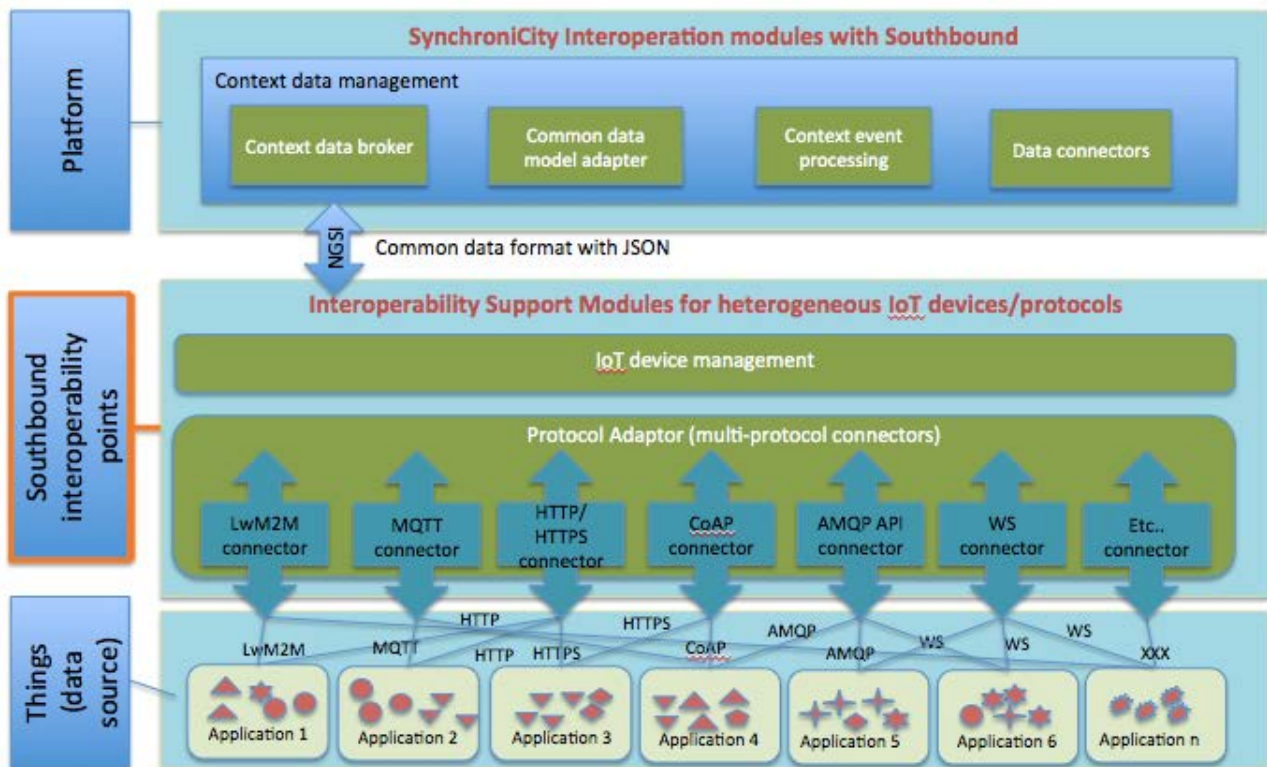


Figure 35. Support of diverse IoT protocols

There is no doubt that it is essential to cover diverse networking technologies for the IoT connectivity and data communication protocols and not to exclude the use of standardized application-specific IoT protocols. In order to accomplish that, it is necessary to introduce protocol adapters that support multiple IoT data communication protocols. Figure 35 illustrates the concept of the support of diverse IoT data communication protocols used in the current RZs and for others to be extended.

Such protocol adapter is a *must* in smart city solutions, which include multiple application domains which naturally connect heterogeneous IoT protocols. SynchroniCity's RZs have customized protocol adapters as follow:

- *FIWARE IoT Agent*: used in Santander, Porto, and Eindhoven. it provides Ultralight 2.0 protocol connector as well as connectors for MQTT, HTTP and AMQP, and communicate with Orion Context Broker with NGSI.
- *DYAMAND* (Dynamic, Adaptive Management of Networks and Devices): used in Antwerp to support multiple IoT protocols. It abstracts the peculiarities of devices, on different levels (network, protocols, languages, data formats and control flow). It is able to represent devices, its data and how to interact with the devices in a normalized way.
- *UDG (Universal Device Gateway)*: used in Carouge for integration of different smart city applications. It supports over 50 different IoT connection and communication protocols including international standards and a wide area of industry standards, as well as legacy support. It also supports NGSI to connect to the Orion context broker.
- *Customized Gateway in Helsinki*: developed in Helsinki to provide API conversion from O-Mi, O-DF that are used in Helsinki Urban Platform to NGSI. The high-level building block on this model is in Table 8. Integration models.
- *Customized Proxy in Seongman*: developed in KETI to provide API conversion from oneM2M Mca to NGSI. The high-level building block on this model is in Table 8. Integration models.

It is noted that Milan has a plan to build a customised NGSI adapter to be used for its WSO2 platform to be compliant with SynchroniCity reference architecture.

As it is illustrated in the Figure 35, the minimum requirement to provide integrated framework is to agree on the common data format as well as an API. The survey results show that JSON, XML, UL2.0 and TXT are used in the RZs. Regarding the data format, it follows the decision of the SynchroniCity's task T2.2 (Enablement of a DSM for IoT enabled Smart City applications), where JSON data format with SynchroniCity's common data models are defined.

Unlike the heterogeneous protocol support, the API for the context management and its data format is a key to build an interoperable solution and it is necessary to use a common data format between southbound modules and the context data management module. There is a minimal set of requirements for smart city solutions to be compliant with SynchroniCity's framework as followings:

- A. The smart city solutions *must* be provided with protocol adapters to support the diverse IoT data communication protocols.
- B. In order to be compliant with the SynchroniCity's reference implementation, the smart city solutions *must* provide NGSI API for communication with the SynchroniCity context data management module.
- C. Considering that the current reference data models are based on the JSON data format, it is *required* to use JSON data format to communicate with SynchroniCity's reference platform.

Each RZ is in different status regarding the fulfilment of these requirements. The following table explains the required work in the southbound integration in the different platforms used in different

RZs. It describes the binding protocols¹ and supporting protocols² in the platform, which can help the choice of the devices when introducing new services and platforms.

Table 8. Integration models

DYAMAND in Antwerp	
The DYAMAND platform is the IoT middleware layer of the IMEC data platform that is used within the Antwerp Smart City project, City-of-Things. This DYAMAND component is very flexible in terms of supporting novel types of sensors and standards via a plug and play architecture. It also supports actuation, device management as well as monitoring and error reporting. Within the City of Things data stack, DYAMAND sends all data to the Kafka Message Buffer.	
Heterogeneous IoT protocol integration	Binding protocols: HTTP, USB, CoaP Supporting protocols from the local development: LoRa/Actiity, UPnP Other protocols including legacy systems: Insteon, Philip Hue, EnOcean, Grove, Z-Wave
Connecting to the Context Data Management	With FIWARE connectors: none. With customised gateway for non-supporting IoT protocols: by means of dedicated microservices that make the translation (has to be set up).
IoT Gateway in Antwerp	
The IoT Gateway is responsible for processing device messages outside the Antwerp Smart City project, City-of-Things. The IoT Gateway manages IoT devices and captures real-time data streams (Southbound IOT devices data ingestion). It transforms the incoming messages to a standardized JSON data model and provides further distribution of these messages through a RESTful pub-sub model.	
Heterogeneous IoT protocol integration	Binding protocols: HTTP (Push and Polling), HTTPS, MQTT Other protocols including legacy systems: should build dedicated adapters (or gateways) to support the protocols.
Connecting to the Context Data Management	IoT gateway pushes its messages to a Nifi instance that provides further message transformation to conform to the NGSI v2 entity model. NiFi pushes the messages to Orion Context Broker through HTTPS.

¹ Here, it means these protocols are directly supported by the platform with its the intercommunication modules.

² Here, it means these protocols are supported by installing extra modules developed by its community.

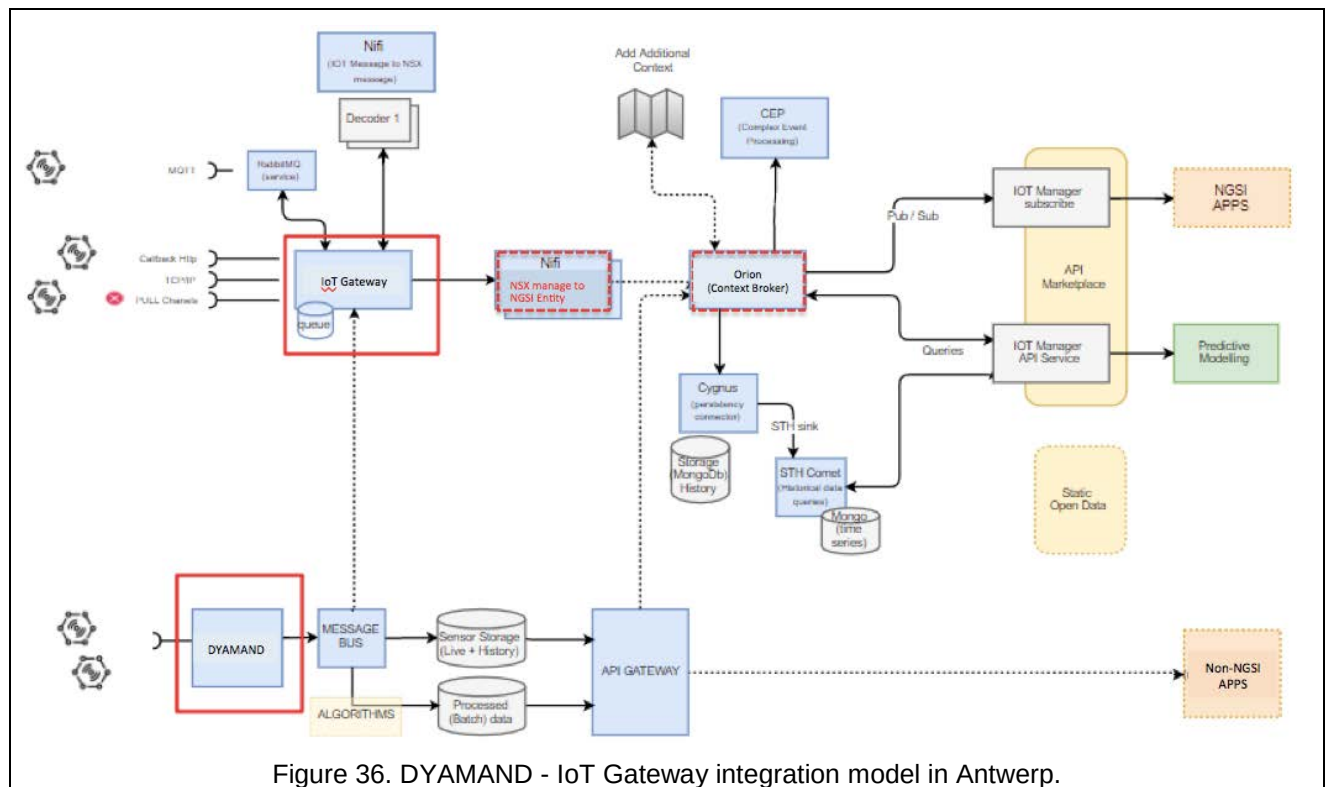


Figure 36. DYAMAND - IoT Gateway integration model in Antwerp.

In Antwerp there are two IoT southbound platforms, DYAMAND and the IoT Gateway. Under the IoT Gateway platform, NiFi instance transforms the messages into NGSI and pushes them to the Orion context broker.

UDG (Universal Device Gateway) in Carouge

UDG supports rich sets of IoT protocols in link connectivity, networking and data communication in diverse application domains (currently over 50 IoT protocols are supported including major standard protocols and protocols used in legacy systems). It is designed to support large-scale IoT services with heterogeneous IoT devices. It also provides its own platform services and the southbound platform modules, which can be separately used as an intermediate solution for supporting seamless interoperability for heterogeneous IoT edge into different IoT platforms in the case of Carouge.

Heterogeneous IoT protocol integration	Binding protocols: about 50 IoT protocols including HTTP, MQTT, LwM2M, CoAP, 6LoWPAN, 6Tisch, AMQP, etc. Other protocols used in specific application domains: should build dedicate connectors to support the protocols.
Connecting to the Context Management Data	Support NGSI. No extra work needed. Illustrated in Figure 37.

In Carouge, at the moment there are not many IoT protocols being used, but the city is extending its services in diverse domains which may require the integration of heterogeneous IoT protocols. The choice of UDG has been made due to its wide range of IoT protocol support and easy integration with IoT platforms.

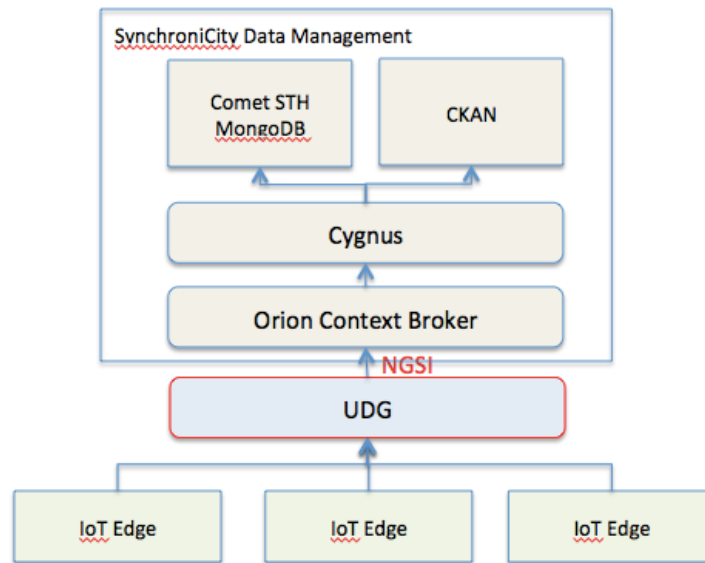


Figure 37. UDG integration model in Carouge

Helsinki Urban Platform in Helsinki

Helsinki Urban Platform is designed to support a rich set of diverse smart city applications built in Helsinki.

Heterogeneous IoT protocol integration

Binding protocols: HTTP, Modbus, AIREAS, etc.

Other protocols used in specific application domains: should build dedicate connectors to support the protocols.

Connecting to the Context Data Management

Need to build a dedicate module or device that support NSGI connecting the Urban Platform to the SynchroniCity context data management module.

Helsinki uses O-Mi, O-DF APIs in its platform, and Helsinki has developed a customized gateway that supports NSGI for sharing the open data within the SynchroniCity framework.

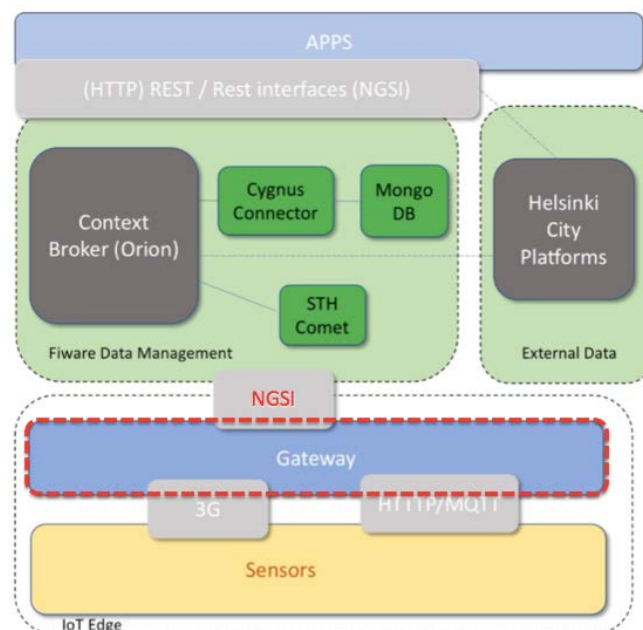
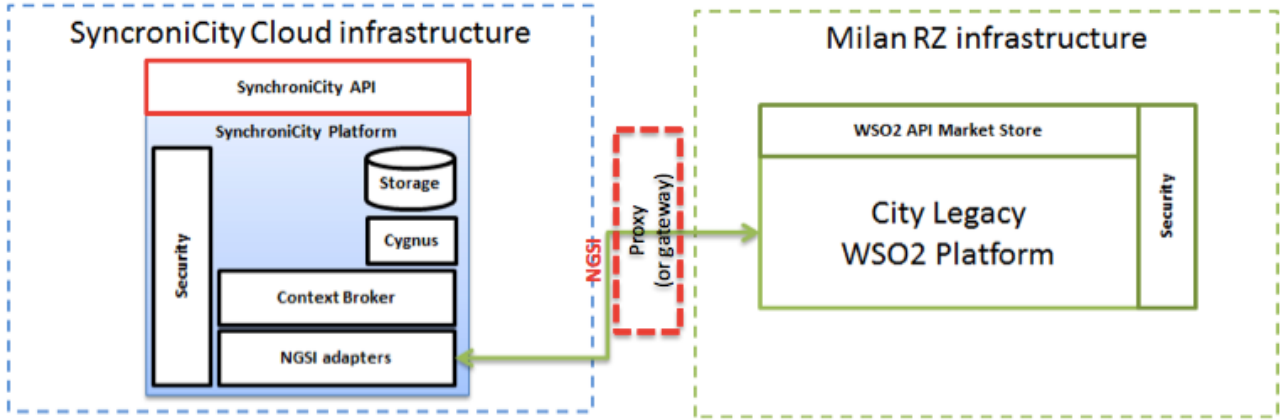
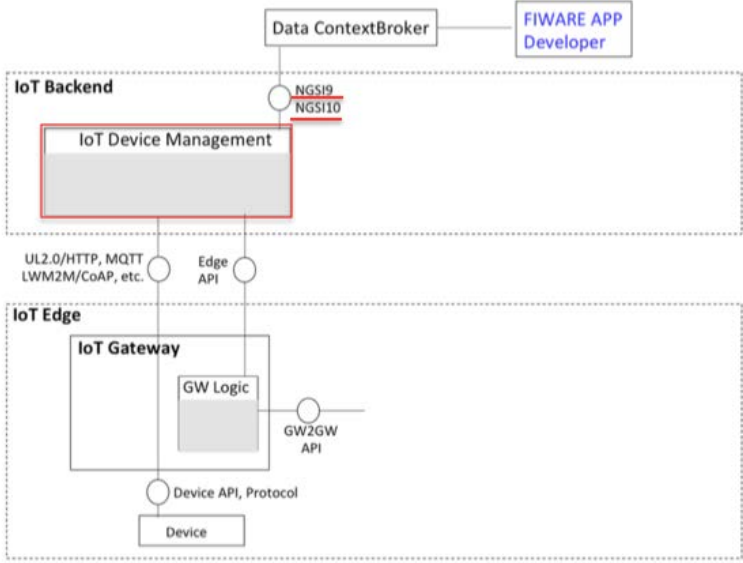
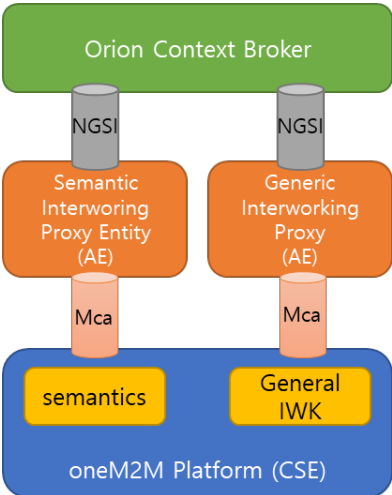


Figure 38. Helsinki Urban Platform integration model

WSO2 in Milan	
WSO2 is a fully open source interoperability platform. WSO2 APIs are released under Apache Software License Version 2.0. WSO2 is composed by several modules such as API manager, Enterprise Integrator, Identity server and Data Analytics server. WSO2 IoT provides technologies for devices to develop connected products as well as integration and analytics capabilities.	
Heterogeneous IoT protocol integration	Binding protocols: customized for Milan applications such as MQTT + TLS / SSL, CoAP + TLS / SSL. Other protocols used in specific application domains: should build dedicate adapters (or gateways) to support the protocols.
Connecting to the Context Data Management	Needs to build a dedicate module or device that support NSGI connecting WSO2 to the SynchroniCity context data management module. It may be done within the context of the Open Call.
 Figure 39. WSO2 Integration model in Milan.	
As Milan uses WSO2 platform which uses different APIs for service data, it is necessary to build a proxy (or gateway, or extend NGSI adapters) to share the open data for creating new services. Milan plans to build the interconnector through open calls.	
FIWARE Platform in Eindhoven, Porto and Santander	
FIWARE Platform is widely used in European smart city platforms by providing diverse functionalities in modular approach to build IoT and smart city services.	
Heterogeneous IoT protocol integration	Binding protocols: HTTP, MQTT. Supporting from the local development: OMA LwM2M, CoAP Other protocols including legacy systems: should build dedicate connectors (or gateways) to support the protocols.
Connecting to the Context Data Management	With FIWARE connectors: NGSI based. No extra work needed. With customised gateway for non-supporting IoT protocols: the gateway should support NGSI. Illustrated in Figure 40.

When the device supports UL2.0/HTTP, MQTT, LWM2M/CoAP, it is directly connected to FIWARE IoT Agents (respectively, LWM2M IoT Agent, JSON IoT Agent, Ultralight IoT Agent.) For other IoT protocols, a dedicated IoT gateway should support the protocols. FIWARE IoT device management support NSI and communicates with the Orion Context Broker.	 Figure 40. FIWARE Integration model
oneM2M in Seongnam	
oneM2M is a common IoT middleware specification for different domains and supports rich southbound intergration. To provide better migration with existing deployment, especially in smart cities, oneM2M interworking technologies have been specified. (e.g. Lwm2M). oneM2M is already deployed in the other metro cities in South Korea, such as Busan and Goyang.	
Heterogeneous IoT protocol integration	Binding protocols: HTTP, CoAP, MQTT, WebSocket. Supporting protocols: BBF TR-069, OMA DM, Lwm2M, AllJoyn, OCF, OSGi
Connecting to the Context Data Management	Needs to build a dedicate module that supports NSGI connecting the oneM2M to SynchroniCity's context broker. Leveraging oneM2M semantic capabilities, it can be designed with semantic interworking or non-semantic generic interworking following oneM2M interworking framework specification.

Seongnam uses oneM2M Mca interface for the southbound interface in oneM2M platform, and it builds interworking proxies as shown in the Figure, in order to support open data sharing based in NSGI interface. (CSE: Common Services Entity, AE: Application Entity)	 Figure 41. oneM2M Integration model
Customised platform in other cities	
Heterogeneous IoT protocol integration	Binding protocols: customized sets based on the city demands. Other protocols used in specific application domains: should build dedicate connectors to support the protocols.
Connecting to the Context Data Management	Needs to build a dedicate module or device that support NSGI connecting the Urban Platform to the SynchroniCity context data management module.

4.2 Integration guidelines

In the Section 4.1, the current RZs' integration models are presented in order to meet the minimal requirement to be integrated into the SynchroniCity's framework. It explains different approaches to support NSGI in RZs which use different platforms. As the table explains, an NSGI adapter is needed in several cases. Thus, this section explains practical implementation guidelines to support southbound integration with the SynchroniCity's platform that requires NSGI API for the context data management.

In particular, three reference implementations are described. In order to give a generic view on developing an NSGI adapter, a basic composition of an Ad-hoc NSGI adapter is described with its message sequence diagram. Two examples of the reference implementations are followed: a guideline to build an adapter adopting FIWARE IDAS Agent Manager [44], and the UDG example of a third-party development of the NSGI adapter that was performed in a SynchroniCity RZ.

It is important to emphasise that the following scenarios have to be considered as reference examples, and that they should be customised for the specific IoT infrastructure already deployed in the RZs.

4.2.1 Generic description of an Ad-hoc NSGI Adapter

As it is explained in the Section 4.1, when the legacy platform does not support NSGI, an ad-hoc NSGI adapter must be developed to integrate with the SynchroniCity's platform. The ad-hoc adapter is composed by two main components as shown in Figure 42:

- A Legacy Client
- A Model Adapter

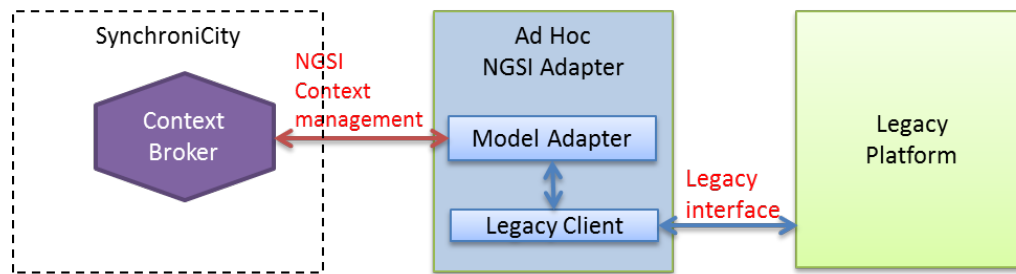


Figure 42. Ad-hoc NGSI adapter.

The Legacy Client will be responsible for accessing the Legacy Platform, moving the data accordingly to the legacy interfaces. Such component has to embed the logic to access the data from the Legacy Platform. For example, polling or subscribing if this functionality is provided by the legacy interface. Some improvements could be performed in this data phase: for instance filtering out-dated information and forward only the information that has been updated since the last call.

The Model Adapter will be in charge of combining and transforming raw legacy data to be compliant with the NGSI meta models, and the Model Adapter will use the NGSI context management API provided by SynchroniCity to update the context information into the broker.

The following sequence diagrams shows how the ad-hoc adapter modules should act in order to correctly update the data on the Context Broker in the SynchroniCity's platform. It is divided into two main cases: data polling (Figure 43) and subscription (Figure 44). In the polling scenario, the client periodically asks the Legacy Platform for data receiving, and receives responses with the original format and data model (called raw data in the diagrams). This data is sent from the client to the Model Adapter and the Model Adapter translates it into a specific SynchroniCity's data model adopting NGSI, and forwards the result to the Context Broker.

The subscription scenario differs from the previous one in the first phase. The data is collected through a direct notification from the Legacy Platform to the client that had previously subscribed the specific topic.

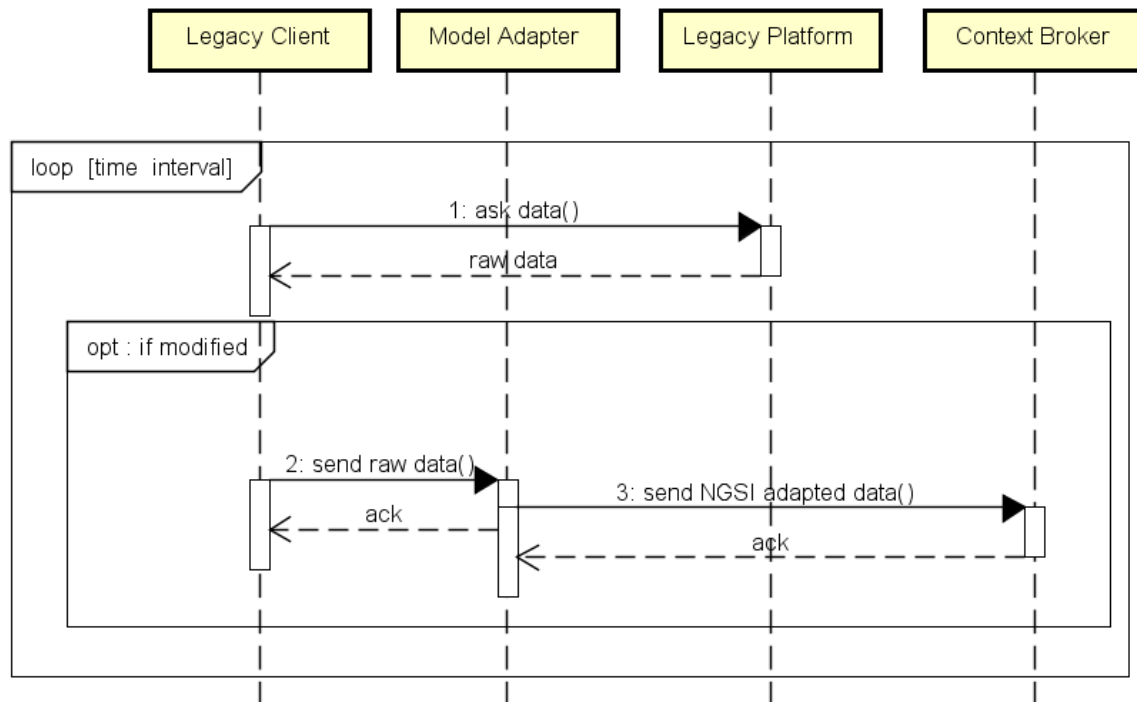


Figure 43. Data polling

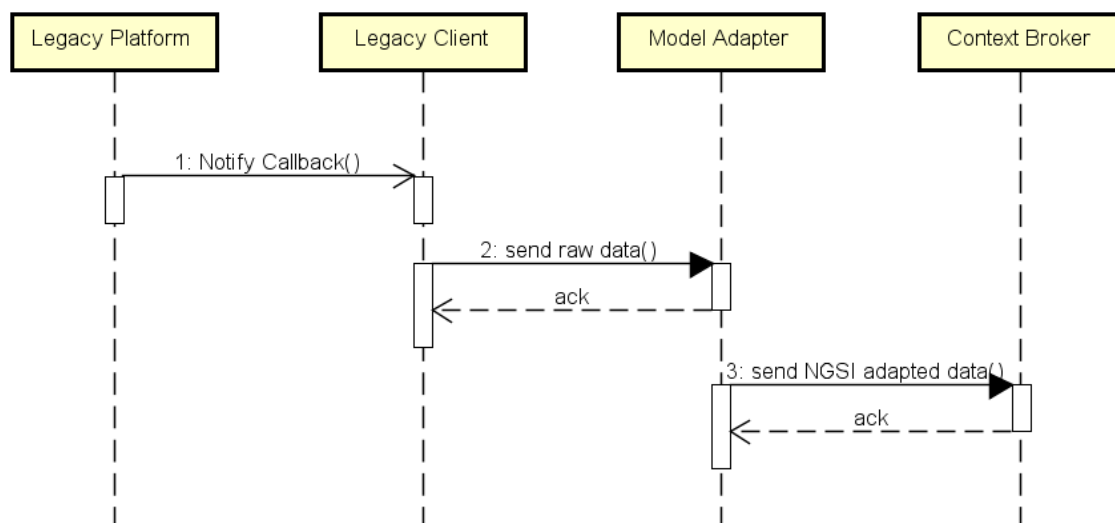


Figure 44. Subscription Scenario

4.2.2 NGSI Adapter Example 1: FIWARE IDAS Agent

The IDAS IoT Agent Manager can interact with multiple IoT Agents that support different southbound protocols shown in Figure 45.

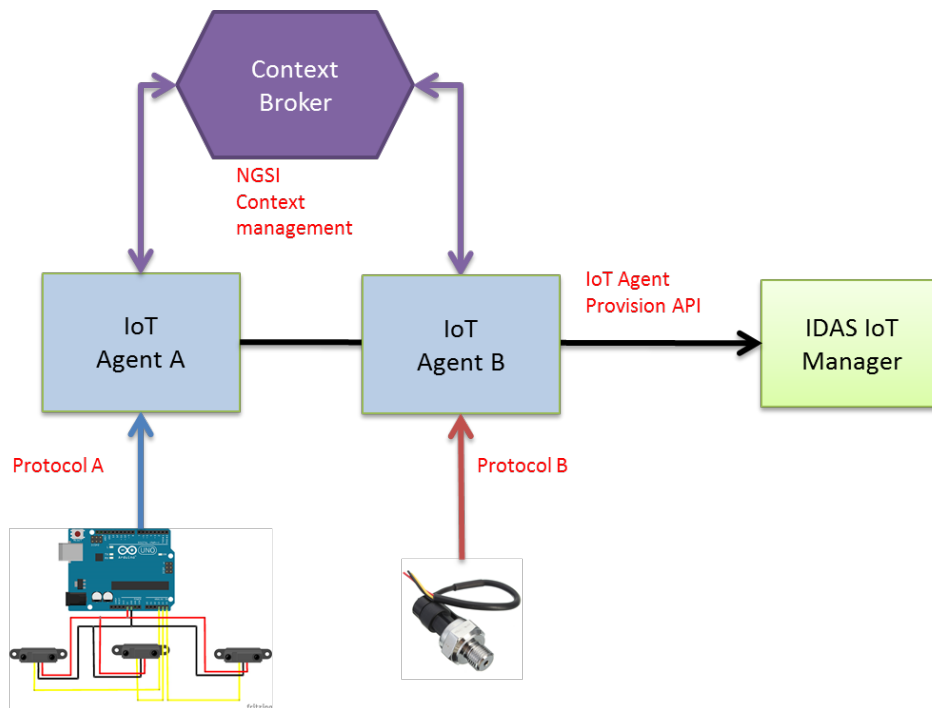


Figure 45. Agent Manager integration

It is a single administration endpoint for agent provisioning tasks, redirecting NGSI requests to the appropriate IoT Agent based on the defined protocol.

Each registered Agent is uniquely identified by the following two parameters:

- Protocol: Name of the protocol served by the IoT Agent
- Resource-API Key: Unique pair of strings used to identify different IoT Agents for the same protocol.

The agent manager exposes the Subscription APIs³ to manage agent provisioning and the services to be forwarded to the appropriate Agent. Whenever a new IoT Agent wants to register itself into the IoT Agent Manager, it must send a subscription request to the following path:

Http method: POST
 {IoTAgentHost}/iot/protocols

It indicates the following information:

- protocol: Name of the protocol served by the IoT Agent.
- description: Textual description for its display in portals.
- iotagent: URL address where requests for this IoT Agent will be redirected.
- resource: Unique string used to identify different IoT Agents for the same protocol.
- services: List of device configurations available in the IoT Agent. The IoT Manager saves a cache for all the configurations, aimed to be used to fasten the operations against the IoT Agent' databases.

The following example shows a registration of an IoT Agent that already has some configuration groups registered in the IoT Agent:

```
{
  "protocol": " Protocol-A-http",
```

³ <https://github.com/telefonicaid/iotagent-manager#-subscription-api>

```

    "description": "Protocol-A",
    "iotagent": "http://aprotocol.rzx.com/iot",
    "resource": "/iot/a",
    "services": [
      {
        "apikey": "801230BJKL23Y9090DSFL123HJK09H324HV8732",
        "token": "8970A9078A803H3BL98PINEQRW8342HBAMS",
        "entity_type": "SensorMachine",
        "resource": "/deviceTest",
        "service": "theService",
        "service_path": "theSubService",
        "attributes": [
          {
            "name": "status",
            "type": "Boolean"
          }
        ]
      }
    ]
  }
]
}

```

This operation can also be used to update the protocol subscriptions. If a protocol creation request arrives to the IoT Agent Manager with the same protocol and resource of an already existing agent, it will override the record with the new information.

To retrieve the list of all the available protocols, with their available endpoints you can use the following RESTful operation:

```

Http method: GET
{IoTAgenHost}/iot/protocols

```

The following example shows a sample response from the server:

```

{
  "count": 2,
  "protocols": [
    {
      "protocol" : "Protocol-A-http",
      "description" : " Protocol-A",
      "endpoints" : [
        { "endpoint" : " http://aprotocol.rzx.com:8080/iot",

```

```
        "identifier" : "idcl2:8080",
        "resource" : "/iot/a"
    }
]
}, {
    "protocol" : "PDI-IoTA-UltraLight",
    "description" : "UL2",
    "endpoints" : [
        { "endpoint" : "http://127.0.0.1:8080/iot",
          "identifier" : "idcl1:8080",
          "resource" : "/iot/d"
        }
    ]
}
]
```

The list accepts to query parameters with the following conditions:

- *limit*: limits the number of entries to return from the query.
- *offset*: skips the given number of entries from the database before returning the list.

4.2.3 NGSI Adapter Example 2: UDG NSGI Adapter

Basically, UDG is composed by different application and protocol modules integrating the different communication protocols and services used by the city of Carouge. This approach permits a high modularity across all the application modules of UDG. Indeed, the application modules can exchange data easily and bidirectionally between them and, at the end, between communication protocols, applications and services.

In the case of Carouge, two sensor manufacturers are based in Geneva: IEM and OrbiWise. Both offer an access to its server through an API. So, UDG provides a specific module for each sensor manufacturer. At the same time, the canton of Geneva has provided open data for the public parking ("OffStreetParking" FIWARE data model) and the current status of the traffic on the different streets of the canton ("TrafficFlowObserved" FIWARE data model). Then, two application modules were created to retrieve the data of the canton, mapping by the way the data models developed by the canton of Geneva to the FIWARE compatible data models. The public transport company in Geneva named TPG (Transports Publics Genevois) already has an API for the open data and this API was incorporated into UDG through an application module.

UDG included a new model to integrate NGSI v2. This module was tested internally using its own instance of Orion context broker. The following figure illustrates the different modules used in the frame of the integration of Carouge.

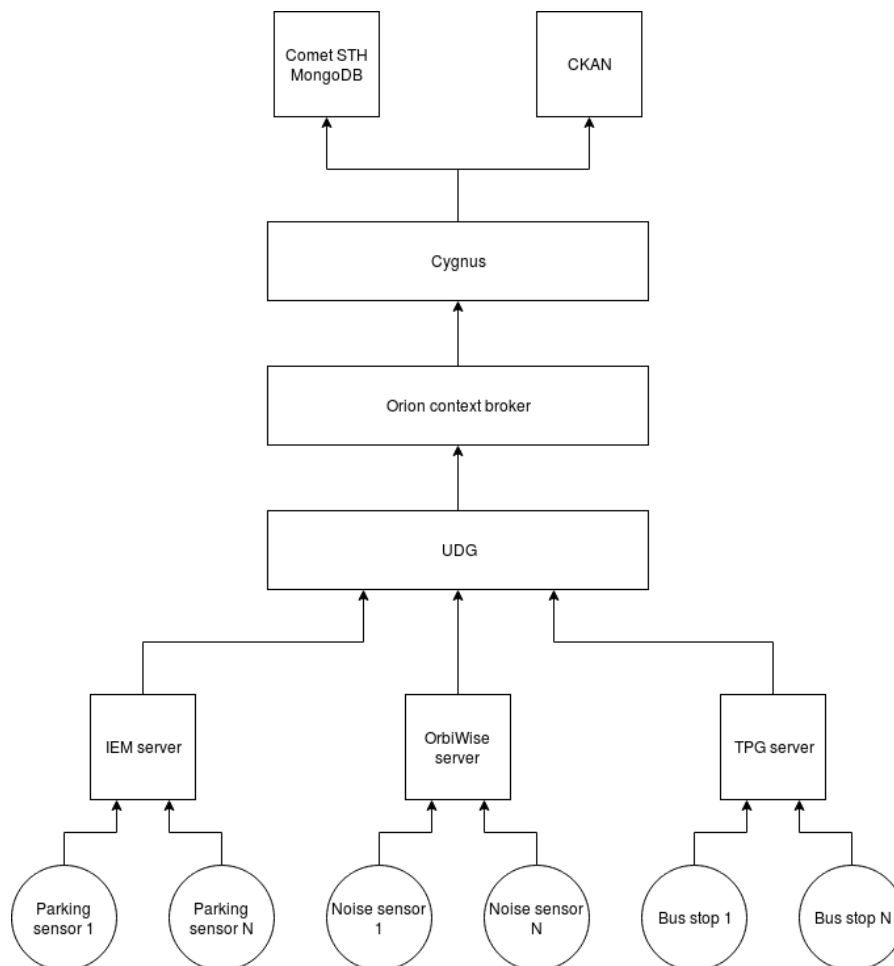


Figure 46. UDG as Carouge integrator

In the framework of the SynchroniCity project and the Carouge integration, UDG is using its own NGSI v2 module to transmit the data of the different sensors and services to an Orion context broker hosted by UDG Alliance, a partner of the SynchroniCity consortium.

The process is quite simple: a specific module receives the data from one of the sensors installed in the city of Carouge or from a server based in Geneva. Then, the data is mapped to a defined FIWARE data model and different calls of the NGSI v2 API are made by the UDG NGSI v2 module. First, a call is made to create the FIWARE/NGSI v2 entity on the Orion context broker, followed by a new call to create a subscription for this new created entity. Then, when the data is received again, the UDG NGSI v2 module updates the entity with the corresponding API call.

The different NGSI v2 API calls used by UDG in the context of SynchroniCity are described below. The first step is the creation of new entity inside the Orion context broker and the corresponding NGSI v2 API call is as the following shows:

```
curl localhost:1026/v2/entities -s -S --header 'Content-Type: application/json' -d @- <<EOF
{
  "id": "Room1",
  "type": "Room",
  "temperature": {
    "value": 23,
    "type": "Float"
  },
  "pressure": {
    "value": 720,
    "type": "Integer"
  }
}
EOF
```

Figure 47. Creation of a NGSI v2 entity

The following figure shows the subscription of the previous created NGSI v2 entity:

```
curl -v localhost:1026/v2/subscriptions -s -S --header 'Content-Type: application/json' \
-d @- <<EOF
{
  "description": "A subscription to get info about Room1",
  "subject": {
    "entities": [
      {
        "id": "Room1",
        "type": "Room"
      }
    ],
    "condition": {
      "attrs": [
        "pressure"
      ]
    }
  },
  "notification": {
    "http": {
      "url": "http://localhost:1028/accumulate"
    },
    "attrs": [
      "temperature"
    ]
  },
  "expires": "2040-01-01T14:00:00.00Z",
  "throttling": 5
}
EOF
```

Figure 48. Subscription to a NGSI v2 entity

Finally, the figure below displays the update of the same entity.

```
curl localhost:1026/v2/entities/Room1/attrs -s -S --header 'Content-Type: application/json' \
-X PATCH -d @- <<EOF
{
  "temperature": {
    "value": 26.5,
    "type": "Float"
  },
  "pressure": {
    "value": 763,
    "type": "Float"
  }
}
EOF
```

Figure 49. Update of a NGSI v2 entity

5 Conclusion

In this deliverable, we discussed minimal interface points to build an interoperable smart city framework in a multi-domain and multi-disciplined environment. Its focus is in the interoperability in the southbound interface of the SynchroniCity's reference architecture and the collection of the current platforms and devices that are deployed and used in the RZs. Diverse IoT devices, products and solutions deployed in the RZs were collected, which helps on achieving one of the most important objectives of Task 2.4: enabling interested cities to access a list of IoT products and solutions easily deployable and fully interoperable with the SynchroniCity's architecture and deployment.

There are common smart city applications and services, while some are unique in some RZs. On developing a same application, different RZs chose different systems and data models with different platforms. In order to minimize the extra work for RZs to be integrated into SynchroniCity, a detailed survey and analysis was made. Apart from a few exceptions, most of them use a few common standard protocols that are interconnected with platforms via proxies or gateways. It was discussed that SynchroniCity is focusing on data sharing that enables regenerating new services. According to it, the focus of southbound integration is to support an interoperable API among data sources (IoT devices) and the context management module in the platform, without putting any limitation on the choice of IoT networking technologies and data communication protocols that are naturally binding to the city's situation and the types of applications.

The analysis of the survey data was given per application, as it is believed that such results are useful for city authorities to check the available options when the city wants to build a new service. Cities can learn from the choices of systems, devices, and related protocols that other cities already made.

As we could foresee, different RZs are using different platforms and systems, which have different binding protocols among IoT devices, and between IoT devices and the platform. As reinvention of a new solution is not the scope or goal of SynchroniCity, this deliverable describes the situation, supporting protocols and the required work to be done per platform that is used in each RZ. For new cities, this summarized description per platform can give a good insight on measuring their effort in their situation. It also includes a generic model to build a customised NSGI adapter with two implementation examples: IDAS Agent and UDG NSGI adapter. IDAS Agent is used in FIWARE platform that uses NSGI API, and UDG NSGI Agent is built on top of UDG IoT management module to support the NSGI API and FIWARE platform. The current RZs have already built an NSGI adapter in a gateway or are planning to build one. For the new cities, the implementation guidelines and examples may give a good starting point when they need to build the adapter.

As a next step, this task will focus on building a southbound market place in order to support Digital Single Market (DSM). The marketplace will include IoT products and solutions interoperable with the SynchroniCity's framework, and also provide use cases of the RZs. It aims to provide interested cities for references to find SynchroniCity compliant products and solutions and help to foster IoT ecosystem.

The design of interoperability will be continuously carried out, both at a protocol level and at a data model level in RZs, which make it possible for platform-agnostic data gathering and sharing within the Digital Single Market (DSM). At the end of this process, a catalogue of devices ready for the integration will be published.

References

- [1] SynchroniCity, D2.1 – Reference Architecture for IoT Enabled Smart Cities, 2017 (available online at <https://tinyurl.com/ychr35kf>)
- [2] <https://ovacen.com/las-smart-cities-se-multiplican-por-cuatro-hasta-el-2025/>
- [3] OASC (Open & Agile Smart Cities), www.oascities.org
- [4] FIWARE, www.fiware.org
- [5] oneM2M, www.onem2m.org
- [6] AXITEA, Security provider, www.axitea.it
- [7] IMEC, www.imec-int.com
- [8] JSON, www.json.org
- [9] Proximus LoRAWAN base station, www.proximus.be
- [10] BIGBelly smart city solution, www.bigbelly.com
- [11] Enovo, www.enevo.com
- [12] Total Waste System (TWS), www.totalwastesystems.nl
- [13] Universal Device Gateway (UDG), IoT interoperability solution, <http://devicegateway.com>
- [14] Orbiwise, www.orbiwise.com
- [15] ETSI EN 302 663, Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band
- [16] Philips City Touch, www.lighting.philips.com/main/systems/lighting-systems/citytouch
- [17] AIREAS, www.aireas.com
- [18] CESVA TA120 - Noise measuring sensor for Smart Solutions, <https://www.cesva.com/en/products/sensors-terminals/TA120/>
- [19] Sentilo, www.sentilo.io/xwiki/bin/view/APIDocs/Overview
- [20] Vaisala air quality sensors, AQT420, <https://www.vaisala.com/en/products/instruments-sensors-and-other-measurement-devices/weather-stations-and-sensors/aqt420>
- [21] Milano Linate airport, www.milanolate-airport.com/en
- [22] Milano Malpensa airport, www.milanomalpensa-airport.com/en
- [23] Orio al Serio international airport • SACBO S.p.A., www.milanbergamoairport.it/en
- [24] Fiware-Orion, <https://fiware-orion.readthedocs.io>
- [25] The Comprehensive Knowledge Archive Network (CKAN), <https://github.com/KSP-CKAN/CKAN>
- [26] iBKS Plus · Accent Systems, <https://accent-systems.com/product/ibks-plus>
- [27] BullsEye NFC - SMARTRAC N.V., www.smartrac-group.com/bullseye-nfc.html
- [28] iBeacon™, Accent Systems, <https://accent-systems.com/support/faq/what-is-ibeacon>
- [29] Eddystone™, [https://en.wikipedia.org/wiki/Eddystone_\(Google\)](https://en.wikipedia.org/wiki/Eddystone_(Google))
- [30] ISO/IEC 14443-3:2011 - Identification cards -- Contactless integrated,

<https://www.iso.org/standard/50942.html>

- [31] Tag Type Technical Specifications - NFC Forum, <https://nfc-forum.org/our-work/specifications-and.../tag-type-technical-specifications>
- [32] Manual_M5 Manual IZAR cp R3.5, Diehl Metering GmbH - FCC ID, <https://fccid.io/.../5A62420C-F5F0-4F86-9D5E-6104F7B5134B>
- [33] Diehl Metering Izar Receiver GPRS/LAN, www.diehl.com/en/diehl-metering/products-solutions/pdb-details.html?tx_diehlproducts_pi1%5Bproduct%5D=2794509&tx_diehlproducts_pi1%5BbackPid%5D=535&tx_diehlproducts_pi1%5Baction%5D=show&tx_diehlproducts_pi1%5Bcontroller%5D=MeteringProduct&cHash=c6b8923ed171114c97336324b3e1382a
- [34] Libelium - Connecting Sensors to the Cloud, www.libelium.com
- [35] ONVIF, www.onvif.org
- [36] Diehl Metering, www.diehl.com/en/diehl-metering/products-solutions/pdb-details.html?tx_diehlproducts_pi1%5Bproduct%5D=21566482578&tx_diehlproducts_pi1%5BbackPid%5D=512&tx_diehlproducts_pi1%5Baction%5D=show&tx_diehlproducts_pi1%5Bcontroller%5D=MeteringProduct&cHash=096c86c6c9e20fa2e5823130c90d8a5e
- [37] Global Reporting Initiative (GRI), https://en.wikipedia.org/wiki/Global_Reporting_Initiative
- [38] ETSI TS 100 392-3-7, Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 3: Interworking at the Inter-System Interface (ISI); Sub-part 7: Speech Format Implementation for Packet Mode Transmission, www.etsi.org/deliver/etsi_ts/100300_100399/1003920307/01.01.01_60/ts_1003920307v010101p.pdf
- [39] DYAMAND, <https://dyamand.ilabt.imec.be>
- [40] Platform WSO2, <https://wso2.com/platform>
- [41] GUPSY - Global Urban Positioning & Sensor sYstem, www.gupsy.com
- [42] DASH7 Alliance, www.dash7-alliance.org
- [43] Q-Park, www.q-park.com
- [44] Backend Device Management - IDAS | FIWARE Catalogue, <https://catalogue.fiware.org/enablers/backend-device-management-idas>
- [45] MongoDB, www.mongodb.com
- [46] EU Smart Santander project, www.smartsantander.eu
- [47] IETF RFC 6347, Datagram Transport Layer Security Version 1.2, <https://tools.ietf.org/pdf/rfc6347.pdf>
- [48] Digital Addressable Lighting Interface (DALI), <https://www.digitalilluminationinterface.org/dali/>
- [49] 0-10 V lighting control, https://en.wikipedia.org/wiki/0-10_V_lighting_control
- [50] SDH (Synchronous Digital Hierarchy), https://en.wikipedia.org/wiki/Synchronous_optical_networking

Annex

Set of tables compiling application information.

Table A1. Reference Zones Analysis: Parking

Application	RZ	IoT Device		Gateway or backend system		Connectivity	Data comm. protocol		Data format	Southbound Interface			
		Model name	Version	Backend IoT device	Version		Name	Version		Device management	Comms	Context Data mgmt	Direct comm. with data storage
Parking	Carouge	IEM Pres to Sense		IEM Pres to 1000 Server		LoRaWAN	HTTPS		JSON	UDG	UDG	Orion Context Broker	CKAN, MongoDB
	Eindhoven	Proprietary via Q-Park		Via Q-Park		Via Q-park	?		n/a				
	Seongnam	Parking sensor	Illegal parking detector	Gateway (Kerlink SPN)		LoRa 1.1			XML		oneM2M Mca Rel-2	oneM2M LoRa IPE	
	Antwerp	FASTPRK		Worldsensing Cloud Sigfox Cloud	(cloud)	Sigfox	REST API	-	JSON	ACPaaS IoT Gateway		REST API (HTTP)	
		A*Sign		Sigfox Cloud	(cloud)	Sigfox	REST API	-	JSON	Romcore SmartCities backend		REST API (HTTP)	
	Santander	Parking sensor		Meshlium (Libelium) 802.15.4/3G/WiFi/Ethernet		802.15.4/3G/Wi-Fi/Ethernet	REST, MQTT, WebSocket		UL2.0	SmartSantander Infrastructure FIWARE IDAS	UL2.0		
	Milan	Smart parking sensors for disabled	Kiunsys			LoRaWAN/NB-IoT	LoRaWAN: MQTT+TLS/SSL NB-IoT: CoAP+TLS/SSL		JSON	InesCloud ??	REST API		
		Smart parking sensors for	Kiunsys			LoRaWAN/NB-IoT	LoRaWAN: MQTT+TLS/SSL		JSON	InesCloud ??	REST API		

		electric vehicles					NB-IoT: CoAP+TLS/SSL						
		Smart parking sensors to monitor no parking area (bus stop, zebra crossing, etc)	Kiunsys			LoRaWAN/NB-IoT	LoRaWAN: MQTT+TLS/SSL NB-IoT: CoAP+TLS/SSL		JSON	InesCloud ??	API REST		

Application	RZ	IoT Device		Gateway or backend system		Connectivity	Data comm. protocol		Data format	Southbound Interface			
		Model name	Version	Backend IoT device	Version		Name	Version		Device management	Communication	Context Data mgmt	Direct comm. with data storage
Noise	Carouge	Orbi Wise		Orbiwise Server		LoRaWAN	HTTPS		JSON	UDG	UDG	Orion Context Broker	CKAN, MongoDB
	Helsinki	Cesva TA120		Custom HTTP endpoint		3G/4G	HTTP		Sentilo JSON UL2.0	Sensorthings API	Sensors POST to custom backend	Custom backend POSTs, NGSI data is constructed and proxied to ORION	N/A
	Santander	Noise sensors		Meshlium (Libelium)		---			n/a		HTTP		

Table A2. Reference Zones Analysis: Noise

Table A3. Reference Zones Analysis: Street Light

Application	RZ	IoT Device		Gateway or backend system		Connectivity	Data comm. protocol		Data format	Southbound Interface			
		Model name	Version	Backend IoT device	Version		Name	Version		Device management	Communication	Context Data mgmt	Direct comm. with data storage
Street Light	Seongnam	Street light actuator / controller		---		---			n/a		---		
	Santander	Meshlium (Libelium) Presence & Luminosity Nodes	?	Meshlium (Libelium)		802.15.4/3G/WiFi/Ethernet	REST, MQTT, WebSocket		UL2.0	SmartSantander Infrastructure FIWARE IDAS	UL2.0	?	?
	Manchester	---		---		LoRa			n/a		---		

Table A4. Reference Zones Analysis: Mobility and Traffic

Application	RZ	IoT Device		Gateway or backend system		Connectivity	Data comm. protocol		Data format	Southbound Interface			
		Model name	Version	Backend IoT device	Version		Name	Version		Device management	Communication	Context Data mgmt	Direct comm. with data storage
Mobility and Traffic	Milan (Mobility)	Metro status		ATM					n/a		SOAP Webservice		Data storage strictly forbidden
		Metro station - Parking status		ATM					n/a		SOAP Webservice		Data storage strictly forbidden

		Around me - Point of interest		ATM				n/a		SOAP Webservice		Data storage strictly forbidden
		BikeMe - Bike sharing station-based service. Bikes availability at each station.	GPS	ATM				n/a		SOAP Webservice		Data storage strictly forbidden
		bus/tram waiting time		ATM				XML		SOAP Webservice		Data storage strictly forbidden
		Local train time table		Trenord				XML		REST API		
		Local train circulation status		Trenord				n/a		RSS		
		Linate and Malpensa airport flights timetables		SEA				n/a		REST API		
		Linate and Malpensa Airport parking availability		SEA				n/a		REST API		
		Bergamo airport flights timetables		SACBO				n/a		REST API		
		Airport-Milan bus connection service	first company	Autostradale				n/a		SOAP Webservice		
		Airport-Milan bus connection service	second company	Air Pullmann				n/a		SOAP Webservice		
		Train network status		Trenitalia				n/a		SOAP Webservice		
	Milan (Traffic)	loops	various types, to measure traffic flow,	SMTCC / SigmaPlus / Utopia		cable/GSM/SDH	proprietary protocol	n/a			no	no

			speed, car dimensio ns. used also to adjust traffic- lights										
		fixed cameras	various types; B&W, color, with OCR, infrared			SDH/Wi-Fi	proprietary protocol		n/a			no	no
		mobile cameras	various types; B&W, color, with OCR, infrared			3G/4G	proprietary protocol		n/a			no	no
		local police agents	road incidents reports						Italian standard incidents report protocol	Database			
	Santander	Meshlium (Libelium) Vehicle counter SENSORS	?	Meshlium (Libelium)		802.15.4/3G/Wi- Fi/Ethernet	REST, MQTT, WebSocket		UL2.0	SmartSantander Infrastructure FIWARE IDAS		?	?
		Meshlium (Libelium) Vehicle speed SENSORS	?	Meshlium (Libelium)		802.15.4/3G/Wi- Fi/Ethernet	REST, MQTT, WebSocket		UL2.0	SmartSantander Infrastructure FIWARE IDAS	https://www.e015.regione.lombardia.it/PE015/esplora-i-contenuti/i-servizi/stato-del-servizio-ferroviario	?	?
	Porto	Vehicle counters		---		DIASER / GERTRUDE			TXT		---		
		Video cameras				ONVIF			n/a		---		

	Eindhoven (mobility)	Smart traffic lights ??		?	?	ITS-G5	?	?	n/a				
		Connected street lights				2G	COAP, DTLS, DALI, 1-10V		Philips CT proprietary relational data models	Philips CityTouch			
	Antwerp (Cycling)	Cycling Push Sensor	Verhaert	Proximus LoRaWAN base station	---	LoRaWAN	LoRaWAN	---	JSON	---	---	---	---
		Bike sharing stations							n/a	ACPaaS IoT Gateway	REST API (HTTP)	(To Be: Orion)	ACPaaS Data Lake
		Camera feed		(Wired)		(Wired)	?		n/a	?	?	?	?

Table A 5. Reference Zones Analysis: Environment

Application	RZ	IoT Device		Gateway or backend system info		Connectivity	Data comm. protocol		Data format	Southbound Interface			
		Model name	Version	Backend IoT device	Version		Name	Version		Device management	Communication	Context Data management	Direct comm. With data storage, if applicable
Environment	Milan	lampPost weather sensor	1.0						JSON /XML ?!	WSO2	REST API	WSO2 context broker	
		cameras	camera LEZ (Low Emission Zone) to monitor vehicles access			3G/4G	proprietary protocol		n/a			no	no
		Weather reports and forecast	report also air-pollution data and UV radiation						n/a	E015	SOAP WebService		

	Seongnam	Air quality sensors		---		---			n/a		---	---	
	Antwerp	Fixed/Mobile Air Quality Sensor	IMEC	Proximus LoRaWAN base station		LoRaWan	LoRaWAN (+Sigfox & DASH7)		JSON				
		Mobile Air Quality Sensor	Gypsy	Proximus LoRaWAN base station		LoRaWan	LoRaWan		JSON				
	Santander	Smart Spot Air quality	v.1.5	Homard	v1.0	Wi-Fi, M2M	LwM2M	v1.0	JSON, TEXT, TLV, Packet	Homard	REST (HTTP)	Orion Context Broker	v2
		Meshlium (Libelium) Noise	?	Meshlium (Libelium)		802.15.4/3G/Wi-Fi/Ethernet	REST, MQTT, WebSocket		UL2.0	SmartSantander Infrastructure FIWARE IDAS	UL2.0	?	?
		Meshlium (Libelium) luminosity	?	Meshlium (Libelium)		802.15.4/3G/Wi-Fi/Ethernet	REST, MQTT, WebSocket		UL2.0	SmartSantander Infrastructure FIWARE IDAS	UL2.0	?	?
		Meshlium (Libelium)CO	?	Meshlium (Libelium)		802.15.4/3G/Wi-Fi/Ethernet	REST, MQTT, WebSocket		UL2.0	SmartSantander Infrastructure FIWARE IDAS	UL2.0	?	?
		Meshlium (Libelium)Temperature	?	Meshlium (Libelium)		802.15.4/3G/Wi-Fi/Ethernet	REST, MQTT, WebSocket		UL2.0	SmartSantander Infrastructure FIWARE IDAS	UL2.0	?	?
		GPRS module MOBILE node	?	GPRS module		GPRS	REST, MQTT, WebSocket		UL2.0	SmartSantander Infrastructure FIWARE IDAS	UL2.0	?	?
	Helsinki	Vaisala AQT420	Air quality	Proprietary		Sensor->gateway: serial, gateway->internet: Wi-Fi	Modbus / ASCII		Proprietary	---	---	---	---
	Eindhoven	AiREAS Airbox	Intelligent Air Measurement system	Airbox		GSM	AIREAS API	v2	JSON		HTTP V2.0	VIA API on open data portal	
	Porto	Citibrain / Monitor SmartNOISESense (noise) SmartMETEOSense (meteorological parameters)	Particles (PM2.5, PM10), gases (Ox, NO2, CO), air	Monitar Sense (www.sense.monitar.pt)		HTTP			JSON	---	---	---	---
						CoAP			JSON	---	---	---	---
						MQTT			JSON	---	---	---	---

		SmartAIRSense (air pollutants)	(temperature, relative humidity), wind (direction, speed), rain, UV and noise										
--	--	--------------------------------	---	--	--	--	--	--	--	--	--	--	--

Table A6. Reference Zones Analysis: Culture

Application	RZ	IoT Device		Gateway or backend system		Connectivity	Data comm. protocol		Data format	Southbound Interface			
		Model name	Version	Backend IoT device	Version		Name	Version		Device management	Communication	Context Data mgmt	Direct comm. with data storage
Culture	Manchester	---		---		---			n/a		---		
	Milan	Street naming	4.0						JSON	WSO2	REST API	WSO2 context broker	
		Street/Area Information	1.0						JSON	WSO2	REST API	WSO2 context broker	
		Events+categories	1.0						JSON	WSO2	REST API	WSO2 context broker	
		Point of interest+categories	1.0						JSON	WSO2	REST API	WSO2 context broker	
		Itinerary+categories	1.0						JSON	WSO2	REST API	WSO2 context broker	
		Free WiFi area locations	1.0						JSON	WSO2	REST API	WSO2 context broker	

		Free WiFi area locations	1.0						JSON	WSO2	REST API	WSO2 context broker	
	Porto (Tourism and News)	Accent Systems iBKS Plus	Beacons (active)	Connectings			iBeacon and Eddystone		n/a		---		
		SMARTRAC BullsEye NFC	Beacons (passive)	Connectings			NFC: ISO/IEC14443 type A, NFC Forum type 2 QR codes		n/a		---		

Table A7. Reference Zones Analysis: Citizen

Application	RZ	IoT Device		Gateway or backend system info		Connectivity	Data comm. protocol		Data format	Southbound Interface			
		Model name	Version	Backend IoT device	Version		Name	Version		Device management	Communication	Context Data management	Direct comm. with data storage
Citizen	Milan	Users mobile phones	Android; iOS; Windows phone; BlackBerry		TIM Mobility scan	3G, 4G			n/a	E015	REST API		
	Santander	Users Mobile phones	Android & iOS			Wi-Fi, 3G, 4G, Bluetooth LE			n/a	SmartSantander Infrastructure FIWARE IDAS		?	NGSI9/NGSI10/NG SIV2

Table A8. Reference Zones Analysis: Energy Management

Application	RZ	IoT Device		Gateway or backend system info		Connectivity	Data comm. protocol		Data format	Southbound Interface			
		Model name	Version	Backend IoT device	Version		Name	Version		Device management	Communication	Context Data management	Direct comm. with data storage
Energy Management Heating System	Milano	temperature sensors	Capetti	Control Unit CTR372 or Flower	Cable/Wi-Fi connection from sensor to control Unit	from control unit to central server: GSM/GPRS	proprietary protocol		n/a			no	no
		temperature probes	Capetti	Control Unit CTR372 or Flower	Cable/Wi-Fi connection from sensor to control Unit	from control unit to central server: GSM/GPRS	proprietary protocol		n/a			no	no

Table A9. Reference Zones Analysis: Safety Alarms

Application	RZ	IoT Device		Gateway or backend system		Connectivity	Data comm. protocol		Data format	Southbound Interface			
		Model name	Version	Backend IoT device	Version		Name	Version		Device management	Communication	Context Data mgmt	Direct comm. with data storage
Safety Alarms	Milan	volumetric sensors		Control Unit	Cable connection from sensor to control Unit	Wi-Fi	proprietary protocol		n/a	Axitea (Managed by Local Police)		no	no
		cameras		Control Unit	Cable connection from sensor to control Unit	Wi-Fi	proprietary protocol		n/a	Axitea (Managed by Local Police)		no	no
		motion detectors		Control Unit	Cable connection from sensor to control Unit	Wi-Fi	proprietary protocol		n/a	Axitea (Managed by Local Police)		no	no

		door alarms		Control Unit	Cable connection from sensor to control Unit	Wi-Fi	proprietary protocol		n/a	Axitea (Managed by Local Police)		no	no
		fire detectors	Urmet	Control Unit	Cable connection from sensor to control Unit	Wi-Fi	proprietary protocol		n/a	Axitea (Managed by Local Police)		no	no
		smoke detectors	Urmet	Control Unit	Cable connection from sensor to control Unit	Wi-Fi	proprietary protocol		n/a	Axitea (Managed by Local Police)		no	no

Table A10. Reference Zones Analysis: Security

Application	RZ	IoT Device		Gateway or backend system info		Connectivity	Data comm. protocol		Data format	Southbound Interface			
		Model name	Version	Backend IoT device	Version		Name	Version		Device management	Communication	Context Data management	Direct comm. with data storage
Security	Milan	security cameras		switch		LoRaWAN / SCTT	proprietary protocol		n/a	Pacis		no	
		mobile GPS & radio		radio	sensor to Control Station	communication station-to-station wired (SDS-SCN); AVL; GRI Protocol	proprietary protocol (Tetra - GRI Protocol)		Tetra - GRI Protocol	Tetra		no	
		vehicles GPS & radio		radio	sensor to Control Station	communication station-to-station wired (SDS-SCN); AVL; GRI Protocol	proprietary protocol (Tetra - GRI Protocol)		Tetra - GRI Protocol	Tetra		no	