

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:	33 months
Project end date:	30.09.2019
Deliverable number:	D2.9
Deliverable title:	Report on Advanced Reference Zones platform deployment and operational plan
Document version:	Ver1.0
WP number:	WP2
Lead beneficiary:	7-ATOS
Main author(s):	Ömer Özdemir (ATOS), In Song Lee (KETI), Ignacio EliceGUI (UC), Filipe Aranda de Sá (Porto), Jaime Ventura (Porto), Martino Maggio (ENG), Giuseppe Ciulla (ENG), Paolo Foschi (MIL), Adrian Slatcher (MAN), Andrea Gaglione (DIGICAT), Juha Nummila (FVH), Antti Nurminen (AALTO), Jan-Joost van Kan (ATOS), Vincent de Waal (Heijmans), Niels Wiersma (EIN), Cedric Crettaz (MI), Koen Cornelissen (DIGIPOLIS)
Internal reviewers:	Arturo Medela (TST), Koen Cornelissen (DIGIPOLIS), Martin Brynskov (AU)
Type of deliverable:	Report
Dissemination level:	Public
Delivery date from Annex 1:	M23
Actual delivery date:	15.01.2020 (M37)



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.

Co-funded by



Executive Summary

As taken from the previous D2.8[3] Executive Summary section, SynchroniCity project represents the first attempt to deliver a Digital Single Market for Europe by piloting its foundations at scale in reference zones across 8 European cities, involving also other cities globally. It addresses how to incentivize and build trust for companies and citizens to actively participate, in finding common co-created IoT (Internet of Things) solutions for cities that meet citizen needs and to create an environment of evidence-based solutions that can easily be replicated in other regions.

Both D2.8 and D2.9 have been issued by the SynchroniCity Work Package 2 (WP2), which is responsible to define and provide the architecture, components and software, initial interaction model and data model, making up a so-called SynchroniCity platform, by leveraging on existing or new solutions, or integrated system based on context management, IoT device management, data management and market place, among others elements, and related to smart cities eco-system, city service domains and IoT (Internet of Things) infrastructure (software and hardware), by providing the major technological elements allowing a high level of interoperability.

This deliverable is an extension of the previous D2.8 where each RZ has reported about their infrastructure platform elements, data sources to be deployed and operated in each city respectively. These activities are based on the initial architecture specification that has been defined and documented in D2.1[1] and further revised in D2.10[2]. Our approach is to collect and present the “as-is” status of the RZs technical environments by this deliverable, which will play an important role for the WP3 analyse, design and development activities.

Abbreviations

AAA	Authentication, Authorization and Accounting
ACI	Access Control Interface
AE	Application Entity
AIOT	Alliance for the Internet of Things Innovation
API	Application Programming Interface
ASI	Application Support Interface
AWS	Amazon Web Services
B2B	Business-to-Business
BE	Business Entities
BSI	British Standards Institution
CB	Context Broker
CDB	Context Data Broker
CDMI	Cloud Data Management Interface
CEP	Complex Event Processing
CKAN	Comprehensive Knowledge Archive Network
CSE	Common Service Entity
D	Deliverable
DCAT	Data Catalogue Vocabulary
DMG	Device Management
DIS	Discovery
DMR	Data Management and Repository
DoW	Description of Work
DoA	Description of Action
EC	European Commission
e.g.	Exempli gratia = for example
ESI	Experimental Support Interface
ETSI	European Telecommunications Standards Institute
EU	European Union
FG-SSC	Focus Group on Smart Sustainable Cities
GA	Grant Agreement
GE	Generic Enabler
GIS	Geographic information system
GPS	Global Positioning System
GPRS	General Packet Radio Service
GMG	Group Management
GUI	Graphical User Interface
HART	Highway Addressable Remote Transducer
HDS	Hadoop Distributed File System
HRI	Helsinki Region Info share
HTML	Hypertext Mark-up Language
HTTP	Hypertext Transfer Protocol
IA	Innovation Action
ICT	Information and communication technology
IDAS	Intelligence Data Advanced Solution
IDE	Integrated Development Environment

IDM	Identity Management
IDS	Intrusion Detection System
i.e.	id Est = that is to say
IE	Integration Environment
IEEE	Institute of Electrical and Electronics Engineers
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IoT	Internet of things
IoTA	Internet of things Agent
IP	Intellectual Property
IP	Internet Protocol
ISO	International Standard Organization
IT	Information Technology
ITU	International Telecommunication Union
ITU-T	ITU Telecommunication Standardization Sector
JSON	JavaScript Object Notation
JWT	JSON Web Token
LTE	Long Term Evolution
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
MVP	Minimal Viable Product
NFC	Near-field communication
NGSI	Next Generation Service Interfaces
NSE	Network Service Entity
MSI	Management Support Interface
OASC	Open & Agile Smart Cities
OASIS	Organization for the Advancement of Structured Information Standards
OATH	Open Authentication
OAuth	Industry-standard protocol for authorization
ODF	Open Data Federation
ODMS	Open Data Management System
OIDC	OpenID Connect
OMA	Open Mobile Alliance
OMG	Object Management Group
PM	Person Month
PPM	Privacy Policy Manager
PEP	Policy Enforcement Point
PDP	Policy Decision Point
QoS	Quality of Service
RBAC	Role-Based Access Control
RDF	Resource Description Framework
REST	Representational state transfer
RFC	Request for Comments
RFID	Radio Frequency Identification
RPS	Receive Packet Steering
RTD	Research and Technological Development

RZ	Reference Zone
SAML	Security Assertion Mark-up Language
SC	SynchroniCity
SCADA	Supervisory Control And Data Acquisition
SDO	Standards Developing Organizations
SDK	Software Development Kit
SLA	Service Level Agreement
SNS	Simple Notification Service
SOA	Service Oriented Architecture
SOA- RM	(OASIS) Reference Model for Service Oriented Architecture
SOAP	Simple Object Access Protocol
SQS	Simple Queue Service
SR	System Requirement
SSH	Secure Shell
SSL	Secure Sockets Layer
SSO	Single Sign-On
STH	Short Time Historic
TCP	Transmission Control Protocol
TGP	Transports Publics Genevois
TLS	Transport Layer Security
UAA	User Management, Authentication and Authorization
UA	User Authentication
UC	Use Case
UDG	Universal Device Gateway
UI	User Interface
UML	Unified Modelling Language
URI	Universal Resource Identifier
URL	Universal Resource Locator
VM	Virtual Machine
VPN	Virtual Private Network
W3C	World Wide Web Consortium
WLAN	Wireless Local Area Network
WP	Work Package
WT	Work Task
WS	Web Service
WSDL	Web Services Description Language
WSN	Wireless Sensor Network
XACML	eXtensible Access Control Mark-up Language
XML	eXtensible Mark-up Language
XMPP	Extensible Messaging and Presence Protocol
XSD	XML Schema Definition

Contents

1	Introduction	10
1.1	Intended audience	10
1.2	Document structure	10
2	Scope and background	10
3	Reference platform deployment and element update	12
3.1	RZ Carouge (CH).....	12
3.1.1	Use case/service content updates.....	12
3.1.2	SynchroniCity framework deployment	13
3.1.3	Data sources and datasets available	13
3.1.4	Advanced Operation, maintenance process	15
3.1.5	Future improvements	15
3.2	RZ Helsinki (FI).....	16
3.2.1	Use case/service content updates.....	16
3.2.2	SynchroniCity framework deployment	16
3.2.3	Data sources and datasets available	16
3.2.4	Advanced operation, maintenance process.....	17
3.2.5	Future improvements	17
3.3	RZ Eindhoven (NL)	18
3.3.1	Use case/service content updates.....	18
3.3.2	SynchroniCity framework deployment	18
3.3.3	Data sources and datasets available	18
3.3.4	Advanced Operation, maintenance process	18
3.3.5	Future improvements	19
3.4	Manchester (UK).....	19
3.4.1	Use case/service content updates.....	19
3.4.2	SynchroniCity framework deployment	19
3.4.3	Data sources and datasets available	21
3.4.4	Advanced Operation, maintenance process	22
3.4.5	Future improvements	22
3.5	RZ Antwerp (BE).....	23
3.5.1	Use case/service content updates.....	23
3.5.2	SynchroniCity framework deployment	23
3.5.3	Data sources and datasets available	25
3.5.4	Advanced Operation, maintenance process	26
3.5.5	Future improvements	26

3.6	RZ Santander (ES)	27
3.6.1	Use case/service content updates.....	27
3.6.2	SynchroniCity framework deployment	30
3.6.3	Data sources and datasets available	32
3.6.4	Advanced Operation, maintenance process	33
3.6.5	Future improvements	34
3.7	RZ Porto (PT)	34
3.7.1	Use case/service content updates.....	34
3.7.2	SynchroniCity framework deployment	35
3.7.3	Data sources and datasets available	36
3.7.4	Advanced Operation, maintenance process	38
3.7.5	Future improvements	38
3.8	RZ Milan (IT).....	39
3.8.1	Use case/service content updates.....	39
3.8.2	SynchroniCity framework deployment	39
3.8.3	Data sources and datasets available	41
3.8.4	Advanced Operation, maintenance process	45
3.8.5	Future improvements	45
3.9	Non-RZ Seongnam (KR).....	47
3.9.1	Use case/service content updates.....	47
3.9.2	SynchroniCity framework deployment	47
3.9.3	Data sources and datasets available	49
3.9.4	Advanced Operation, maintenance process	49
3.9.5	Future improvements	50
4	Conclusion	50
5	References	52

List of Figures

Figure 1 Carouge components architecture	13
Figure 2 Manchester RZ framework deployment and EPs to access Manchester's resources	21
Figure 3 Antwerp RZ overview of technology	24
Figure 4 Antwerp RZ components architecture	24
Figure 5 POC CKAN- architecture within RZ Antwerp	27
Figure 6 Santander Scenario diagram.....	28
Figure 7 Santander use case orchestration.....	30
Figure 8 Santander RZ framework deployment and end points (EPs) to access Santander resources	32
Figure 9 Porto RZ framework deployment and end points (EPs) to access Porto's resources.....	36
Figure 10 Integration between Milan infrastructure and SynchroniCity Platform	40
Figure 11 User Service Application Screenshot	47
Figure 12 Seongnam Server Platform Architecture	48
Figure 13 The admin dashboard with KETI Parking Lot information	50

List of Tables

Table 1 Carouge RZ themes and services	12
Table 2 List of Entities used in Carouge Orion Context Broker.....	14
Table 3 Default values for Carouge Orion Context Broker.....	15
Table 4 Helsinki City Data Infrastructure Elements	17
Table 5 Eindhoven data sources.....	18
Table 6 Manchester RZ themes and services.	19
Table 7 List of available datasets in Manchester RZ	22
Table 8 Antwerp RZ themes and services.....	23
Table 9 Antwerp RZ Data sources and datasets	26
Table 10 List of available datasets in Santander RZ, compiled by Task 4.2	33
Table 11 Porto Atomic Services status.....	35
Table 12 Porto RZ updated service list	38
Table 13 Legacy Open Datasets and target NGSI Data Models.....	43
Table 14 Legacy API sources and target NGSI Data Models	44
Table 15 Seongnam - data to be uploaded	45
Table 16 SeongNam Non-RZ - List of datasets available	49

1 Introduction

This deliverable reports on the advance reference zones platform deployment and operational plan as carried out under WP2 (“IoT Enabled Framework for Smart Cities”). It is the final version of the previous D2.8 where all RZ has described their initial approach based on the SynchroniCity reference architecture elements, data models and deployment & operation plans.

D2.8 was built on an initial set of guidelines and plans for every RZ involved in the SynchroniCity project: with this document we aim to create a final report on the platform activities and elements which would play an important role for the Open Call participants also for the technical partners that act as the key actors in the whole project life cycle.

The information that has been collected and presented in this document is a direct source for the WP3 use case applications and thematic.

1.1 Intended audience

This document is relevant to the software engineers, programmers and developers who are directly involved in the development, participating effectively on the design and implementation of the SynchroniCity platform and the underlying components and sub-systems who want to know more about some technical information and specification intrinsic to the SynchroniCity platform.

At the technical level this document is relevant to: system architects; information systems designers; system developers and application developers; software engineers; integrators; other audiences who provide design services and applications using relevant standards and the recommendations of standards bodies like IETF, ITU, ISO, W3C, etc.

1.2 Document structure

The content of this document is structured into the following chapters:

- We start with the introduction section which includes purpose, overview and the intended audience.
- Section 2 provides an overview of what has been achieved with the D2.8 and what has been expected with the D2.9.
- Section 3 lists all the updates and changes that has been going on in each RZ. Each RZ section has been divided into 5 sub-sections where we have collected and documented the progress about the SynchroniCity platform deployment and technical activities that have been realized since the time we have delivered D2.8.
- Conclusion chapter summarizes the tasks that have been carried out by T2.5 and two deliverables that have been produced so far by describing the work performed and the conclusions reached.

2 Scope and background

As commented in the previous D2.8– (Section 2 - Scope and Background) - the aim of the SynchroniCity project is to build a distributed platform that provides services and functionalities related to smart city ecosystems to the cities. The SynchroniCity environment provides both third parties and application providers a set of services that are necessary to build applications related to a set of preselected themes and service domains that are in the

interest of the cities. To build the SynchroniCity platform and make it available to cities, it is necessary to perform technical activities such as development, integration and deployment tasks as well as operations and maintenance activities.

These technical activities mentioned above have been initially listed in the previous D2.8 as each RZ has documented their data sources, working/legacy environments and technical components and approach under 3 main areas:

- Reference architecture modules and RZ platform elements
 - o RZ legacy system overview
 - o High level theme and service domains
 - o City infrastructure and technical elements
- Integration and deployment process
- Operation and maintenance process

D2.8 helped RZs to define a clear strategy for adopting the SynchroniCity reference framework and technical methodologies to realize their use cases in respect of the guidelines defined by the project boundaries.

This document is an extension to D2.8 and intends to provide the advanced or updated activities regarding the interfaces, data model specifications, operation and deployment plans.

3 Reference platform deployment and element update

This section covers updates, changes and improvements around the platform elements, data models, use cases and operational plan of each RZ, these changes cover the period between D2.8 has been delivered and D2.9 was started.

Each RZ was asked to provide as much as practical information and updates regarding their way of integration with the SynchroniCity reference framework. There are mainly 5 sub-sections that have been covered by the RZs in this aspect:

- Use case/service content updates
 - o Specific use case element updates in each Reference Zone
- SynchroniCity framework deployment
 - o Final deployment picture of the SynchroniCity framework with the updated components and integrations
- Data sources and datasets
 - o Data-set and data-model updates
- Platform maintenance
 - o Platform maintenance updates
- Future improvements
 - o Planning for the short-term future activities

3.1 RZ Carouge (CH)

3.1.1 Use case/service content updates

The following Table 1 illustrates all the services and the associated infrastructure related to the Carouge RZ that is supporting the SynchroniCity use cases deployment and the Open Call developments.

Service	Service provider	Infrastructure
Smart parking	IEM	Parking slots with sensors
Noise monitoring	OrbiWise	Sound level sensors
Online schedule for the public transports	TPG	Bus stops
Off street parking	SITG	Public parking entrances and exits
Traffic flow observed	SITG	Road sections
Air quality observed	SABRA (Canton of Geneva)	Air quality for PM10, NO2 and O3

Table 1 Carouge RZ themes and services

In the context of the multimodal transportation theme, the Carouge reference zone is using the parking estimator service to determine the availability and the provision of the different parking slots across the smart city of Carouge. The same software of the parking estimator service is also installed in Carouge and its usage is currently investigated in the frame of the noise monitoring near the bars and the restaurants.

3.1.2 SynchroniCity framework deployment

The architecture of the Carouge reference zone is shown in the following Figure 1:

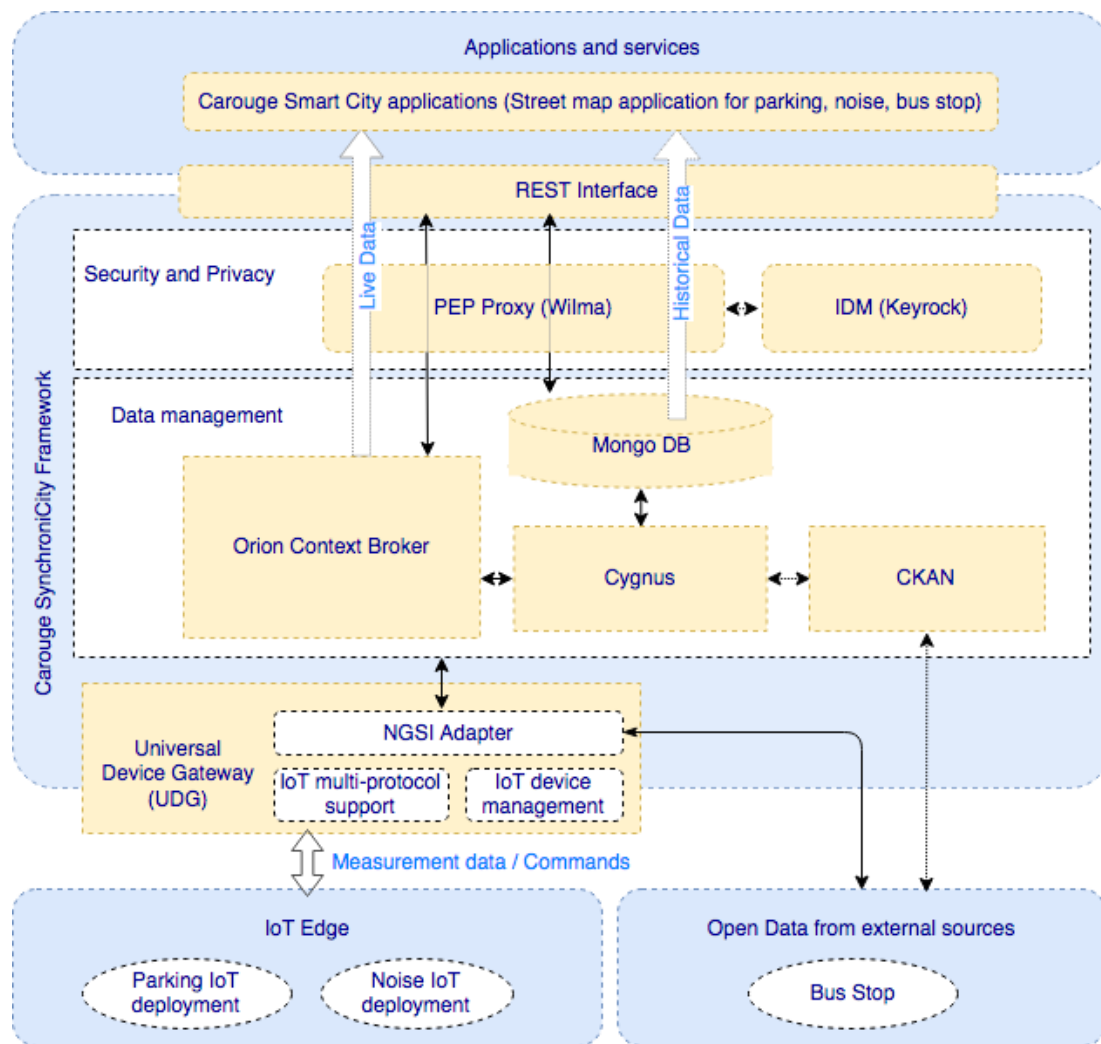


Figure 1 Carouge components architecture

The user access was improved the last months notably when the configuration of KeyRock and Wilma were modified. The KeyRock API is available through the URL¹. On the same way, the PEP Proxy, also known as Wilma, is reachable through the URL². Each specific application or service is also secured through Wilma and KeyRock like Orion Context Broker and Comet STH.

The utilisation of KeyRock and Wilma protects the Orion Context Broker of the Carouge reference zone and permits to avoid unauthorized people to access private or sensible data stored in the Orion Context Broker database.

3.1.3 Data sources and datasets available

All the data models of each kind of datasets were updated to correspond to the data models defined in the frame of the SynchroniCity project. The current data models used by the Carouge reference zone are described in the following Table 2:

¹ <https://keyrock.cityreport.org:443/>

² <https://wilma.cityreport.org:4000/>

Entity Type	Description	Fiware-ServicePath	Fiware-Service
OffStreetParking	A parking site with entries and exits. The update is every 15 minutes.	Carouge	/OffStreetParking
OnStreetParking	A street parking which contains one or more parking spots. The update is every 15 minutes	Carouge	/OnStreetParking
ParkingSpot	An individual parking spot.	Carouge	/ParkingSpot
AirQualityObservedPM10	Sensors detecting PM10 particles. The update is every 7 days	Carouge	/AirQualityObservedPM10
AirQualityObservedNO2	Sensors detecting NO2 particles. The update is every 24 hours	Carouge	/AirQualityObservedNO2
AirQualityObservedO3	Sensors detecting O3 particles. The update is every 24 hours	Carouge	/AirQualityObservedO3
NoiseLevelObserved	Noise pressure level sensors. The update is every 15 minutes.	Carouge	/NoiseLevelObserved
TrafficFlowObserved	Traffic flow conditions for a place at a time. The update is every 15 minutes.	Carouge	/TrafficFlowObserved
Vehicle	Real time tracking of vehicles that are used for public transportation (buses, trains, etc.). The update is every 30 seconds	Carouge	/Vehicle

Table 2 List of Entities used in Carouge Orion Context Broker

In general entities can be queried with:

```
curl http://orioncontextbroker:1026/v2/entities -H 'Fiware-Service: $FIWARE_SERVICE' -H 'Fiware-ServicePath: $FIWARE_SERVICEPATH'
```

Entity type filter can be specified if needed:

```
curl
http://orioncontextbroker:1026/v2/entities?type=$ENTITY_TYPE -H
'Fiware-Service: $FIWARE_SERVICE' -H 'Fiware-ServicePath:
$FIWARE_SERVICEPATH'
```

Default values to be used for making queries to Orion Context Broker is shown in Table 3:

Fiware-service	Fiware-ServicePath
carouge	/OffStreetParking

Table 3 Default values for Carouge Orion Context Broker

Generally, for date the format is ISO 8601 yyyy-MM-dd'T'HH:mm:ssZ.

Location is represented by GeoJSON geometry object "location": {"type": "Point", "coordinates": [longitude, latitude]}

3.1.4 Advanced Operation, maintenance process

The maintenance process is realized by the Swiss partners and concerns the monitoring of the server and the software components, the server and its virtual machines are constantly updated and maintained.

3.1.5 Future improvements

Carouge reference zone is improving the running of the services in the context of the WP3 on its own virtual machines, for the parking estimator service and the noise monitoring part. The data models for the datasets related to the IEM parking sensors are currently in phase of update and test in the Carouge reference zone, but IEM is working on its own servers and so, IEM cannot provide access to these sensors temporarily.

3.2 RZ Helsinki (FI)

Helsinki has been one of the open data pioneers in Europe and opened data since 2011.

The Helsinki Region Info share service³ aims to make regional information quickly and easily accessible to all. Essentially, HRI is a web service for fast and easy access to open data sources between the cities of Helsinki, Espoo, Vantaa and Kauniainen. The data published is mainly statistical, giving a comprehensive and diverse outlook on different urban phenomena, such as living conditions, economics and well-being, employment and transport. A good proportion of the data material offered by the service is GIS based. 600 data sets and 100 APIs have been published in HRI so far and amounts are increasing.

3.2.1 Use case/service content updates

The Helsinki RZ belongs to the Multimodal Transportation theme. Helsinki has long been supporting Open Data and Open APIs. The aim is to move out from closed silos and enable innovative services. In this view, authorities provide basic data, services and tools, on top of which third parties and cities can create their novel services and applications

3.2.2 SynchroniCity framework deployment

SynchroniCity platform is currently running and accepting requests.

To provide NGSI support, as SynchroniCity requirement, Helsinki RZ deploys its own instance of the FIWARE Orion Context Broker. This instance is currently up and running, offering the complete NGSIv2 interface published by FIWARE.

Helsinki RZ publishes its datasets according agreed FIWARE data models.

The process to deploy software agents which are used to consume, transform and publish legacy data is to be developed with the technical subcontractor.

Currently data is delivered to the Context Broker using custom proof-of-concept level gateway (Django web application), which receives data from sensors via HTTP. Application converts custom data formats to NGSI API v2 format and then sends it to the Context Broker. The NGINX Proxy handles basic authentication and allows only POST and PATCH HTTP verbs.

The Helsinki RZ framework is based on the integration of the RZ data sources via the NGSIv2 API in pursue of a large-scale piloting facility. These sources are read, parsed and stored via sensor agents in a fast memory cache for real time usage. Then, Northbound APIs provide this data via NGSIv2 API, both for the RZ Pilot use case and as general open data.

The implementation of this service manager may be based on either the SynchroniCity platform with selected FIWARE components such as Orion Context Broker.

3.2.3 Data sources and datasets available

IoT infrastructure (shown in Table 4) in Helsinki is taking form as devices are being deployed in the city. Devices are generally operated by the organizations using the data in their operations. The data is planned to be imported into the SynchroniCity platform via custom software agents that subscribe to the data sources. This presupposes that all data that we wish to import to the SynchroniCity platform must be provided openly by the respective organizations.

³ <http://dev.hsl.fi>

Data Name	Data Model	Key attribute	Data provider (owner of sensors)
Air quality	AirQualityObserved	pm2.5, pm10, humidity, temperature	FVH
		SO2, NO2, O3, CO, TRS, pm2.5, pm10, humidity, temperature	FMI
Noise	NoiseLevelObserved	LAeq;LAmx	FVH
Bike Hiring	BikeHireDockingStation	availableBikeNumber	HSL

Table 4 Helsinki City Data Infrastructure Elements

3.2.4 Advanced operation, maintenance process

Maintenance is done by Aalto and contains server health monitoring, critical security updates, automatic resource monitoring with alerts and external watchdog processes, which check periodically that services are up and running.

3.2.5 Future improvements

Helsinki has a strategic support for Open Data and Open Source software. Helsinki will pursue this by engaging Open Source software in many activities and enabling open access to new data sources. Helsinki will also continue supporting standardized and harmonized network interfaces, although not all data sources will be available in all possible APIs. During projects such as SynchroniCity, Helsinki will continue evaluating these APIs, related tools and their capabilities to provide as good Open Data service as possible.

Regarding extensions for the current Open Data in SynchroniCity, we are interested in data management and processing. For example, the current fleet of noise sensors will be increased, and new data made available. One specific issue regarding noise data is edge computing. How should noise data be handled? A typical noise sensor averages noise over time, yielding simply a number in decibels. However, this does not consider the frequency distribution or other characteristics of the signal. What irritates our citizens? What causes most health issues? We wish not simply providing open data but provide useful data. Edge computing and related data processing methods are here essential. We also wish to augment the data with related metadata, like the requirements for INSPIRE compliant data.

To yield predictions or facilitate policy and decision making, we also need to provide historical views to our data. We are looking into proper validation, storage and analysis methods regarding air quality data, which has the challenge of a very large scale. Now, it appears best to simply store each hourly update in the native NetCDF form, although we are looking at database solutions.

The SynchroniCity Platform with its FIWARE component Cygnus might suit for storing historical noise data. First, we will improve the reliability and general robustness of the existing pipeline for noise data (from sensors to Orion). We will next investigate the possibility of using Cygnus for storing historical noise data.

For the needs of the Open Call, the security framework will be improved. Potentially we will need to manage write access, although for read access we wish to leave all data sources open for everyone. We already have KeyRock installed and running and will add necessary components as necessary.

3.3 RZ Eindhoven (NL)

3.3.1 Use case/service content updates

The Eindhoven use-cases revolve around promoting cycling as an emission-free method of urban and regional mobility. Eindhoven has a lot of bicycles and many cars, which create conflicts mainly on intersections on the ring road. For both modalities, we aim to provide an optimal flow.

The use cases on data-driven bicycle mobility aim to improve bicycle mobility in the city by giving positive user feedback, improving flow for cyclists while balancing motorized traffic, and giving mobility advisors deeper insight into cycling patterns and flow.

The two use cases consist of giving feedback to cyclists and mobility advisors based on the numbers of cyclists passing. This is in line with the previous report D2.8. As an update, the sensors used can in theory provide more detailed information on type of transport used (e.g. Distinction between e-bikes and regular bikes). We will evaluate this and incorporate it in the use case if possible.

In the four advanced use cases, we will link bicycle sensors to intersection Traffic Light Controllers to optimize flow and get even more insight. Now of reporting (March '19) we are focused on getting the required platform modules operational. We expect limited content updates on the advanced scenarios after this activity is concluded.

3.3.2 SynchroniCity framework deployment

The Synchronicity framework in Eindhoven pilot has been deployed in a data center of Atos NL, where an instance of Context Broker is available to process the data from thermal cameras and air quality indicated below. The defined use cases will access the data collected by CB to provide the described services as indicated in D3.6.

3.3.3 Data sources and datasets available

The set of data sources for the Eindhoven RZ is shown in the Table 5 below.

Data Source	Source Type	Description	NGSI Data Model	Accessibility
Air quality data	JSON	Real time air quality data	AirQualityObserved	Public
Thermal camera data	JSON	FLIR thermal camera for collecting real time bicycle information	TrafficFlowObserved	Private

Table 5 Eindhoven data sources

3.3.4 Advanced Operation, maintenance process

The Eindhoven pilot has been supported by Atos (Spain and NL) and Heijmans partners. Atos Spain has been developing the described use cases and facilitating the adoption of Synchronicity framework due to their technical role in the project as service provider. Heijmans has been providing the thermal camera data through its FLIR provider. Atos NL is the responsible for pilot deployment and operation at node in Netherlands, and it will be in charge of future maintenance of the pilot once

project was over. The pilot is part of a wider strategy of the city around digitalization and wellbeing policy.

3.3.5 Future improvements

More advance use cases will be defined based on other data sources which will enrich the current pilot and will take fully advantage of SynchroniCity framework.

3.4 Manchester (UK)

3.4.1 Use case/service content updates

The following Table 6 illustrates all the services and the associated infrastructure related to the Manchester RZ that is supporting the SynchroniCity use cases deployment and the Open Call developments.

Service	Service Provider	Infrastructure
Agile governance	Bronze Labs	n/a
Active Travel Insights	Vivacity iSensing Tracsis	5 x WIFI sensors 5 x cameras Both on street, and attached to existing street furniture
Smart Cycling	See.Sense BT	250+ smart bike lights distributed directly to users.
Smart Waste Management	Neighbourly	n/a

Table 6 Manchester RZ themes and services.

In the context of the Community Policy Suite theme, Manchester RZ is using the agile governance service to quickly adapt the deployment of policy to the local needs of a specific area of the city.

In the context of the Human Centric Traffic Management theme, Manchester RZ is using the sustainable mobility service fed by information from cameras and environmental sensors to encourage more non-motorized transport on the roads.

3.4.2 SynchroniCity framework deployment

As for the deployment of the SynchroniCity framework, Manchester RZ followed the architecture and components detailed in deliverable D2.8.

Orion Context Broker version 2.0.0 supports our data management core operations, providing all the NGSiv2 compliant API and the access to all the Manchester RZ's context information. Cygnus version 1.10 connects the Orion Context Broker to a Mongo DB via both a Mongo DB sink and a

STH sink (the latter used to aggregate context data) to store historical data which are accessible via the STH Comet API.

As for the security layer, the deployment relies on the FIWARE KeyRock IdM version 7 using the OAuth 2.0 interface. It supports users' management and the definition of the roles required to grant access to context information and historical data. The PEP Proxy complementing the IdM is the Wilma Plus, developed by DigiCat and based on the FIWARE PEP Proxy Wilma. Wilma Plus implements a role-based access control (RBAC) in order to authorize users to access data.

Manchester RZ's IoT infrastructure has been adapted to be SynchroniCity compliant by converting the original data models (based on HyperCat) in the SynchroniCity ones. To that aim, NGSI adaptors have been implemented in Node-RED by DigiCat.

An instance of the SynchroniCity IoT data marketplace fully integrated with the components mentioned above has been deployed in Manchester RZ. The marketplace hosts all the data offerings available and provides mechanisms to buy and access them.

Figure 2 shows a diagram of the Manchester RZ framework deployment and the end points (EPs) to access Manchester's resources. The EPs have been enabled to allow SynchroniCity open call pilots to access and exploit the framework. They are listed in the following:

- EP1: provides access to Manchester RZ IdM and allows end users to create their own SynchroniCity user profiles to access the other endpoints. It supports the OAuth 2.0 flow to authenticate users.
- EP2: provides access to context information stored in the Orion Context Broker. It fully supports the NGSIv2 API. Protected with OAuth 2.0.
- EP3: provides access to historical data via the STH Comet API. Protected with OAuth 2.0.
- EP4: refers to the instance of the SynchroniCity IoT data marketplace developed by DigiCat running in Manchester RZ. Data providers can register their data offerings in the marketplace and make them discoverable by data consumers (service developers).

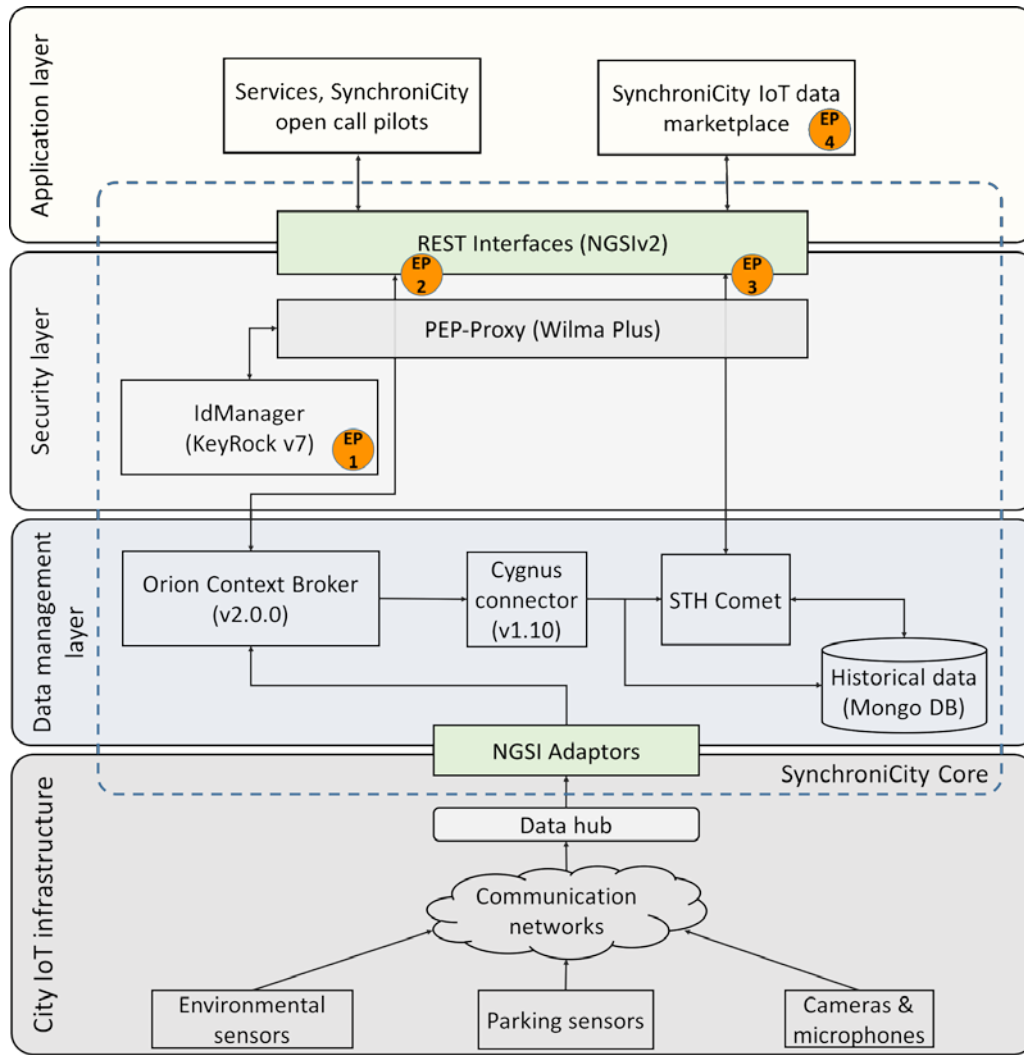


Figure 2 Manchester RZ framework deployment and EPs to access Manchester's resources

3.4.3 Data sources and datasets available

Main Manchester RZ data sources were deeply described in D2.8. To complement the use case and to expand the available information, we have added two more relevant source:

- People, cycle and vehicle counting: Provided by one of the open call pilots and will be made SynchroniCity compliant during the development of the pilot. They will complement the same type of data already SynchroniCity compliant and available in Manchester RZ's IoT infrastructure.
- Points of interest (Poi): Provided by one of the open call pilots and will be made SynchroniCity compliant during the development of the pilot.

The data model refinement process is documented on D2.3[2] and so, the "valid" SynchroniCity data models can be followed in the SynchroniCity data models repository⁴. Manchester RZ manages all its available datasets according these defined data models, exposing them through its SynchroniCity NGSI interface.

⁴ <https://gitlab.com/synchronicity-iot/synchronicity-data-models>

Entity Type
AirQualityObserved
BikeHireDockingStation
CrowdFlowObserved
NoiseLevelObserved
OffStreetParking
OnStreetParking
TrafficFlowObserved
WeatherObserved

Table 7 List of available datasets in Manchester RZ

Table 7 presents all the currently available datasets that can be read through the Synchronicity NGSI API.

3.4.4 Advanced Operation, maintenance process

Manchester City Council (MCC) is the coordinating partner for the Manchester RZ. Although the city is the public sector partner on SynchroniCity, the RZ will access assets and services that are provided by a range of public and private sector partners. The assets from two major projects, Triangulum5 (2015-20), funded by the EU's Horizon 2020 Programme, and CityVerve (2016-18), funded by the UK innovation agency Innovate UK.

The management of these programs is discreet from SynchroniCity, but the coordination across all three projects is managed by the Resources and Programmes team within City Policy, Partnerships and Research within Manchester City Council.

The Manchester instance of the SynchroniCity framework is hosted and supported by Digital Catapult. This entity is responsible, during SynchroniCity project to execute and maintain the whole framework and each of its components. In the same line, Digital Catapult will provide the required support to third parties when using the platform. Any updates on this SynchroniCity framework and/or its components will be executed/supervised also by Digital Catapult.

3.4.5 Future improvements

Continuing to develop the operation of the city data platform provided by BT which federates datasets from several sources. This platform⁶ was developed during the CityVerve project. The platform will continue in this form until the end of 2019.

The hosted SynchroniCity framework from Digital Catapult will integrate further with the BT platform during 2019, but with an intention that either/both can be developed separately or together to sustain Manchester's data architecture following on from the project.

Separately the city is working with University of Manchester to develop a sustainable 3-year urban data observatory and the aim is to utilize the Synchronicity platform either as part of this, or to be supported by it.

⁵ <http://www.triangulum-project.eu/>

⁶ <https://portal.bt-hypercat.com>

3.5 RZ Antwerp (BE)

3.5.1 Use case/service content updates

The following Table 8 illustrates all the services and the associated infrastructure related to the Antwerp RZ that is supporting the SynchroniCity use cases deployment and the Open Call developments.

Service	Service provider	Infrastructure
Bike hiring stations	ClearChannel	Bike hiring stations usage sensors
Bike traffic flow observed	Ring Ring	Mobile app for bike usage

Table 8 Antwerp RZ themes and services

In the context of the Human-Centric-Management theme, the Antwerp reference zone is using the Route visualisation service to visualise Bicycle patterns within the city of Antwerp.

3.5.2 SynchroniCity framework deployment

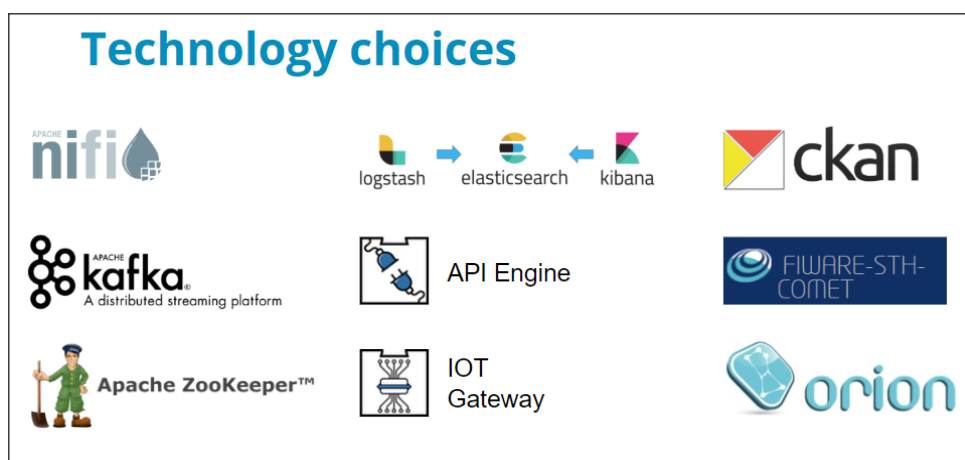
The IoT-architecture of the Antwerp reference zone is aligned with the IT-strategy of Antwerp which forms the basis of the City Platform ACPaaS

^{vii} (Antwerp City Platform as a Service).

ACPaaS is a collection of reusable software blocks called engines, such as notification, digital signature, workflow, payments and profile management. These blocks are ready to use and can be combined in various ways to get a new application with the desired functionalities. Or linked with other existing systems.

As a generic IoT back-end an engine cluster was constructed based on the components defined in the SynchroniCity reference architecture.

The various open source software components shown in the Figure 3 below. It was a conscious choice by the organisation to manage the cost of the architecture, be vendor independent, etc.



^{vii} <https://youtu.be/w3kQlxww20k>

Figure 3 Antwerp RZ overview of technology

The architecture of the Antwerp reference zone is built with these technology stack shown in the following Figure 4:

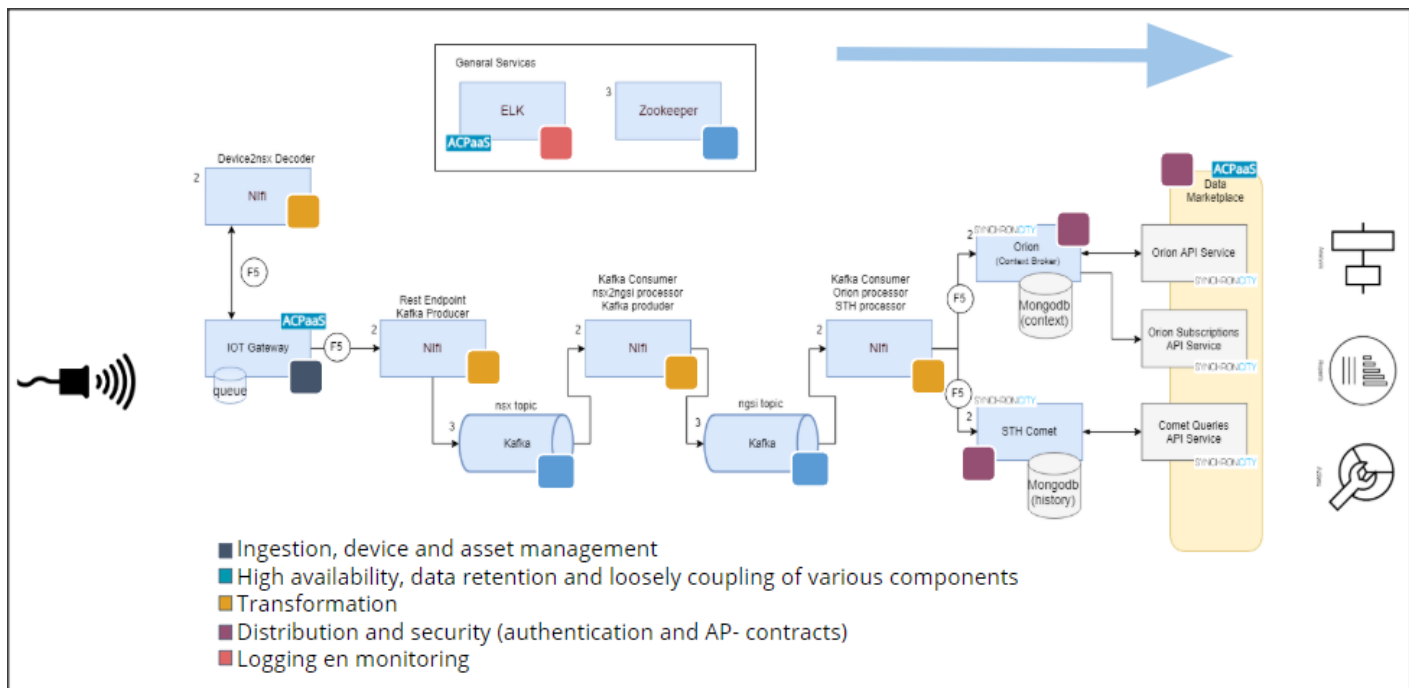


Figure 4 Antwerp RZ components architecture

The different components are further described in detail below:

- ELK Stack: Part of the ACPaaS-platform and is a powerful solution for logs, searching, indexing and building dashboards
- Nifi: Facilitates the dataflow between different software systems and has powerful transformation possibilities and can handle a lot of protocols
- Apache Zookeeper and Kafka: Mandatory to set up Kafka in a clustered architecture. Apache Kafka allows publishing streams of messages and subscribe to them; these messages can also be saved. Kafka is the component that protects us from data loss due to its clustered design, disassociating various parts from each other. The Kafka architecture has many advantages and explaining them all would take us too far. I added an article at the back for reference.

Two FIWARE components are used to externally access the data:

- Orion Context Broker: Maintains the last state of an entity and allows certain queries over devices, their attributes, types and even geographically
- STH Comet: Maintains a time series of the changes within an entity and allows it to be retrieved with windowing and (limited) aggregations

Two ACPaaS-engines are reused in the IoT architecture:

- API Engine: Consists of a marketplace and an underlying API gateway, which protects discoverability and access to the datasets
- The IoT Gateway: Initial IoT legacy component which takes care of the ingestion and device management within the broader platform

3.5.3 Data sources and datasets available

The set of data sources for the Antwerp reference zone is shown in the Table 9 below. It states the domain, data source, status, type and data model.

The publication of the various data sources is an ongoing transition process within the organisation whether to publish the data as private/open and what is the added value of doing so.

Domain	Data Source	Status	Type	Data model
Environment	Air Quality sensor VMM	Published	Open	AirQuality
Environment	Weerstations - Smart Zone	Published	Open	AirQuality
Transportation	RingRing - Bike tracking app data	Ongoing	Open	Traffic flow observed
Transportation	Velo bike hiring stations - Occupancy rate	Ongoing	Open	Bike hiring docking station
Waste Management	BigBelly - filling rate and temperature	Ongoing	Private	Waste container
Water Management	Smart Fountain sensordata	Medium term	Private	tbd
Street Lighting	Smart lighting sensor - Cronos	Medium term	Private	Street lighting data models
Street Lighting	Smart lighting sensor - Smartnodes	Medium term	Private	Street lighting data models
Environment	Pluviometers	Medium term	tbd	Weather observed
Environment	Flowmeters	Medium term	tbd	Weather observed
Environment	Groundwater level meter	Medium term	tbd	tbd
Environment	Content measurement of tanks	Medium term	tbd	tbd
Waste Management	Filling level of garbage trucks	Long term	Private	tbd
Parking	Loading and unloading quays	Short term	Private	Parking harmonized data models
Environment	Air quality - SmartZone	Short term	Open	AirQuality
Traffic	Traffic light data	Long term	Private	tbd

Waste Management	Sorting streets	Short term	tbd	Waste container
Street Lighting	Smart lighting (Basketball case, weather, ...)	Medium term	tbd	Street lighting data models
Parking	Events from e-parking signs	Medium term	tbd	Parking harmonized data models
Environment	Pedestrian crossing events (camera images)	Medium term	tbd	tbd

Table 9 Antwerp RZ Data sources and datasets

3.5.4 Advanced Operation, maintenance process

The operation and maintenance process is monitored using various components in the ELK-stack and integrated in the dev-ops process within the organisation. It is realized by collaboration of the support team in combination with external partners in 2nd and 3rd line support. The various components are constantly monitored and maintained and will be updated when necessary.

3.5.5 Future improvements

- POC regarding the publication of large (historical) IoT Datasets.

For the publication of large (historical) IoT datasets the RZ Antwerp is investigating the added value of CKAN based on the following reasons:

- Available possible NGSI plugins
- The go-to applicative for open source open data publication in the market
- Security compatible with the legacy systems within the RZ Antwerp

The architecture of the POC CKAN-setup is shown in the following Figure 5:

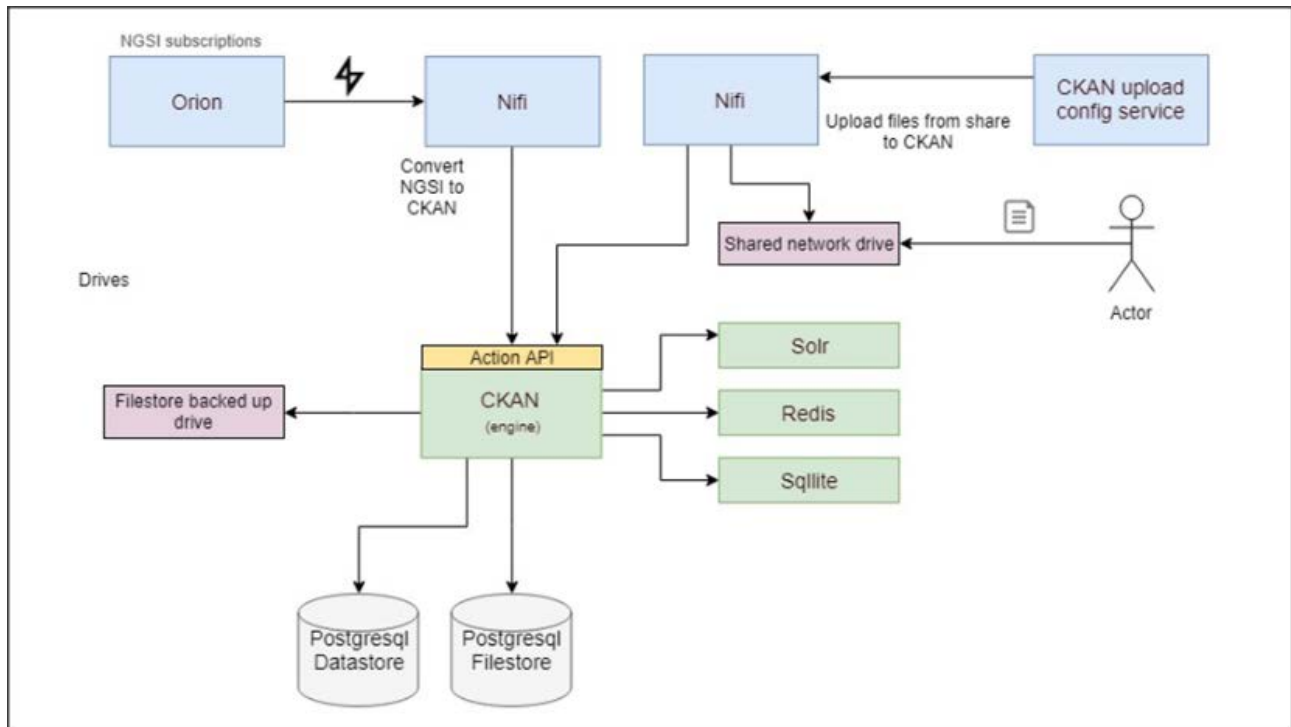


Figure 5 POC CKAN- architecture within RZ Antwerp

- Expose additional datasets

At this moment only, a small number of datasets are exposed through the IoT data broker/marketplace.

It is the ambition of the RZ to increase the number of published datasets on the various internal and external marketplaces to align with the data strategy of the city and operational data owners.

- Further mature the existing IoT Backend within the RZ Antwerp

To cover future and potential use cases regarding traffic light actuation, autonomous drone swarm surveillance and image processing for various operational processes the IoT Backend must step up to a next maturity level.

This not only to minimize the latency of the message/event throughput but also covering the reliability, resilience and robustness of the end to end IoT-process. Also, the DevOps-process will be continuously improved where/when necessary.

3.6 RZ Santander (ES)

3.6.1 Use case/service content updates

As described in D3.5[6], the Santander's RZ main use case combines two complementary scenarios focused on multimodal mobility/transportation:

- Smart Parking app that helps the end user to quickly find a parking lot, combining current parking status with historical available data to extrapolate in time the parking possibility in different city areas.
- Multimodal Navigation within Santander City exploits urban buses info (lines, stops, routes), arrival estimations and mobility to provide a multimodal route adapted to user's requirements (time, cost, carbon footprint, etc).

Figure 6 represents this combination, showing the interaction between the outcomes of parking app and the incomes of the multimodal navigator.

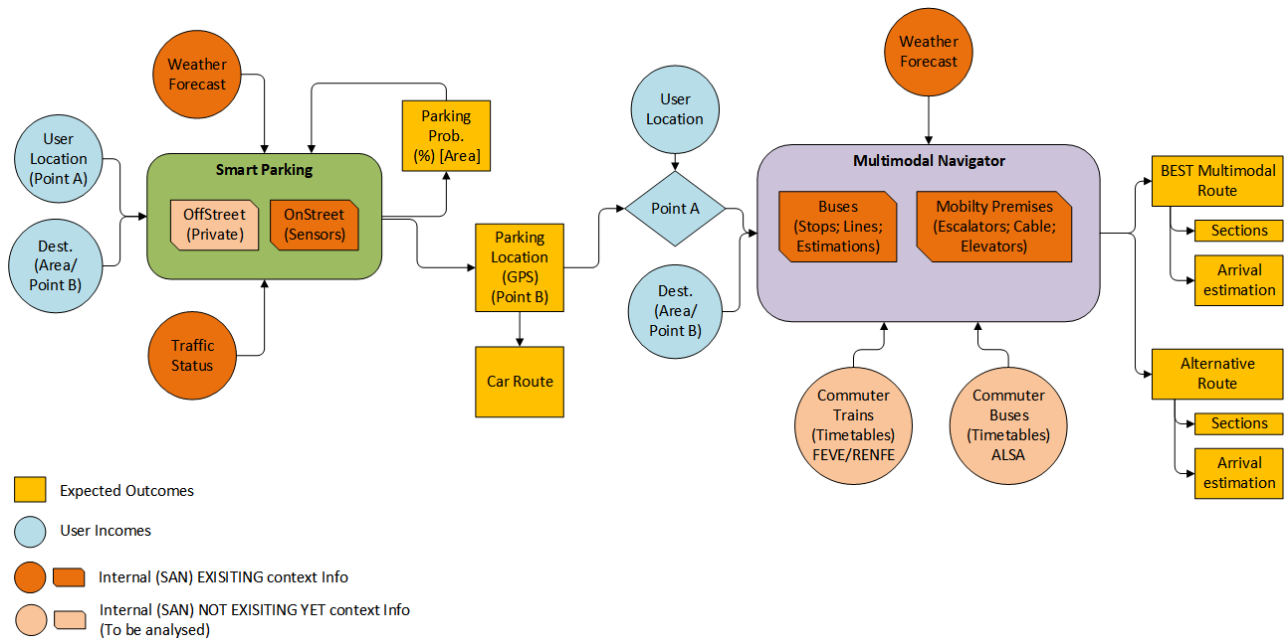


Figure 6 Santander Scenario diagram

Embedded in these two applications, a set of atomic services developed within SynchroniCity and described in detail in D3.3[5] provides the required core to implement the use case functionalities:

- **Routing Service:** Based on OpenTripPlanner^{viii}, it finds suitable routes combining taxi stops, buses info and bicycle routes (among other available possibilities) between 2 points and according user's preferences.
- **Parking estimator:** Using Artificial Intelligence (AI) techniques with historical and current parking data, it predicts the probability of finding a free parking spot in a specified area and in a given time window.
- **GTFS Fetcher:** Extracting data from GTFS TransitFeedFile entities (see D2.3[2]) it imports GTFS uploaded files (static timetables and related info) into OpenTripPlanner platform, making this available for routing calculation (within the Routing Service)
- **GTFS-RT loader from NGSi:** Service consumes Real Time Urban Transport entities (so far ArrivalEstimations) and generates GTFS-RT feeds from it. These feeds will be later consumed by the Routing Service to calculate/update the requested routes.

The final Santander use case and the involved atomic services will be fed by the IoT edge and the datasets deployed in Santander City. These will be available through the SynchroniCity Context Broker (NGSiV2 API) and following SynchroniCity data models. In this sense, the involved datasets are:

- **SmartParking:**
 - o OnStreetParking, an on street, free entry (but might be metered) parking zone which contains at least one more adjacent parking spots.
 - o ParkingSpot, an individual, usually monitored, parking spot.
- **Public Transport Information System (from Santander City).** These datasets are migrating to UrbanMobility^{ix} data models, to complement/feed GTFS compatible apps (and SynchroniCity Atomic Services):

^{viii} <http://www.opentripplanner.org/>

^{ix} <https://gitlab.com/synchronicity-iot/synchronicity-data-models/tree/master/UrbanMobility>

- o busLine entity (migrating to “gtfs:Route”), a bus line in which there are bus stops (migrating to “gtfs:Stop”) (mandatory) and buses (optional).
 - o busStop (migrating to “gtfs:Stop”) entity, representing a bus stop which belongs to a bus line.
 - o Estimation (migrating to “ArrivalEstimation”) entity, providing estimation of arrivals of buses in relation to a combination of busStop-busLine. This will map GTFS-RT (Real Time) data.
 - o GtfsTransitFeedFile, that provides the link to a GTFS transit feed file including all static data (timetables) related to e.g. a transportation company/service.
- Bike Hiring System (TUSBIC):
 - o BikeHireDockingStation entities, with location and status (e.g. the number of available bikes/free docks, etc.) of the docking stations.

To this minimum required datasets, new ones will be added, during the use case deployment to enrich the available info and the services/functionalities offered. These are the Mobility premises entities, that will add new routing possibilities, the WeatherObserved and WeatherForecast entities or the Traffic information, to provide routes fitted to users' requirements or better estimations.

The complete orchestration of the data models, data sources and atomic services required to build the Santander use case is drawn in Figure 7. This also shows the relationship with the Santander SynchroniCity framework, updated in next section.

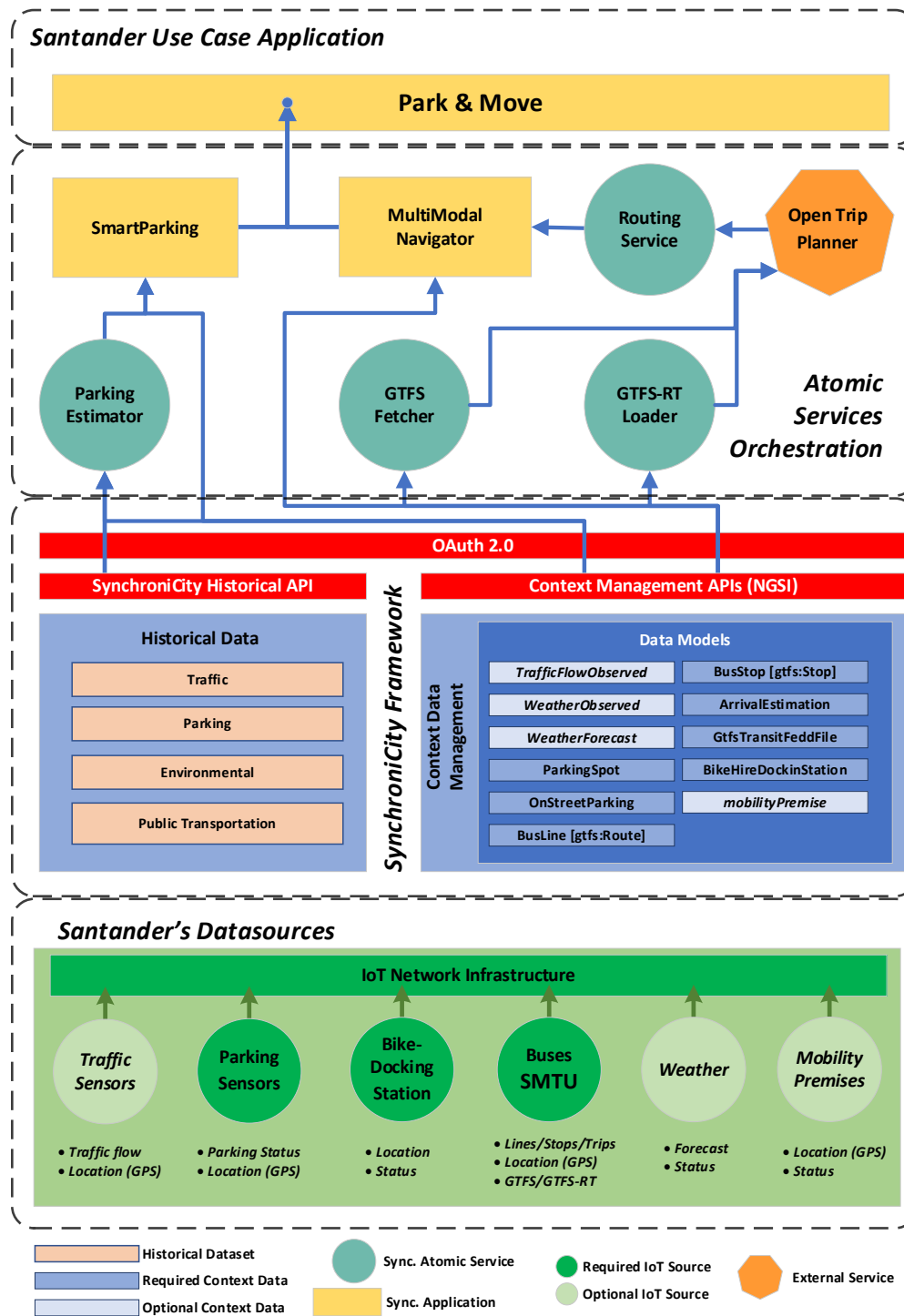


Figure 7 Santander use case orchestration

3.6.2 SynchroniCity framework deployment

Referring to Santander's SynchroniCity deployed framework, the reference zone followed the architecture and components detailed on D2.8, adapting this to the components' new versions released and our internal testing, to guarantee the maximum support to SynchroniCity agreed interoperability points and standards. In this line:

Santander's IoT Edge practically remains the same as described in D2.8. SmartSantander IoT agents are currently being revisited and upgraded to improve their efficiency and adaptation to SynchroniCity data models. They all will be migrated to node.js environment.

Orion Context Broker version 1.15 supports our data management core. This provides all the NGSiv2 compliant API, and the access to all the context information related to Santander City. Cygnus version 1.9 connects our context broker with a Mongo database that keeps all the historical data. We've developed our own implementation of the SynchroniCity Historical API on top of Mongo APIs and Cygnus configuration capabilities. This is available through the Synchronicity GitLabx.

The security layer, and so, the OAuth 2.0 interface, relies on FIWARE KeyRock IdM version 6. This supports the users' management and the definition of the roles, required to grant access to context information and historical data. The used PEP proxy that complements the IdM is the Wilma Plus, supported by DigiCat and based on FIWARE Wilma component. For the access policies, it implements a Role-based access control (RBAC) developed also by DigiCat.

On top of this, Santander has deployed an instance of the SynchroniCity Data Marketplace, fully integrated with our IdM and our data access (context information and historical data) what provides the end user with a complete window to all what's deployed in Santander RZ. It also provides the different ways to register into the Santander Framework and retrieves required datasets to create their apps.

To allow SynchroniCity pilots and Open Call experiments to access and exploit Santander Framework, a set of End Points have been enabled. These are:

EP1^{xi}, accesses to Santander RZ Identity Manager. This will allow end-users to create their own SynchroniCity user profiles to access the rest of Santander EPs. It also supports the OAuth 2.0 flow to authenticate users.

EP2^{xii} represents the access to the Context Information stored in Santander Context Broker. It fully supports the NGSiv2 API. Protected with OAuth 2.0.

EP3^{xiii} supports the Synchronicity defined Historical API, containing the historical data referred to the context information managed by the Context Broker. Protected with OAuth 2.0.

EP4^{xiv} shows the Santander instance of the Synchronicity Data Marketplace, developed by DigiCat. This provides a quick view of all datasets available (context information and historical data) in Santander RZ, as well as the mechanisms to buy and access them.

Deployment and EPs are shown in Figure 8.

^x <https://gitlab.com/synchronicity-iot/platform-deployment-docker>

^{xi} <https://auth.san.synchronicity-iot.eu>

^{xii} <https://context.san.synchronicity-iot.eu>

^{xiii} <https://historical.san.synchronicity-iot.eu/>

^{xiv} <https://marketplace.san.synchronicity-iot.eu>

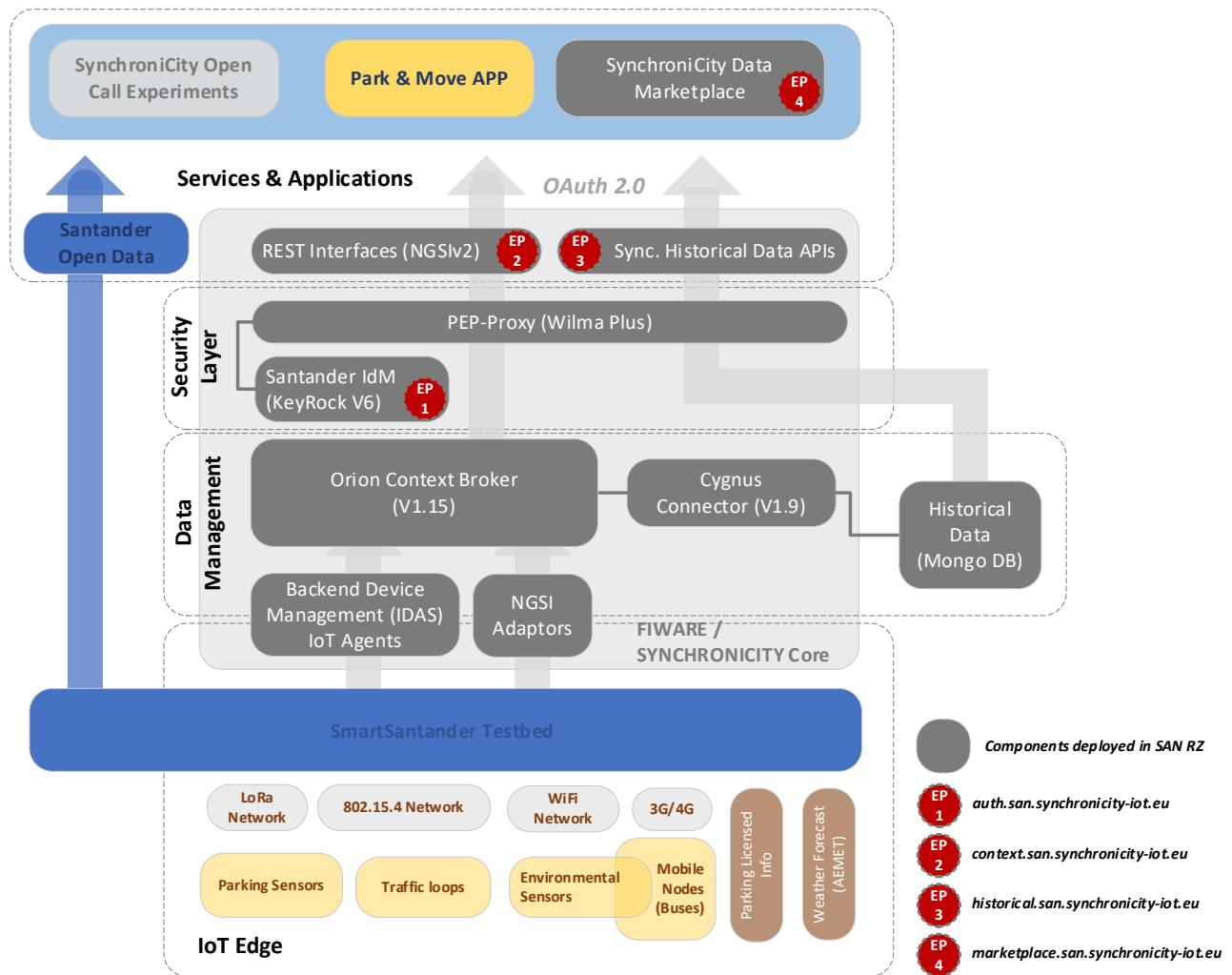


Figure 8 Santander RZ framework deployment and end points (EPs) to access Santander resources

3.6.3 Data sources and datasets available

Main Santander RZ data sources where deeply described in D2.8. To complement the use case and to expand the available information, we've added two more relevant sources:

- Santander City Urban Bus Service (SMTU^{xv}) GTFS and GTFS-RT (Real Time) data:
 - o GTFS Feed File: a zip file compliant with General Transit Feed Specification^{xvi} that contains the buses timetables and its linked location info, among others.
 - o GTFS-RT: updated (real time) by the SMTU service, provides the arrival estimation times of the different buses to the corresponding bus stops.
- Weather Forecast, referred to Santander City and provided by the AEMET^{xvii}. This is distributed on several forecast entities, referred to today, tomorrow, current and next hour and current and next 6 hours' time slot.

These datasets are currently on internal validation process before been added to the Santander SynchroniCity Framework.

The data model refinement process is documented on D2.3[2] and so, the "valid" SynchroniCity Data models can be followed in the SynchroniCity data models' site^{xviii}. Santander RZ manages all its

^{xv} <http://www.tusantander.es/>

^{xvi} <https://developers.google.com/transit/gtfs/>

^{xvii} <http://www.aemet.es>

^{xviii} <https://gitlab.com/synchronicity-iot/synchronicity-data-models>

available datasets according these defined data models, exposing them as open data through its SynchroniCity NGSI interface..

Entity Type
AirQualityObserved
Beach
BikeHireDockingStation
BusArrivalEstimation
BusLine
BusStop
Device
GreenspaceRecord
Museum
NoiseLevelObserved
OnStreetParking
ParkingSpot
PointOfInterest
Shop
TrafficFlowObserved
Vehicle
WeatherForecast
WeatherObserved

Table 10 List of available datasets in Santander RZ, compiled by Task 4.2

Table 10 presents all the currently available datasets that can be read through the Synchronicity NGSI API.

3.6.4 Advanced Operation, maintenance process

Santander City IoT edge, covering the IoT sensors, devices and IoT infrastructure played in SynchroniCity mostly belongs to the City of Santander. In this sense this infrastructure will be evolved and maintained by the Santander Municipality (and corresponding services, as the SMTU). Some minor deployments, such the LoRa network and linked devices, currently belong to the University of Cantabria, so the maintenance activities related to these infrastructures will rely on this entity.

The Santander instance and shared datasets of the SynchroniCity framework are hosted and supported by the University of Cantabria. The University of Cantabria will provide the required

support to third parties when using the platform. Any updates on this SynchroniCity framework and/or its components will be executed/supervised also by the University of Cantabria.

3.6.5 Future improvements

SynchroniCity framework core components are deployed in Santander Reference Zone, these are, the Context Broker, the developed IoT Agents (as Smart Santander legacy sources' adaptors to comply with Synchronicity interoperability requirements), the Identity Manager, security interface and the historical data storage blocks, will be fully supported and maintained during SynchroniCity and after the project ends. Santander City will rely on this framework for future deployments so, in this sense:

- For framework components, new versions will be tested and updated, always obeying and enforcing the defined interoperability points:
 - o Orion Context Broker will be updated to version 2.0, and in the future, NGSI-LD will be supported
 - o QuantumLeap^{xix} environment and FIWARE Draco^{xx} components will be evaluated to evolve the current deployed historical data framework
 - o FIWARE IdM KeyRock will be updated to version 7 to support new integrations with different authentication/authorisation entities, such as IDAS^{xxi} that will expand the framework capabilities
- For data models, Santander will follow the guidelines described in D2.3[2] to polish existing datasets and new ones coming for new IoT deployments. The enrichment of our shared datasets is one of our priorities.
- For SynchroniCity deployed use cases, Park & Move will be exploited beyond SynchroniCity. New datasets (such WeatherForecast or AirQualityObserved) will be analysed so they could be added to the developed predictive algorithms. This will enrich the use case and will give the chance to new applications and scenarios.

3.7 RZ Porto (PT)

Since deliverable D2.8, some important changes to Porto's infrastructure have been implemented. Not only new components have been installed (such as the ones responsible for security), but also due to policy changes, some infrastructure components were set to be unavailable in time for the project (such as the ones supporting bike sharing).

This section presents the status for the Atomic Services, IoT architecture and data models.

3.7.1 Use case/service content updates

Porto is focused on two use cases, one for assisting mobility (Multimodal Transportation), and one for policy validation (Community Policy Suite).

For policy validation, the use case is completely handled by the company Bronze Labs. The development is under way^{xxii}, and Porto is providing the appropriate specification, support and data. This application doesn't use any Atomic Services and no special data is required (apart from the one already available at the SynchroniCity platform).

As for assisting mobility, Porto aims to provide a multimodal transportation application (MMT) for route planning and navigation, using different means of transportation, providing bus/metro timetable information, providing lines and stops information, etc., all based on user's preferences.

^{xix} <https://quantumleap.readthedocs.io/en/latest/>

^{xx} <https://fiware-draco.readthedocs.io/en/latest/>

^{xxi} <https://ec.europa.eu/digital-single-market/trust-services-and-eid>

^{xxii} The final solution is a customization of the framework (www.bronzelabs.co.uk/framework42)

As mentioned in deliverable D3.3, Porto decided to use Digitransit^{xxiii} – an open source journey planner – for the base of its MMT application and, at this time, it has a minimal working instance^{xxiv}.

This instance relies information from (Atomic) Services as described in the Table 11 below.

Service	Atomic	Description	Status
Routing Service	Yes	Based on OpenTripPlanner ^{xxv} it finds suitable routes combining taxi stops, buses info and bicycle routes (among other available possibilities) between two points and according to user's preferences.	Installed
(reverse) geo-coder Service	No	Based on Pelias ^{xxvi} , taking an input text, such as an address or the name of a place, it returns a latitude/longitude location on the Earth's surface for that place (and vice-versa).	Installed
GTFS Fetcher	Yes	Extracts data from GTFS TransitFeedFile entities ^{xxvii} and imports GTFS uploaded files (static timetables and related info) into the OpenTripPlanner platform, making this available for routing calculation (within the Routing Service).	Installed
GTFS-RT loader from NGSI	Yes	This service consumes Real Time Urban Transport entities (so far, ArrivalEstimations) and generates GTFS-RT feeds from it. These feeds will be later consumed by the Routing Service to calculate/update the requested routes.	To be installed

Table 11 Porto Atomic Services status

3.7.2 SynchroniCity framework deployment

As for the SynchroniCity framework deployment, Porto followed the architecture and components detailed on deliverable D2.8, with a few changes and updates, namely:

- Installation of QuantumLeap, as a replacement of the original component for holding the historical data (Cygnus).
- Installation of an instance for publishing some of datasets as open data, namely CKAN.
- Installation of a visualization tool based on WireCloud, which is not supposed to be a visualization tool to be used by the end-user. Although it is being used for creating maps and supporting data preview in the open data portal, it mainly supports internal usage (manual checks).
- Implementation of the following security layer components:
 - o FIWARE KeyRock IdM for users' management and the definition of their roles, required to grant access to context information and historical data.
 - o A PEP proxy to enforce authentication provided by KeyRock. For this, we're using Wilma Plus, supported by DigiCat (which is based on FIWARE Wilma component implementation).

The Figure 9 below is an update of the original one presented in D2.8. It also shows the components responsible for implementing the End Points (EPs):

^{xxiii} <https://digitransit.fi/en>

^{xxiv} <http://mmt.portodigital.pt>

^{xxv} www.opentripplanner.org

^{xxvi} <https://github.com/pelias/pelias>

^{xxvii} (Catalogue of OASC Shared Data Models for Smart City domains)

- The context broker^{xxviii}
- The short-term historic data (initially installed, but to be deprecated soon)^{xxix}.
- The long-term historic data, all historic data is stored^{xxx}. It will be served by the current short-term historic end-point, once it is deprecated (for backwards compatibility)
- The marketplace component. As previously mentioned, it is set to be installed^{xxxi}
- The identity manager, responsible for authentication of the users (with user/password) and applications (with API keys)^{xxxii}

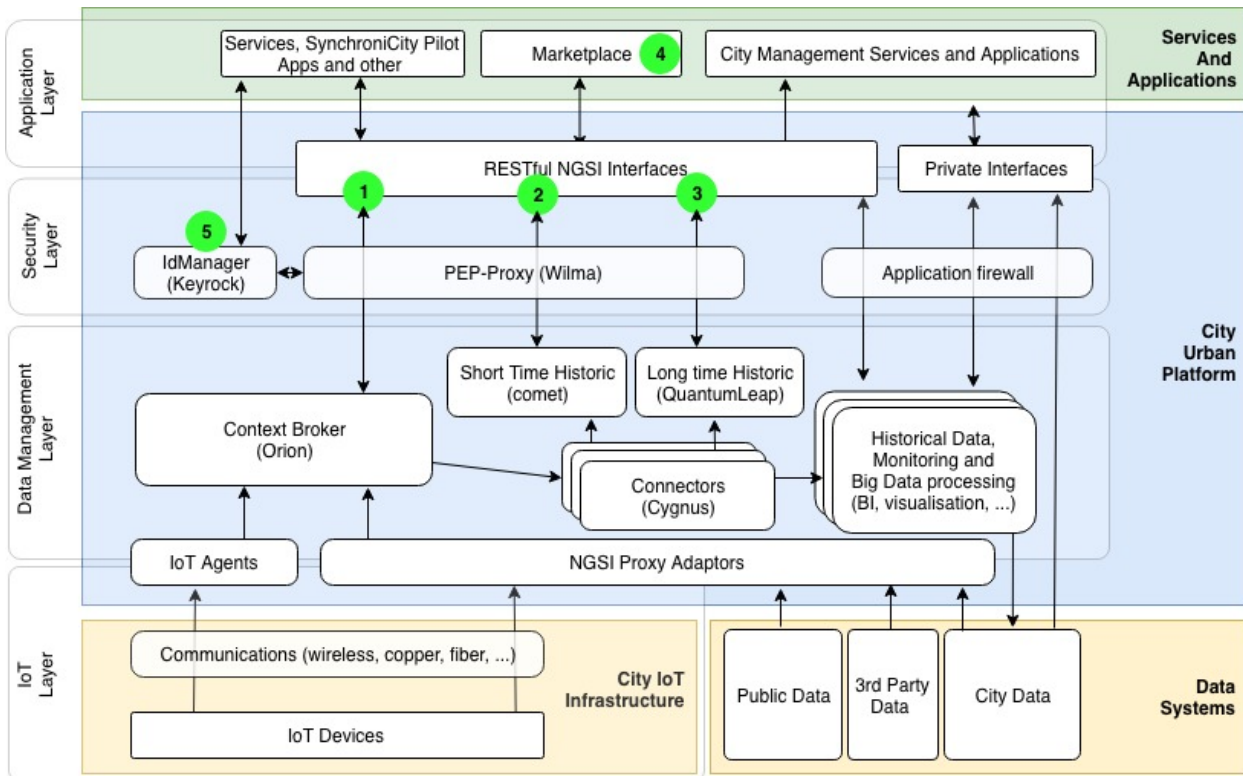


Figure 9 Porto RZ framework deployment and end points (EPs) to access Porto's resources.

3.7.3 Data sources and datasets available

Given that the list of data sources initially presented in D2.8 have major changes, here is an updated list of services shown in Table 12 below.

Data type	Data model	Description and source
Environmental monitoring	AirQualityObserved	Air pollutants (PM2.5 and PM10 particles, ozone, nitrogen dioxide and carbon monoxide), available through the Porto's environmental sensor network (ESN).
	WeatherObserved	Meteorological parameters (solar radiation, ultraviolet radiation, wind direction, wind speed, rainfall, atmospheric pressure, air temperature and air relative

^{xxviii} <https://broker.fiware.urbanplatform.portodigital.pt>

^{xxix} <https://history-data.fiware.urbanplatform.portodigital.pt>

^{xxx} <https://ql.urbanplatform.portodigital.pt>

^{xxxi} <http://marketplace.urbanplatform.portodigital.pt>

^{xxxii} <http://idm.urbanplatform.portodigital.pt>

		humidity), available through the Porto's environmental sensor network (ESN).
	NoiseLevelObserved	Noise, available through the Porto's environmental sensor network (ESN).
	WeatherForecast	Weather forecast for the city of Porto, provided by IPMA ^{xxxiii} . This is distributed on several forecast entities, referred to every hour of the current and next day (today and tomorrow, respectively) and three-hour timeslots for the day after (the day after tomorrow).
Transportation	TrafficFlowObserved	Vehicle counts data, provided by the Mobility Department of the Municipality of Porto, thought a NGSI adapter.
	Vehicle	Used to represent specific vehicles belonging to the Municipality of Porto, provided by the Fleet Management Department, thought a NGSI adapter.
POIs and events	PointOfInterest	Represents any relevant point in the city which might be of interest to inhabitants, visitants or tourists; provided by Porto Digital's own database (and updated by the Tourism Department of the Municipality of Porto).
	Event, as specified in link ^{xxxiv}	Represents the cultural events happening in the city, provided the Tourism Department of the Municipality of Porto.
Urban Mobility	gtfsTransitFeedFile	Holds the location and valid period of existing GTFS zip files (static timetables), provided and managed by Porto Digital.
	BusArrivalEstimation	Holds the estimate time for buses/metro to arrive at a stop.
	GTFS:line	Represents the public transportation line (bus or metro), provided by the transportation service companies.
	GTFS:stop	Represents the public transportation stops (bus or metro), provided by the transportation service companies.
Device	Device	Holds the location, specs and usage parameters for the wireless access points, provided by Porto Digital.
Electrical Vehicle Charging Station	EVChargingStation	Holds the location, specs and usage parameters for the electrical vehicle charging stations, provided by the Fleet Management Department of the Municipality of Porto.

^{xxxiii} www.ipma.pt

^{xxxiv} <http://schema.org/Event>

Table 12 Porto RZ updated service list

Please note that “Bike Sharing” model was removed, as now it is not possible to confirm whether the information will be available.

Except for the explicitly mentioned data models, all are the ones approved by the SynchroniCity group, mentioned in the SynchroniCity data models Git repository^{xxxv}.

3.7.4 Advanced Operation, maintenance process

Porto's team responsible for deploying and managing the SynchroniCity framework is especially concerned with its performance and reliability.

A set of identified changes need to be done, and they will be released soon.

3.7.5 Future improvements

As previously stated, most of the components of the framework have been deployed. Porto will continue testing its components, with a special focus on performance and reliability. Short and medium-term improvements will include:

- Upgrading all the components to the latest stable version, with special focus on:
 - o Major version update of Orion Context Broker to 2.x, which was delayed since NGSI v1 is deprecated in this version.
 - o Update QuantumLeap.
- Install the NGSI-LD Wrapper^{xxxvi}, to start testing the usage of NGSI-LD.
- Install the Marketplace component.
- Test and evaluate Draco^{xxxvii}.

As for the data models, there are still a few improvements to be made for the existing data, namely the one for the Events which hasn't been adopted yet. On this matter, new datasets will be available to support the Open Call Pilots.

Finally, regarding Porto's use cases, Porto will proceed implementing the missing functional/non-functional requirements for the Multimodal Transportation app and will continue to assist Bronze Labs in developing the Community Policy Suite application.

^{xxxv} <https://gitlab.com/synchronicity-iot/synchronicity-data-models>

^{xxxvi} https://github.com/Fiware/NGSI-LD_Wrapper

^{xxxvii} <https://fiware-draco.readthedocs.io/en/latest/>

3.8 RZ Milan (IT)

3.8.1 Use case/service content updates

Milan RZ is involved into the developing of two internal applications upon the Human Centric Traffic Management (HCTM) theme and the Multi Modal Transportation (MMT) theme.

The HCTM application is a supporting tool for bicycles routes planners of the Municipality that will exploit data coming from the SynchroniCity platform shown on the graphic interface of the GeoPortal, owned by the Municipality. The HCTM application will exploit the Route service to create desired lines from Origin-Destination matrix of bike sharing usage.

The MMT application is a Multimodal Navigator, especially designed for the needs of disabled people, that will be fed with information coming from the SynchroniCity platform integrated with data coming from other external data-sources like the E015 ecosystem^{xxxviii}. The MMT application will be designed to exploit the Parking Estimation service when the parking sensors will be deployed in the city (note: Parking sensors were originally planned to be deployed in May 2018. Then deployment was postponed to December 2018).

At mid of January 2019 both HCTM and MMT development is at an initial stage of development.

Beyond the internal applications, Milan RZ is supporting four different pilot applications developed by SMEs selected by the Open Call. These applications are:

- **Blue Alpaca:** This application consists in a chat bot controlled by an Artificial Intelligence that will answer to citizen questions about data coming from the city. While during the Open Call selection stage the application was designed to exploit Event data, at the end of 2018 the Municipality announced that the Event API involved were no-longer fed, so in January 2019 the application needs to be re-scoped to exploit Point of Interests data (i.e. pharmacies positions and opening time);
- **Kimap:** This solution provides useful information about accessibility in the city for disabled people. The SME in charge to develop this solution will firstly create a baseline ground-truth dataset by exploring the city with his agents. Data generated will feed the SynchroniCity platform and eventually could be used to integrate the MMT application;
- **Kiss my bike:** This solution involves anti-theft devices for bikes that will provide GPS position data that will feed the Synchronicity platform. Eventually these data could be used for the HCTM application;
- **Linc:** This solution involves the deployment of electric usage sensors in buildings, to get spectral appliance signatures of electricity usage. Generated sensor data will feed the SynchroniCity platform.

3.8.2 SynchroniCity framework deployment

Starting from first stage of the two-step deployment plan agreed between Municipality of Milan and ENG, as reported in D2.8, the main challenge was to adapt existing legacy data sources to NGSI entities, compliant to data models defined within the SynchroniCity project.

The interoperability platform of Milan infrastructure provides data sources through different technologies, such as REST (JSON) or SOAP (XML) web services, as well as Milan's datasets coming from the CKAN Open Data Portal.

Therefore, an initial phase of study of the available services was performed, according to the usability needs and licenses issued by the Municipality; this led to the development of several NGSI Adapters that feed a dedicated instance of the Context Broker for Milan.

^{xxxviii} <http://www.e015.regione.lombardia.it/>

The second deployment step, the second hypothesis proposed by ENG and reported in D2.8, involved the instantiation of core components of the SynchroniCity reference architecture in a dedicated platform for Milan RZ. The platform has been deployed as a fully dedicated VM instance running on the ENG Cloud infrastructure, and it has been integrated with the legacy WSO2 platform of the Municipality of Milan. Another critical step was the integration of legacy WSO2 security layer, to use the OAuth2 protocol required by SynchroniCity reference architecture. This approach, adopted to keep the legacy security systems instead of using the SynchroniCity ones (Pep Proxy and KeyRock IdM), allowed to re-expose SynchroniCity APIs through the Milan APIStore [9], exploiting its own authorization mechanism.

Figure 10 shows how the main components of SynchroniCity architecture have been integrated with the existent Milan platform (City Legacy WSO2 Platform).

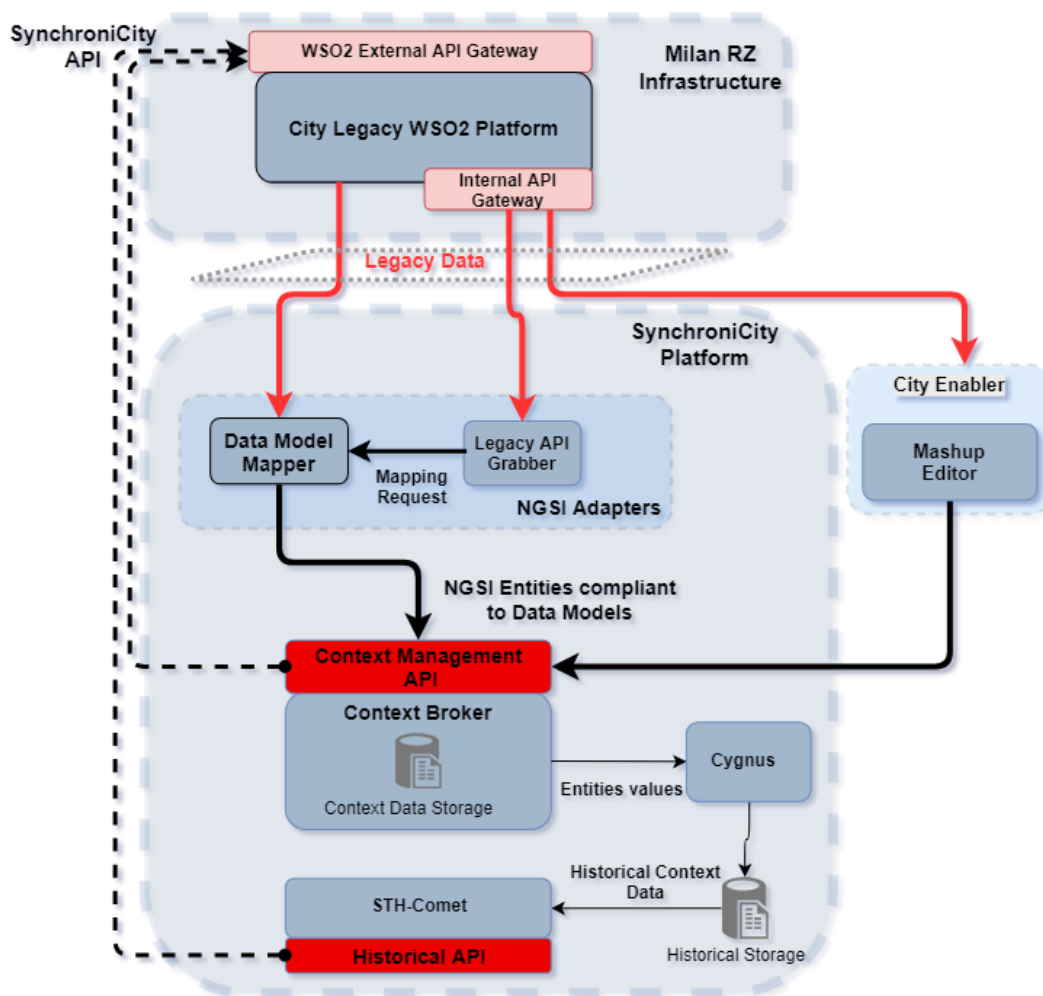


Figure 10 Integration between Milan infrastructure and SynchroniCity Platform

As shown in the Figure 10, the connection bridge between the two platforms is represented by the NGSI Adapters components. They oversee transforming legacy data, coming from the Milan's platforms (WSO2 API Store and Open Data Portals), to the entities in NGSI format, which must be compliant to the agreed SynchroniCity Data Models. They consist of three components:

- Data Model Mapper
- Legacy API Grabber
- Data Mashup Editor from City Digital Enabler

The Data Model Mapper [10] is a NodeJS tool that allows to convert input data, such as JSON and CSV files, into NGSI entities format, according to the JSON schema of the specific SynchroniCity Data Model. It allows to map each object/row of the source data to a NGSI entity, which will be validated and automatically written or updated to the specific instance of the Context Broker. To correctly map the input fields/columns to the JSON fields that compose the specific NGSI Data Model, a mapping file, whose JSON syntax follows simple rules, must be compiled and provided to the tool. Specific JSON mapping files were created for every kind of source data provided by Milan RZ platform. It could happen that, sometimes, original data sources need to be pre-processed before being mapped into an NGSI Model: it is the case, for instance, when text values must be translated from Italian to English, or when it is not possible to determine a 1 to 1 mapping between original and NGSI attributes in a data-models, requiring the aggregation of several field values.

The Legacy API Grabber is the subcomponent in charge of invoking Legacy APIs, provided by Milan RZ, with appropriate authenticated requests, by using access tokens provided by the Municipality, according to the Oauth2 protocol. Then, retrieved legacy data are sent in a mapping request, through HTTP protocol, to the Data Model Mapper. The Grabber is composed of following sub-components:

- Lamppost API Grabber: NodeJS tool used to invoke Milan Lamppost API and to retrieve resulting Weather measurement, every 5 minutes.
- RL (Regione Lombardia) API Grabber: Java tool used to retrieve Air Quality and Weather measurements provided by the Socrata APIs of Regione Lombardia Open Data Portal [12]
- Generic API Grabber: Tool used to invoke other generic APIs, such as the E015 ones.

The following two typologies of legacy data, coming from Milan RZ platform, have been mapped:

- Open Data: CSV, JSON and GEOJSON files, directly issued to the Mapper.
- Legacy APIs: firstly, invoked by the Legacy API Grabber component and then sent to the Data Model Mapper.

Thus, a dedicated Context Broker instance will hold all the NGSI Context Entities resulting from the mapping process and validated against the SynchroniCity Data Models.

The data historicizing has been achieved by using subscription mechanism provided by the Context Broker and Cygnus Data Connector. A subset of mapped NGSI entities flowing to the Context Broker, are notified to Cygnus, which in turn stores their values to the Historical Data Storage. Then, a custom STH-Comet component can read these data to expose them as raw historical data. This is a customized version of the FIWARE STH-Comet that implements the Historical Data API defined within the SynchroniCity project.

Both Context Management (NGSI) and Historical APIs are exposed by corresponding SynchroniCity components, but only accessible by the Milan WSO2 API Manager (API Store), to be secured and publicly re-exposed with endpoints hold by the Municipality of Milan. These endpoints indeed will be the final entry points for SynchroniCity APIs implemented in the Milan RZ deployment.

The followed approach meets the initial Municipality's requirement of avoiding changing either its own internal data models or the WSO2 Identity Manager, which could impact on existing processes; at the same time, the integration achieved allows to expose the existing legacy data, translated and managed by SynchroniCity core components, following the SynchroniCity Data Models and APIs.

3.8.3 Data sources and datasets available

As anticipated in the previous section, the mapping process made it possible to transform following legacy data sources to NGSI Context entities:

- Open Data: CSV, GeoJSON and JSON files retrieved from the Open Data platform of Milan Municipality [11]
- Legacy API: Lamppost API from Milan Municipality, Air Quality and Weather API from Regione [12]

Table 13 reports the list of specific Open Datasets and the corresponding target NGSI Data Models to which they have been mapped.

Data Source	Source Type	Description	NGSI Data Model	Accessibility
BikeMe Hire Docking Stations	GeoJSON	Bike Sharing Station data	BikeHireDockingStation	Public
Bike Sharing Racks	CSV	Bike Sharing Station data	BikeHireDockingStation	Public
Pharmacies	GeoJSON	Information about Pharmacies position and contacts	PointOfInterest	Public
Archaeological evidences	CSV	Information about Archaeological Artefacts position and museum locations	PointOfInterest	Public
Artworks	CSV	Information about Archaeological Artefacts position and museum locations	PointOfInterest	Public
Cultural heritages	CSV	Information about Cultural Heritages position and details	PointOfInterest	Public
Retail activity	JSON	Information about Retail Business Activities position and details	PointOfInterest	Public
Kiosk	GeoJSON	Information about Newsstands position and details	PointOfInterest	Public
Restricted Traffic Area	GeoJSON	Information about Milan Restricted Traffic Areas (ZTL)	RestrictedTrafficArea	Public
Public/Multimodal Parking	GeoJSON	Information about public and multimodal Parking	OffStreetParking	Public
Kindergartens	GeoJSON	Information about Kindergartens position and details	PointOfInterest	Public
Primary/High Schools	GeoJSON	Information about Primary, Secondary	PointOfInterest	Public

		and High Schools position and details		
Parks and gardens	GeoJSON	Information about public parks and gardens position and details	Garden	Public
Sport fields and centers	GeoJSON	Information about sport fields and centers position and details	PointOfInterest	Public
Drinking fountains	GeoJSON	Information about drinking fountains position	PointOfInterest	Public
Public Toilets	GeoJSON	Information about public toilets position	PointOfInterest	Public
General Registry Office	GeoJSON	Information about General Registry Offices position and details	PointOfInterest	Public
Cemeteries	GeoJSON	Information about Cemeteries in Milan	PointOfInterest	Public
Swimming Pools	GeoJSON	Information about Swimming Pools position and details	PointOfInterest	Public
Electric Vehicle Charging Stations	GeoJSON	Information about public EV Charging Station position and details	EVChargingStation	Public
Car Sharing Stations	Open Dataset	Information about public Car Sharing Station position and details	CarSharingStation	Public

Table 13 Legacy Open Datasets and target NGSI Data Models

Table 14 reports the list of specific Legacy API sources and the corresponding target NGSI Data Models to which they have been mapped.

Data Source	Source Type	Description	NGSI Data Model	Accessibility
Toponomastica API	SOAP/XML	Information about Milan places and street names	Road/RoadSegment	Public

Lamppost API	REST/JSON	Information about humidity, pressure and temperature from 7 installed weather sensors, every 5 minutes	WeatherObserved	Public
Regione Lombardia Meteorological API	Socrata API	Information about position of Meteorological Stations placed in Milan metropolitan Area	PointOfInterest	Public
Regione Lombardia Meteorological API	Socrata API	Meteorological measurements in Milan metropolitan Area, every 6 hours	PointOfInterest	Public
Regione Lombardia Air Quality API	Socrata API	Information about position of Air Quality Stations placed in Milan metropolitan Area	PointOfInterest	Public
Regione Lombardia Air Quality API	Socrata API	Air Quality measurements in Milan metropolitan Area, every 6 hours	PointOfInterest	Public

Table 14 Legacy API sources and target NGSI Data Models

To complete the overall overview about data provided by Milan RZ, the following Table 15 shows data that is currently not available but will be uploaded in the SynchroniCity platform within February 2019 for the developing of the internal applications. Some of these datasets are not public and will be used only for internal usage.

Data Source	Source Type	Description	NGSI Data Model	Accessibility
Bike Routes	GeoJSON	Bike routes	Road/RoadSegment	Internal / Private
Blu / Yellow lines – Parking slots	GeoJSON	Information about public parking slots (for resident or under payment) localization	OnStreetParking	Internal / Private
Bus and Metro lines and station	Open data	Static localization and information about bus and metro lines and stations	Urban Mobility	Public

Metro and bus time tables	JSON	Seasonal public transportation time tables	Urban Mobility	Public
Local trains lines and Stations	RL GTFS	Static localization and information about local trains and stations	Urban Mobility	Public
Car sharing stations	GeoJSON	Localization and information about car sharing stations in Milan	CarSharing Station	Public

Table 15 Seongnam - data to be uploaded

3.8.4 Advanced Operation, maintenance process

During the lifespan of the SynchroniCity project, the maintenance process is realized by both ENG and the Municipality of Milan sides. ENG, hosting the cloud instance of the platform for Milan RZ, is in charge to constantly maintain and update the SynchroniCity platform.

The Interoperability Department of the Municipality of Milan is in charge to publicly expose the APIs of the platform and authorize access, by means its API Store, to users for project internal testing and data-validation according to WP4, or internal/external Pilots applications. Issues and troubles experienced by the users are collected by the Municipality of Milan and reported to ENG, if related to the platform.

3.8.5 Future improvements

Regarding the data availability, currently, the Municipality of Milan is continuing its process of data-discovery in order to enrich the data offer for the Pilots, since new datasets are continuously discovered or specifically requested by the SMEs winning the Open Call (e.g. at mid-January 2019 one of the pilots is asking whether is possible to have the dataset of the location of family doctors in the Reference Zone and the Municipality started the process to get the datasets and make it available).

Regarding the technical side, the focus will be in the improvement of the Legacy API Grabber, for the Generic API part. This module will be more configurable, to easily invoke additional services with generally different interfaces (not only the E15 ones), without the need of changing the code implementation.

The Data Model Mapper will be also extended to use a mapping dictionary that will enable to do a mapping also in the values of the fields. This requirement emerged during the mapping of the legacy data of the Municipality, where the source category fields had custom values (in Italian) that had to be translated (in English) into values of the Factual Taxonomy[13], according to the PointOfInterest specification. Furthermore, more advanced mapping directives will be implemented; such as, the “skipIf”, which skips the mapping of a field if source fields have a specified (forbidden) value. The adoption of a more powerful mapping syntax that could better support these advanced features, such as YAML, will also be evaluated, to be used in addition to the existing JSON notation of the current mapping files.

Finally, is under discussion the possibility of implementing a module that will be able to automatize the historicization of selected datasets, either during the mapping or for those already saved as NGSI entities in the Orion Context Broker. Such module, thus, would manage the subscription mechanism between the Orion Context Broker and the Cygnus Data Connector.

During the testing phase, the Municipality will evaluate how to support the Pilot's applications, and, more generally, to maintain the overall SynchroniCity platform, after the end of the project.

3.9 Non-RZ Seongnam (KR)

Seongnam, one of the smart city testbeds in the SynchroniCity, is focusing on deploying testbeds using both FIWARE and oneM2M standard. The testbed provides services to the citizens in the theme of multimodal transportation, more detailed explanation is provided in D2.8

3.9.1 Use case/service content updates

Seongnam features a commuter parking service which recommends to the users regarding the parking lot availability and provides information about local bus schedule, and metro train schedules.



Figure 11 User Service Application Screenshot

SynchroniCity Korean consortium intended to serve identical data to user in two different platforms. As a server platform, oneM2M and FIWARE are chosen. The oneM2M, global IoT standard, is widely used IoT standard in Korea, and FIWARE is generally used in Europe. Two smartphone applications have been developed. One of the applications retrieves data from the oneM2M based platform while the other gets its data from the Orion Context Broker.

3.9.2 SynchroniCity framework deployment

Seongnam used the base architecture provided by the SynchroniCity and extended it to interoperate oneM2M standard platforms. Some of the new modules that Seongnam used with the base architecture are Mobius a oneM2M platform, Mca-NGSI^{xxxix} Adaptor, Mca-CKAN Adaptor and CKAN.

^{xxxix} In oneM2M standard, Mca is communication flow between an application entity and a server platform entity

In the platform architecture, CKAN[Ref] has been added. It helps users to obtain the offered data set such as parking lot, intercity bus, metro information in CKAN web interface.

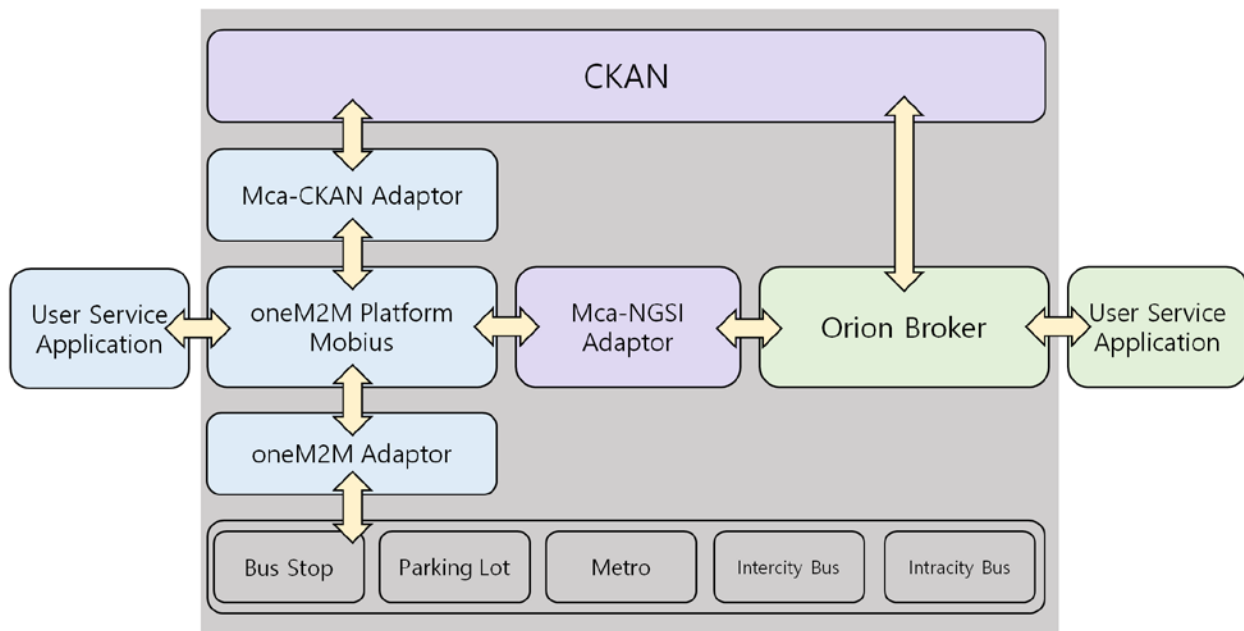


Figure 12 Seongnam Server Platform Architecture

Figure 12 illustrates Seongnam SynchroniCity Framework. Once the data is measured or collected at the sensor level, it is stored in the Mobius by oneM2M Adaptor. The Mca-CKAN adaptor and the Mca-NGSI Adaptor pushes data to CKAN and Orion Context Broker respectively. Users can view and obtain data itself from CKAN. Currently implementation of data collection from testbed, and publishing data to oneM2M platform and Orion Broker is finished and we are in the process of completing implementing application service to control the testbeds via the platform.

Descriptions for each module are as follow:

Mobius: Mobius is an open source IoT server platform based on the oneM2M standard. Mobius provides common services functions as middleware to IoT applications of different service domains. In oneM2M standards, common services functions include data management, device management, discovery, group management, registration, security and subscription/notification.

Mca-NGSI Adaptor: Mca-NGSI Adaptor is an interworking proxy for the Mobius and Orion Broker. Note that interworking proxy is the term defined in oneM2M which basically translates protocols from/to oneM2M and another. Leveraging subscription/notification feature in Mobius, whenever the data is created/updated in the Mobius, it sends notifications to the Mca-NGSI Adaptor. The Mca-NGSI Adaptor then modifies the data format to fit Orion Context Broker and requests to store it to the Orion Context Broker over NGSI.

Mca-CKAN Adaptor: This adaptor gets the data from Mobius via notifications like the Mca-NGSI Adaptor. However rather than re-formatting the data, it publishes the data to CKAN with predefined privacy settings and under predefined datasets and organisations.

CKAN: CKAN is an open source software tool for making open data websites. Once data is published to CKAN, users can use its features to browse and find the data they need and preview it using maps, graphs and tables.

3.9.3 Data sources and datasets available

Six data models are defined for the commuter parking service deployed in Seongnam. Some data models that referenced are extended to accommodate different aspects in Korea. The details are described below.

Entity Type	Reference	Data Source	Accessibility	Number of entity instance
OffStreetParking	OffStreetParking	Proprietary parking service server	public	2
ParkingSpot	ParkingSpot	Proprietary parking service server	public	231
BusStop	BusStop	GyeongGi Bus Information: http://www.gbis.go.kr/	public	4
Intra_bus	BusLine	http://www.gbis.go.kr/	public	96
Inter_bus	BusLine	Open Data Portal: https://www.data.go.kr/dataset/15000590/openapi.do	public	105
Metro	-	Open Data Portal: https://www.data.go.kr/dataset/15000754/openapi.do	public	2

Table 16 SeongNam Non-RZ - List of datasets available

OffStreetParking and ParkingSpot entities are used without modifying the FIWARE data model.

BusStop: To distinguish two different bus stops pertaining to the bus route direction, the direction attribute has been added.

BusLine: This represents both intercity bus and intracity bus line (or route). For intercity bus, destination, duration, departure time etc. attributes are added. The etc. attribute has human readable additional information. Similarly, intracity bus has new attributes like origin, destination, direction, operation hour for weekdays and holidays.

Metro: Metro entity is newly defined, since there is no metro model in SynchroniCity.

JSON schema files are available on the GitHub repository, and note that the schemas are compatible with oneM2M flexContainer resource types that can store data models above.

3.9.4 Advanced Operation, maintenance process

Since the service deployment consists of many components in a chain, it is not easy to see where the issue arises. Sensor data is gathered firstly from the local parking server and provided to the application via oneM2M adaptor, oneM2M platform, Mca-NGSI adaptor and Orion Broker.

For maintenance, LoRa MGMT Tool, a web-based admin dashboard was developed to monitor the deployment. Figure 13 is a picture the screen of the admin dashboard. This shows the running status of each module (e.g. oneM2M adaptor) as well as real-time sensor data. In the case of oneM2M Adaptor, located between the Mobius and Parking sensors, heartbeat data is sent to Mobius every 5 minutes, LoRa MGMT checks this heartbeat to see if the adaptor is running without a problem.

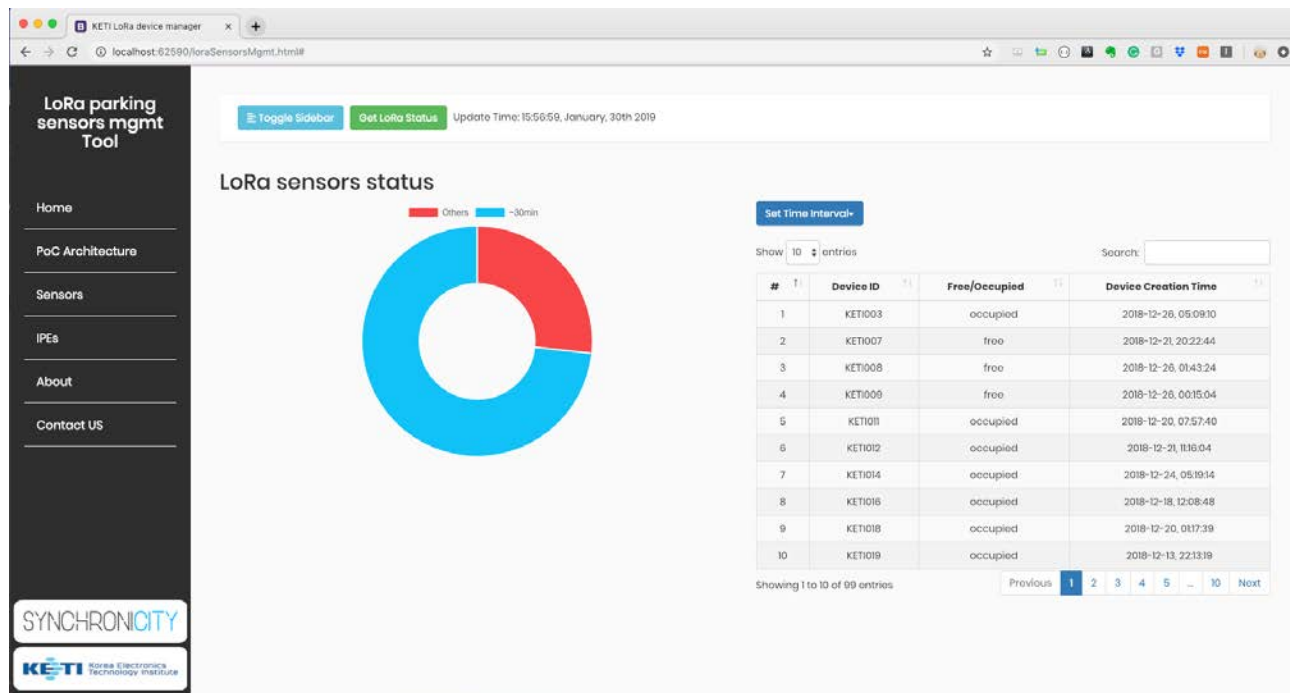


Figure 13 The admin dashboard with KETI Parking Lot information

3.9.5 Future improvements

For the future improvements, a data marketplace will be developed on the top of oneM2M platform. The data marketplace will be based on the SynchroniCity one in term of functionalities (e.g. SLA management). Also, in the same way we would like to forward to the Orion Context Broker based on the SynchroniCity platform, the NGSI-LD interface is planned to be supported from the oneM2M platform.

With those two extensions, Korean consortium will provide oneM2M standard based SynchroniCity framework. This plan would extend the digital single market of SynchroniCity covering oneM2M standard echo system.

4 Conclusion

This deliverable defines the final version of the deployment and operation activities through RZs by following the architectural guidelines offered in T2.1. Each RZ and their respective technical partners carried out these heterogeneous activities in their cities both for the Open Call and WP3 activities. Every RZ has presented their current environment schemas, defined data-sets and commented about integration and deployment process by giving as much as information available.

To do so, the main activities have been divided into 4 main sections:

- Specific reference zone infrastructure and platform elements: We asked RZs to detail as-is and legacy systems in their environments to have an in detail understanding of the systems and platforms.
- Integration guide: This section details the transformation and adaptation activities of the legacy systems (datasets, sensors, DBs, APIs, etc.) with the reference architecture based on the components and standards defined in the previous deliverables.
- Deployment process: Each RZ is asked to provide the very first approach to deploy the customized reference architecture based on the pilot needs.

- Operation and maintenance process: Day to day operations to keep the SynchroniCity platform operable and functional.

This document is expected to be a practical guideline for RZ technical reference check as it creates a big picture of the platform and use case related activities or questions.

It is important to underline that, due to the different activities to be performed during the piloting phase and open call projects execution, the city deployments described in this document could change and be improved in terms of both services and datasets provided. Information about these changes will be included in other WP3 deliverables, in particular D3.8 – “Report on reference zones IoT service deployment and operations”.

5 References

The following references are used as background documents for the preparation of this document. References are categorized standards (i.e. standards and specifications from the consortium working groups or alliances and specifications or drafts standardization bodies) and other documents, publications and technical or scientific books.

1. SynchroniCity project. Deliverable D2.3 "Catalogue of OASC Shared Data Models for Smart City domains", 2019
2. SynchroniCity project. Deliverable D2.10 "Reference Architecture for IoT Enabled Smart Cities, Update", 2018
3. SynchroniCity project. Deliverable D2.8 "Report on Basic Reference Zones platform deployment and operational plan", 2018
4. SynchroniCity project. Deliverable D3.1 "Documentation of service designs", 2018
5. SynchroniCity project. Deliverable D3.3 "Suite of atomic implementations - advanced", 2019
6. SynchroniCity project. Deliverable D3.5 "Customized IoT service prototypes for lead ref. zones"
7. SynchroniCity project. Deliverable D4.2 "Technical Validation Phase I", 2018
8. SynchroniCity project. Deliverable D4.3 "Technical Validation Phase II", 2019
9. C. d. Milano, "Milano API Store," [Online]. Available: <https://apisp.comune.milano.it/store/>.
10. ENG, "SynchroniCity Data Model Mapper Repository," 2018. Available: <https://gitlab.com/synchronicity-iot/data-model-mapper>.
11. "City Enabler for Digital Urban Services," Available: <http://digitalenabler.eng.it/>.
12. "Portale Open Data | Comune di Milano," Available: <https://dati.comune.milano.it/>.
13. FIWARE, "Places - Factual taxonomy," Available: https://github.com/Factual/places/blob/master/categories/factual_taxonomy.json.