

# SYNCHRONICITY



|                             |                                                                                      |
|-----------------------------|--------------------------------------------------------------------------------------|
| GA no:                      | 732240                                                                               |
| Action full title:          | SynchroniCity: Delivering an IoT enabled Digital Single Market for Europe and Beyond |
| Call/Topic:                 | Large Scale Pilots                                                                   |
| Type of action:             | Innovation Action (IA)                                                               |
| Starting date of action:    | 01.01.2017                                                                           |
| Project duration:           | 36 months                                                                            |
| Project end date:           | 31.12.2019                                                                           |
| Deliverable number:         | D4.3                                                                                 |
| Deliverable title:          | Technical Validation (Phase 2)                                                       |
| Document version:           | Ver1.0                                                                               |
| WP number:                  | WP4                                                                                  |
| Lead beneficiary:           | 9-UC                                                                                 |
| Main author(s):             | Ignacio Elicegui, Laura Rodriguez, Luis Díez, Luis Muñoz (UC)                        |
| Internal reviewers:         | Flavio Cirilo (NEC), Nuria de Lama (ATOS), Martin Brynskov (AU)                      |
| Type of deliverable:        | R                                                                                    |
| Dissemination level:        | PU                                                                                   |
| Delivery date from Annex 1: | M27                                                                                  |
| 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



Switzerland



South Korea

## Executive Summary

In SynchroniCity project, WP2 copes with the design and specification of the architecture as well as the development of the corresponding interfaces and enablers. WP3 addresses the development of the atomic components which should be used in either an individual or composed basis for deploying services which can be used by some applications demanded by the Reference Zones (RZ) and pilots participating in the open call. Within this context, WP4 focuses on the specification of the methodology for validating the above developments encompassing functional, technical, user, market and replication characteristics. With such aim and during the first year of the project, WP4 delivered D4.1 [1] which describes the methodology that has been adopted for the validation of the above-mentioned attributes. Such methodology has been applied here on different context, for the purpose of:

1. Validating first the so named “leading zones”. Those are Reference Zones (RZs) characterised by advanced degree of maturity in terms of components deployment and stability, both on the city platforms and smart city applications.
2. Validating next all RZs extending the functionalities of the tools as well as the components.
3. In-depth analysis carried out on the atomic services described in D3.3 [2] and its composition.

The reader will realize immediately the important additional functionalities and tools that have been developed compared to the ones presented in D4.2 [3]. Thus, besides the data model validator two additional ones have been developed. The first is coping with the validation of the Context Management API; the second addresses Historical API testing. Furthermore, the security framework, based on the OAuth2.0 protocol, is also checked. This means that all Minimum Interoperability Modules (MIM) [4], in order to conform with SynchroniCity architecture, are tested in all reference zones.

In particular, the MIM Validator exposes a REST service that enables to carry out periodic (synchronous) checks against preconfigured endpoints and punctual (asynchronous) validations of both, APIs and data models. An instance of this service has been deployed and configured with the endpoints and access data provided by the RZs. This service is continuously validating the deployed SynchroniCity frameworks.

The second singular contribution compared to the first phase of the technical validation, compiled in D4.2, refers to the atomic services [2] assessment. Such constituents are grouped around three scenarios:

- Routing which is made of four components: i) NGSI Urban Mobility to GTFS; ii) GTFS Fetcher; iii) GTFS RT-Loader; iv) Routing Service.
- Estimation: Two atomic services have been developed in this framework. One provides the probability to find an available parking lot in a geographical area (Parking Probability Estimator). The second (Traffic Flow Estimator) predicts traffic intensity in different city arteries.
- Visualization: Also, two modules have been developed in this case. One called Smartcities dashboard and the second the Metrics visualizer based on Grafana.

An exhaustive validation has been carried out both in terms of the deployment status on each RZ and the functionalities provided by the corresponding atomic module.

In summary, the document provides a detailed description of the validation process along the SynchroniCity architecture [4] which will be complemented with an external one based on the pilots selected in the open call (incoming D4.5 - Technical Validation of the SME projects).

## Abbreviations

|         |                                                |
|---------|------------------------------------------------|
| API     | Application Programming Interface              |
| AGPL    | Affero General Public License                  |
| AI      | Artificial Intelligence                        |
| AS      | Atomic Service                                 |
| Auth.   | Authentication                                 |
| CB      | Context Broker                                 |
| CIM     | Context Information Management                 |
| CLI     | Command-Line Interface                         |
| CO      | Carbon monoxide                                |
| D       | Deliverable                                    |
| DB      | Data Base                                      |
| DBMS    | Data Base Management System                    |
| EC      | European Commission                            |
| ETSI    | European Telecommunications Standard Institute |
| EV      | Electrical Vehicle                             |
| FR      | Functional Requirement                         |
| GNU     | GNU Not Unix                                   |
| GPS     | Global Positioning System                      |
| GS      | Group Specification                            |
| GTFS    | General Transit Feed Specification             |
| GTFS RT | General Transit Feed Specification Real Time   |
| HTTP    | Hypertext Transfer Protocol                    |
| ID      | Identifier                                     |
| IdM     | Identity Manager                               |
| IoT     | Internet of Things                             |
| ISG     | Industry Specification Groups                  |
| JSON    | JavaScript Object Notation                     |
| KPI     | Key Performance Indicator                      |
| M       | Month                                          |
| MIM     | Minimal Interoperability Modules               |
| Mob.    | Mobility                                       |
| NGSI    | Next Generation Service Interface              |
| NGSI-LD | NGSI Linking Data                              |
| NGSIv2  | FIWARE-NGSI version 2                          |
| NO2     | Nitrogen dioxide                               |
| NoSQL   | No Structured Query Language                   |
| OASC    | Open & Agile Smart Cities                      |
| OAUTH2  | Open Authorization 2                           |
| OMA     | Open Mobile Alliance                           |

|        |                                     |
|--------|-------------------------------------|
| OTP    | Open Trip Planner                   |
| PDP    | Policy Decision Point               |
| PEP    | Policy Enforcement Point            |
| POI    | Point of Interest                   |
| REST   | Representational State Transfer     |
| RFC    | Request for Comments                |
| RZ     | Reference Zone                      |
| SAN    | Santander reference Zone            |
| SC_DMV | SynchroniCity Data Models Validator |
| SQL    | Structured Query Language           |
| STH    | Short-Term Historic                 |
| Sync.  | SynchroniCity                       |
| TUS    | Santander Urban Transport           |
| UI     | User Interface                      |
| URL    | Uniform Resource Locator            |
| URN    | Uniform Resource Name               |
| UTC    | Coordinated Universal Time          |
| WEB UI | Web User Interface                  |
| WP     | Work Package                        |

## Contents

|       |                                                  |    |
|-------|--------------------------------------------------|----|
| 1     | Introduction .....                               | 10 |
| 2     | Synchronicity Framework validation update .....  | 11 |
| 2.1   | Framework components to be validated.....        | 11 |
| 2.1.1 | Synchronicity defined Data models .....          | 12 |
| 2.1.2 | SynchroniCity Interfaces .....                   | 13 |
| 2.1.3 | Atomic Services (AS) .....                       | 14 |
| 2.2   | Components' validation tools & mechanisms .....  | 14 |
| 2.2.1 | SynchroniCity MIM Validator .....                | 15 |
| 2.2.2 | Atomic Services' scenarios .....                 | 16 |
| 3     | MIMs validation results per Reference Zone ..... | 23 |
| 3.1   | Porto's Reference Zone MIMs validation.....      | 24 |
| 3.1.1 | Endpoints, interfaces and core components.....   | 24 |
| 3.1.2 | Data models and available datasets.....          | 24 |
| 3.2   | Santander's Reference Zone MIMs validation.....  | 25 |
| 3.2.1 | Endpoints, interfaces and core components.....   | 25 |
| 3.2.2 | Data models and available datasets.....          | 26 |
| 3.3   | Antwerp's Reference Zone MIMs validation .....   | 26 |
| 3.3.1 | Endpoints, interfaces and core components.....   | 26 |
| 3.3.2 | Data models and available datasets.....          | 27 |
| 3.4   | Eindhoven's Reference Zone MIMs validation ..... | 27 |
| 3.4.1 | Endpoints, interfaces and core components.....   | 27 |
| 3.4.2 | Data models and available datasets.....          | 28 |
| 3.5   | Helsinki's Reference Zone MIMs validation.....   | 28 |
| 3.5.1 | Endpoints, interfaces and components.....        | 28 |
| 3.5.2 | Data models and available datasets.....          | 28 |
| 3.6   | Manchester's Reference Zone MIMs validation..... | 29 |
| 3.6.1 | Endpoints, interfaces and core components.....   | 29 |
| 3.6.2 | Data models and available datasets.....          | 29 |
| 3.7   | Milan's Reference Zone MIMs validation.....      | 29 |
| 3.7.1 | Endpoints, interfaces and core components.....   | 29 |
| 3.7.2 | Data models and available datasets.....          | 30 |
| 3.8   | Carouge's Reference Zone MIMs validation .....   | 31 |
| 3.8.1 | Endpoints, interfaces and core components.....   | 31 |
| 3.8.2 | Data models and available datasets.....          | 31 |
| 4     | Atomic Services Validation.....                  | 32 |

|       |                                                        |    |
|-------|--------------------------------------------------------|----|
| 4.1   | Scenario 1 (Routing) Validation .....                  | 32 |
| 4.1.1 | NGSI Urban Mobility to GTFS .....                      | 33 |
| 4.1.2 | GTFS Fetcher .....                                     | 36 |
| 4.1.3 | GTFS-RT Loader .....                                   | 39 |
| 4.1.4 | Routing Service.....                                   | 43 |
| 4.2   | Scenario 2 (Estimation) Validation .....               | 48 |
| 4.2.1 | Parking Probability Estimator .....                    | 49 |
| 4.2.2 | Traffic Flow Estimator.....                            | 53 |
| 4.3   | Scenario 3 (City data Visualization) Validation.....   | 57 |
| 4.3.1 | Smartcities Dashboard .....                            | 57 |
| 4.3.2 | Metrics Visualizer (Grafana).....                      | 62 |
| 5     | Results and Conclusions.....                           | 68 |
| 5.1   | MIMs validation update (view of overall results) ..... | 68 |
| 5.2   | Synchronicity Atomic Services .....                    | 70 |
| 5.3   | Validation process assessment.....                     | 71 |
| 5.4   | KPIs.....                                              | 73 |
|       | REFERENCES .....                                       | 75 |

## List of Figures

|            |                                                                                                             |    |
|------------|-------------------------------------------------------------------------------------------------------------|----|
| Figure 1.  | SynchroniCity Framework: Architecture's Interoperability Points & Interfaces (MIMs) + atomic services ..... | 11 |
| Figure 2.  | Routing scenario composition .....                                                                          | 19 |
| Figure 3.  | Estimation scenario composition .....                                                                       | 20 |
| Figure 4.  | City Data Visualization composition.....                                                                    | 22 |
| Figure 5.  | Routing Scenario Santander's Layout .....                                                                   | 33 |
| Figure 6.  | NGSI Urban Mobility to GTFS operations.....                                                                 | 33 |
| Figure 7.  | GTFS Fetcher operations .....                                                                               | 36 |
| Figure 8.  | GTFS-RT Loader from NGSI operations .....                                                                   | 39 |
| Figure 9.  | Routing Service orchestration .....                                                                         | 43 |
| Figure 10. | AS Multimodal route requested to Routing Service (Web User Interface) .....                                 | 45 |
| Figure 11. | Estimation scenario layout for validation .....                                                             | 48 |
| Figure 12. | Parking Estimator deployment .....                                                                          | 49 |
| Figure 13. | Traffic Estimator deployment .....                                                                          | 53 |
| Figure 14. | Visualization scenario layout for validation .....                                                          | 57 |
| Figure 15. | Smartcities Dashboard composition.....                                                                      | 58 |
| Figure 16. | Capture of the Smartcities Dashboard deployed in the test environment .....                                 | 59 |
| Figure 17. | Metrics Visualizer service composition.....                                                                 | 62 |
| Figure 18. | Data series stored by Quantum Leap on Crate DB component.....                                               | 64 |
| Figure 19. | Graphics (time series) created by the Grafana dashboard component.....                                      | 64 |

## List of Tables

|           |                                                                                                                                                                    |    |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Table 1.  | First set of approved SynchroniCity Data models (April 17th, 2019).....                                                                                            | 13 |
| Table 2.  | List of SynchroniCity atomic services to be validated .....                                                                                                        | 14 |
| Table 3.  | Documentation validation schema.....                                                                                                                               | 17 |
| Table 4.  | SynchroniCity atomic services integration and orchestration validation .....                                                                                       | 17 |
| Table 5.  | Template for functional requirement validation outcome .....                                                                                                       | 18 |
| Table 6.  | Atomic service colour codes for validation tables .....                                                                                                            | 18 |
| Table 7.  | Colour code showing validated component result/status .....                                                                                                        | 23 |
| Table 8.  | Porto's deployed components (Synchronicity Architecture) .....                                                                                                     | 24 |
| Table 9.  | Evolution of the Porto's data model validation. The table depicts the status in the first and second validations, as well as the increment between them.....       | 25 |
| Table 10. | Santander's deployed components (Synchronicity Architecture) .....                                                                                                 | 25 |
| Table 11. | Evolution of the Santander's data model validation. The table depicts the status in the first and second validations, as well as the increment between them.....   | 26 |
| Table 12. | Antwerp's deployed components (Synchronicity Architecture) .....                                                                                                   | 27 |
| Table 13. | Evolution of the Antwerp's data model validation. The table depicts the status in the first and second validations, as well as the increment between them.....     | 27 |
| Table 14. | Eindhoven's deployed components (Synchronicity Architecture).....                                                                                                  | 27 |
| Table 15. | Evolution of the Eindhoven's data model validation. The table depicts the status in the first and second validations, as well as the increment between them.....   | 28 |
| Table 16. | Helsinki's deployed components (Synchronicity Architecture) .....                                                                                                  | 28 |
| Table 17. | Evolution of the Helsinki's data model validation. The table depicts the status in the first and second validations, as well as the increment between them.....    | 28 |
| Table 18. | Manchester's deployed components (Synchronicity Architecture) .....                                                                                                | 29 |
| Table 19. | Evolution of the Manchester's data model validation. The table depicts the status in the first and second validations, as well as the increment between them. .... | 29 |
| Table 20. | Milan's deployed components (Synchronicity Architecture) .....                                                                                                     | 30 |
| Table 21. | Evolution of the Milan's data model validation. The table depicts the status in the first and second validations, as well as the increment between them.....       | 31 |
| Table 22. | Carouge's deployed components (Synchronicity Architecture).....                                                                                                    | 31 |
| Table 23. | Evolution of the Carouge's data model validation. The table depicts the status in the first and second validations, as well as the increment between them.....     | 31 |
| Table 24. | Probability estimations returned by Parking Prob. Estimator service .....                                                                                          | 51 |
| Table 25. | Validated SynchroniCity components per RZ, as of April the 17th, 2019. ....                                                                                        | 68 |
| Table 26. | Available SynchroniCity compliant datasets as of April the 17th, 2019. It is also shown the overall number of data assets according to its data model. ....        | 70 |
| Table 27. | WP3 listed SynchroniCity atomic service validation summary. ....                                                                                                   | 71 |
| Table 28. | Technical validation (phase I) KPIs .....                                                                                                                          | 73 |
| Table 29. | KPIs global values .....                                                                                                                                           | 74 |

Table 30. KPIs values per RZ ..... 74

# 1 Introduction

---

Work Package (WP) 4 is responsible for validating the outcomes of SynchroniCity in terms of architecture, deployments, services, pilots and user's engagement. In this sense, and as it was introduced in D4.2 [3], task 4.2 focuses on the technical results achieved within WP2 and WP3, summarised in their corresponding available documents D2.9 [5] and D3.3 [2]. The first round of the technical validation (phase I) of the components and deployments carried out within these two work packages was presented in D4.2, centred then on the SynchroniCity architecture deployed and data models adopted in each RZ. The evolution of these deployments and the services developed on top of this framework are the targets of the validation phase II, whose results are presented along this text.

In this sense, the D4.3 Technical Validation Phase II objective is to present the process, tools and scenarios used to validate the technical outcomes of SynchroniCity and to provide the reader with the status of the RZs in terms of framework deployments. This is done by dividing this presentation into two complementary branches: Firstly, this document lists the SynchroniCity reference framework components (MIMs) deployed in each RZ, detailing which have passed the validation process and which have not. This analysis also checks the SynchroniCity compliant data models available on each instance, presenting the evolution compared to status at phase I. This way, the reader can have a quick and updated view of the SynchroniCity deployment in each involved city with their corresponding progress on this; secondly, the text centres on the atomic services developed on top of the SynchroniCity framework. Besides analysing the provided software and documentation to deploy the services and the developed functionalities, the proposed validation includes an orchestration of these services, which involves their complete deployment and integration with a living SynchroniCity framework plus the interaction with other related services. To perform this, WP4 defined a set of three scenarios that relates atomic services, data sources and outcomes to check different orchestrations and integration capabilities.

The different sections of this document present, as it has been introduced, first (section 2) the components included in the technical validation, as an update on what was already written on D4.2:

- The SynchroniCity MIMs (NGSI, Historical data Access and OAuth APIs)
- The SynchroniCity Data Models, available on the SynchroniCity GitLab

And, as a new important feature in the second validation phase:

- The SynchroniCity atomic services developed within the scope of WP3

Following, the tools (and the methodology) used to validate these components are detailed:

- For the MIMs and the Data Models, the initial tool used in D4.2 has been evolved to support all the MIMs and get integrated with the SynchroniCity GitLab to include new updates on data sources. It also provides REST interfaces to automate the validation process.
- For the atomic services, the already mentioned scenarios definition has been described

The validation process ends showing the results and status of the Reference Zones (section 3) related to their adoption of the SynchroniCity framework, and the scenario's performance (section 4), integrating the atomic services with the Santander SynchroniCity Instance.

Finally (section 5), the assessment processes carried out to polish both, RZ frameworks and atomic services deployments, together with developers and technical leaders are mentioned, including a set of KPIs and results tables to summarize the outcomes of the Phase II validation process.

This deliverable closes this way the Task 4.2, addressing the technical validation of the SynchroniCity internal developments in terms of tools and mechanisms but will still provide support to RZs to improve their corresponding deployments and new atomic services to polish the shared code and documentation. It also resets the technical guidelines to further validate the SynchroniCity pilots in D4.5, within Task 4.4.

## 2 Synchronicity Framework validation update

As shown in Figure 1, the SynchroniCity framework, in this second validation round, includes a set of atomic services, specially designed to exploit SynchroniCity resources and MIMs whilst provides service developers with extra tools to build city services on top of SynchroniCity.

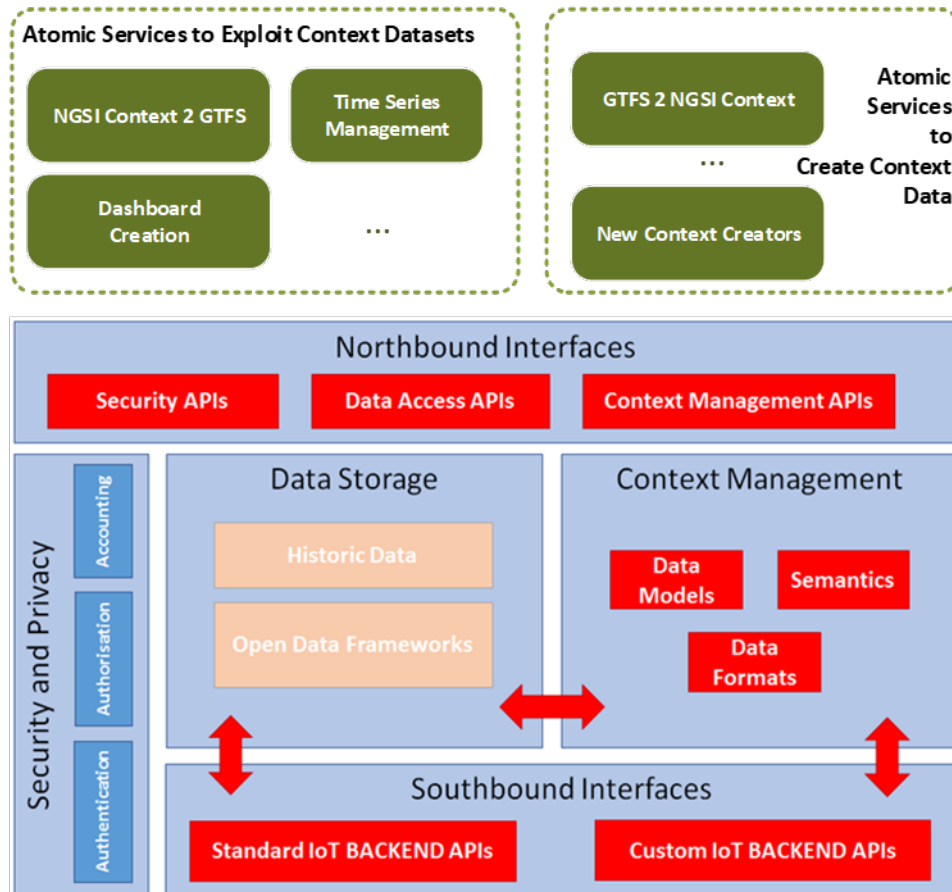


Figure 1. SynchroniCity Framework: Architecture's Interoperability Points & Interfaces (MIMs) + atomic services

SynchroniCity architecture components and MIMs are explained and updated in deliverable D2.10 [4]. In terms of validations, these have been referenced in D4.2 [3]. This section will refresh the list of target components to be checked and updates the tools to execute the corresponding analysis. The main effort of this second technical validation is focused on the execution and orchestration of the atomic services defined in WP3 [2].

### 2.1 Framework components to be validated

This document is oriented to the technical work done in WP2, involving the SynchroniCity architecture reference implementation, and to the WP3, checking the specific services developed on top. In this sense, it updates the status of the SynchroniCity adoption in each RZ, whilst checks the orchestration capabilities it offers to external developers. The validation process will cover three main layers: the updated SynchroniCity data models set, the MIMs available at the RZ and the deployment of the developed WP3's atomic services.

To execute this validation, the tools introduced in D4.2 [3] have been evolved and a set of scenarios to check the services has been designed and deployed.

### 2.1.1 Synchronicity defined Data models

Task 2.2 harmonises a set of data models in the context of the SynchroniCity Architecture. This work is reflected on D2.3 [6] and updated, with new inclusions and modifications, in the data models section on the SynchroniCity GitLab [7]. Table 1 shows the data models approved within the SynchroniCity context at the time this validation was done.

| Area            | Data Model Name        | Description                                                                                                                      | Source |
|-----------------|------------------------|----------------------------------------------------------------------------------------------------------------------------------|--------|
| Environment     | AirQualityObserved     | It represents an observation of air quality conditions at a certain place and time                                               | G      |
|                 | NoiseLevelObserved     | It represents an observation of those parameters that estimate noise pressure levels at a certain place and time                 | F      |
| PointOfInterest | PointOfInterest        | A harmonised geographic description of a Point of Interest                                                                       | G      |
|                 | Beach                  | A harmonised geographic description of a beach                                                                                   | F      |
|                 | Museum                 | A harmonised geographic description of a museum                                                                                  | F      |
|                 | Store                  | A harmonised geographic description of a store                                                                                   | DS     |
| Transportation  | BikeHireDockingStation | A bike hire docking station where subscribed users can hire and return a bike.                                                   | F      |
|                 | TrafficFlowObserved    | An observation of traffic flow conditions at a certain place and time.                                                           | F      |
|                 | EVChargingStation      | A public charging station supplying energy to electrical vehicles (EV).                                                          | DS     |
|                 | CrowdFlowObserved      | An observation of people movement at a certain place and time.                                                                   | DS     |
|                 | Vehicle                | Real time tracking of the vehicles that are used for public transportation (buses, trains, etc.)                                 | F      |
|                 | CarSharingStation      | A site dedicated to car sharing with either direct access from a road or with suitable and clearly marked access points.         | DS     |
|                 | RestrictedTrafficArea  | An area of a city in which the traffic generated by cars or any other kind of vehicles is subjected to limitation.               | DS     |
|                 | Road                   | A harmonised geographic and contextual description of a road. Roads are made up of one or more RoadSegment entities.             | F      |
|                 | RoadSegment            | A harmonised geographic and contextual description of a road segment. A collection of road segments are used to describe a Road. | F      |
| Weather         | WeatherObserved        | An observation of weather conditions at a certain place and time.                                                                | G      |
|                 | WeatherForecast        | Harmonised description of a Weather Forecast.                                                                                    | G      |
| Urban Mobility  | GtfsAgency             | Maps agency.txt from GTFS specification [8]                                                                                      | DS     |
|                 | GtfsStop               | Maps stop.txt from GTFS specification                                                                                            | DS     |
|                 | GtfsStation            | It is a GTFS stop which location_type is equal to 1.                                                                             | DS     |
|                 | GtfsAccessPoint        | It is a GTFS stop which location_type is equal to 2.                                                                             | DS     |

|                  |                      |                                                                                                                                                      |    |
|------------------|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|                  | GtfsRoute            | Maps routes.txt from GTFS specification                                                                                                              | DS |
|                  | GtfsTrip             | Maps trips.txt from GTFS specification                                                                                                               | DS |
|                  | GtfsShape            | Maps shapes.txt from GTFS specification                                                                                                              | DS |
|                  | GtfsStopTime         | Maps stop_times.txt from GTFS specification                                                                                                          | DS |
|                  | GtfsService          | It represents a transportation service which is available for one or more routes at certain dates.                                                   | DS |
|                  | GtfsCalendarRule     | Maps calendar.txt from GTFS specification                                                                                                            | DS |
|                  | GtfsCalendarDateRule | Maps calendar_dates.txt from GTFS specification                                                                                                      | DS |
|                  | GtfsFrequency        | Maps frequencies.txt from GTFS specification                                                                                                         | DS |
|                  | GtfsTransferRule     | Maps transfers.txt from GTFS specification                                                                                                           | DS |
|                  | ArrivalEstimation    | It captures the estimated arrival time of a public transport vehicle reaching a particular stop, whilst the vehicle is servicing a particular route. | DS |
|                  | GtfsTransitFeedFile  | Provides the Location (URL) of the ZIP file containing a valid GTFS transit feed.                                                                    | DS |
|                  | PublicTransportRoute | Provides generic model for public transport route.                                                                                                   | DS |
|                  | PublicTransportStop  | Provides generic model for public transport stop.                                                                                                    | DS |
| Parking          | ParkingSpot          | An individual, usually monitored, parking spot/parking lot                                                                                           | F  |
|                  | OffStreetParking     | An offstreet parking site with explicit entries and exits                                                                                            | F  |
|                  | OnStreetParking      | An on street, free entry (but might be metered) parking zone which contains at least one or more adjacent parking spots                              | F  |
| Parks & Gardens  | Garden               | A distinguishable planned space, usually outdoors, set aside for the display, cultivation, and enjoyment of plants and other forms of nature.        | F  |
|                  | GreenspaceRecord     | Harmonised description of the conditions recorded on a particular area or point inside a greenspace (flower bed, garden, etc.)                       | F  |
| Waste Management | WasteContainer       | Instance of waste container placed at a particular place with its dynamic properties (e.g. fillingLevel)                                             | F  |
|                  | WasteContainerModel  | Static properties of a waste container (big belly)                                                                                                   | F  |
| Device           | Device               | An apparatus (hardware + software + firmware) intended to accomplish a particular task (sensing the environment, actuating, etc.).                   | F  |

Table 1. First set of approved SynchroniCity Data models (April 17th, 2019). Source: G (GSMA), F (FIWARE), DS (Developed by SynchroniCity)

### 2.1.2 SynchroniCity Interfaces

The SynchroniCity defined interfaces directly link with the defined SynchroniCity MIMs (Minimal Interoperability Mechanisms) and refers to the implementation provided by each city of the APIs defined in SynchroniCity. These APIs were listed in D4.2 [3] and deeply described in D2.10 [4].

Here is the complete list of these APIs with the reference to the latest updates on SynchroniCity APIary [9].

- **Synchronicity NGSI interface for contextual information**, based on the NGSI API standard [10] and the preliminary ETSI NGSI-LD API (GS CIM 004) [11]
- **SynchroniCity Historical data access API**, based on FIWARE Short Term Historic Comet component and designed to match NGSI-LD short term historical data access
- **SynchroniCity Security API** is compliant with the OAuth2 standard described in RFC 6749 [12] and supports all four grant types defined there. The support of this API depends on each RZ framework deployment. The SynchroniCity reference implementation refers to the FIWARE IdM [13].

### 2.1.3 Atomic Services (AS)

WP3 defines [2] a set of atomic services as building blocks on top of the SynchroniCity framework, consuming SynchroniCity resources. All the atomic services are delivered in a way that are easily customized and deployed as presented in the deliverable D3.4 [14].

This document will reflect only the validation of those atomic services included in D3.3 [2] and listed in Table 2, however, the creation of new SynchroniCity services is a live cycle that increases its number day by day. The complete list is shared through the SynchroniCity GitLab [15].

| AS Name                  | Description                                                                                                                                                                                |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Routing                  | Open Trip Planner (OTP) [16] based, provides the end-user with multimodal routes, considering data about bus and taxi stops, city bikes and bicycling routes, plus disruption information. |
| Parking Prob. Estimator  | Using Artificial Intelligence techniques, it predicts the probability of finding a free parking spot within a particular area of a RZ in 15, 30, 45 and 60 minutes.                        |
| Traffic Flow Estimator   | By exploiting traffic flow sensors, it trains a model that predicts the intensity of traffic in the next hour, giving a slotted array of outputs (i.e. 15, 30, 45 and 60 minutes).         |
| SmartCity Dashboard      | User-friendly interface that visualises the devices (location and relevant information) that compose the SynchroniCity deployment in a Reference Zone.                                     |
| Metrics Visualizer       | Custom deployment of Grafana [17] software working on top of SynchroniCity framework, exploiting all the dashboard creation capabilities of this software.                                 |
| GTFS Fetcher             | Component that fetches GTFS files (static timetables and related info) from NGSI GtfsTransitFeedFile SynchroniCity context entities and loads them into the SynchroniCity Routing AS.      |
| NGSI Urban Mob. 2 GTFS   | Converts Urban Mobility NGSI entities to GTFS feed format, performing the inverse mapping done when defining the NGSI counterpart.                                                         |
| GTFS-RT Loader from NGSI | Consumes information from SynchroniCity ArrivalEstimation and GtfsTransitFeedFile data models to generate compliant GTFS-RT feeds.                                                         |

Table 2. List of SynchroniCity atomic services to be validated

## 2.2 Components' validation tools & mechanisms

As described in the previous section, on this second round, technical validation targets are the SynchroniCity data models, the defined SynchroniCity MIMs or interfaces and the developed SynchroniCity atomic services.

Focused on the SynchroniCity architecture and covering the data models and interfaces, WP4 has designed a tool to automatically check and validate these components. Its first version, the SynchroniCity data model validator presented in D4.2 [3], was used to execute the first validation round. This tool has evolved covering new requirements from RZs and architecture implementations and offering new configuration possibilities. This way, the new SynchroniCity MIM validator [18] provides a new user interface, in the form of a RESTful API, able to integrate the SynchroniCity MIMs' validation process in different development environments.

Although all SynchroniCity atomic services can run alone on top of a SynchroniCity framework (or requiring just a subset of SynchroniCity components), in order to validate their replicability and integration capabilities, we've defined three scenarios based on atomic services listed in WP3 D3.3 [2]. These scenarios, based on services' related functionalities, orchestrate different outcomes to compose elaborated tools that enrich the SynchroniCity framework. Each scenario relies on a SynchroniCity instance to support data gathering. This way, we also validate the essence of the atomic services as part of more elaborated services and/or final applications.

### 2.2.1 SynchroniCity MIM Validator

The SynchroniCity MIM validator is distributed as a SynchroniCity service through the SynchroniCity Gitlab [18]. It is an evolution of the previous SynchroniCity data model validator (SC\_DMV), used in D4.2 to evaluate the initial deployments of the SynchroniCity framework on RZs. This service is intended to validate the SynchroniCity interoperability points, covering:

- SynchroniCity Context Management API (based on NGSIv2) and the context query/retrieve, context creation/update, context resource discovery and context data subscription functionalities.
- Data models defined within SynchroniCity, relying on a JSON Schema Validator, to check structures, attributes and value ranges according given specifications and supporting new data model updates.
- Historical API defined in SynchroniCity, to query historical data linked to given entities and checking the retrieved data structures.
- OAUTH2 interface, where present, allowing the service configuration to access protected context brokers and adapt to each RZ OAuth implementation.

MIM Validator is developed in Node.js and distributed as Open Software within the SynchroniCity environment. The way to deploy and configure this service is shown in the project GitLab [18] and its interface is fully documented through Swagger [19]. The configuration options, among others, allows the user to specify:

- Endpoints of the components to be validated, so each RZ can point to their corresponding context brokers and historical resources
- OAuth tokens to be used or the parameters to obtain a valid OAuth2.0 token, so this feature can also be checked during the data access validation

This validation tool exposes now a REST service that permits to carry out periodic (synchronous) checks against preconfigured endpoints and punctual (asynchronous) validations of both, APIs and data models. Apart from retrieving the updated list of SynchroniCity compliant data models (it is synchronised with the data models definition and update process) used for validation, it also implements:

- Access to periodic validation
  - o NGSIv2 API (Context Management access validation)
    - /cities/ngsiApi: returns the cities for which there is periodic NGSI API validation configured
    - /cities/{CITY\_NAME}/ngsiApi: returns the current status of the NGSI API validation for a given city
  - o Data models API

- /cities/dataModels: returns the cities (including their services) for which there is periodic data model validation
- /cities/{CITY\_NAME}/dataModels/list: returns the type of datasets of a given city/service/service-path
- /cities/{CITY\_NAME}/dataModels: returns the validation of a given city/service/service-path
- /cities/{CITY\_NAME}/dataModels/{DATAMODEL\_NAME}: returns the validation of one dataset of a given city/service/service-path
- o Historical API
  - /cities/historicalApi: returns an array of the validated cities and the last time they were updated
  - /cities/{CITY\_NAME}/historicalApi: returns the current status of the historical API validation for a given city
- Online validation
  - o /online/ngsiApi: performs the NGSI API validation for an endpoint
  - o /online/historicalApi: performs the historical API validation for an endpoint
  - o /online/dataModel: performs the data models validation for an endpoint
  - o /online/dataModel/{DATAMODEL\_NAME}: performs the data model validation for a given datamodel for an endpoint

An instance of this service has been deployed and configured with the endpoints provided by the RZs to execute the validation process against their corresponding deployed SynchroniCity frameworks (described in section 3).

### 2.2.2 Atomic Services' scenarios

As agreed within WP3, the Synchronicity atomic services are distributed (when possible) following software containers approach (Docker [20]). Most of them expose their outcomes through REST APIs and/or SQL databases, so they require specific TCP/IP configurations and/or some Data Base Management System (DBMS) knowledge. In this sense, the end user is supposed to have initial skills in Docker systems, networks and data bases.

The focus of this evaluation will be put on three main aspects:

- 1) The **documentation** shared through the SynchroniCity Gitlab [15] and its presentation of the atomic services. These are related to the description of their functionalities (what they exactly provide); components included; components required (from Sync. Framework and from external sources); installation/deployment steps, etc. Key targets and dimensions are shown in Table 3

| Documentation validation schema |                              |                                                                                                    |
|---------------------------------|------------------------------|----------------------------------------------------------------------------------------------------|
| Key Target                      | Dimensions validated         | Characteristics to be considered                                                                   |
| Identification                  | Commercial info              | Version, Licensing mode.                                                                           |
|                                 | Technical Requirements       | Software/Hardware requirements.<br>Supported platforms/Operating Systems.<br>Data sources required |
|                                 | Potential end users/Services | Description of upper services/end users that could benefit from the AS                             |
|                                 | AS qualitative description   | Summary of the basic characteristics and inputs/outputs                                            |
| Deployment description          | Set up & Running             | Easy description of the different steps & configuration options.                                   |
|                                 | Usage                        | API/Instructions documentation                                                                     |

|         |                                                               |                                                                                            |
|---------|---------------------------------------------------------------|--------------------------------------------------------------------------------------------|
|         |                                                               | Errors documentation/guidance<br>Self-contained / properly referenced<br>Graphical support |
| Results | End-User Outcomes                                             | Dashboards/Web/Graphical final services offered                                            |
|         | Reusable Outcomes<br>(Integration/orchestration capabilities) | Supported protocols/standards<br>Replicability/Reusability                                 |

Table 3. Documentation validation schema

- 2) **Deployment and Integration with the SynchroniCity framework** (and its components) plus the other considered atomic services (orchestration) are tested here (according Table 4 criteria).

| SynchroniCity Integration & Orchestration validation schema |                            |                                                                                        |
|-------------------------------------------------------------|----------------------------|----------------------------------------------------------------------------------------|
| Key Target                                                  | Dimensions validated       | Characteristics to be considered                                                       |
| Deployment Procedure                                        | Set up & Running execution | Skills required.<br>Average Installation time.<br>Deployment documentation adjustment. |
| Supported MIMS                                              | NGSI                       | Context Info (in/out)                                                                  |
|                                                             | OAuth                      | OAuth flows supported                                                                  |
|                                                             | Historical Data            | Use of the Sync. Historical API                                                        |
|                                                             | Data models                | List of Sync. data models supported (in/out)                                           |
| Features Supported                                          | Data access management     | Support NGSIv2 data management features (FIWARE Services and FIWARE paths)             |
|                                                             | Orchestration              | Link with other Sync. Based services.<br>Standard protocols supported                  |

Table 4. SynchroniCity atomic services integration and orchestration validation

- 3) Final set of **functionalities** and features offered by the atomic services, including those enhanced ones due to the integration with other atomic services: WP3 compiled a set of Functional Requirements (FRs) for every application theme: human-centric traffic management; multimodal transportation; and community policy suite. The complete list of FRs linked to Application themes is reported in D3.1 [21]. The initial set of SynchroniCity atomic services were developed to support the final applications on these three main themes, providing specific functionalities and data exploitation features in an open and standard fashion. In this sense, each atomic service supports a set of FRs of the Application themes lists. This analysis was introduced in D3.2 [22] and continued in D3.3 [2]. The FRs linked to each atomic service on this last document was used as the reference to validate the functionalities of each one.

| Theme   |                |          |
|---------|----------------|----------|
| FR code | FR description | Comments |

| FR-N°       | Description of the FR as it is in D3.3                            | Comments from D3.3 |
|-------------|-------------------------------------------------------------------|--------------------|
| Validated   | Outcomes                                                          |                    |
| Colour Code | Brief notes about the validation process and the obtained results |                    |

Table 5. Template for functional requirement validation outcome

The codes to colour the boxes (Table 6) in the above described validation tables refer to:

| Colour Code | Description                                                                                                                                                                                                                                    |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | Acceptable fulfilment of the referred functionality/feature in the way it is described in the doc.                                                                                                                                             |
|             | It provides the referred functionality but with some deficiencies or not totally according the way described in the doc.<br>Missing requested or detected feature/functionality that would significantly improve the integration/orchestration |
|             | Mayor deficiencies found in the support of the referred functionality and/or in the provided doc. This functionality/feature is not properly provided                                                                                          |
|             | The referred functionality/feature couldn't be validated                                                                                                                                                                                       |

Table 6. Atomic service colour codes for validation tables

All these aspects will be checked through the creation and execution of the three following scenarios.

### 2.2.2.1 Scenario 1: ROUTING

The Routing scenario will combine atomic services related with General/Google Transit Feed Specification (GTFS [8]) with the Routing service, running on top of a SynchroniCity context broker: the services will use the NGSI SynchroniCity MIM to retrieve context information and build GTFS and GTFS-RT [23] that will be used by the Routing atomic service to improve the results offered to the end user (or services above).

Figure 2 shows the atomic services involved in this scenario:

- **GTFS Fetcher:** developed to feed Open Trip Planner (OTP [16]) Routing service, it monitors GTFSTransitFeedFile [24] entities in the Context broker to update and reload OTP data bases and maps.
- **NGSI Urban Mobility to GTFS:** reads Urban Mobility entities [25] stored in the Context Broker to generate a new GTFS file. Next, it creates a new GTFSTransitFeedFile context entity to share this file.
- **GTFS-RT Loader from NGSI:** Generates GTFS-RT feeds based on GTFSTransitFeedFile and linked ArrivalEstimation context entities in the Context Broker. These feeds can be later used by GTFS-RT compliant routing services.
- **Routing Service:** based on Open Trip Planner, it consumes GTFS and GTFS-RT data from SynchroniCity atomic services and provides multimodal routes, supporting different customization options.

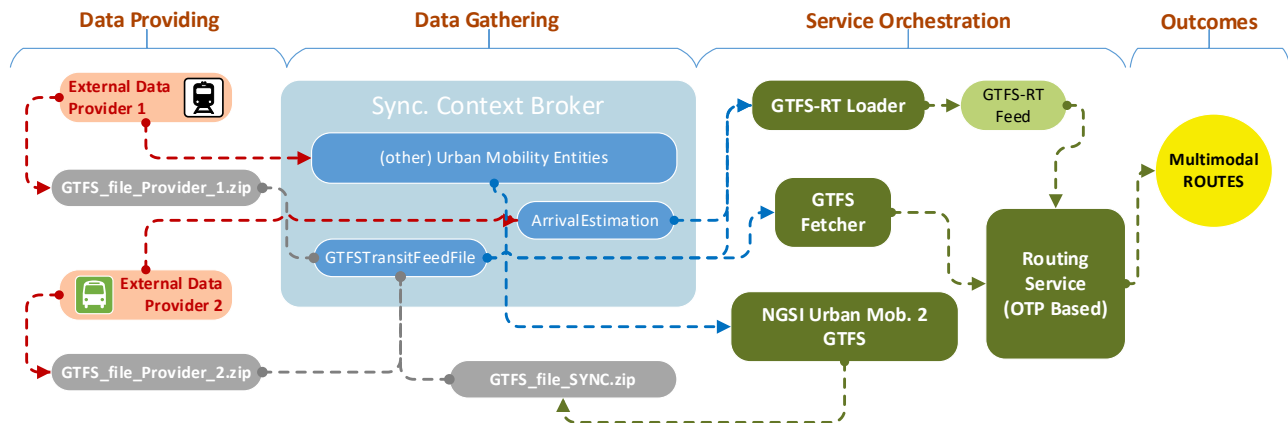


Figure 2. Routing scenario composition

Once the above listed atomic services have been properly deployed, following the linked documentation on top of a SynchroniCity instance (SynchroniCity Context Broker is needed), the validation process will follow with the stages shown in Figure 2.

- **Data Providing:** here, data from Transport Service Providers is needed. An external provider supplies GTFS compliant data (GTFS Transit feed files packed in .zip format and/or GTFS-RT information) or just raw information about time tables, lines, stops and estimations. To play the whole composition, at least one GTFS zip file and related estimations raw data is needed, plus a minimum set of Urban Mobility Entities, to compose a new GTFS zip file.
- **Data Gathering:** all required data should be uploaded to the SynchroniCity Context Broker to be later exploited by the atomic services, using the framework NGSI interface:
  - o GTFS\_file\_Provider\_X.zip files will be registered as GTFSTransitFeedFile [24] entities (one entity per each file).
  - o Raw information about real time updates must be uploaded according ArrivalEstimation data model (for estimations).
  - o Raw data related to stops, lines, etc. should be updated following the SynchroniCity Urban Mobility data models.
- **Service Orchestration:** with the Data Gathering completed, the atomic services can operate:
  - o Available raw data related to Urban Mobility is captured by NGSI Urban Mob. 2 GTFS service to create a new GTFS feed file in .zip format (GTFS\_file\_SYNC.zip). This new information is reflected in a new GTFSTransitFeedFile entity shared through the Context Broker.
  - o GTFS Fetcher is subscribed to all GTFSTransitFeedFile entities in the Context Broker. When a new one appears or changes on existing ones are detected, this service updated the Routing Service with the new GTFS zip files and rebuilds its base maps, so the new routes can consider new GTFS info.
  - o When new updates are detected on ArrivalEstimation entities (these reflects real time changes on time arrivals linked to a given GTFS feed file), GTFS-RT Loader updates a created GTFS-RT feed related to an existing GTFS Transit Feed File. Routing Service has been previously configured to read this feed every X seconds, so this real time info would be also available for calculated routes.
- **Outcomes:** main outcomes of this scenario are the routes offered by the Routing Service to the end user (or upper services) both, through the web interface and through the API. These multimodal routes should consider all the GTFS feed files stored in its corresponding SynchroniCity framework (those directly provided by the external parties and those built by the NGSI Urban Mob. 2 GTFS atomic service. In the same line, the arrival times (and routes, legs and trips durations) should reflect both, the GTFS feeds original timetables and the real time updates provided by the GTFS-RT data supplied by the GTFS-RT Loader service.

### 2.2.2.2 Scenario 2: ESTIMATION

This scenario evaluates in parallel two atomic services that are conceived in the same way, and the results provided are highly related. The scenario will manage current (context) and historical parking and traffic data to estimate traffic status and parking probability. The involved atomic services, represented in Figure 3, are:

- **Parking Estimator:** consumes OnStreetParking and/or OffStreetParking entities available in the SynchroniCity framework of the linked RZs, both, through the NGSI API, to read current parking status and through the Sync. Historical API to get the historical related data. Based on this data it estimates the parking probability for a given city area and in a specified (short) time to come.
- **Traffic Flow Estimator:** consumes TrafficFlowObserved entities (context info) and related historical data available in SynchroniCity frameworks to generate an estimation of the traffic intensity flow in the short term (i.e. in the next 15, 30, 45 or 60 minutes)

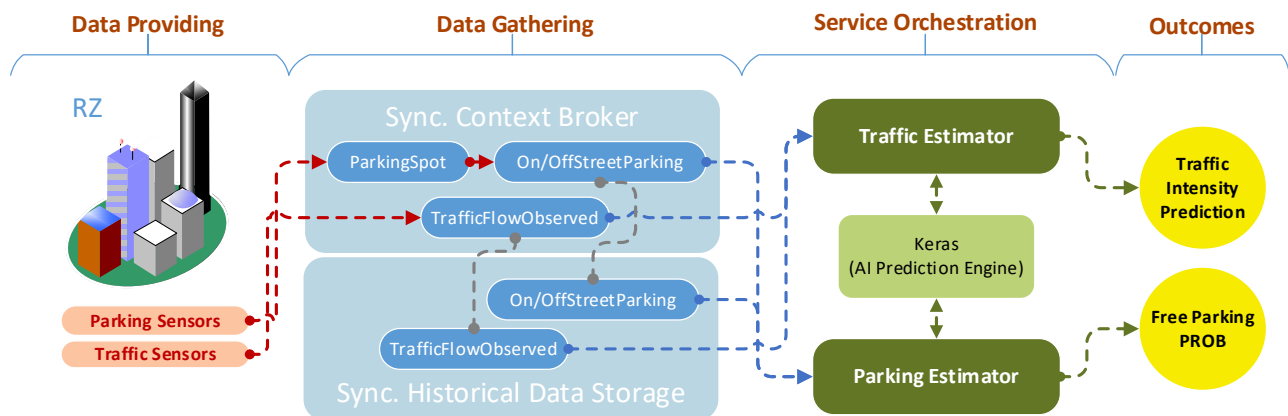


Figure 3. Estimation scenario composition

According to the available documentation of these services, context data and historical information are required, so they should be deployed on top of a Synchronicity instance that provides the NGSI and the Historical APIs. The required entities are On/OffstreetParking and TrafficFlowObservation respectively. The layout of the Estimation scenario is described in Figure 3 and will consist on:

- **Data Providing:** although there could be different external data providers for parking data or traffic intensity, we will consider here the Reference Zones as the information providers. City parking sensors and traffic loops are the devices capturing the relevant data. Other sources, as traffic cams or directly parking areas can also be valid sources. On the other hand, city areas (to link traffic status or group parking probabilities) are also defined by the Reference Zones as part of the OnStreetParking entities.
- **Data Gathering:** this scenario considers two different information sets:
  - o context data provided by the different sources should be uploaded to the Synchronicity Context Broker through the corresponding NGSI interface:
    - TrafficSensors are registered as TrafficFlowObserved [26] entities, including (among others) traffic intensity and road occupancy data.
    - Parking Sensors, mapped as ParkingSpot [27], record the status (free/occupied) of parking lots.
    - On/OffstreetParking [28] entities represent parking areas and the free available parking lots within. This information can be directly provided by an external source (e.g. a parking facility manager) or be composed from the ParkingSpot linked entities (as in the shown layout in Figure 3).
  - o historical data related to On/OffStreetParking and TrafficFlowObserved entities are stored by the Synchronicity framework and offered through the Synchronicity Historical API [29].

- **Service Orchestration:** with the Data Gathering completed, both atomic services proceed in a similar way:
  - o Reads and get subscribed to the available required entities (Off/OnStreetParking for the parking estimation, TrafficFlowObserved for traffic flow estimation)
  - o Historical data related to the read entities and notifications from the context broker are forwarded to the Artificial Intelligence (AI) prediction engine, which elaborates the corresponding predictions.
  - o AI engine updates the estimations on the corresponding stored entities (TrafficFlowObservations internally managed by the traffic flow estimator, On/OffStreetParking internally managed by the parking estimator) so it can be served by the corresponding service when requested.
- **Outcomes:** this scenario will provide several results:
  - o By means of the information retrieved from the different parking related context entities (i.e. OnStreetParking and OffStreetParking), parking estimator will:
    - Provide a (near) real-time status of a parking area
    - Generate an estimation of the probability of finding a free parking spot in the short term (i.e. in the next 15, 30 and 45 minutes)
  - o By means of the information retrieved from the different traffic intensity context entities, traffic estimator will:
    - generate an estimation of the traffic intensity flow (according to the Sync. TrafficFlowObserved data model intensity attribute definition) in the short term (i.e. in the next 15, 30, 45 or 60 minutes).

### 2.2.2.3 Scenario 3: City Data VISUALIZATION

This scenario is composed by the atomic services that exploits SynchroniCity context information to provide the end-user with different visualizations of the available datasets. Both are oriented to build dashboards to show, in a friendly way, the SynchroniCity resources. These atomic services are Smartcities Dashboard and Metrics Visualizer (Grafana):

- **Smartcities Dashboard:** links with a Sync. Context Broker (CB) and (optional) with a Historical data component supporting Short Term Historic (STH) Comet [30] to capture all the SynchroniCity compliant entities (including location attribute). Using Google maps API [31], it will show them on a web map, enabling the end user to quickly request the real time info linked to a selected entity. It also exposes two RESTful APIs, one for real time data access and another one (if linked to an historical component) to access entities historical data.
- **Metrics Visualizer:** it subscribes to Sync. Context entities to keep time series data and location information. It uses the Grafana analytics framework to offer the end user a wide set of tools to elaborate dashboards with these resources.

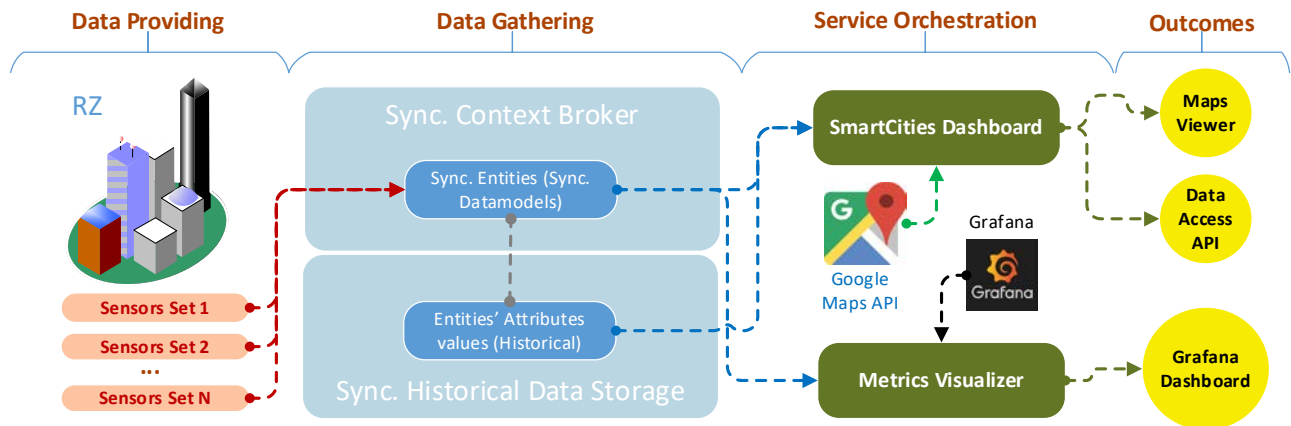


Figure 4. City Data Visualization composition

According to the documentation, both services will consume SynchroniCity context data, so they require a SynchroniCity framework integration. This layout, described in Figure 4, consists of:

- **Data Providing and Gathering:** the RZs will feed their corresponding Sync. Context Broker. Both services will be focused on measurements (CO, traffic flow, temperature, humidity, etc.) captured in a given observation time and referred to a location. Historical component is optional and will be replaced in future versions by an internal historical storage (for the Smartcity dashboard service).
- **Service Orchestration:** with the Data Gathering completed:
  - o Smartcities Dashboard can operate in two different ways:
    - It can be directly linked to a given instance of a Sync. Compliant CB, so it can access to all entities there stored.
    - It can be linked to an internal Orion CB and feed it through NGSI subscriptions. This way, the service will show only the subscribed entities.
  - o Metrics Visualizer uses subscriptions. This way, it captures the data from the desired entities to create the internal time series.
- **Outcomes:** this scenario will provide the following:
  - o Smartcities dashboard will create a web map so the end user can have a quick look of what is available in the Sync. Instance. By clicking on a selected entity, the web user interface will show its available context information.
  - o The Metrics visualizer exploits the Grafana Analytics platform and available plugins to create different dashboards with the captured context info. This service documents two specific plugins for time series and maps visualizations.

### 3 MIMs validation results per Reference Zone

This section updates the D4.2 validation process carried out together with the RZ in M19 (July 2018). This part of the whole validation process covers the core of the SynchroniCity frameworks deployed within the project scope on involved cities and shows the actual evolution one by one in terms of actual available interfaces and datasets. The progress on SynchroniCity components and IoT deployments per RZ compared to past D4.2 is shown in D2.9 [5], so this text is not exposing these details.

An instance of the SynchroniCity MIM validator tool, described in section 2.2.1, has been deployed and configured according to the Reference Zones provided data. For each city (RZ) **interfaces validation**:

- Endpoint to access its Context Broker (mandatory), supporting NGSIv2 context management API. It is used to check and validate the **SynchroniCity Context Management API**. This shall also include the NGSIv2 services and service paths parameters to access corresponding SynchroniCity datasets.
- Endpoint to access its **SynchroniCity compliant Historical data access API**, deployed by the RZ. When present, it allows the Historical Data access MIM validation.
- Endpoint to access its Identity Management (IdM) instance, to manage the authentication and identification process (when required). The way to manage the corresponding OAuth2.0 tokens to check the **SynchroniCity security layer** is also included (when required) in the configuration file.

For the data models and datasets validation process:

- Link to the repositories with the latest **SynchroniCity approved data models**, to get the JSON schemas to validate the datasets on each RZ. As many of the SynchroniCity current data models derive from the FIWARE ones, two repositories are configured:
  - o repository of SynchroniCity data models. (<https://gitlab.com/synchronicity-iot/synchronicity-data-models.git>)
  - o repository of FIWARE data models(<https://github.com/FIWARE/dataModels.git>).

The online validation process, mentioned in section 2.2.1, returns detailed results of the status of the different APIs and the datasets found on each configured SynchroniCity framework instance. These results show what is SynchroniCity compliant and what is not, pointing to what can be failing. This is a technical validation, so the outcomes have this technical profile and are useful for the RZs' assessment. The current (April 2019) status of each RZ has been summarised and coded with colours:

For Interfaces validation (Table 7):

|  |                                                                                                                                                                                                                                      |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | The component/interface satisfactory passed the corresponding validation process, with minor remarks (if any).                                                                                                                       |
|  | The component/interface is present/reachable but failed the corresponding validation process because of required mismatched functionalities and/or not SynchroniCity compliant. In any case, these reported issues can be rectified. |
|  | This component/interface is mandatory and is not present or not compliant at all with SynchroniCity specifications.                                                                                                                  |
|  | The component/interface has not been validated because external issues (e.g. not reachable, not deployed, need specific credentials that have not been provided, etc.)                                                               |

Table 7. Colour code showing validated component result/status

The uploaded instance has also been programmed to automatically launch and update every day this process, thus the deployments status can be summarised and published in the SynchroniCity web [32].

The results from the last online validation executed on each Reference Zone are shown in the next subsections.

For the provided endpoints validation, Table 7 colour codes are used to remark its status. When “Auth. Required” column is marked as “Yes”, an OAuth token is required, and the SynchroniCity security layer was also validated.

On each dataset validation, a comparison table is provided, showing the status of the first technical validation [3], the current validation results and the increment on available datasets. The “Total” column represents the total number of entities from each SynchroniCity Data Model available in the SynchroniCity instance. The “Valid” column shows how many of them are fully compliant with the corresponding approved schema.

The main conclusions extracted from these outcomes are shown in section 5.

### 3.1 Porto’s Reference Zone MIMs validation

#### 3.1.1 Endpoints, interfaces and core components

| PORTO's Reference Zone SynchroniCity Framework INSTANCE |                                                                                                                       |                                     |                |                                      |
|---------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|-------------------------------------|----------------|--------------------------------------|
| Interfaces (Interoperability Points)                    | Endpoint                                                                                                              | Description                         | Auth. Required | Reference standards / Specifications |
| Context Information Management                          | <a href="https://broker.fiware.urbanplatform.portodigital.pt">https://broker.fiware.urbanplatform.portodigital.pt</a> | Orion Context Broker endpoint       | YES            | NGSI Interface: NGSIv2               |
| Identity Management                                     | <a href="http://idm.urbanplatform.portodigital.pt/">http://idm.urbanplatform.portodigital.pt/</a>                     | IdM Keyrock                         | YES            | OAuth 2.0 (Supported)                |
| Historical Data                                         | <a href="https://ql.urbanplatform.portodigital.pt/">https://ql.urbanplatform.portodigital.pt/</a>                     | QuantumLeap REST API implementation | YES            | SynchroniCity Historical API         |

Table 8. Porto’s deployed components (Synchronicity Architecture)

#### 3.1.2 Data models and available datasets

| PORTO's Reference Zone data model validation |                  |       |                   |       |           |       |
|----------------------------------------------|------------------|-------|-------------------|-------|-----------|-------|
| Dataset                                      | First Validation |       | Second Validation |       | Increment |       |
|                                              | Total            | Valid | Total             | Valid | Total     | Valid |
| AirQualityObserved                           | 1                | 1     | 5                 | 5     | 4         | 4     |
| ArrivalEstimation                            |                  |       | 8                 | 0     | 8         | 0     |
| Device                                       | 211              | 0     | 191               | 155   | -20       | 155   |
| GtfsTransitFeedFile                          |                  |       | 1                 | 1     | 1         | 1     |

|                      |      |     |      |      |      |      |
|----------------------|------|-----|------|------|------|------|
| NoiseLevelObserved   | 9    | 9   | 15   | 13   | 6    | 4    |
| OffStreetParking     |      |     | 1    | 1    | 1    | 1    |
| PointOfInterest      | 2545 | 0   | 3062 | 2696 | 517  | 2696 |
| TrafficFlowObserved  | 210  | 0   | 115  | 115  | -95  | 115  |
| Vehicle              | 246  | 0   | 246  | 0    | 0    | 0    |
| WeatherForecast      | 115  | 115 | 131  | 131  | 16   | 16   |
| WeatherObserved      | 7    | 1   | 17   | 12   | 10   | 11   |
| GtfsAgency           |      |     | 1    | 1    | 1    | 1    |
| GtfsCalendarDateRule |      |     | 6    | 6    | 6    | 6    |
| GtfsCalendarRule     |      |     | 15   | 15   | 15   | 15   |
| GtfsRoute            |      |     | 73   | 73   | 73   | 73   |
| GtfsService          |      |     | 15   | 15   | 15   | 15   |
| GtfsStop             |      |     | 2468 | 2468 | 2468 | 2468 |

Table 9. Evolution of the Porto's data model validation. The table depicts the status in the first and second validations, as well as the increment between them.

## 3.2 Santander's Reference Zone MIMs validation

### 3.2.1 Endpoints, interfaces and core components

| SANTANDER's Reference Zone SynchroniCity Framework INSTANCE |                                                                                                         |                                     |                |                                      |
|-------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|-------------------------------------|----------------|--------------------------------------|
| Interfaces (Interoperability Points)                        | Endpoint                                                                                                | Description                         | Auth. Required | Reference standards / Specifications |
| Context Information Management                              | <a href="https://context.san.synchronicity-iot.eu">https://context.san.synchronicity-iot.eu</a>         | Orion Context Broker endpoint       | YES            | NGSI Interface: NGSIv2               |
| Identity Management                                         | <a href="https://auth.san.synchronicity-iot.eu">https://auth.san.synchronicity-iot.eu</a>               | IdM Keyrock                         | YES            | OAuth 2.0                            |
| Historical Data                                             | <a href="https://historical.san.synchronicity-iot.eu">https://historical.san.synchronicity-iot.eu</a>   | QuantumLeap REST API implementation | YES            | SynchroniCity Historical API         |
| Marketplace                                                 | <a href="https://marketplace.san.synchronicity-iot.eu">https://marketplace.san.synchronicity-iot.eu</a> | SynchroniCity Marketplace           | YES            | TCM                                  |

Table 10. Santander's deployed components (Synchronicity Architecture)

### 3.2.2 Data models and available datasets

| Santander's Reference Zone data model validation |                  |       |                   |       |           |       |
|--------------------------------------------------|------------------|-------|-------------------|-------|-----------|-------|
| Dataset                                          | First Validation |       | Second Validation |       | Increment |       |
|                                                  | Total            | Valid | Total             | Valid | Total     | Valid |
| AirQualityObserved                               | 65               | 65    | 77                | 77    | 12        | 12    |
| ArrivalEstimation                                | 982              | 982   | 25                | 25    | -957      | -957  |
| Beach                                            | 13               | 13    | 13                | 13    | 0         | 0     |
| BikeHireDockingStation                           | 17               | 17    | 16                | 16    | -1        | -1    |
| Device                                           | 283              | 283   | 325               | 325   | 42        | 42    |
| GreenspaceRecord                                 | 18               | 18    | 18                | 18    | 0         | 0     |
| GtfsTransitFeedFile                              |                  |       | 2                 | 2     | 2         | 2     |
| Museum                                           | 10               | 10    | 10                | 10    | 0         | 0     |
| NoiseLevelObserved                               | 19               | 19    | 16                | 16    | -3        | -3    |
| OnStreetParking                                  | 23               | 19    | 23                | 23    | 0         | 4     |
| ParkingSpot                                      | 262              | 262   | 323               | 323   | 61        | 61    |
| PointOfInterest                                  | 214              | 214   | 214               | 214   | 0         | 0     |
| PublicTransportRoute                             | 35               | 35    | 32                | 32    | -3        | -3    |
| PublicTransportStop                              | 446              | 446   | 449               | 449   | 3         | 3     |
| Store                                            | 2170             | 2170  | 2170              | 2170  | 0         | 0     |
| TrafficFlowObserved                              | 465              | 311   | 311               | 311   | -154      | 0     |
| Vehicle                                          |                  |       | 92                | 92    | 92        | 92    |
| WeatherForecast                                  |                  |       | 6                 | 6     | 6         | 6     |
| WeatherObserved                                  | 181              | 181   | 207               | 207   | 26        | 26    |

Table 11. Evolution of the Santander's data model validation. The table depicts the status in the first and second validations, as well as the increment between them.

## 3.3 Antwerp's Reference Zone MIMs validation

### 3.3.1 Endpoints, interfaces and core components

| ANTWERP's Reference Zone SynchroniCity Framework INSTANCE |          |             |                |                                      |
|-----------------------------------------------------------|----------|-------------|----------------|--------------------------------------|
| Interfaces (Interoperability Points)                      | Endpoint | Description | Auth. Required | Reference standards / Specifications |

|                                |                                                                                                                                                             |                                     |     |                              |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|-----|------------------------------|
| Context Information Management | <a href="https://ext-api-gw-p.antwerpen.be/sirus/orion-admin-api-a">https://ext-api-gw-p.antwerpen.be/sirus/orion-admin-api-a</a>                           | Orion Context Broker endpoint       | YES | NGSI Interface: NGSIv2       |
| Identity Management            | <a href="https://ext-api-gw-p.antwerpen.be/sirus/orion-consumer-api-a/v2/oauth2">https://ext-api-gw-p.antwerpen.be/sirus/orion-consumer-api-a/v2/oauth2</a> | WSO2                                | YES | OAuth 2.0                    |
| Historical Data                | <a href="https://ext-api-gw-p.antwerpen.be/sirus/comet-consumer-api-a">https://ext-api-gw-p.antwerpen.be/sirus/comet-consumer-api-a</a>                     | QuantumLeap REST API implementation | YES | SynchroniCity Historical API |
| Marketplace                    | <a href="https://ext-api-store.antwerpen.be">https://ext-api-store.antwerpen.be</a>                                                                         | Kong Marketplace                    | YES | n.a.                         |

Table 12. Antwerp's deployed components (Synchronicity Architecture)

### 3.3.2 Data models and available datasets

| Antwerp's Reference Zone data model validation |                  |       |                   |       |           |       |
|------------------------------------------------|------------------|-------|-------------------|-------|-----------|-------|
| Dataset                                        | First Validation |       | Second Validation |       | Increment |       |
|                                                | Total            | Valid | Total             | Valid | Total     | Valid |
| AirQualityObserved                             |                  |       | 12                | 0     | 12        | 0     |
| WasteContainer                                 | 127              | 107   | 159               | 62    | 32        | -45   |

Table 13. Evolution of the Antwerp's data model validation. The table depicts the status in the first and second validations, as well as the increment between them.

## 3.4 Eindhoven's Reference Zone MIMs validation

### 3.4.1 Endpoints, interfaces and core components

| EINDHOVEN's Reference Zone SynchroniCity Framework INSTANCE |                                                                                         |                               |                |                                      |
|-------------------------------------------------------------|-----------------------------------------------------------------------------------------|-------------------------------|----------------|--------------------------------------|
| Interfaces (Interoperability Points)                        | Endpoint                                                                                | Description                   | Auth. Required | Reference standards / Specifications |
| Context Information Management                              | <a href="http://mycity.techlab atos.net/orion">http://mycity.techlab atos.net/orion</a> | Orion Context Broker endpoint | NO             | NGSI Interface: NGSIv2               |
| Identity Management                                         |                                                                                         |                               |                |                                      |
| Historical Data                                             |                                                                                         |                               |                |                                      |

Table 14. Eindhoven's deployed components (Synchronicity Architecture)

### 3.4.2 Data models and available datasets

| EINDHOVEN's Reference Zone data model validation |                  |       |                   |       |           |       |
|--------------------------------------------------|------------------|-------|-------------------|-------|-----------|-------|
| Dataset                                          | First Validation |       | Second Validation |       | Increment |       |
|                                                  | Total            | Valid | Total             | Valid | Total     | Valid |
| AirQualityObserved                               | 35               | 35    | 35                | 35    | 0         | 0     |

Table 15. Evolution of the Eindhoven's data model validation. The table depicts the status in the first and second validations, as well as the increment between them.

## 3.5 Helsinki's Reference Zone MIMs validation

### 3.5.1 Endpoints, interfaces and components

| HELSINKI's Reference Zone Synchronicity Framework INSTANCE |                                                                                           |                               |                |                                      |
|------------------------------------------------------------|-------------------------------------------------------------------------------------------|-------------------------------|----------------|--------------------------------------|
| Interfaces (Interoperability Points)                       | Endpoint                                                                                  | Description                   | Auth. Required | Reference standards / Specifications |
| Context Information Management                             | <a href="https://synchronicity.cs.hut.fi/orion">https://synchronicity.cs.hut.fi/orion</a> | Orion Context Broker endpoint | NO             | NGSI Interface: NGSIv2               |
| Identity Management                                        |                                                                                           |                               |                |                                      |
| Historical Data                                            |                                                                                           |                               |                |                                      |

Table 16. Helsinki's deployed components (Synchronicity Architecture)

### 3.5.2 Data models and available datasets

| HELSINKI's Reference Zone data model validation |                  |       |                   |       |           |       |
|-------------------------------------------------|------------------|-------|-------------------|-------|-----------|-------|
| Dataset                                         | First Validation |       | Second Validation |       | Increment |       |
|                                                 | Total            | Valid | Total             | Valid | Total     | Valid |
| BikeHireDockingStation                          | 255              | 255   | 347               | 347   | 92        | 92    |
| AirQualityObserved                              |                  |       | 52                | 41    | 52        | 41    |
| NoiseLevelObserved                              |                  |       | 7                 | 7     | 7         | 7     |
| WeatherObserved                                 |                  |       | 14                | 14    | 14        | 14    |

Table 17. Evolution of the Helsinki's data model validation. The table depicts the status in the first and second validations, as well as the increment between them.

### 3.6 Manchester's Reference Zone MIMs validation

#### 3.6.1 Endpoints, interfaces and core components

| MANCHESTER's Reference Zone SynchroniCity Framework INSTANCE |                                                                     |                               |                |                                      |
|--------------------------------------------------------------|---------------------------------------------------------------------|-------------------------------|----------------|--------------------------------------|
| Interfaces (Interoperability Points)                         | Endpoint                                                            | Description                   | Auth. Required | Reference standards / Specifications |
| Context Information Management                               | <a href="http://54.229.156.165:1026">http://54.229.156.165:1026</a> | Orion Context Broker endpoint | NO             | NGSI Interface: NGSIv2               |
| Identity Management                                          |                                                                     |                               |                |                                      |
| Historical Data                                              | <a href="http://18.202.7.94:8666">http://18.202.7.94:8666</a>       | Proprietary                   | NO             | SynchroniCity Historical API         |

Table 18. Manchester's deployed components (Synchronicity Architecture)

#### 3.6.2 Data models and available datasets

| MANCHESTER's Reference Zone data model validation |                  |       |                   |       |           |       |
|---------------------------------------------------|------------------|-------|-------------------|-------|-----------|-------|
| Dataset                                           | First Validation |       | Second Validation |       | Increment |       |
|                                                   | Total            | Valid | Total             | Valid | Total     | Valid |
| AirQualityObserved                                | 67               | 47    | 3                 | 3     | -64       | -44   |
| BikeHireDockingStation                            | 1                | 1     | 1                 | 1     | 0         | 0     |
| CrowdFlowObserved                                 | 8                | 0     | 40                | 40    | 32        | 40    |
| NoiseLevelObserved                                | 6                | 6     | 6                 | 0     | 0         | -6    |
| OffStreetParking                                  | 1                | 1     | 80                | 9     | 79        | 8     |
| OnStreetParking                                   | 40               | 0     | 464               | 464   | 424       | 464   |
| TrafficFlowObserved                               |                  |       | 12                | 12    | 12        | 12    |
| WeatherObserved                                   | 6                | 6     | 6                 | 6     | 0         | 0     |

Table 19. Evolution of the Manchester's data model validation. The table depicts the status in the first and second validations, as well as the increment between them.

### 3.7 Milan's Reference Zone MIMs validation

#### 3.7.1 Endpoints, interfaces and core components

| MILAN's Reference Zone SynchroniCity Framework INSTANCE |          |             |                |                                      |
|---------------------------------------------------------|----------|-------------|----------------|--------------------------------------|
| Interfaces (Interoperability Points)                    | Endpoint | Description | Auth. Required | Reference standards / Specifications |

|                                |                                                                                                                             |                               |     |                              |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------|-------------------------------|-----|------------------------------|
| Context Information Management | <a href="https://api.comune.milano.it/synchronicity/context/1.0">https://api.comune.milano.it/synchronicity/context/1.0</a> | Orion Context Broker endpoint | YES | NGSI Interface: NGSIv2       |
| Identity Management            | <a href="https://api.comune.milano.it:443/token">https://api.comune.milano.it:443/token</a>                                 | WSO2                          | YES | OAuth 2.0                    |
| Historical Data                | <a href="https://api.comune.milano.it/synchronicity/comet/1.0">https://api.comune.milano.it/synchronicity/comet/1.0</a>     | Proprietary                   | YES | SynchroniCity Historical API |
| Marketplace                    | <a href="https://apis.comune.milano.it/store/">https://apis.comune.milano.it/store/</a>                                     | WSO2 based                    | YES | n.a.                         |

Table 20. Milan's deployed components (Synchronicity Architecture)

### 3.7.2 Data models and available datasets

| MILAN's Reference Zone data model validation |                  |       |                   |       |           |       |
|----------------------------------------------|------------------|-------|-------------------|-------|-----------|-------|
| Dataset                                      | First Validation |       | Second Validation |       | Increment |       |
|                                              | Total            | Valid | Total             | Valid | Total     | Valid |
| AirQualityObserved                           |                  |       | 852               | 852   | 852       | 852   |
| BikeHireDockingStation                       | 3                | 0     | 767               | 767   | 764       | 767   |
| CarSharingStation                            |                  |       | 113               | 113   | 113       | 113   |
| EVChargingStation                            |                  |       | 63                | 63    | 63        | 63    |
| Garden                                       |                  |       | 1064              | 1064  | 1064      | 1064  |
| GtfsAgency                                   |                  |       | 4                 | 4     | 4         | 4     |
| GtfsCalendarRule                             |                  |       | 1414              | 1414  | 1414      | 1414  |
| GtfsFrequency                                |                  |       | 1414              | 1414  | 1414      | 1414  |
| GtfsRoute                                    |                  |       | 582               | 582   | 582       | 582   |
| GtfsService                                  |                  |       | 1414              | 1414  | 1414      | 1414  |
| GtfsShape                                    |                  |       | 582               | 582   | 582       | 582   |
| GtfsStation                                  |                  |       | 6886              | 6886  | 6886      | 6886  |
| GtfsStop                                     |                  |       | 10618             | 10618 | 10618     | 10618 |
| GtfsStopTime                                 |                  |       | 71356             | 71356 | 71356     | 71356 |
| GtfsTransitFeedFile                          |                  |       | 4                 | 0     | 4         | 0     |
| GtfsTrip                                     |                  |       | 1414              | 1414  | 1414      | 1414  |
| OffStreetParking                             |                  |       | 74                | 74    | 74        | 74    |
| PointOfInterest                              |                  |       | 75999             | 75999 | 75999     | 75999 |
| RestrictedTrafficArea                        |                  |       | 177               | 177   | 177       | 177   |
| Road                                         |                  |       | 4331              | 4331  | 4331      | 4331  |

|                        |  |  |      |      |      |      |
|------------------------|--|--|------|------|------|------|
| Vehicle                |  |  | 2    | 0    | 2    | 0    |
| WeatherObserved        |  |  | 4253 | 4253 | 4253 | 4253 |
| AirQualityObserved     |  |  | 852  | 852  | 852  | 852  |
| BikeHireDockingStation |  |  | 767  | 767  | 767  | 767  |

Table 21. Evolution of the Milan's data model validation. The table depicts the status in the first and second validations, as well as the increment between them.

### 3.8 Carouge's Reference Zone MIMs validation

#### 3.8.1 Endpoints, interfaces and core components

| CAROUGE's Reference Zone SynchroniCity Framework INSTANCE |                                                                                                           |                               |                |                                      |
|-----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|-------------------------------|----------------|--------------------------------------|
| Interfaces (Interoperability Points)                      | Endpoint                                                                                                  | Description                   | Auth. Required | Reference standards / Specifications |
| Context Information Management                            | <a href="https://orion.cityreport.org:5005">https://orion.cityreport.org:5005</a>                         | Orion Context Broker endpoint | YES            | NGSI Interface: NGSIv2               |
| Identity Management                                       | <a href="https://keyrock.cityreport.org:443">https://keyrock.cityreport.org:443</a>                       | WSO2                          | YES            | OAuth 2.0                            |
| Historical Data                                           | <a href="https://orion.cityreport.org:5005/v2/entities">https://orion.cityreport.org:5005/v2/entities</a> | Proprietary                   | YES            | SynchroniCity Historical API         |

Table 22. Carouge's deployed components (Synchronicity Architecture)

#### 3.8.2 Data models and available datasets

| CAROUGE's Reference Zone data model validation |                  |       |                   |       |           |       |
|------------------------------------------------|------------------|-------|-------------------|-------|-----------|-------|
| Dataset                                        | First Validation |       | Second Validation |       | Increment |       |
|                                                | Total            | Valid | Total             | Valid | Total     | Valid |
| AirQualityObserved                             |                  |       | 12                | 12    | 12        | 12    |
| OffStreetParking                               |                  |       | 25                | 24    | 25        | 24    |
| OnStreetParking                                |                  |       | 16                | 16    | 16        | 16    |
| ParkingSpot                                    |                  |       | 1                 | 0     | 1         | 0     |
| TrafficFlowObserved                            |                  |       | 254               | 254   | 254       | 254   |
| Vehicle                                        |                  |       | 18                | 18    | 18        | 18    |

Table 23. Evolution of the Carouge's data model validation. The table depicts the status in the first and second validations, as well as the increment between them.

## 4 Atomic Services Validation

---

The Synchronicity atomic service validation process deployed the D3.3 [2] listed services following the documented steps to build the three described scenarios in section 2.2.2. Besides the evaluation of each AS, this process also validates the orchestration capabilities of these services on top of the SynchroniCity framework. It is worth noting that the validation is performed playing the role of an external user, meaning that the knowledge of the platform and the services themselves is limited. It puts the focus on the following aspects:

- Service description and deployment documentation
- Service deployment and integration with a given SynchroniCity Framework. As an external user, it is assumed that authorization integration is a desirable feature, regardless of the initial consideration in the service design.
- APIs, functionalities, features and covered functional requirements.

### 4.1 Scenario 1 (Routing) Validation

This scenario combines GTFS [8] information feeds with the SynchroniCity OTP-based routing service to provide multimodal routes. It will test and validate the capabilities of Synchronicity framework to support GTFS files management and GTFS-Real Time feeds generation and later use this information to calculate and update routes within the city.

For the layout of the scenario (Figure 5), the Santander Synchronicity instance will be used, providing:

- GtfsTransitFeedFile Entity [*urn:ngsi-Id:GtfsTransitFeedFile:santander:TUS*] which includes the link to the Santander Urban Transport Buses (TUS) GTFS feed file.
- Urban Mobility entities representing the ferry service (Reginas [33]) that crosses Santander's Bay
  - o *urn:ngsi-Id:GtfsAgency:santander:Ferry:Reginas*
  - o *urn:ngsi-Id:GtfsStop:santander:Ferry:Embarcadero\_SAN*
  - o *urn:ngsi-Id:GtfsStop:santander:Ferry:Embarcadero\_PED*
  - o *urn:ngsi-Id:GtfsStop:santander:Ferry:Embarcadero\_SOM*
  - o *urn:ngsi-Id:GtfsRoute:santander:Ferry:Reginas:Bahia*
  - o *urn:ngsi-Id:GtfsTrip:santander:Ferry:Reginas:Bahia:0*
  - o *urn:ngsi-Id:GtfsTrip:santander:Ferry:Reginas:Bahia:1*
  - o *urn:ngsi-Id:GtfsShape:santander:Ferry:Reginas:Bahia:Shape0*
  - o *urn:ngsi-Id:GtfsShape:santander:Ferry:Reginas:Bahia:Shape1*
  - o *urn:ngsi-Id:GtfsService:santander:Ferry:Reginas:Regular*
  - o *urn:ngsi-Id:GtfsCalendarRule:santander:Ferry:Reginas:san\_somo*
  - o *urn:ngsi-Id:GtfsFrequency:santander:Ferry:Reginas:Regular:san\_somo*
  - o *urn:ngsi-Id:GtfsFrequency:santander:Ferry:Reginas:Regular:somo\_san*
  - o *urn:ngsi-Id:GtfsStopTime:santander:Ferry:Reginas:Regular:san\_somo:Embarcadero\_SAN:1*
  - o *urn:ngsi-Id:GtfsStopTime:santander:Ferry:Reginas:Regular:san\_somo:Embarcadero\_PED:1*
  - o *urn:ngsi-Id:GtfsStopTime:santander:Ferry:Reginas:Regular:san\_somo:Embarcadero\_SOM:1*
  - o *urn:ngsi-Id:GtfsStopTime:santander:Ferry:Reginas:Regular:somo\_san:Embarcadero\_SOM:1*
  - o *urn:ngsi-Id:GtfsStopTime:santander:Ferry:Reginas:Regular:somo\_san:Embarcadero\_PED:1*
  - o *urn:ngsi-Id:GtfsStopTime:santander:Ferry:Reginas:Regular:somo\_san:Embarcadero\_SAN:1*
  - o *urn:ngsi-Id:GtfsCalendarDateRule:santander:Ferry:RuleTest1*

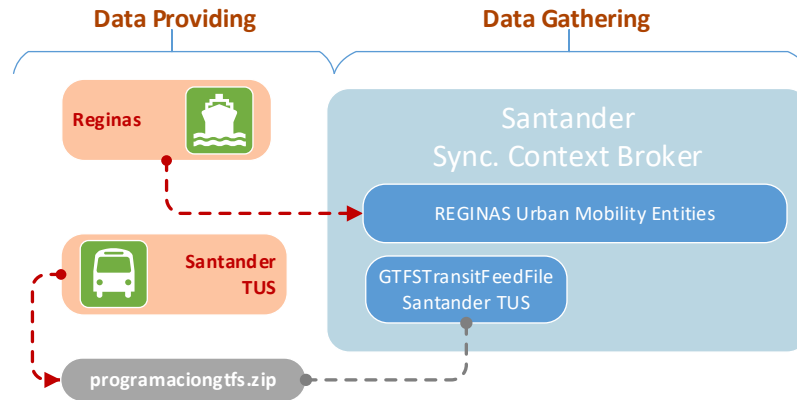


Figure 5. Routing Scenario Santander's Layout

#### 4.1.1 NGSI Urban Mobility to GTFS

This atomic service will produce a new GTFS Feed File following the steps in Figure 6:

- (A-1) the service reads (through NGSI API) the Urban Mobility Entities related to the same GtfsAgency [*Reginas GTFS agency*].
- (A-2) it generates a GTFS transit feed file (in .zip format) with the required .txt files and containing the related context information [*result\_urn\_ngsi-Id\_GtfsAgency\_santander\_Ferry\_Reginas.zip*]
- (A-3) the zip file is mapped in the Context Broker through an GtfsTransitFeedFile entity [*urn:ngsi-Id:GtfsTransitFeedFile:santander:Ferry:Reginas*]

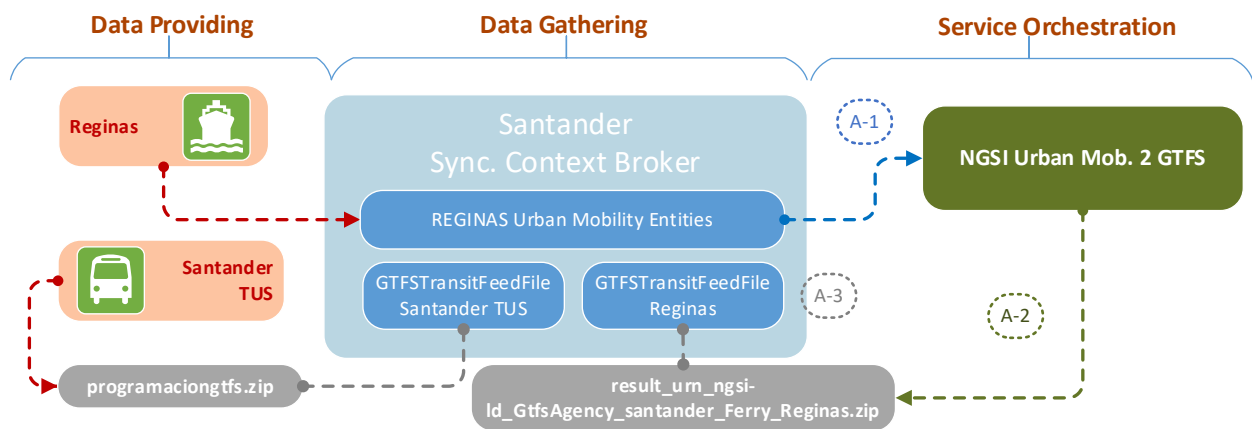


Figure 6. NGSI Urban Mobility to GTFS operations

According documentation, the service is configured with this parameters [*config.js*]

```
//ID of the GtfsAgency entity in the Context Broker to feed the GTFS file
agencyId: 'urn:ngsi-Id:GtfsAgency:santander:Ferry:Reginas'

//Sync. Orion Instance with all the required entities
orionUrl: 'http://193.144.201.58:1026', // Orion Instance access (Internal UC one)
orionAuthHeaderName: 'x-auth-token', // Authorization Header name (OAuth)
orionAuthToken: 'test-token', // Authorization token (OAuth)
fiwareService: 'AtomicServices', // FIWARE-Service header to be put in the Orion request
fiwareServicePath: '', // FIWARE-ServicePath header to be put in the Orion request (not used in our tests)
```

The docker installation process has been used to create an instance of this service. It deploys on the test machine [193.144.201.50] an API REST, listening on port [8082] with two main functionalities:

- a) Retrieves information about the Urban Mobility entities stored in the referred Context Broker instance
  - a. /api/urbanmobility/agencies → GtfsAgency entities registered
  - b. /api/urbanmobility/routes/{agencyId} → GtfsRoute entities stored
  - c. /api/urbanmobility/trips/{routeId} → GtfsTrip entities linked to routes
- b) Creates a GTFS Transit Feed File, in .zip format, with all required GTFS .txt files to represent a Gtfs Agency transportation info.
  - a. /api/gtfs/agencies/{agencyId}

To obtain the GTFS Feed File:

```
curl -X GET \
  http://193.144.201.50:8082/api/gtfs/agencies/urn:ngsi-Id:GtfsAgency:santander:Ferry:Reginas
```

returns [result\_urn\_ngsi-Id\_GtfsAgency\_santander\_Ferry\_Reginas.zip] file which includes mapped:

- agency.txt ← urn:ngsi-Id:GtfsAgency:santander:Ferry:Reginas
- calendar.txt ← urn:ngsi-Id:GtfsCalendarRule:santander:Ferry:Reginas:san\_somo
- calendar\_dates.txt ← urn:ngsi-Id:GtfsCalendarDateRule:santander:Ferry:RuleTest1
- frecuencies.txt ← urn:ngsi-Id:GtfsFrequency:santander:Ferry:Reginas:Regular:X
- routes.txt ← urn:ngsi-Id:GtfsRoute:santander:Ferry:Reginas:Bahia
- shapes.txt ← urn:ngsi-Id:GtfsShape:santander:Ferry:Reginas:Bahia:X
- stop\_times.txt ← urn:ngsi-Id:GtfsStopTime:santander:Ferry:Reginas:Regular:X:X
- stops.txt ← urn:ngsi-Id:GtfsStop:santander:Ferry:X
- trips.txt ← urn:ngsi-Id:GtfsTrip:santander:Ferry:Reginas:Bahia:X

This file is shared through [http://193.144.201.50:10000/result\_urn\_ngsi-Id\_GtfsAgency\_santander\_Ferry\_Reginas.zip] and updated on the Context Broker instance using the GtfsTransitFeedFile entity:

```
{
  "id": "urn:ngsi-Id:GtfsTransitFeedFile:santander:Ferry:Reginas",
  "type": "GtfsTransitFeedFile",
  "dateCreated": "2019-03-27T15:00:05.00Z",
  "dateModified": "2019-03-28T00:00:05.00Z",
  "description": "Santander City Ferry Los Reginas",
  "endDate": "2019-07-29T23:59:59.00Z",
  "feedFileContent": [
    "agency.txt",
    "calendar.txt",
    "calendar_dates.txt",
    "routes.txt",
    "shapes.txt",
    "stops_times.txt",
    "stops.txt",
    "trips.txt",
    "frecuencies.txt"
  ],
  "feedFileURL": "http://193.144.201.50:10000/result_urn_ngsi-Id_GtfsAgency_santander_Ferry_Reginas.zip",
  "publisher": {
    "name": "Los Reginas",
    "publisherURL": "http://www.losreginas.com"
  },
  "referenceZone": "Santander Municipality - 39075",
  "source": "http://www.losreginas.com",
  "startDate": "2019-01-01T06:00:00.00Z",
  "version": {
    "id": "Los Reginas",
    "dateReleased": "2019-03-27T00:00:00.00Z"
  }
}
```

#### 4.1.1.1 Documentation

| Atomic Service Ref: <a href="https://gitlab.com/synchronicity-iot/ngsiurbanmobility-to-gtfs">https://gitlab.com/synchronicity-iot/ngsiurbanmobility-to-gtfs</a> |                                                            |                                                                                                                                                                                                                                                                      |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Key Target                                                                                                                                                      | Dimensions validated                                       | Comments                                                                                                                                                                                                                                                             |
| Identification                                                                                                                                                  | Commercial info                                            | Affero General Public License (GPL) version 3 indicated. No info about versions is provided                                                                                                                                                                          |
|                                                                                                                                                                 | Technical Requirements                                     | Just NodeJS (>= 10.15.2) needs to be supported.                                                                                                                                                                                                                      |
|                                                                                                                                                                 | Potential end users/Services                               | No end users' description. This AS provides support to upper services (Routing).                                                                                                                                                                                     |
|                                                                                                                                                                 | AS qualitative description                                 | Full description of the features and operating modes are provided.                                                                                                                                                                                                   |
| Deployment description                                                                                                                                          | Set up & Running                                           | Two installation modes documented (Code and Docker). Clear set up and configuration instructions. Setup contact provided                                                                                                                                             |
|                                                                                                                                                                 | Usage                                                      | Operates as RESTful service. Full Apiary description provided (RESTful Interface): <ul style="list-style-type: none"> <li>- Query/retrieve urban mob. Entities</li> <li>- GtfsTransitFeedFile generation</li> </ul> API errors documented.<br>Some examples provided |
| Results                                                                                                                                                         | End-User Outcomes                                          | RESTful API                                                                                                                                                                                                                                                          |
|                                                                                                                                                                 | Reusable Outcomes (Integration/orchestration capabilities) | Consumes/Returns Sync. NGSI data models. Produces GTFS standard Transfer Feed files.<br>Requires NGSIv2 compliant Context Broker.                                                                                                                                    |

#### 4.1.1.2 Deployment & Integration with Synchronicity Framework

| Docker Deployment: <a href="https://hub.docker.com/r/synchronicityiot/ngsiurbanmobility-to-gtfs">https://hub.docker.com/r/synchronicityiot/ngsiurbanmobility-to-gtfs</a> |                            |                                                                                                                             |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| Key Target                                                                                                                                                               | Dimensions validated       | Comments                                                                                                                    |
| Deployment Procedure                                                                                                                                                     | Set up & Running execution | Simple Docker deployment according documentation (no further skills required).<br>Easy configuration through config.js file |
| Supported MIMs                                                                                                                                                           | NGSI                       | NGSIv2 compliant (info sources required to be stored in a Sync. Compliant CB)                                               |
|                                                                                                                                                                          | OAuth                      | Supports OAuth 2.0 tokens (to request Sync. Frameworks)                                                                     |
|                                                                                                                                                                          | Historical Data            | Neither required nor supported.                                                                                             |
|                                                                                                                                                                          | Data models                | Works with all Sync. Urban Mobility [25] data models                                                                        |
| Features Supported                                                                                                                                                       | Data access management     | Full support of FIWARE Services and FIWARE ServicePaths.<br>Generates compliant GTFS transfer feed files                    |

|  |               |                                                                                                                    |
|--|---------------|--------------------------------------------------------------------------------------------------------------------|
|  | Orchestration | Possible orchestration through NGSI API to add new GTFS feeds.<br>Provides GTFS data to compliant Routing Services |
|--|---------------|--------------------------------------------------------------------------------------------------------------------|

#### 4.1.1.3 Functional Requirements & Provided functionalities

This service does not directly address any functional requirement listed in D3.1 [21] but provides upper services with the tools to address some specific ones:

| Multimodal Transportation Requirements |                                                                                                                                                                                                                                           |                                                                                                                                                                                                           |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FR code                                | FR description                                                                                                                                                                                                                            | Comments                                                                                                                                                                                                  |
| FR-5                                   | The user should be able to enter the departure point, destination point and optionally other intermediates points of an itinerary writing their addresses, GPS coordinates or Point of Interest (PoI).                                    | This service allows the Routing atomic service to recognize PoI and other geographical items, such as stations, bus stops, etc., provided by a NGSI compliant data source (such as Orion Context Broker). |
| FR-12                                  | The user should be able to view on a map the points of interest located in an area near to the route.                                                                                                                                     | This service allows to provision PoI (e.g.: station, bus stops, etc.) described in NGSI as GTFS files to GTFS compliant services (such as Routing Services)                                               |
| FR-13                                  | The application may consider real-time information (deleted rides, delays, etc.) to better plan and update the journey                                                                                                                    | This service offers to application not compatible to NGSI a convenient way to get data about transportation from NGSI compliant sources (such as Orion Context Broker).                                   |
| Validated                              | Outcomes                                                                                                                                                                                                                                  |                                                                                                                                                                                                           |
|                                        | Service creates a GTFS Transit feed file including the Stops (Urban Mob. GtfsStop entities) registered in the CB. This file can be loaded by the Routing service so Bus/Metro/Ferry (etc.) stops could be located on the provided routes. |                                                                                                                                                                                                           |

#### 4.1.2 GTFS Fetcher

This atomic service is directly linked to the Routing Service based on Open Trip Planner and fetches GTFS feed files (in .zip format) according links stored in GtfsTransitFeedFile entities stored in the CB as shown in Figure 7:

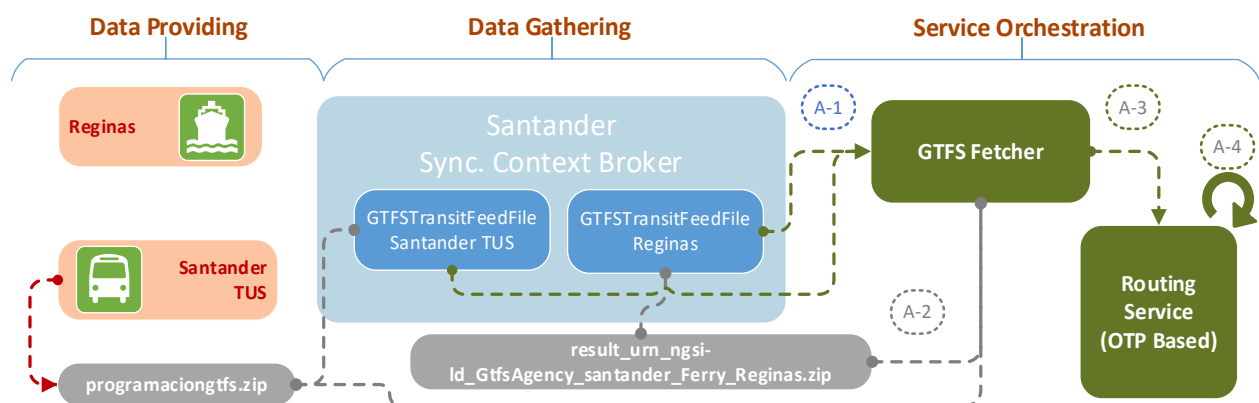


Figure 7. GTFS Fetcher operations

- (A-1) the service is subscribed (through NGSI API) to the GTFSTransitFeedFile entities that include the links to the GTFS .zip files to be later included in the route calculation. In our case `[urn:ngsi-Id:GtfsTransitFeedFile:santander:Ferry:Reginas]` and `[urn:ngsi-Id:GtfsTransitFeedFile:santander:TUS]`.
- (A-2) When the subscription is done and on every update on the affected entities, the linked zip files (`[programaciongtfs.zip]` and `[result_urn_ngsi-Id_GtfsAgency_santander_Ferry_Reginas.zip]`) are fetched and
- (A-3) uploaded in the corresponding Routing Service folder.
- (A-4) Then, the service reloads the maps and the internal databases of the Routing Service (according OTP documentation) so the new provided routes can include the updated GTFS information.

The complete service includes a fetching action to retrieve GTFS zip files, an OTP-Routing based service update with this information and a conversion from GTFS txt files (in the retrieved zip file) into Urban Mobility context entities to be stored in the pointed context broker. This routing scenario will focus only on the two first functionalities: GTFS zip files fetching and Routing service update.

GTFS-Fetcher SynchroniCity atomic service is distributed as a docker [20]. Configuration of this service can be done through the provided `[gtfs-fetcher/deployment/docker-compose.yml]` file (only gtfs-fetcher service is shown):

```
gtfs-fetcher:
  build: ../
  environment:
    - ORION_URL=http://193.144.201.58:10026 # SAN test Synch. Orion CB
    - FIWARE_SERVICE='AtomicServices' #FIWARE Service for scenarios
    - OUTPUT_FOLDER=./output
    - ORION_ENTITY_PATTERNS=[".*GtfsTransitFeedFile.*"] # To retrieve all GTFS zip files
    - IGNORE_EXPIRED_GTFS=false
    - PLUGIN_COMMAND_ARRAY=["python plugins/otp/otp_plugin_example.py"] # Deploys only the Gtfs-Fetcher
```

Once the configured docker-compose file is executed, two services are uploaded:

- **gtfs-fetcher** service: on CLI mode (by default) will access the given context broker to read ALL GtfsTransitFeedFile entities (as configured), download the pointed .ZIP files and reload the OTP maps.
- **otp** service: deploys an Open Trip Planner [16] instance to support the routes calculation (listening on port 8080 by default). This instance can be replaced by another OTP already available.

When the gtfs-fetcher is called

```
docker-compose exec gtfs-fetcher python gtfs_fetcher_run_once.py
```

.zip files linked by the present GtfsTransitFeedFiles are downloaded and stored in the `[otp-build-files/]` folder:

- `urn:ngsi-Id:GtfsTransitFeedFile:santander:TUS` → `programaciongtfs.zip`
- `urn:ngsi-Id:GtfsTransitFeedFile:santander:Ferry:Reginas` → `result_urn_ngsi-Id_GtfsAgency_santander_Ferry_Reginas.zip`

Automatically, OTP is called to reload the maps supporting the route calculation, so this updated GTFS information is considered.

#### 4.1.2.1 Documentation

Atomic Service Ref: <https://gitlab.com/synchronicity-iot/gtfs-fetcher>

| Key Target | Dimensions validated | Comments |
|------------|----------------------|----------|
|------------|----------------------|----------|

|                        |                                                            |                                                                                                                                                                             |
|------------------------|------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Identification         | Commercial info                                            | Distributed as free software (GNU General Public License V3)                                                                                                                |
|                        | Technical Requirements                                     | Built on top of Python 2.7. Supported by the docker deployment.                                                                                                             |
|                        | Potential end users/Services                               | No end users' description. This AS provides support to upper services (Routing).                                                                                            |
|                        | AS qualitative description                                 | Functional description needs to be elaborated.                                                                                                                              |
| Deployment description | Set up & Running                                           | Docker deployment documented. Instructions are enough to deploy the services but sometimes is confusing. This can be improved. Graphical doc is included.                   |
|                        | Usage                                                      | Offers a Python CLI to execute the fetching. Errors not documented. No further functionalities are documented.                                                              |
| Results                | End-User Outcomes                                          | Python-based plugins that can be configured by the end user. CLI to trigger fetching operations. It works only with OTP-based routing service.                              |
|                        | Reusable Outcomes (Integration/orchestration capabilities) | Consumes/Returns Sync. NGSI data models (Urban Mobility Sync. data models).<br>Manage GTFS zip files WITH OTP Routing service.<br>Requires NGSIv2 compliant Context Broker. |

#### 4.1.2.2 Deployment & Integration with Synchronicity Framework

| Docker Deployment: <a href="https://gitlab.com/synchronicity-iot/gtfs-fetcher/tree/master/deployment">https://gitlab.com/synchronicity-iot/gtfs-fetcher/tree/master/deployment</a> |                            |                                                                                                                                                                                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Key Target                                                                                                                                                                         | Dimensions validated       | Comments                                                                                                                                                                                     |
| Deployment Procedure                                                                                                                                                               | Set up & Running execution | Simple Docker deployment according documentation (main configuration parameters through docker-compose.yml file)<br>Hard to follow without solid knowledge of OTP.<br>Errors not documented. |
| Supported MIMs                                                                                                                                                                     | NGSI                       | NGSIv2 compliant (info sources required to be stored in a Sync. Compliant CB)                                                                                                                |
|                                                                                                                                                                                    | OAuth                      | No Security API is supported                                                                                                                                                                 |
|                                                                                                                                                                                    | Historical Data            | Neither required nor supported.                                                                                                                                                              |
|                                                                                                                                                                                    | Data models                | Works with all Sync. Urban Mobility [25] data models                                                                                                                                         |
| Features Supported                                                                                                                                                                 | Data access management     | Full support of FIWARE Services and FIWARE ServicePaths.<br>Manages Urban Mobility Sync. entities                                                                                            |
|                                                                                                                                                                                    | Orchestration              | Possible orchestration through NGSI API to add new GTFS feeds.<br>Supports only OTP-Based routing services                                                                                   |

#### 4.1.2.3 Functional Requirements & Provided functionalities

| Multimodal Transportation Requirements |                                                                                                                                                                                                                                                  |                                                                                                                                                                                |
|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FR code                                | FR description                                                                                                                                                                                                                                   | Comments                                                                                                                                                                       |
| FR-5                                   | The user should be able to enter the departure point, destination point and optionally other intermediates points of an itinerary writing their addresses, GPS coordinates or Point of Interest (Pol).                                           | This service allows the Routing atomic service to recognize Pol and other geographical items, such as stations, bus stops, etc, provided by a compliant GTFS Transit Feed File |
| FR-12                                  | The user should be able to view on a map the points of interest located in an area near to the route.                                                                                                                                            | GTFS Fetcher allows to provision Pol encapsulated in GTFS to GTFS compliant services (such as Routing Services)                                                                |
| Validated                              | Outcomes                                                                                                                                                                                                                                         |                                                                                                                                                                                |
|                                        | Service succesfully imports GTFS Transit feed files including the Stops (Urban Mob. GtfsStop entities) provided by third parties through GtfsTransitFeedFile entities registered in the CB. It also reloads the maps on the OTP Routing service. |                                                                                                                                                                                |

#### 4.1.3 GTFS-RT Loader

GTFS-RT (Real-Time) is a feed specification that allows public transportation agencies to provide real-time updates about their fleet to application developers. It is an extension to GTFS. This service provides notifications of real-time information in a widely used format, GTFS-RT. The atomic service consumes information following the Synchronicity data models using an NSGI API and generates GTFS-RT feeds from it.

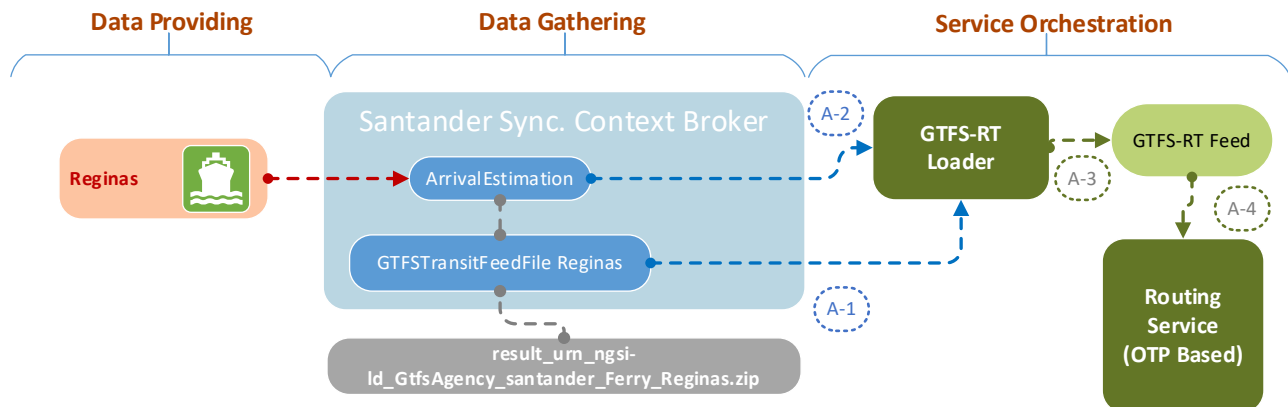


Figure 8. GTFS-RT Loader from NSGI operations

As shown in Figure 8, the steps are:

- (A-1) The GTFS Real Time updates are always related to a main GTFS feed file with all information related to the transport company. For this test, we will use the file generated for the Regina's Ferry provider. `[urn:ngsi-Id:GtfsTransitFeedFile:santander:Ferry:Reginas]` is registered in the GTFS-RT loader service.
- (A-2) ArrivalEstimation entities (from Synchronicity Urban Mobility data models) linked to the registered GTFS feed file are generated containing time delays related to included trips. GTFS-RT service is subscribed and aware of these entities.
- (A-3) GTFS-RT loader generates and updates a GTFS-RT feed including this information, according GTFS specifications. This feed is exposed through a REST resource

- (A-4) Out of the scope of this AS, the generated GTFS-RT feed is provided to a Routing Service supporting this technology. New routes calculated by this service will include now real time updates on arrival times.

GTFS-RT loader from NGSI SynchroniCity atomic service is distributed as a Dockerfile and Source code. Configuration of this service can be done through the provided `[config.js]` file:

```
config.httpPort = 9090; // service API port, if not set it uses 8080
config.httpPort_admin = 9091; // administration API port, if not set it uses 8081
config.tripUpdatesProtoPath = './protos/gtfs-realtime.proto'; // .proto file for TripUpdates.
config.contextBroker = 'http://193.144.201.58:10026'; // URL of the associated context broker
config.fiwareService = 'AtomicServices' //identifies de FIWARE Service to get access to the required entities
config.notifyProtocol = 'http'; // 'http' or 'https', by default it takes 'http'
config.notifyAddress = '193.144.201.50'; // IP address or domain (used to send the notification)
config.gtfsrtRefresh = 300; // GTFS-RT refreshing period in seconds, if not set it uses 300 seconds
config.notifyThrottling = 30; // notification throttling in seconds, if not set it uses 30 seconds
```

Following docker guidelines, the deployed GTFS-RT Loader service offers two ports:

- Management API listening on port 9091
- Service API listening on port 9090

As first step, the `[urn:ngsi-Id:GtfsTransitFeedFile:santander:Ferry:Reginas]` entity, containing the GTFS information of the Santander Ferry service is registered in the service.

```
POST 193.144.201.50:9091/management/feed/urn:ngsi-Id:GtfsTransitFeedFile:santander:Ferry:Reginas
```

This creates a subscription in the configured CB to all ArrivalEstimation entities linked to the registered GtfsTransitFeedFile entity.

```
{
  "id": "5cae08a38dc9b1af761f3400",
  "description": "NSGI/GTFS-RT loader subscription for TripUpdates of urn:ngsi-Id:GtfsTransitFeedFile:santander:Ferry:Reginas",
  "expires": "2020-04-10T15:18:40.00Z",
  "status": "active",
  "subject": {
    "entities": [
      {
        "idPattern": ".*",
        "type": "ArrivalEstimation"
      }
    ],
    "condition": {
      "attrs": [
        "lastUpdatedAt"
      ],
      "expression": {
        "q": "refGtfsTransitFeedFile==urn:ngsi-Id:GtfsTransitFeedFile:santander:Ferry:Reginas"
      }
    }
  },
  "notification": {
    "timesSent": 1,
    "lastNotification": "2019-04-10T15:15:47.00Z",
    "attrs": [
      "refGtfsTransitFeedFile",
      "routeId",
      "stopId",
      "lastUpdatedAt",
      "arrivalEstimationUpdate"
    ],
    "attrsFormat": "keyValues",
    "http": {
      "url": "http://193.144.201.50:9091/notify"
    },
    "lastSuccess": "2019-04-10T15:15:47.00Z"
  },
}
```

```

"throttling": 30
}

```

If a new *ArrivalEstimation* entity is now created or updated and linked to *[urn:ngsi-Id:GtfsTransitFeedFile:santander:Ferry:Reginas]* in the referred CB,

```

{
  "id": "urn:ngsi-Id:ArrivalEstimation:santander:Ferry:Reginas:Bahia:0:est00",
  "type": "ArrivalEstimation",
  "refGtfsTransitFeedFile": {
    "value": "urn:ngsi-Id:GtfsTransitFeedFile:santander:Ferry:Reginas"
  },
  "routeId": {
    "value": "urn:ngsi-Id:GtfsRoute:santander:Ferry:Reginas:Bahia"
  },
  "stopId": {
    "value": "urn:ngsi-Id:GtfsStop:santander:Ferry:Embarcadero_SOMO"
  },
  "lastUpdatedAt": {
    "value": "2019-04-10T15:14:00.238Z",
    "type": "DateTime"
  },
  "arrivalEstimationUpdate": {
    "value": [
      {
        "tripId": "urn:ngsi-Id:GtfsTrip:santander:Ferry:Reginas:Bahia:0",
        "arrivalDelay": -600
      },
      {
        "tripId": "urn:ngsi-Id:GtfsTrip:santander:Ferry:Reginas:Bahia:1",
        "arrivalDelay": 600
      }
    ]
  }
}

```

a notification would be sent to the GTFS-RT Loader service

```

gtfsrt_from_ngsi_1 | [2019-04-10T16:04:04.022Z [gtfsrt-ngsi-loader] info: Processing entity urn:ngsi-Id:ArrivalEstimation:santander:Ferry:Reginas:Bahia:0:est00

```

which will process the trip updates and automatically generates the GTFS-RT binary file. This GTFS-RT feed is served by the GTFS-RT Loader through

```

GET http://193.144.201.50:9090/gtfsrt/urn:ngsi-Id:GtfsTransitFeedFile:santander:Ferry:Reginas

```

The log of the service shows this serving action:

```

gtfsrt_from_ngsi_1 | [2019-04-10T16:33:01.701Z [gtfsrt-ngsi-loader] debug: Serving feed urn:ngsi-Id:GtfsTransitFeedFile:santander:Ferry:Reginas

```

#### 4.1.3.1 Documentation

| Atomic Service Ref: <a href="https://gitlab.com/synchronicity-iot/gtfsrt-loader-from-nsgi">https://gitlab.com/synchronicity-iot/gtfsrt-loader-from-nsgi</a> |                              |                                                                                            |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------|
| Key Target                                                                                                                                                  | Dimensions validated         | Comments                                                                                   |
| Identification                                                                                                                                              | Commercial info              | Version 1.0. AGPL v3                                                                       |
|                                                                                                                                                             | Technical Requirements       | Built on top Node JS (version 11.4.0). Supported by the docker deployment.                 |
|                                                                                                                                                             | Potential end users/Services | To be used by GTFS-RT compliant services. It provides support to upper services (Routing). |

|                        |                                                            |                                                                                                                                                                                      |
|------------------------|------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                        | AS qualitative description                                 | Complete description of the provided functionalities. Linked to GTFS-RT specifications.<br>Detailed description of the protocols and formats the outcomes are provided               |
| Deployment description | Set up & Running                                           | Native service (NodeJS) or Docker deployments and setups are provided. Easy to follow configuration instructions enriched with examples.                                             |
|                        | Usage                                                      | RESTful management and Service APIs. Documented. Test examples provided.                                                                                                             |
| Results                | End-User Outcomes                                          | RESTful APIs to fully operate with all service functionalities.<br>JSON format and Sync. data models used to manage data incomes.<br>GTFS-RT binary feeds as exploitable outcomes.   |
|                        | Reusable Outcomes (Integration/orchestration capabilities) | Consumes/Returns Sync. NGSI data models (ArrivalEstimation & GtfsTransitFeedFile).<br>Requires NGSIv2 compliant Context Broker.<br>Compliant with any end-service supporting GTFS-RT |

#### 4.1.3.2 Deployment & Integration with Synchronicity Framework

| Docker Deployment: <a href="https://gitlab.com/synchronicity-iot/gtfsrt-loader-from-nsqi/tree/master#docker">https://gitlab.com/synchronicity-iot/gtfsrt-loader-from-nsqi/tree/master#docker</a> |                            |                                                                                                                                                                                     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Key Target                                                                                                                                                                                       | Dimensions validated       | Comments                                                                                                                                                                            |
| Deployment Procedure                                                                                                                                                                             | Set up & Running execution | Simple Docker deployment according documentation. All configuration can be done through the config.js file. Example provided.<br>Required GTFS-RT basic knowledge (links provided). |
| Supported MIMs                                                                                                                                                                                   | NGSI                       | NGSIv2 compliant (required info sources must be provided by a Sync. Compliant CB)                                                                                                   |
|                                                                                                                                                                                                  | OAuth                      | No Security API is supported                                                                                                                                                        |
|                                                                                                                                                                                                  | Historical Data            | Neither required nor supported                                                                                                                                                      |
|                                                                                                                                                                                                  | Data models                | Works with ArrivalEstimation and GtfsTransitFeedFile Sync. data models                                                                                                              |
| Features Supported                                                                                                                                                                               | Data access management     | Full support of FIWARE Services and FIWARE ServicePaths.<br>Manages access to ArrivalEstimation and GtfsTransitFeedFile entities                                                    |
|                                                                                                                                                                                                  | Orchestration              | Possible orchestration through NGSI API to add new GTFS feeds.<br>Supports all end-services GTFS-RT compliant                                                                       |

#### 4.1.3.3 Functional Requirements & Provided functionalities

This atomic service addresses functional requirements FR-8 and FR-13 of the multi-modal transportation application theme. In particular, the service provides real-time feeds in a widely used format (GTFS-RT) generated from SynchroniCity data.

| Multimodal Transportation Requirements |                                                                                                                                                                                                                                                        |                                                                                                                                                        |
|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| FR code                                | FR description                                                                                                                                                                                                                                         | Comments                                                                                                                                               |
| FR-8                                   | <i>The application must permit to plan routes by taking into account user preferences (panoramic itinerary, air quality, lowest cost, lowest time, etc.) and requirements</i>                                                                          | <i>GTFS-RT Loader atomic service allows real-time data (such as time delays) announced via NGSI to be injected to Routing atomic service as input.</i> |
| FR-13                                  | <i>The user should be able to view on a map the points of interest located in an area near to the route.</i>                                                                                                                                           | <i>The GTFS-RT loader enables the final service to address this requirement.</i>                                                                       |
| Validated                              | Outcomes                                                                                                                                                                                                                                               |                                                                                                                                                        |
|                                        | Service succesfully creates GTFS-RT compliant feeds with real time info provided by the Context Entities (through ArrivalEstimation entities). These feeds are reusable by ANY service supporting GTFS-RT protocol (as the Sync. Routing Service does) |                                                                                                                                                        |

#### 4.1.4 Routing Service

The Synchronicity Routing Service provides the end user with multimodal routes that potentially can combine buses, bicycles, taxis and any other public/private transportation service, even pedestrian routes. This service is based on Open Trip Planner (OTP) software and can exploit different data formats (GTFS, GTFS-RT, shape files, Open Streep Maps tiles, etc.) to calculate different route alternatives.

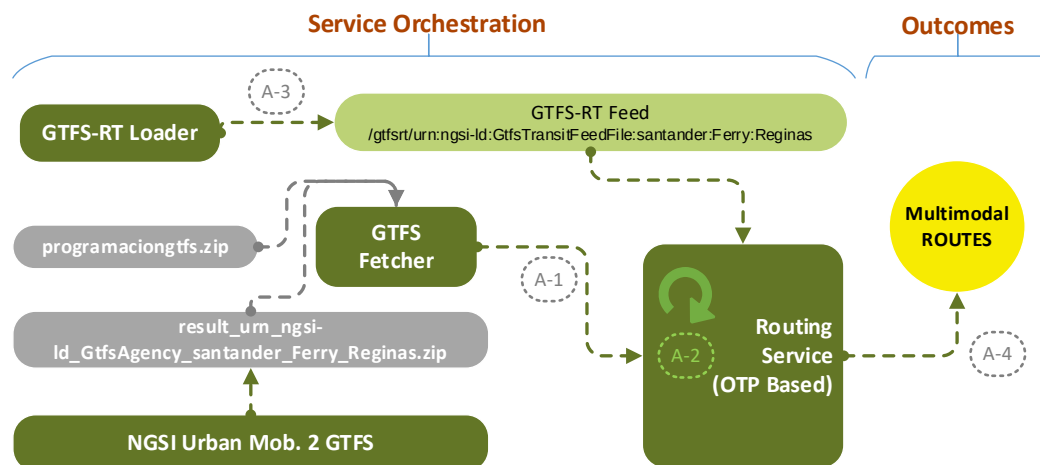


Figure 9. Routing Service orchestration

As shown in Figure 9, this multiformat support has been used by SynchroniCity atomic services to extract and inject useful context information into the route calculation process:

- NGSI Urban Mob. To GTFS: creates the GTFS.zip file from the Urban Mobility entities related to the Santander's Ferry service (Reginas) and stores it in a GtfsTransitFeedFile in the Sync. CB.

- GTFS Fetcher: reads and downloads (in the Routing Service routers folder) the GTFS Transit Feed Files (in zip format) available through the registered GtfsTransitFeedFile entities in the CB (data from Santander Buses and Santander Ferries). Once this is done, reloads the maps of the routing service, so these new updates are available for next calculated routes.
- GTFS-RT Loader: creates a GTFS-RT (real time) feed updating the arrival estimations on the trips for the Santander's Ferry service. This standard binary file is configured in the Routing Service through the OTP [*routing-config.json*] file:

```
{ "updaters": [ // Polling for GTFS-RT TripUpdates) {
  "type": "stop-time-updater",
  "frequencySec": 10,
  "sourceType": "gtfs-http",
  "url": "http://193.144.201.50:9090/gtfsrt/urn:ngsi-Id:GtfsTransitFeedFile:santander:Ferry:Reginas",
  "feedId": "1" } ]
}
```

Considering all referred atomic services running, the Routing service is deployed following the docker instructions provided:

- Graphical interface is deployed on [*http://193.144.201.50:8080*]
- RESTful API is deployed on [*http://193.144.201.50:8080/otp*]

The routing scenario orchestrator is launched according the following steps:

- (A-1) GTFS-Fetcher is called; GTFS Transit feed files provided directly by the Bus service (programaciongtfs.zip) and the one created by the NGSI Urban Mob. to GTFS atomic service (result\_urn\_ngsi-Id\_GtfsAgency\_santander\_Ferry\_Reginas.zip) are downloaded to the Routing Service.
- (A-2) GTFS-Fetcher triggers the new map generation, so the Routing service can consider the new GTFS feeds
- (A-3) A one hour (3600 secs) delay in Santander's Ferry service is introduced by an ArrivalEstimation entity linked to [*urn:ngsi-Id:GtfsTransitFeedFile:santander:Ferry:Reginas*] GTFS feed.

```
POST /v2/entities/urn:ngsi-Id:ArrivalEstimation:santander:Ferry:Reginas:Bahia:0:est01/attrs
{
  "refGtfsTransitFeedFile": {
    "value": "urn:ngsi-Id:GtfsTransitFeedFile:santander:Ferry:Reginas"
  },
  "routeId": {
    "value": "urn:ngsi-Id:GtfsRoute:santander:Ferry:Reginas:Bahia"
  },
  "stopId": {
    "value": "urn:ngsi-Id:GtfsStop:santander:Ferry:Embarcadero_SOMO"
  },
  "lastUpdatedAt": {
    "value": "2019-04-10T16:35:00.238Z",
    "type": "DateTime"
  },
  "arrivalEstimationUpdate": {
    "value": [
      {
        "tripId": "urn:ngsi-Id:GtfsTrip:santander:Ferry:Reginas:Bahia:0",
        "arrivalDelay": -3600
      },
      {
        "tripId": "urn:ngsi-Id:GtfsTrip:santander:Ferry:Reginas:Bahia:1",
        "arrivalDelay": -3600
      }
    ]
  }
}
```

GTFS-RT Loader updates corresponding GTFS-RT feed and serves it to the Routing Service.

- (A-4) Using the Routing service graphical web user interface, a multimodal route is requested, including Santander buses, ferries and pedestrian sections. The result of this request is shown in Figure 10. The ferry section starts with a 1 hour delay, as pointed by the GTFS-RT feed.

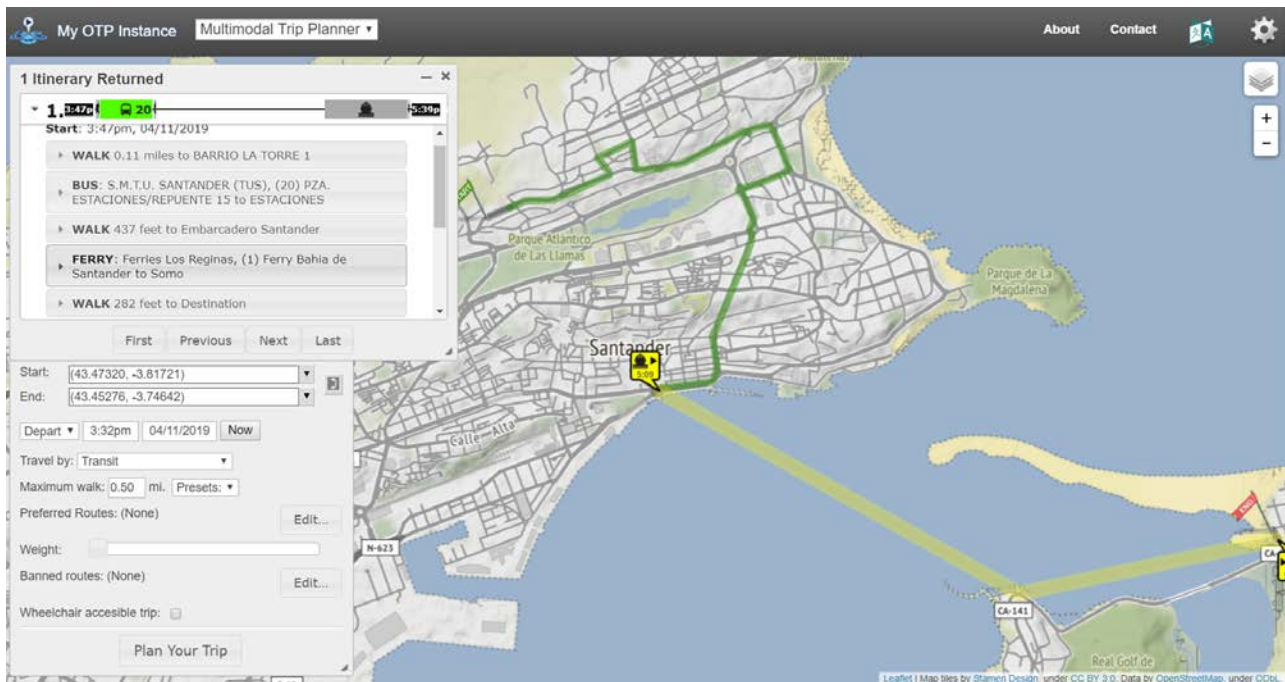


Figure 10. AS Multimodal route requested to Routing Service (Web User Interface)

#### 4.1.4.1 Documentation

| Atomic Service Ref: <a href="https://gitlab.com/synchronicity-iot/routing-atomic-service">https://gitlab.com/synchronicity-iot/routing-atomic-service</a> |                              |                                                                                                                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Key Target                                                                                                                                                | Dimensions validated         | Comments                                                                                                                                                                                                                                                                   |
| Identification                                                                                                                                            | Commercial info              | Licensing or version information not provided                                                                                                                                                                                                                              |
|                                                                                                                                                           | Technical Requirements       | Not specified.                                                                                                                                                                                                                                                             |
|                                                                                                                                                           | Potential end users/Services | Users requesting multimodal routes (via GraphQL) and/or services exploiting routes info (through OTP API)                                                                                                                                                                  |
|                                                                                                                                                           | AS qualitative description   | Description of the provided functionalities (Multimodal routes). Link to OTP expanded functionalities<br>Description of the protocols (GTFS and GTFS-RT) required to include Sync. Datasources.<br>Missing description (links) to the API and the outcomes provided by it. |
| Deployment description                                                                                                                                    | Set up & Running             | Docker deployment and setup documentation are provided.<br>Easy to follow configuration instructions for a basic deployment.<br>Missing documentation (or links) to configure/use expanded OTP capabilities.                                                               |

|         |                                                               |                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|         | Usage                                                         | RESTful API to manage OTP and request multimodal routes, but it lacks examples.<br>Graphical Interface to query/retrieve multimodal routes (no examples provided)                                                                                                                                                                                                                                  |
| Results | End-User Outcomes                                             | RESTful APIs to fully operate with all service functionalities <ul style="list-style-type: none"> <li>- OTP Management &amp; conf.</li> <li>- Query/retrieve multimodal routes</li> <li>- Configure "routers" (service providers)</li> </ul> GTFS-RT binary feeds as exploitable incomes.<br>GTFS Transit feed files compliant<br>Complete graphical interface to query/retrieve multimodal routes |
|         | Reusable Outcomes<br>(Integration/orchestration capabilities) | Consumes GTFS and GTFS-RT feeds<br>Produces multimodal routes in JSON format                                                                                                                                                                                                                                                                                                                       |

#### 4.1.4.2 Deployment & Integration with Synchronicity Framework

| Docker Deployment: <a href="https://gitlab.com/synchronicity-iot/routing-atomic-service#installation">https://gitlab.com/synchronicity-iot/routing-atomic-service#installation</a> |                            |                                                                                                                                                                                                                                                                                                               |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Key Target                                                                                                                                                                         | Dimensions validated       | Comments                                                                                                                                                                                                                                                                                                      |
| Deployment Procedure                                                                                                                                                               | Set up & Running execution | Simple Docker deployment according documentation.<br>Examples about how to configure it for Helsinki area. Missing some links to expand how to do it for different regions.<br>Required some OTP knowledge to expand configuration possibilities (e.g. how to include GTFS-RT feeds or Bicycle dockings info) |
| Supported MIMs                                                                                                                                                                     | NGSI                       | No direct NGSIv2 access required                                                                                                                                                                                                                                                                              |
|                                                                                                                                                                                    | OAuth                      | No (Sync.) Security API is required                                                                                                                                                                                                                                                                           |
|                                                                                                                                                                                    | Historical Data            | Neither required nor supported                                                                                                                                                                                                                                                                                |
|                                                                                                                                                                                    | Data models                | Not exploiting Sync. data models                                                                                                                                                                                                                                                                              |
| Features Supported                                                                                                                                                                 | Data access management     | No data directly required from Sync. Framework to operate                                                                                                                                                                                                                                                     |
|                                                                                                                                                                                    | Orchestration              | Exploits GTFS and GTFS-RT provided by Sync. ASs                                                                                                                                                                                                                                                               |

#### 4.1.4.3 Functional Requirements & Provided functionalities

| Multimodal Transportation Requirements |                                                                                                                    |                                                 |
|----------------------------------------|--------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| FR code                                | FR description                                                                                                     | Comments                                        |
| FR-4                                   | <i>The user should be able to enter (or provide by the GPS sensor embedded in his mobile device) the departure</i> | <i>Accepted as input by the routing service</i> |

|                  |                                                                                                                                                                                                                                                                                                   |                                                                                                                                                          |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | <i>point, destination point and optionally other intermediates points of an itinerary on a map.</i>                                                                                                                                                                                               |                                                                                                                                                          |
| <b>FR-6</b>      | <i>The user should be able to select among different traveler profiles (e.g.: by walk, drive a car, drive a bike, use public transportation, etc.).</i>                                                                                                                                           | <i>Accepted as input by the routing service</i>                                                                                                          |
| <b>FR-7</b>      | <i>The user must be able to combine multiple transportation modes and mobility infrastructures (e.g.: metro + bike sharing for last mile instead of bus).</i>                                                                                                                                     | <i>Accepted as input by the routing service</i>                                                                                                          |
| <b>FR-9</b>      | <i>The user should be able to view on map alternative routes and select the one he/she wants to use.</i>                                                                                                                                                                                          | <i>Alternative route supported by the routing service</i>                                                                                                |
| <b>FR-15</b>     | <i>The application should be able to take into account user related information in order to suggest routes based on his/her preferences.</i>                                                                                                                                                      | <i>Accepted as input by the routing service</i>                                                                                                          |
| <b>Validated</b> | <b>Outcomes</b>                                                                                                                                                                                                                                                                                   |                                                                                                                                                          |
|                  | <i>These functional requirements are covered (and offered) by the OTP itself. It allows multiple user options (both, through its exposed API and the Web User Interface), including the combination of multiple transportation options, alternatives selection and GPS information capturing.</i> |                                                                                                                                                          |
| <b>FR-5</b>      | <i>The user should be able to enter the departure point, destination point and optionally other intermediates points of an itinerary writing their addresses, GPS coordinates or Point of Interest (PoI).</i>                                                                                     | <i>Accepted as input by the routing service in cooperation with the "GTFS fetcher", NGSI Urban Mobility to GTFS" and GTFS-RT Loader" atomic services</i> |
| <b>Validated</b> | <b>Outcomes</b>                                                                                                                                                                                                                                                                                   |                                                                                                                                                          |
|                  | <i>Service successfully consumes GTFS and GTFS-RT feed files. It shows and includes in the provided routes the points included in these files. It also shows the information related to these points and allows them to be selected as departure/end/intermediate points.</i>                     |                                                                                                                                                          |
| <b>FR-8</b>      | <i>The application must permit to plan routes by taking into account user preferences (panoramic itinerary, air quality, lowest cost, lowest time, etc.) and requirements</i>                                                                                                                     | <i>Accepted as input by the routing service and supported by the GTFS-RT Loader atomic service</i>                                                       |
| <b>FR-13</b>     | <i>The user should be able to view on a map the points of interest located in an area near to the route.</i>                                                                                                                                                                                      | <i>The GTFS-RT loader enables the final service to address this requirement.</i>                                                                         |
| <b>Validated</b> | <b>Outcomes</b>                                                                                                                                                                                                                                                                                   |                                                                                                                                                          |
|                  | <i>Service successfully consumes GTFS-RT compliant feeds with real time info provided by the Context Entities (through ArrivalEstimation entities). These feeds are provided by the GTFS-RT loader Sync. AS)</i>                                                                                  |                                                                                                                                                          |

## 4.2 Scenario 2 (Estimation) Validation

The estimation scenario provides information gathered from parking sensors and traffic loops to Parking and Traffic estimator atomic services to calculate parking probabilities and traffic status. Both services combined can offer an overview of current and incoming traffic situation on a city. These atomic services exploit context resources and historical data to feed AI and machine learning engines and are clear examples about how Synchronicity framework can support advanced software components with multiple application areas.

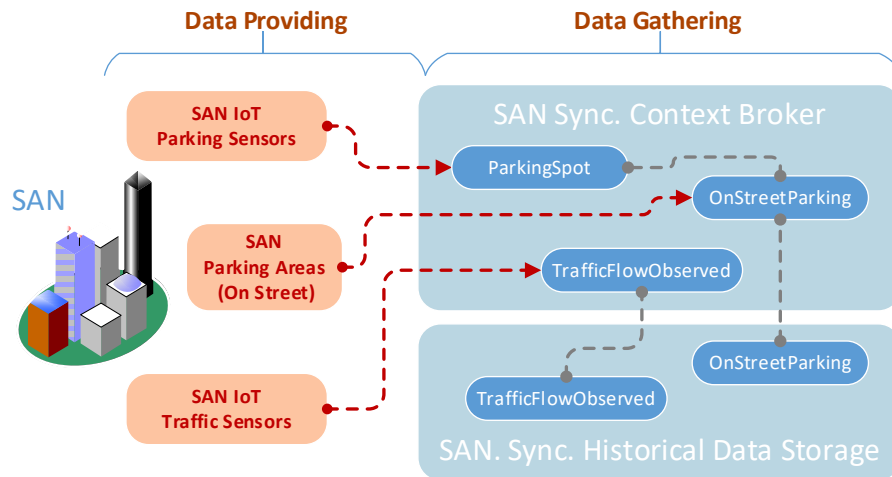


Figure 11. Estimation scenario layout for validation

The Estimation scenario layout (Figure 11) deployed in Santander to validate the Parking and Traffic estimators include:

- ParkingSpot (323 context entities) mapping the current status (free/occupied) of 323 on street parking lots.
- OnStreetParking (23 context entities) grouping the status of ParkingSpots and defining 23 parking areas in the city
- TrafficFlowObserved (311 context entities) showing the traffic status reported by 322 traffic loops deployed all along the city (covering the main city streets)

These context information sources will, in turn, feed:

- OnStreetParking historical data, keeping the historical status of the defined parking areas in Santander
- TrafficFlowObserved historical data, with the traffic occupancy of the main roads of Santander

Context data is offered through the SynchroniCity NGSI API. Historical Data is served through the defined SynchroniCity Historical Data Access API.

As set in both AS descriptions, due to the machine learning and AI components (Keras [34] and TensorFlow [35] required to implement the data analysis, these are high hardware demanding pieces of code (memory and computing mainly). To ensure the proper support of these components, an increase of the virtual memory size should be done.

```
sysctl -w vm.max_map_count=262144
```

Besides this, it is worth remarking the AI algorithms needed to estimate parking probability and extrapolate traffic flows are still under development. All the data gathering, pre-processing and presenting layers have been already set up and the efforts now are concentrated on the machine learning techniques applied to these datasets to obtain the best results.

### 4.2.1 Parking Probability Estimator

Parking Probability Estimator calculates the probability of finding a free parking spot within a pre-defined area in the city given by OnStreetParking and/or OffStreetParking Context entities. With the information retrieved from these entities, this service will:

- 1- Provide a (near) real-time status of a particular parking area
- 2- Generate an estimation of the probability of finding a free parking lot in the short term (i.e. in the next 15, 30 and 45 minutes)

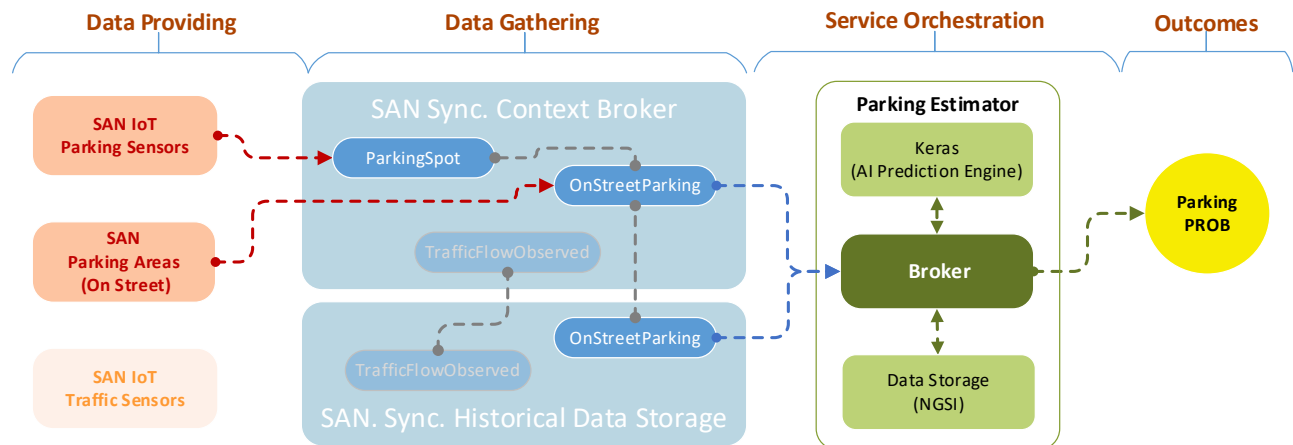


Figure 12. Parking Estimator deployment

The Parking Estimator deployment (Figure 12) consists of three main components:

- The Broker (or parking-estimator-api) builds a (REST) API that extracts data from the OnStreetParking and/or OffStreetParking entities, distributes it and harvests the estimated probabilities.
- The AI prediction engine based on Keras framework [Ref] is the core module that produces the estimations. It includes a training process to polish the predictions obtained.
- The NGSI Context Data storage relies on an internal Orion instance to keep track of the parking context elements and store the prediction results.

This AS allows two ways of deployment: using source code to compile the components; or set up the provided docker images. As this is a complex atomic service, the recommended installation will rely on the docker composition.

Paying attention to the guidelines, the hardware and software environments are configured for the testing environment

Config.js: links the broker with the local CB and Historical data providers

```

rzs: {
  brokers: [{
    rz: 'AS_Test',
    orion_host: 'http://193.144.201.58:10026', //Test environment Orion CB (Santander Parking data)
    historic_host: 'http://193.144.201.58:8080', //Test environment Historical data API (SAN Historical Data)
    auth: false,
    fiware_service: 'AtomicServices',
    area: [ // Santander location area (coordinates)
      [-4.198940, 43.332055],
      [-3.369473, 43.686628]
    ],
  }],
}
```

Config.ini: to configure the Keras and the AI environment

```
[API]
api_name = parkingestimator
endpoint = parking-estimator ;it will use the internal docker network names
[ELK]
elastic_host = elasticsearch ;it will use the internal docker network names
elastic_port = 9200
```

Finally, the environment variables are set to properly link to the data sources:

```
environment:
  SUBSCRIPTIONS_ENDPOINT: http:193.144.201.58:10026 # Mandatory parameter
  ORION_HOST: orion # Default option for the INTERNAL NGSI storage (containers on same host)
  AI_PREDICTION_ENGINE_HOST: ai-prediction-engine # Default option (containers on same host)
```

Running the docker-compose.yml file deploys:

- Parking estimator (port 3000) that supports the synchronicity-parking-estimator RESTful API [36].

Parking estimator reads data from Santander's Orion CB. A "List Parking areas" method (included in [36]) call retrieves (in the summary view) the 23 OnStreetParking areas available in the CB.

GET parkingestimator:3000/parkingareas

```
"summary": {
  "current_page": 1,
  "items_per_page": 23, // 23 OnStreetParking areas defined for Santander City
  "items": 23,
  "pages": 1
},
```

As pointed in the description of the Parking Estimator, the system (automatically) is now subscribed to the 23 available OnStreetParking entities and getting trained by analysing:

- Historical data related to the OnStreetParking entities gathered through the provided Sync. Historical data API (this source ONLY will be exploited were available)
- Notifications about current status of OnStreetParking entities received (and stored) by the component itself (creating its own historical dataset)

After one day of system training (it requires from a minimum set of previous 50 historical records or 50 received notifications to start calculating probabilities), we select one of the city areas (OnStreetParking entities) and request for the estimations.

GET parkingestimator:3000/parkingareas/urn:ngsi-Id:OnStreetParking:santander:parking:onStreet:StaLuciaEast/freeSpotEstimation

```
"freeSpotEstimation": { // Set of parking prob. estimations
  "type": "StructuredValue",
  "value": [
    6.896551724137931,
    8.433002644571765,
    8.433002644571765,
    8.433002644571765,
    8.433002644571765
  ],
  "metadata": {
    "estimationTimes": {
      "type": "StructuredValue",
      "value": [
        "2019-04-12T15:14:17.301Z",
```

|                                                                                                                                                                                                                   |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> "2019-04-12T15:29:17.301Z", "2019-04-12T15:44:17.301Z", "2019-04-12T15:59:17.301Z", "2019-04-12T16:14:17.301Z" ] } }, </pre>                                                                                |
| <pre> "location": { // Location of the City Area   "type": "geo:json",   "value": {     "type": "LineString",     "coordinates": [       [ -3.79858, 43.464448],       [ -3.796919, 43.464694]     ]   } } </pre> |

So, considering the request has been done on 12/04/2019 at 15:14 UTC, the probabilities of finding a free parking lot in Sta. Lucia East city area are shown in Table 24.

| Parking probability (%) | at                                       |
|-------------------------|------------------------------------------|
| 6.896551724137931       | 2019-04-12T15:14:17.301Z (now)           |
| 8.433002644571765       | 2019-04-12T15:29:17.301Z (in 15 minutes) |
| 8.433002644571765       | 2019-04-12T15:44:17.301Z (in 30 minutes) |
| 8.433002644571765       | 2019-04-12T15:59:17.301Z (in 45 minutes) |
| 8.433002644571765       | 2019-04-12T16:14:17.301Z (in 60 minutes) |

Table 24. Probability estimations returned by Parking Prob. Estimator service

As mentioned in the Santander's scenario layout, by the time this validation was done, the AI component for Paking estimation analysis is not completely finished, so the results are still not accurate.

#### 4.2.1.1 Documentation

| Atomic Service Ref: <a href="https://gitlab.com/synchronicity-iot/parking-estimator">https://gitlab.com/synchronicity-iot/parking-estimator</a> |                              |                                                                                                                                                                                                        |
|-------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Key Target                                                                                                                                      | Dimensions validated         | Comments                                                                                                                                                                                               |
| Identification                                                                                                                                  | Commercial info              | Licences and versions of each required component are provided                                                                                                                                          |
|                                                                                                                                                 | Technical Requirements       | Minimum hardware and software requirements are detailed                                                                                                                                                |
|                                                                                                                                                 | Potential end users/Services | To be used by upper services exploiting parking estimations.                                                                                                                                           |
|                                                                                                                                                 | AS qualitative description   | Description of the provided functionalities (Parking estimations for given time slots and for pre-defined city areas).<br>Description of the protocols and technologies used to implement the service. |

|                        |                                                            |                                                                                                                                                                                                                                                                        |
|------------------------|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Deployment description | Set up & Running                                           | Source code (on main components), Docker deployment and setup documentation are provided.<br><br>Easy to follow configuration instructions for a basic deployment.                                                                                                     |
|                        | Usage                                                      | RESTful API to query/retrieve parking areas, historical data and parking probabilities.                                                                                                                                                                                |
| Results                | End-User Outcomes                                          | RESTful API providing JSON outcomes for Query/retrieve Parking probability estimations <ul style="list-style-type: none"> <li>- List Parking Areas (covered by the service)</li> <li>- Detail each parking area</li> <li>- Get free parking spot estimation</li> </ul> |
|                        | Reusable Outcomes (Integration/orchestration capabilities) | Consumes NGSI On/OffStreetParking entities and NGSI ParkingSpot entities<br><br>Produces On/OffStreetParking entities including new "freeSpotEstimation" attribute.                                                                                                    |

#### 4.2.1.2 Deployment & Integration with Synchronicity Framework

| Docker Deployment: <a href="https://hub.docker.com/r/synchronicityiot/parking-estimator-api">https://hub.docker.com/r/synchronicityiot/parking-estimator-api</a> |                            |                                                                                                                    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|--------------------------------------------------------------------------------------------------------------------|
| Key Target                                                                                                                                                       | Dimensions validated       | Comments                                                                                                           |
| Deployment Procedure                                                                                                                                             | Set up & Running execution | Simple Docker deployment according documentation.<br>No extra skills are required                                  |
| Supported MIMs                                                                                                                                                   | NGSI                       | NGSiv2 compliant (info sources required to be stored in a Sync. Compliant CB)                                      |
|                                                                                                                                                                  | OAuth                      | Supports OAuth 2.0 tokens (to request Sync. Frameworks)                                                            |
|                                                                                                                                                                  | Historical Data            | Historical data related to On/OffStreetParking exploited (where available)                                         |
|                                                                                                                                                                  | Data models                | Requires from On/OffStreetParking NGSI entities                                                                    |
| Features Supported                                                                                                                                               | Data access management     | Full support of FIWARE Services and FIWARE ServicePaths.<br>Generates compliant Sync. data models (and attributes) |
|                                                                                                                                                                  | Orchestration              | Possible orchestration through NGSI API and On/OffStreetParking entities                                           |

#### 4.2.1.3 Functional Requirements & Provided functionalities

This baseline service targets the FR-20 of the Multi-Modal transportation application theme related since it estimates the availability of parking spots in the near futures.

| Multimodal Transportation Requirements |                                                                                      |                                                                                                              |
|----------------------------------------|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| FR code                                | FR description                                                                       | Comments                                                                                                     |
| FR-20                                  | <i>The application may be able to estimate when and where it will be possible to</i> | <i>The parking probability estimator service has been conceived specifically for addressing this purpose</i> |

|                  |                                                                                                                                                                                                                                                          |  |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
|                  | <i>find a free parking lot.</i>                                                                                                                                                                                                                          |  |
| <b>Validated</b> | <b>Outcomes</b>                                                                                                                                                                                                                                          |  |
|                  | <i>The service succesfully provides parking probability estimation attributes and enrich corresponding On/OffStreetParking city areas. However, the estimation algorithms are still under development so the provided estimations shall be improved.</i> |  |

## 4.2.2 Traffic Flow Estimator

Traffic Estimator service is a parallel development to that Parking Estimator described above that estimate the value of forthcoming events (in this case, those ones that have to do with traffic intensity). The purposes of this service are:

- To predict the traffic intensity within an area on the city, linked to existing TrafficFlowObserved entities.
- To generate an estimation of the traffic intensity in the short term (i.e. in the next 15, 30 and 45 minutes)

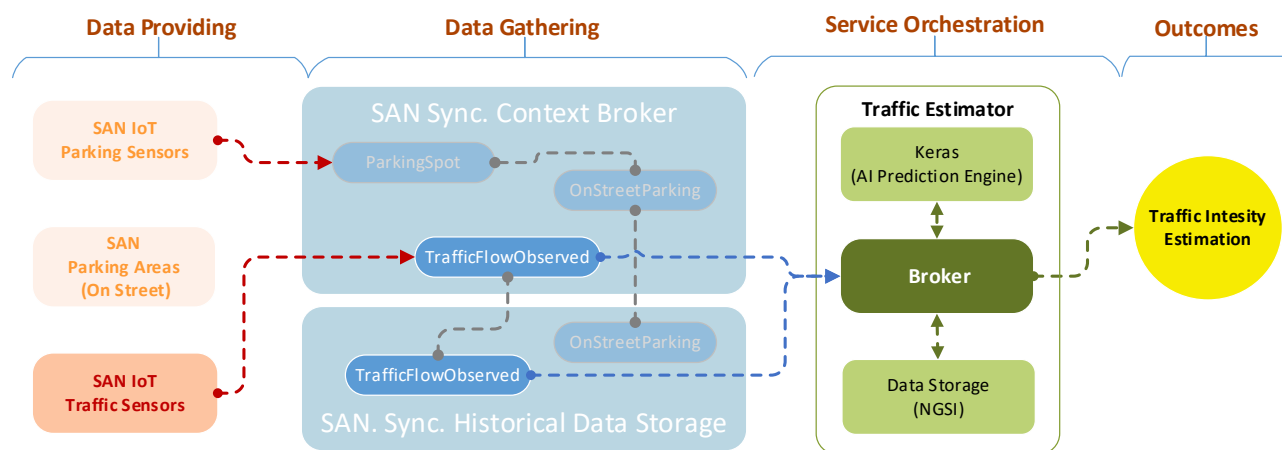


Figure 13. Traffic Estimator deployment

The Traffic Estimator deployment (Figure 13) has a similar composition to the Parking Estimator, but the Broker will manage TrafficFlowObserved entities and the Prediction Engine is configured to work with traffic intensity.

This AS allows two ways of deployment: using source code to compile the components; or set up the provided docker images. As this is a complex atomic service, the recommended installation will rely on the docker composition.

Paying attention to the guidelines, the hardware and software environments are configured for the testing environment

Config.js: links the broker with the local CB and Historical data providers

```

rzs: {
  brokers: [{
    rz: 'AS_Test',
    orion_host: 'http://193.144.201.58:10026', //Test environment Orion CB (Santander Parking data)
    historic_host: 'http://193.144.201.58:8080', //Test environment Historical data API (SAN Historical Data)
    auth: false,
    fiware_service: 'AtomicServices',
    area: [ // Santander location area (coordinates)
      [-4.198940, 43.332055],
      [-3.369473, 43.686628]
    ]
  }]
}

```

```

    },
  }
],
}

```

Config.ini: to configure the Keras and the AI environment

```

[API]
api_name = trafficestimator
endpoint = traffic-estimator ;it will use the internal docker network names
[ELK]
elastic_host = elasticsearch ;it will use the internal docker network names
elastic_port = 9200

```

Finally, the environment variables are set to properly link to the data sources:

```

environment:
  SUBSCRIPTIONS_ENDPOINT: http:193.144.201.58:10026 # Mandatory parameter
  ORION_HOST: orion # Default option for the INTERNAL NGSI storage (containers on same host)
  AI_PREDICTION_ENGINE_HOST: ai-prediction-engine # Default option (containers on same host)

```

Running the docker-compose.yml file deploys:

- Traffic estimator (port 3000) that supports the synchronicity-traffic-flow-estimator RESTful API [37].

Traffic estimator links with Test Orion CB. Traffic-flow-estimation API can list Traffic areas (TrafficFlowObserved entities) retrieved from the SAN Context Broker (311 TrafficFlowObserved entities available)

GET trafficestimator:3000/trafficareas

```

"summary": {
  "current_page": 1,
  "items_per_page": 311, // 311 TrafficFlowObserved entities defined for Santander City
  "items": 311,
  "pages": 1
},

```

As pointed in the description of the Traffic Estimator, the system (automatically) is now subscribed to the 311 available Traffic Flow entities and getting trained by analysing:

- Historical data related to the TrafficFlowObserved entities gathered through the provided Sync. Historical data API (this source ONLY will be exploited were available)
- Notifications about updates on TrafficFlowObserved entities received (and stored) by the component itself (creating its own historical dataset)

After one day of system training (it requires from a minimum set of previous 50 historical records or 50 received notifications to start calculating probabilities), we select one of the city areas (TrafficFlowObserved entities) and request for the estimations

GET trafficestimator:3000/trafficflows/urn:ngsi-Id:TrafficFlowObserved:santander:traffic:flow:1001/estimatedconditions

```

"TrafficFlowConditionEstimation": {
  "type": "StructuredValue",
  "value": [
    240,
    215.33210992813113,
    215.33210992813113,
    215.33210992813113,

```

```

    215.33210992813113
  ],
  "metadata": {
    "estimationTimes": {
      "type": "StructuredValue",
      "value": [
        "2019-04-17T09:58:56.206Z",
        "2019-04-17T10:13:56.206Z",
        "2019-04-17T10:28:56.206Z",
        "2019-04-17T10:43:56.206Z",
        "2019-04-17T11:13:56.206Z"
      ]
    }
  }
}

```

So, considering the request has been done on 17/04/2019 at 09:58 UTC, estimated traffic flow in the area covered by the sensor 1001 would be:

| Estimated total number of vehicles within 5 minutes time window (for Santander deployment) | at                       |
|--------------------------------------------------------------------------------------------|--------------------------|
| 240                                                                                        | 2019-04-17T09:58:56.206Z |
| 215.33210992813113                                                                         | 2019-04-17T10:13:56.206Z |
| 215.33210992813113                                                                         | 2019-04-17T10:28:56.206Z |
| 215.33210992813113                                                                         | 2019-04-17T10:43:56.206Z |
| 215.33210992813113                                                                         | 2019-04-17T11:13:56.206Z |

The traffic flow estimated value corresponds to the number of vehicles counted within a given time window. Traffic loops in Santander are configured to send new data each 5 minutes, so the traffic intensity attribute provided by SAN entities refers to the number of vehicles counted during last 5 minutes.

As happens with the Parking Prob. Estimator service, the AI modules of this Traffic Flow estimator AS are also under development, so the results offered so far are not still accurate.

#### 4.2.2.1 Documentation

| Atomic Service Ref: <a href="https://gitlab.com/synchronicity-iot/parking-estimator">https://gitlab.com/synchronicity-iot/parking-estimator</a> |                              |                                                                                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| Key Target                                                                                                                                      | Dimensions validated         | Comments                                                                                                                                       |
| Identification                                                                                                                                  | Commercial info              | Licences and versions of each required component are provided                                                                                  |
|                                                                                                                                                 | Technical Requirements       | Minimum hardware and software requirements are detailed                                                                                        |
|                                                                                                                                                 | Potential end users/Services | To be used by upper services exploiting traffic flow estimations.                                                                              |
|                                                                                                                                                 | AS qualitative description   | Description of the provided functionalities (traffic flow estimations for given time slots referred to existing TrafficFlowObserved entities). |

|                        |                                                            |                                                                                                                                                                                                                                                                  |
|------------------------|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                        |                                                            | Description of the protocols and technologies used to implement the service.                                                                                                                                                                                     |
| Deployment description | Set up & Running                                           | Source code (on main components), Docker deployment and setup documentation are provided.<br><br>Easy to follow configuration instructions for a basic deployment.                                                                                               |
|                        | Usage                                                      | RESTful API to query/retrieve traffic flow estimations, historical data and parking probabilities.                                                                                                                                                               |
| Results                | End-User Outcomes                                          | RESTful API providing JSON outcomes for Query/retrieve Traffic Estimations <ul style="list-style-type: none"> <li>- List Traffic Flow (covered) entities</li> <li>- Detail each Traffic Flow entity</li> <li>- Get Traffic Flow Intensity estimations</li> </ul> |
|                        | Reusable Outcomes (Integration/orchestration capabilities) | Consumes NGSI TrafficFlowObserved<br><br>Produces TrafficFlowObserved entities including new "TrafficFlowConditionEstimation" attribute.                                                                                                                         |

#### 4.2.2.2 Deployment & Integration with Synchronicity Framework

| Docker Deployment: <a href="https://hub.docker.com/r/synchronicityiot/parking-estimator-api">https://hub.docker.com/r/synchronicityiot/parking-estimator-api</a> |                            |                                                                                                                    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|--------------------------------------------------------------------------------------------------------------------|
| Key Target                                                                                                                                                       | Dimensions validated       | Comments                                                                                                           |
| Deployment Procedure                                                                                                                                             | Set up & Running execution | Simple Docker deployment according documentation.<br><br>No extra skills are required                              |
| Supported MIMs                                                                                                                                                   | NGSI                       | NGSIV2 compliant (info sources required to be stored in a Sync. Compliant CB)                                      |
|                                                                                                                                                                  | OAuth                      | Supports OAuth 2.0 tokens (to interact with Sync. Frameworks)                                                      |
|                                                                                                                                                                  | Historical Data            | Historical data related to TrafficFlowObserved exploited (where available)                                         |
|                                                                                                                                                                  | Data models                | Requires from TrafficFlowObserved NGSI entities                                                                    |
| Features Supported                                                                                                                                               | Data access management     | Full support of FIWARE Services and FIWARE ServicePaths.<br>Generates compliant Sync. data models (and attributes) |
|                                                                                                                                                                  | Orchestration              | Possible orchestration through NGSI API and TrafficFlowObserved entities                                           |

#### 4.2.2.3 Functional Requirements & Provided functionalities

This atomic service targets the FR-26 of the Multi-Modal transportation application theme related since it enriches the information provided by the TrafficFlowObserved entities.

| Multimodal Transportation Requirements |                |          |
|----------------------------------------|----------------|----------|
| FR code                                | FR description | Comments |

|                  |                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FR-26</b>     | <i>Traffic sensors must at least provide their GPS location and the traffic flow intensity</i>                                                                                                                                    | <i>Most of the traffic sensors used by RZs are providing only the traffic intensity of the exact location statically mapped by the respective RZs' administrator. The estimation of the traffic flow intensity from a point to another is, instead, addressed by the Traffic flow estimator atomic service</i> |
| <b>Validated</b> | <b>Outcomes</b>                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                |
|                  | <i>The service succesfully provides estimation attribute and enrich corresponding TrafficFlowObserved entities. However, the estimation algorithms are still under development so the provided estimations shall be improved.</i> |                                                                                                                                                                                                                                                                                                                |

### 4.3 Scenario 3 (City data Visualization) Validation

This scenario is composed by two services that, using different approaches, both provides a view of the Context available data. The first one, the dashboard, consist in an easy way to overview what a SynchroniCity instance is exposing and have a quick access to the last values of each entity. The second one, the Metrics visualizer, links the Grafana [17] dashboard with the Synchronicity context data, providing the end-user with a set of powerful tools to analyse what's happening within the city.

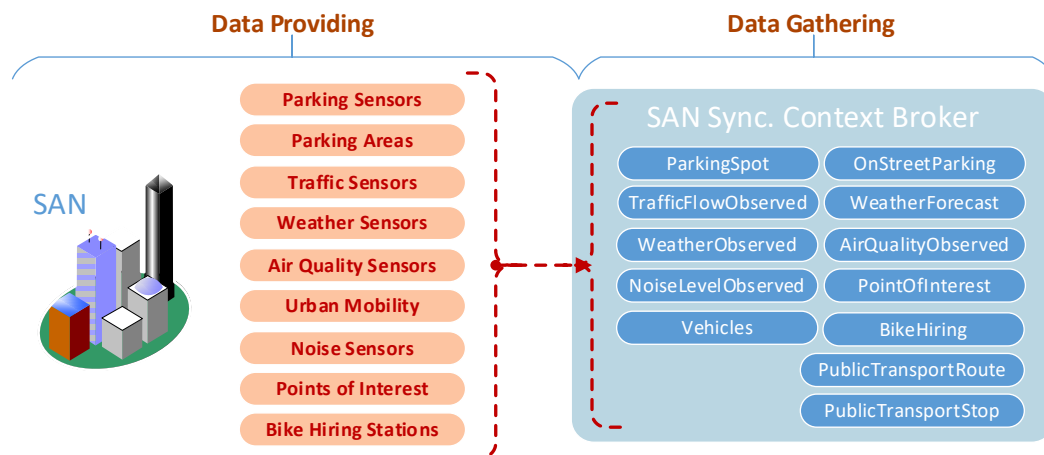


Figure 14. Visualization scenario layout for validation

The scenario composition (Figure 14) for this validation is simple: the Santander SynchroniCity context broker is exposed to these two services through a special user that has reading access to all SynchroniCity compliant entities.

#### 4.3.1 Smartcities Dashboard

Smart Cities Dashboard consists on a web service interface that provides end-users with a simple and user-friendly dashboard representing the SynchroniCity compliant Context Entities available on a given Synchronicity framework (represented by its Context Broker). This interface shows location and real-time entity information. It uses Google Maps API to create the maps supporting the graphical presentation of the Context Information.

The atomic service is provided in a docker installation docker-compose file. Docker images: [synchronicityiot/dashboard-backend] and [synchronicityiot/dashboard-frontend] are distributed through the SynchroniCity docker hub [38] and maintained by the Service developer.

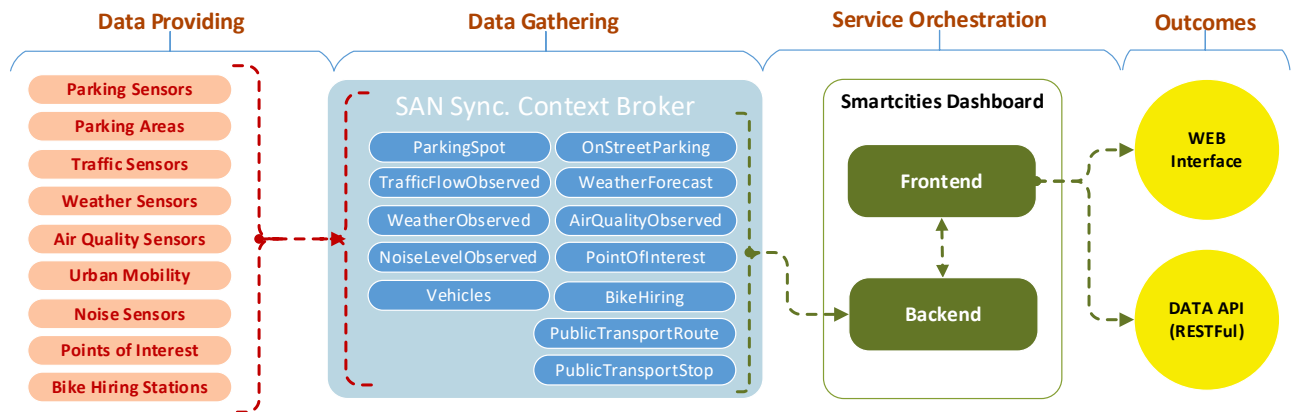


Figure 15. Smartcities Dashboard composition

The proposed configuration of this service (Figure 15) uses a centralised instance of Orion Context Broker to capture data from different SynchroniCity instances, as a federation of RZs. To simplify the functionality (and deployment) validations, a single context broker, replicating the Santander set of Context Entities is used, linking this one directly to the AS.

Configuration of both services (shown in Figure 15) is done by modifying this docker-compose (environment section) file according instructions:

Dashboard-backend:

```
environment:
  - ORION_URL=http://193.144.201.58:10026 #SAN Test Orion
  - ORION_SERVICE= AtomicServices
  - ORION_SERVICEPATH=
  - DJANGO_DEBUG=True
  - DJANGO_ALLOWED_HOSTS="*"
```

Dashboard-frontend:

```
ports:
  - "8004:80" # Port to deploy the RESTful API and the WEB interface
environment:
  - SITE_URL=193.144.201.50 # Local server supporting the AS
  - EXTERNAL_URL=193.144.201.50:8004 # Access to AS
  - GOOGLE_MAPS_API=FlzoPyVxdrygquxk1-gpc6Xlti7oy8cSWStmGxc # Provided by HOPU. Each deployment MUST
    obtain its own Google Maps Key
  - SYNCHRONICITY_BACKEND=synchronicity-backend:8000 # link with the backend
```

Once deployed, this service opens the configured port 8004 to offer:

- [http://Test\\_Server:8004/](http://Test_Server:8004/) the web interface showing all the captured entities in the pointed Context Broker. A simple menu let the user mark only those interesting entities. Figure 16 shows data referred to a selected entity (TrafficFlowObserved) among those provided by the Santander test environment.
- [http://Test\\_Server:8004/api/v1](http://Test_Server:8004/api/v1) the RestFul API that allows the end-user/service to query/retrieve RESTful resources. At the moment of writing this validation results, this API is still being developed and documented.
  - o Last value: GET /api/v1/real-time-data → retrieve the devices stored as entities in Orion Context Broker
  - o Historical data: GET /api/v1/historical-data/<device\_id> → retrieve advanced information from STH Comet related to device with device\_id

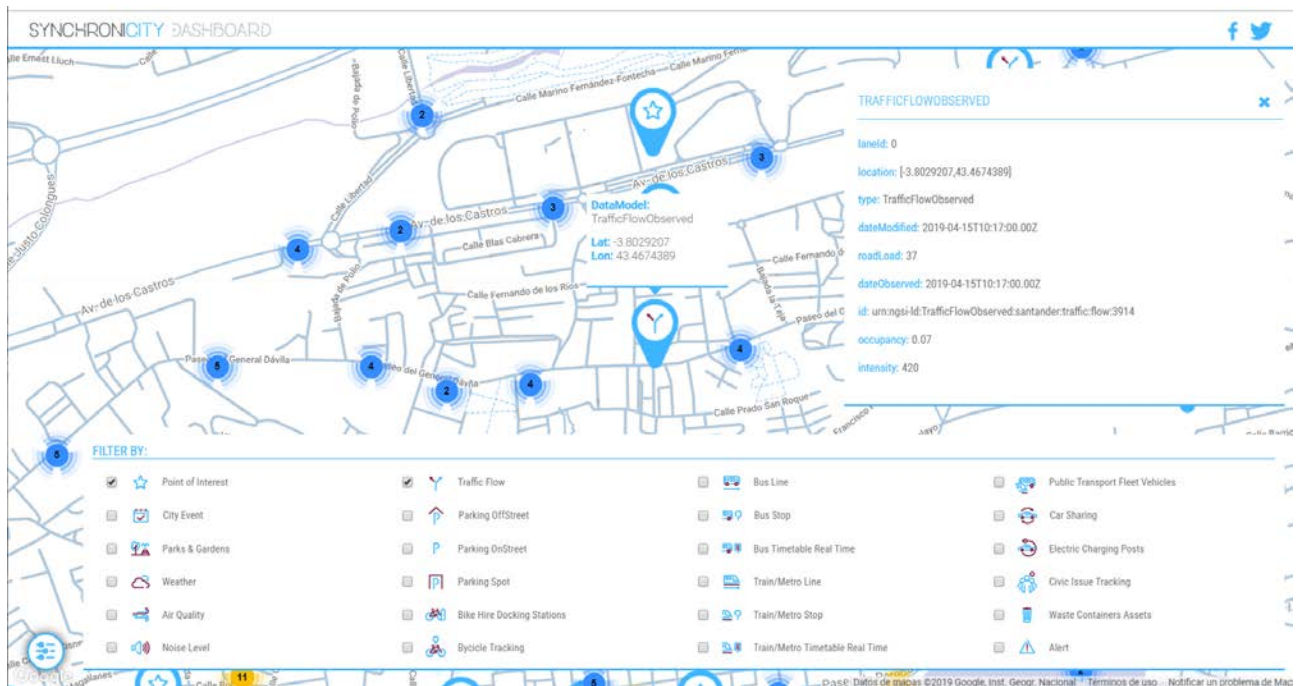


Figure 16. Capture of the Smartcities Dashboard deployed in the test environment

All the captures Context Entities are shown in real time taking advantage of the Orion's publish/subscribe mechanism. In this way, updated values for sensors and location are assured. Each point representing an entity can be clicked deploying all the captured information related to it.

#### 4.3.1.1 Documentation

| Atomic Service Ref: <a href="https://gitlab.com/synchronicity-iot/SmartCitiesDashboardService">https://gitlab.com/synchronicity-iot/SmartCitiesDashboardService</a> |                              |                                                                                                                              |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| Key Target                                                                                                                                                          | Dimensions validated         | Comments                                                                                                                     |
| Identification                                                                                                                                                      | Commercial info              | Licences and versions of each required component are provided. Apache License 2.0. Supported by AS Provider.                 |
|                                                                                                                                                                     | Technical Requirements       | No special hardware nor Software requirements pointed.                                                                       |
|                                                                                                                                                                     | Potential end users/Services | WEB interface oriented to end-users<br>APIs for upper services exploitation                                                  |
|                                                                                                                                                                     | AS qualitative description   | Full Description of the provided functionalities related to the Web Interface.<br>Sort description of the APIs features.     |
| Deployment description                                                                                                                                              | Set up & Running             | Docker deployment and setup documentation are provided.<br>Easy to follow configuration instructions for a basic deployment. |
|                                                                                                                                                                     | Usage                        | Web interface easy and intuitive<br>APIs not properly documented                                                             |
| Results                                                                                                                                                             | End-User Outcomes            | Main Outcome (Web Dashboard) detailed:<br>- Geolocation of entities and Info related                                         |

|  |                                                               |                                                                                                                                                                   |
|--|---------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  |                                                               | API providing: <ul style="list-style-type: none"> <li>- Real-time info of the entities within a map</li> <li>- Historical Data of an identified entity</li> </ul> |
|  | Reusable Outcomes<br>(Integration/orchestration capabilities) | Consumes NGSI entities<br><br>Retrieves (through API) Context Entities compliant with Sync. data models                                                           |

#### 4.3.1.2 Deployment & Integration with Synchronicity Framework

| Docker Deployment: <a href="https://gitlab.com/synchronicity-iot/SmartCitiesDashboardService#service-deployment">https://gitlab.com/synchronicity-iot/SmartCitiesDashboardService#service-deployment</a> |                            |                                                                                                                                                   |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Key Target                                                                                                                                                                                               | Dimensions validated       | Comments                                                                                                                                          |
| Deployment Procedure                                                                                                                                                                                     | Set up & Running execution | Simple Docker deployment according documentation (no further skills required).<br><br>Easy configuration through docker-compose file provided     |
| Supported MIMs                                                                                                                                                                                           | NGSI                       | NGSIV2 compliant (info sources required to be stored in a Sync. Compliant CB)                                                                     |
|                                                                                                                                                                                                          | OAuth                      | Does not support OAuth tokens (to request Sync. Frameworks)                                                                                       |
|                                                                                                                                                                                                          | Historical Data            | No Sync. historical data required nor supported                                                                                                   |
|                                                                                                                                                                                                          | Data models                | Works with all Sync. data models                                                                                                                  |
| Features Supported                                                                                                                                                                                       | Data access management     | Full support of FIWARE Services and FIWARE Service Paths.<br><br>Able to simultaneously retrieve and show entities from different Context Brokers |
|                                                                                                                                                                                                          | Orchestration              | Able to group and offer (RESTful API) entities from different Sync. Frameworks (Using Sync. compliant data models)                                |

#### 4.3.1.3 Functional Requirements & Provided functionalities

| Human-Centric Traffic Management Requirements |                                                                                                                                                                                 |                                                                                           |
|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| FR code                                       | FR description                                                                                                                                                                  | Comments                                                                                  |
| FR-1                                          | System must process data from multiple sources/sensors in real-time                                                                                                             | This visualization dashboard is capable of showing all the data available given as input. |
| Validated                                     | Outcomes                                                                                                                                                                        |                                                                                           |
|                                               | Backend successfully integrates with Sync. CB and reads all Sync. Compliant entities.<br><br>Frontend (WEB-UI & API) successfully presents all datasets gathered by the Backend |                                                                                           |
| FR-6                                          | The system must provide flexibility in terms of viewing, or having access to, certain datasets (e.g. through dashboard widgets)                                                 | The dashboard is versatile and can be customized                                          |
| FR-7                                          | The system may provide a map as the main                                                                                                                                        | This dashboard service is capable of                                                      |

|                  |                                                                                                                                                                                                                                                                                                             |                                                            |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
|                  | <i>way of interaction to superimpose datasets</i>                                                                                                                                                                                                                                                           | <i>superimpose data.</i>                                   |
| <b>Validated</b> | <b>Outcomes</b>                                                                                                                                                                                                                                                                                             |                                                            |
|                  | <i>WEB-UI uses Google Maps API to draw a map and superimpose entities location on it. It provides the end user with a tool to select which datasets will be presented. It also superimpose the data related to a selected entity.</i>                                                                       |                                                            |
| <b>FR-8</b>      | <i>The system should provide different views on the superimposed datasets</i>                                                                                                                                                                                                                               | <i>This dashboard service offers many different views.</i> |
| <b>Validated</b> | <b>Outcomes</b>                                                                                                                                                                                                                                                                                             |                                                            |
|                  | <i>WEB-UI works only works superimposing the entities read on top of the corresponding map. It allows the user to move all along the map (covering the whole world) looking for deployed (and registered) entities. It also offers the possibility of reading the information linked to a given sensor.</i> |                                                            |

| Multimodal Transportation Requirements |                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                     |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| FR code                                | FR description                                                                                                                                                                                                                                                                                                                                                                                                    | Comments                                                                                                            |
| <b>FR-12</b>                           | <i>The user should be able to view on a map the points of interest located in an area near to the route.</i>                                                                                                                                                                                                                                                                                                      | <i>POIs can be visualized in the map</i>                                                                            |
| <b>FR-16</b>                           | <i>The application must provide users with a georeferenced search of parking areas.</i>                                                                                                                                                                                                                                                                                                                           | <i>Changing the focus on the map affects the geographic scope of the query request.</i>                             |
| <b>FR-19</b>                           | <i>The application should be able to indicate in real-time where free parking lots Users, Parking are available and at which cost/minute.</i>                                                                                                                                                                                                                                                                     | <i>The data available, such as parking spots status and parking costs, can be shown on the map</i>                  |
| <b>Validated</b>                       | <b>Outcomes</b>                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                     |
|                                        | <i>WEB-UI presents, geolocalized in a map view, all entities (Sync. Compliant) available in the linked CB:</i> <ul style="list-style-type: none"> <li>- <i>User can select &amp; view only Poi entities</i></li> <li>- <i>User can move along the provided map seeing only ParkingSpots and OnStreetParking entities. OffStreetParking couldn't be validated since SAN does not have such entities</i></li> </ul> |                                                                                                                     |
| <b>FR-18</b>                           | <i>The application must be able to indicate which transport possibilities are currently available near the selected parking area (e.g. estimated arrival times for nearby bus lines).</i>                                                                                                                                                                                                                         | <i>The data available, such as public transportation stops and estimated arrival times, can be shown on the map</i> |
| <b>Validated</b>                       | <b>Outcomes</b>                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                     |
|                                        | <i>WEB-UI presents, geolocalized in a map view, all entities (Sync. Compliant) available in the linked CB:</i> <ul style="list-style-type: none"> <li>- <i>New Urban Mobility PublicTransportationStop and ArrivalEstimationTime entities have not been integrated yet (in progress)</i></li> </ul>                                                                                                               |                                                                                                                     |

| Community Policy Suite Requirements |                |          |
|-------------------------------------|----------------|----------|
| FR code                             | FR description | Comments |

|                  |                                                                                                                                         |                                         |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| <b>FR-4</b>      | <i>UI requirements; system should be intuitive and easy to use</i>                                                                      | <i>Accepted as input by the service</i> |
| <b>Validated</b> | <b>Outcomes</b>                                                                                                                         |                                         |
|                  | <i>WEB-UI presented as a simple and intuitive User Interface. No special skills nor technical knowledge is required to work with it</i> |                                         |

### 4.3.2 Metrics Visualizer (Grafana)

This AS developed specific plugins to link SynchroniCity Context Broker providers with the Grafana [17] powerful analytics open platform, so SynchroniCity compliant context entities can be part of the different dashboards designed for multiple applications.

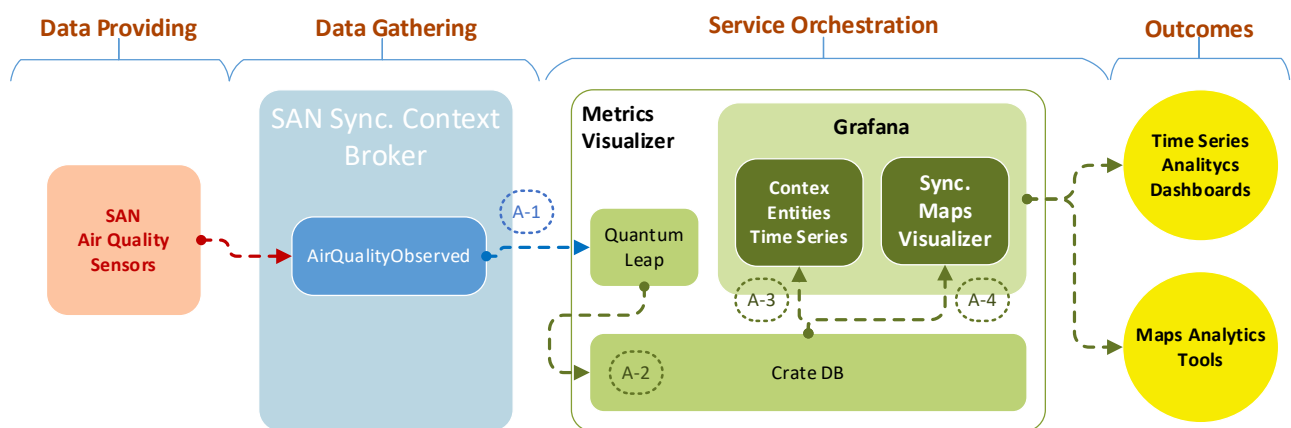


Figure 17. Metrics Visualizer service composition

Potentially, the Metrics Visualizer service can manage any SynchroniCity compliant entity, but it is oriented to entities providing quantitative parameters measurements related to time, such CO, NO2, speed, temperature, etc. for the time series analysis and/or location attributes for the map's analytics tools. For the sake of simplicity, the validation scenario (Figure 17) will link the metrics visualizer service with only one Santander dataset: the AirQualityObserved entities, which covers both requirements to check the service features.

Metrics visualizer composition relies on three main components:

- FIWARE Quantum Leap [39] connector that captures NGSI notifications to feed Time series databases.
- Crate DB [40], a distributed SQL database built on top of a NoSQL foundation to store the captured NGSI context information and to create historical datasets.
- Grafana [17] analytics platform to build the components of the designed dashboards. The testing environment relies on a set of Grafana plugins: environment is complemented with two SynchroniCity developed plugins:
  - o Smart spot worldmap panel developed within SynchroniCity project, provides the complement to map SynchroniCity context entities
  - o Crate-Datasource & smart-spot supporting the time series representations for Context Entities

A docker-compose file is provided to easily deploy and configure these three building blocks. Configuration options allow the user to select the ports these components will be listening on. In our testing environment:

- Port 8668 for Quantum Leap notifications

- Ports 4200 and 4300 for CrateDB data access and management
- Port 3000 to access Grafana platform

Grafana composition automatically downloads the required plugins to build the platform.

Once deployed, the steps to execute the scenario are:

- (A-1) Create the subscription on the Orion CB to link the entities in the SynchroniCity framework with the Metrics Visualizer QuantumLeap

POST <http://193.144.201.58:10026/v2/subscriptions>

```
{
  "description": "GRAFANA test AIRQ",
  "subject": {
    "entities": [
      {
        "idPattern": "urn:ngsi-Id:AirQualityObserved:santander:.*", // Entities set to be captured
        "type": "AirQualityObserved"
      }
    ],
    "condition": {
      "attrs": [ //Attr. that triggers the notifications
        "dateObserved"
      ]
    }
  },
  "notification": {
    "http": { // QuantumLeap notifications endpoint
      "url": http://193.144.201.50:8668/v2/notify
    },
    "attrs": [ //parameters (attributes) to be monitorized
      "relativeHumidity",
      "temperature",
      "location"
    ]
  }
}
```

- (A-2) AirQualityObserved data is notified to the QuantumLeap and forwarded to the Crate DB to be stored (Figure 18).

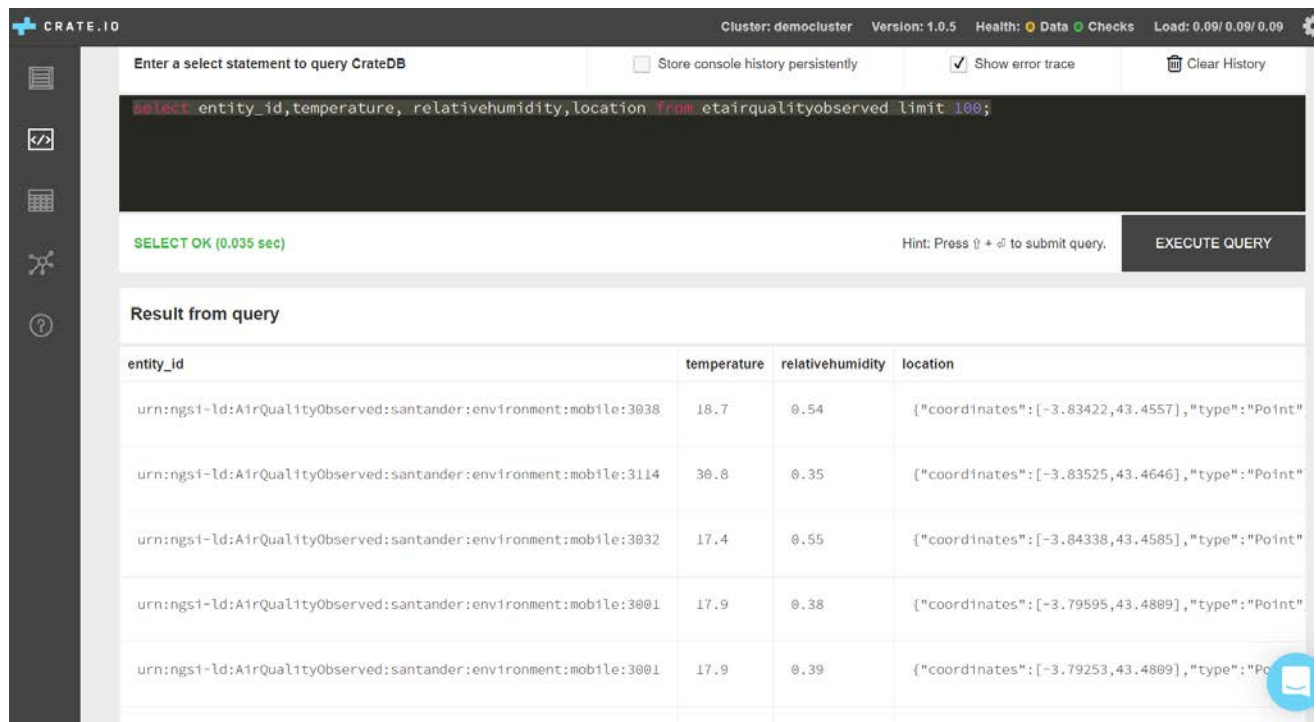


Figure 18. Data series stored by Quantum Leap on Crate DB component

- (A-3) Creating and configuring the Grafana queries according the documentation a test dashboard is created representing the captured relative humidity and temperature reported by SAN Air Quality Sensors (Figure 19).

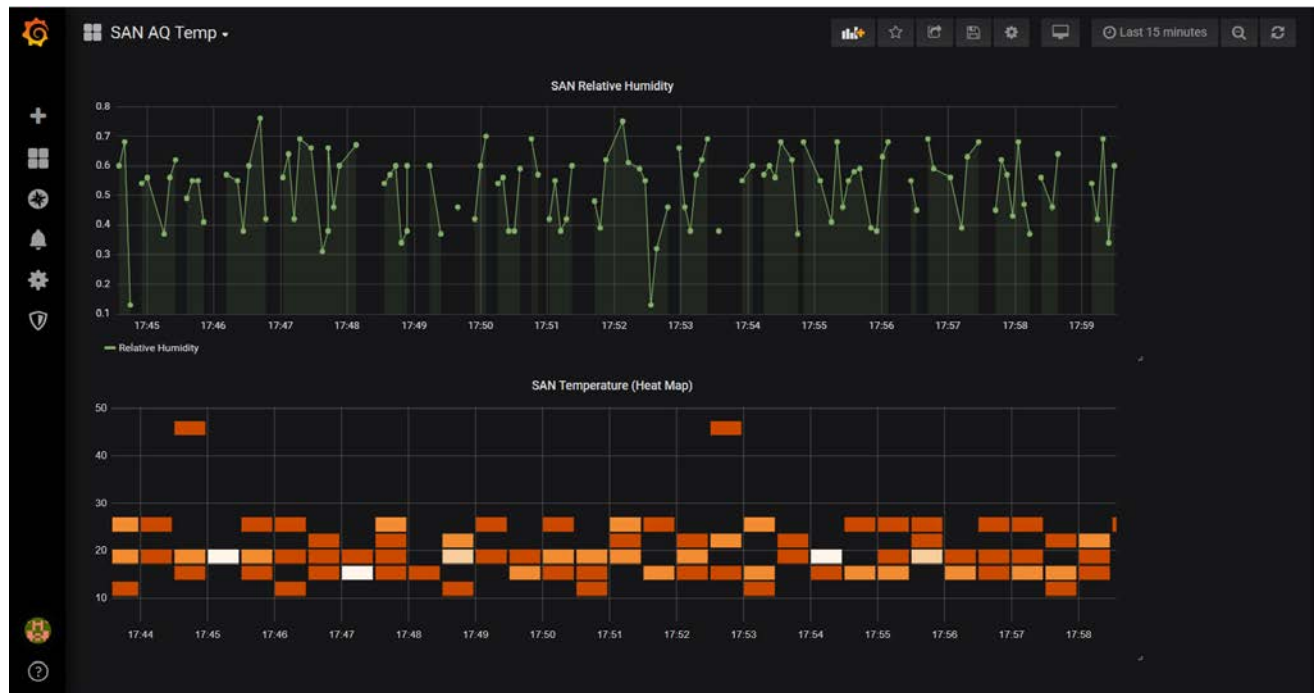


Figure 19. Graphics (time series) created by the Grafana dashboard component

- (A-4) at the time of writing this evaluation, the maps visualization plugin is being finalised and the documentation is not still available, so no validation process could be executed on this feature.

#### 4.3.2.1 Documentation

| Atomic Service Ref: <a href="https://gitlab.com/synchronicity-iot/GrafanaAtomicService">https://gitlab.com/synchronicity-iot/GrafanaAtomicService</a> |                                                            |                                                                                                                                                                                                                                 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Key Target                                                                                                                                            | Dimensions validated                                       | Comments                                                                                                                                                                                                                        |
| Identification                                                                                                                                        | Commercial info                                            | Versions of different components are not provided.<br>Licensing model is not specified                                                                                                                                          |
|                                                                                                                                                       | Technical Requirements                                     | No special hardware nor Software requirements pointed.                                                                                                                                                                          |
|                                                                                                                                                       | Potential end users/Services                               | Data base experienced end-users<br>Dashboard managers<br>Grafana end-users                                                                                                                                                      |
|                                                                                                                                                       | AS qualitative description                                 | Description of the provided functionalities provided.<br>Graphical support also provided                                                                                                                                        |
| Deployment description                                                                                                                                | Set up & Running                                           | Docker deployment is provided. <u>Must be improved to show latest modifications</u><br>Easy to follow configuration instructions for a basic deployment (for Time Series tools)<br><u>No description provided for Maps tool</u> |
|                                                                                                                                                       | Usage                                                      | Familiar for Grafana users and DB managers<br>Time Series usage described<br><u>Maps tool not described</u>                                                                                                                     |
| Results                                                                                                                                               | End-User Outcomes                                          | Graphical Time Series of captured Context Information.<br>Multiple possibilities of visualization<br><u>No description of Maps outcomes</u>                                                                                     |
|                                                                                                                                                       | Reusable Outcomes (Integration/orchestration capabilities) | Consumes NGSI entities<br>End-user visualization tools. No initially reusable by upper services                                                                                                                                 |

#### 4.3.2.2 Deployment & Integration with Synchronicity Framework

| Docker Deployment: <a href="https://gitlab.com/synchronicity-iot/GrafanaAtomicService/blob/master/docker-compose/docker-compose.yml">https://gitlab.com/synchronicity-iot/GrafanaAtomicService/blob/master/docker-compose/docker-compose.yml</a> |                            |                                                                                                                                                                       |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Key Target                                                                                                                                                                                                                                       | Dimensions validated       | Comments                                                                                                                                                              |
| Deployment Procedure                                                                                                                                                                                                                             | Set up & Running execution | Simple Docker deployment according documentation (no further skills required).<br>Easy configuration through docker-compose file provided<br><u>Missing MAPS tool</u> |
| Supported MIMs                                                                                                                                                                                                                                   | NGSI                       | NGSIV2 compliant (info sources required to be stored in a Sync. Compliant CB)                                                                                         |

|                    |                        |                                                                                                                                              |
|--------------------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
|                    | OAuth                  | Does not support OAuth tokens (to request Sync. Frameworks)                                                                                  |
|                    | Historical Data        | No Sync. historical data required nor supported                                                                                              |
|                    | Data models            | Works with all Sync. data models                                                                                                             |
| Features Supported | Data access management | Full support of FIWARE Services and FIWARE ServicePaths.<br>Able to simultaneously retrieve and show entities from different Context Brokers |
|                    | Orchestration          | Able to show data from different SynchroniCity Frameworks                                                                                    |

#### 4.3.2.3 Functional Requirements & Provided functionalities

| Human-Centric Traffic Management Requirements |                                                                                                                                                                                                                                                                                                                |                                                                                           |
|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| FR code                                       | FR description                                                                                                                                                                                                                                                                                                 | Comments                                                                                  |
| FR-1                                          | System must process data from multiple sources/sensors in real-time                                                                                                                                                                                                                                            | This visualization dashboard is capable of showing all the data available given as input. |
| FR-6                                          | The system must provide flexibility in terms of viewing, or having access to, certain datasets (e.g. through dashboard widgets)                                                                                                                                                                                | The dashboard is versatile and can be customized                                          |
| Validated                                     | Outcomes                                                                                                                                                                                                                                                                                                       |                                                                                           |
|                                               | <p>QuantumLeap component can receive notifications from different entities in different CBs (configuring different sources) with near-real time data</p> <p>CrateDB can store in different tables data from different sources</p> <p>Grafana can create multiple visualizations combining this datasources</p> |                                                                                           |
| FR-10                                         | The system may provide a map as the main way of interaction to superimpose datasets                                                                                                                                                                                                                            | This dashboard service is capable of superimpose data.                                    |
| FR-11                                         | The system may allow for annotation specified time windows                                                                                                                                                                                                                                                     | The dashboard allows user to customize desired annotations                                |
| Validated                                     | Outcomes                                                                                                                                                                                                                                                                                                       |                                                                                           |
|                                               | Grafana visualization tools for time series allows the customization of the dashboards                                                                                                                                                                                                                         |                                                                                           |

| Multimodal Transportation Requirements |                                                                                                                                           |                                                                                                                                                                                                          |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FR code                                | FR description                                                                                                                            | Comments                                                                                                                                                                                                 |
| FR-2                                   | Editable dashboards; based on city / team requirements user should be able to amend and lock (based on hierarchy) the dashboard "widgets" | The dashboard provides an authentication system based on organizations where users have different roles which will give them different levels of privileges limiting the ability to modify the dashboard |
| FR-4                                   | UI requirements; system should be intuitive and easy to use                                                                               | The system provides three levels of permissions (admin, edit and view) where users who have an edit rank or higher can                                                                                   |

|                  |                                                                                                                                                                    |                                                                                                                                                                              |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  |                                                                                                                                                                    | <i>generate boards in a simple way and adapt them to the new needs in an agile way.</i>                                                                                      |
| <b>Validated</b> | <b>Outcomes</b>                                                                                                                                                    |                                                                                                                                                                              |
|                  | <i>Grafana has a user/password access control that allow also role configuration. Diferent defined dashboards and so, data access are protected.</i>               |                                                                                                                                                                              |
| <b>FR-3</b>      | <i>Data management; system should be able to take custody of the data and edit for visualization</i>                                                               | <i>The system supports to export the time series data using json or csv format in order to facilitate users to manipulate or apply advanced techniques of data analysis.</i> |
| <b>Validated</b> | <b>Outcomes</b>                                                                                                                                                    |                                                                                                                                                                              |
|                  | <i>Time series (and captured data) is stored in a CrateDB, protected by user/password access control. This component allows data import &amp; export as a DBMS</i> |                                                                                                                                                                              |

## 5 Results and Conclusions

As it was done in D4.2 [3], this section summarises the results of the second round of technical validations, detailed in sections 2, 3 and 4 of this document, and provides a quick look at the current SynchroniCity framework deployment status plus the services developed on top of it.

### 5.1 MIMs validation update (view of overall results)

The first overview (Table 25) shows the components and APIs available in each Reference Zone as a result of the outcomes of section 3.

| Synch. MIMs               | Reference Zones |                |                  |           |           |          |             |            |
|---------------------------|-----------------|----------------|------------------|-----------|-----------|----------|-------------|------------|
|                           | Porto           | Santander      | Antwerp          | Carouge   | Eindhoven | Helsinki | Manchester  | Milan      |
| NSGI Interface            | Orion CB        | Orion CB       | Orion CB         | Orion CB  | Orion CB  | Orion CB | Orion CB    | Orion CB   |
| Security (OAuth2)         | IdM KR          | IdM KR         | WSO2 IdM         | IdM KR    |           |          |             | WSO2 IdM   |
| Historical Data Interface | Quantum Leap    | Cygnus         | Comet            | Comet STH |           |          | Proprietary | Cygnus     |
| Data Market Place         |                 | SC MarketPlace | Kong MarketPlace |           |           |          |             | WSO2 based |

|  |                                            |
|--|--------------------------------------------|
|  | Deployed and validated component           |
|  | Not Available/Not Deployed (Not validated) |

Table 25. Validated SynchroniCity components per RZ, as of April the 17th, 2019.

In this second phase, all the core project cities have deployed their own context broker, supporting the minimum requirement to deploy pilots on top: the NSGI compliant API. Many of them have also completed their initial deployment plans, including historical data access and authentication/identification mechanisms. This allows SynchroniCity to evaluate the replicability of the pilots addressed in the Open Call.

Table 26 reflects the available datasets together with their compliance with SynchroniCity Data Models per city. The table shows the environmental data (AirQualityObserved, WaetherObserved and WeatherForecast) as the most extended info among cities. Milan, Santander and Porto are the most active ones in terms of published resources. In general terms, cities are progressing well, highlighting the adaptation and refinement of their data to get full SynchroniCity compliance, but some efforts are still required to increase the smart city resources. Table 26 has its live version, updated twice a day by the SynchroniCity MIM Validator here [18].

| Dataset            | Reference Zones |           |         |         |           |          |            |       | Entities Count |
|--------------------|-----------------|-----------|---------|---------|-----------|----------|------------|-------|----------------|
|                    | Porto           | Santander | Antwerp | Carouge | Eindhoven | Helsinki | Manchester | Milan |                |
| AirQualityObserved |                 |           |         |         |           |          |            |       | 1048           |
| ArrivalEstimation  |                 |           |         |         |           |          |            |       | 33             |
| Beach              |                 |           |         |         |           |          |            |       | 13             |

|                        |  |  |  |  |  |  |  |  |       |
|------------------------|--|--|--|--|--|--|--|--|-------|
| BikeHireDockingStation |  |  |  |  |  |  |  |  | 1131  |
| CarSharingStation      |  |  |  |  |  |  |  |  | 113   |
| CrowdFlowObserved      |  |  |  |  |  |  |  |  | 40    |
| Device                 |  |  |  |  |  |  |  |  | 516   |
| EVChargingStation      |  |  |  |  |  |  |  |  | 63    |
| Garden                 |  |  |  |  |  |  |  |  | 1064  |
| GreenspaceRecord       |  |  |  |  |  |  |  |  | 18    |
| GtfsAgency             |  |  |  |  |  |  |  |  | 5     |
| GtfsCalendarDateRule   |  |  |  |  |  |  |  |  | 6     |
| GtfsCalendarRule       |  |  |  |  |  |  |  |  | 1429  |
| GtfsFrequency          |  |  |  |  |  |  |  |  | 1414  |
| GtfsRoute              |  |  |  |  |  |  |  |  | 655   |
| GtfsService            |  |  |  |  |  |  |  |  | 1429  |
| GtfsShape              |  |  |  |  |  |  |  |  | 582   |
| GtfsStation            |  |  |  |  |  |  |  |  | 6886  |
| GtfsStop               |  |  |  |  |  |  |  |  | 13086 |
| GtfsStopTime           |  |  |  |  |  |  |  |  | 71356 |
| GtfsTransitFeedFile    |  |  |  |  |  |  |  |  | 7     |
| GtfsTrip               |  |  |  |  |  |  |  |  | 1414  |
| Museum                 |  |  |  |  |  |  |  |  | 10    |
| NoiseLevelObserved     |  |  |  |  |  |  |  |  | 44    |
| OffStreetParking       |  |  |  |  |  |  |  |  | 180   |
| OnStreetParking        |  |  |  |  |  |  |  |  | 503   |
| ParkingSpot            |  |  |  |  |  |  |  |  | 324   |
| PointOfInterest        |  |  |  |  |  |  |  |  | 79275 |
| PublicTransportRoute   |  |  |  |  |  |  |  |  | 32    |
| PublicTransportStop    |  |  |  |  |  |  |  |  | 449   |
| RestrictedTrafficArea  |  |  |  |  |  |  |  |  | 177   |
| Road                   |  |  |  |  |  |  |  |  | 4331  |
| Store                  |  |  |  |  |  |  |  |  | 2170  |
| TrafficFlowObserved    |  |  |  |  |  |  |  |  | 692   |
| Vehicle                |  |  |  |  |  |  |  |  | 358   |
| WasteContainer         |  |  |  |  |  |  |  |  | 159   |

|                 |  |  |  |  |  |  |  |  |      |
|-----------------|--|--|--|--|--|--|--|--|------|
| WeatherForecast |  |  |  |  |  |  |  |  | 137  |
| WeatherObserved |  |  |  |  |  |  |  |  | 4497 |

|  |                                       |
|--|---------------------------------------|
|  | Full SynchroniCity Compliant datasets |
|  | Dataset Partially compliant           |

Table 26. Available SynchroniCity compliant datasets as of April the 17th, 2019. It is also shown the overall number of data assets according to its data model.

Table 26 has been composed using the outcomes shown in section 3. Some of the tables referred to the cities (Table 10, Table 12, Table 15, Table 20 and Table 22) present less validated entities in some of the available SynchroniCity data models compared with the phase I status. This is due to different assesment activites done during phase II to polish the RZs shared datasets plus IoT deployments reorganization that, in some situations caused the disappearance or fusion of several sensor devices. This is reflected on the dataset tables as less available entities.

## 5.2 Synchronicity Atomic Services

The main novelty included in this second phase, as shown in section 4, is related to the validation of the atomic services developed in WP3 and presented in D3.3 [2]. This document only considered those AS listed in D3.3 but, since D3.3 was released, new services have been included in the SynchroniCity framework. The complete set can be followed in the SynchroniCity GitLab [15]. In relation to those services that have been validated, a set of scenarios were proposed to check, in addition to the defined functionalities, their capabilities to be integrated with other SynchroniCity services and orchestrate outcomes to get or support advanced services. In this sense, documentation provided by AS developers and code distributed through the project GitLab was enough to deploy all of them. Nevertheless, some documentation should be improved to get better results on deployment times and/or get improved features. The estimation services, as they involve advanced features such machine learning techniques, required more time to be finalised, but open new interesting IoT data exploitation lines.

| Atomic Services Validation Summary |                                                                                                             |                                                                                                                                                  |                                                                      |
|------------------------------------|-------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|
| Atomic Service                     | Validated Areas                                                                                             |                                                                                                                                                  |                                                                      |
|                                    | Documentation                                                                                               | Deployment                                                                                                                                       | Functionalities                                                      |
| NGSI Urban Mobility to GTFS        | Complete documentation (GitHub and DockerHub)                                                               | Docker and Code fully documented. APIary provided.                                                                                               | Addressed functionalities covered                                    |
| GTFS Fetcher                       | Need to improve: Functionalities' description<br>CLI based operations                                       | Need Docker and OTP previous knowledge to solve deployment issues. This will be solved in next documentation review                              | Main functionalities successfully achieved                           |
| GTFS-RT Loader                     | Full documentation provided. Docker and code installation documented                                        | Easy to follow installation steps. Docker & code provided. API described. Examples included. OAUTH support suggested for further implementations | Full requirements covered                                            |
| Routing Service                    | Documentation about advanced functionalities not provided (nor referenced). OTP API not properly documented | Basic deployment well documented. Missing references about how to operate with advanced features                                                 | Multimodal routes fully supported (both, graphical and API provided) |

|                                |                                               |                                                                                                   |                                                                    |
|--------------------------------|-----------------------------------------------|---------------------------------------------------------------------------------------------------|--------------------------------------------------------------------|
| <b>Parking Prob. Estimator</b> | Full documentation provided                   | Easy to follow installation steps. Apiary included                                                | Estimations not Achieved (under development)                       |
| <b>Traffic Flow Estimator</b>  | Full documentation provided                   | Easy to follow installation steps. Apiary included                                                | Estimations not Achieved (under development)                       |
| <b>Smartcities Dashboard</b>   | API Documentation is missing.                 | Easy deployment. Minor issues to be improved related to API usability. OAUTH support is suggested | WEB-UI fully operational. API is still partially under development |
| <b>Metrics Visualizer</b>      | Maps Visualization tool not documented enough | Maps Visualization tool not easy to deploy                                                        | Required functionalities (FRs) covered by the Time Series Tool.    |

Table 27. WP3 listed SynchroniCity atomic service validation summary.

### 5.3 Validation process assessment

As it was done for the first validation round (D4.2), this second set of results has been obtained through an iterative process, involving the technical responsible staff of the Reference Zones and the atomic services developers. In this sense, the iterations were divided in two different assessment actions: support for SynchroniCity compliant framework deployment; and atomic services documentation and set-up improvement.

Related to the framework deployment assessment, the SynchroniCity MIM Validator tool, described in section 2.2.1 was distributed among project technical staff so they can check at any time the compliance level of their corresponding deployed APIs and Datasets. This text is an outcome example of the SynchroniCity MIM Validator for the Historical API of Porto. It checks all the SynchroniCity Historical Data Access API defined features, and, in case of any error, it will report the user which one is failing, helping the developers to solve the issue.

```
{ "city": "porto",
  "fiwareService": "",
  "fiwareServicePath": "",
  "entityType": null,
  "entityId": "urn:ngsi-ld:AirQualityObserved:porto:environment:ubiwhere:5b97a1a6e521e3053085c067",
  "attribute": "PM10",
  "lastUpdate": "2019-02-11T10:31:13.785Z",
  "afterLastn": {
    "status": false,
    "msg": "It returns values after last"
  },
  "beforeLastn": {
    "status": "ok"
  },
  "betweenLastn": {
    "status": "ok"
  },
  "paging": {
    "status": "ok"
  },
  "afterPaging": {
    "status": "ok"
  },
  "beforePaging": {
```

```

    "status": "ok"
  },
  "betweenPaging": {
    "status": "ok"
  }
}

```

For the datasets analysed, the tool returns the available data models, pointing out how many were found per SynchroniCity Data Model, how many of them are fully compliant and how many are not, showing what are the issues found with the wrong entities. This is an example of an error found in one Reference Zone.

```

{
  "city": "helsinki",
  "fiwareService": "",
  "fiwareServicePath": "",
  "lastUpdate": "2019-04-17T10:24:14.065Z",
  "AirQualityObserved": {
    "total": 52,
    "success": 41,
    "fail": 11,
    "detail": [
      {
        "status": "ENTITY_ERROR",
        "entityId": "fmi-100662",
        "detail": [
          {
            "dataPath": "",
            "params": {
              "missingProperty": "dateObserved"
            },
            "message": "should have required property 'dateObserved'"
          }
        ]
      }
    ]
  },
  [...]
}

```

To keep track of this assessment process, WP4 team periodically executes these validations on each Reference Zone and notifies the corresponding technical responsible people. Every two weeks, a meeting is held with the WP4 partners where summary tables as the ones shown in section 5.1 are shared and discussed. These regular meetings are also used to provide support on detected issues.

For the atomic service validation assessment, WP4 team deployed and run the services to build the scenarios described in section 4. An external approach was followed to complete this process: WP4 used only the documentation and tools provided by the developers through the SynchroniCity GitLab, as an external party would do. This allowed WP4 to check the open available documentation and source code. Based on this, several meetings with atomic services developers and providers were held to suggest improvements and fix detected issues both, in documentations and in distributed code.

Beyond this second technical validation round, the assessment process will continue to improve SynchroniCity frameworks deployment: Reference Zones will set up new components and data models running their corresponding validation process assessed by WP4 and, in turn, periodically WP4 will execute the whole process on all available endpoints to have an updated view of the overall status.

## 5.4 KPIs

Most of the technical validation KPIs presented in this document, have been previously introduced in D4.2 [3], but an additional KPI has been included to measure the number of atomic services implemented. This number is closely related to the project KPI 'Service Implementation' listed in D4.1 [1]. Table 28 summarizes all of them.

| # | KPI                    | Description                                                                                  | Indicator unit | Measurement      |
|---|------------------------|----------------------------------------------------------------------------------------------|----------------|------------------|
| 1 | SynchroniCity datasets | Number of available datasets according to SynchroniCity data models                          | #              | Total and per RZ |
| 2 | Datasets shared        | Number of datasets that are shared by 2 or more RZs                                          | #              | Total            |
| 3 | Entities               | Number of available entities (through NGSI interface)                                        | #              | Total and per RZ |
| 4 | SynchroniCity cities   | Number of RZs with a NGSI interface exposing datasets according to SynchroniCity data models | #              | Total            |
| 5 | Atomic services        | Number of atomic services implemented                                                        | #              | Total            |

Table 28. Technical validation (phase I) KPIs

The purpose of this section is to provide the current values of the identified KPIs and compare them with the ones previously measured in D4.1 to follow the evolution of the technical framework through the project lifecycle

Based on the validation results presented in sections 5.1 and 5.2, we can convey that the 8 RZs have an open NGSI interface to expose SynchroniCity datasets. These RZs expose 38 different datasets according to SynchroniCity data models, where 16 of them are at least common to 2 RZs. This leads to a total number of 80 datasets and 194778 entities that are fully compliant with SynchroniCity data models offered by the 8 RZs to the pilots. Complete results are presented in Table 29.

| # | KPI                    | Description                                                                                                    | First Validation | Second Validation | Increment |
|---|------------------------|----------------------------------------------------------------------------------------------------------------|------------------|-------------------|-----------|
| 1 | SynchroniCity datasets | Number of available different datasets according to SynchroniCity data models                                  | 19               | <b>38</b>         | 19        |
|   |                        | Number of available datasets according to SynchroniCity data models, including those duplicated in several RZs | 28               | <b>80</b>         | 50        |
| 2 | Datasets shared        | Number of datasets that are shared by 2 or more RZs                                                            | 5                | <b>16</b>         | 11        |
| 3 | Entities               | Number of available entities (including partial validated entities)                                            | 9096             | <b>195646</b>     | 186543    |
|   |                        | Number of available entities (full SynchroniCity compliant)                                                    | 5629             | <b>194778</b>     | 189149    |

|   |                      |                                                                                           |   |          |   |
|---|----------------------|-------------------------------------------------------------------------------------------|---|----------|---|
| 4 | SynchroniCity cities | Number of RZs with a NGSI interface exposing datasets according SynchroniCity data models | 7 | <b>8</b> | 1 |
| 5 | Atomic Services      | Number of atomic services implemented                                                     | - | <b>8</b> | 8 |

Table 29. KPIs global values

Table 30 shows how entities and datasets are distributed among the 8 core pilot cities.

| # | KPI                    |                | POR  | SAN  | ANT | CAR | EIN | HEL | MAN | MIL    |
|---|------------------------|----------------|------|------|-----|-----|-----|-----|-----|--------|
| 1 | SynchroniCity datasets |                | 17   | 19   | 2   | 7   | 1   | 4   | 8   | 22     |
| 3 | Entities               | Total          | 6370 | 4329 | 330 | 326 | 35  | 420 | 612 | 183383 |
|   |                        | Full compliant | 5707 | 4329 | 124 | 324 | 35  | 409 | 535 | 183377 |

Table 30. KPIs values per RZ

It should be considered that these KPIs refer to the status of the technical framework before the execution of the open call pilots. D4.5 will compile the results obtained after the open call piloting phase, where it is expected that these KPIs increase their values.

## REFERENCES

---

1. SynchroniCity Consortium, «D4.1 – Validation Methodology Description». SynchroniCity, 2017.
2. SynchroniCity Consortium, «D3.3 – Suite of atomic implementations - advanced». SynchroniCity, 2019.
3. SynchroniCity Consortium, «D4.2 – Technical Validation (Phase 2)». SynchroniCity, 2018.
4. SynchroniCity Consortium, «D2.10 – Reference Architecture for IoT Enabled Smart Cities, Update». SynchroniCity, 2018.
5. SynchroniCity Consortium, «D2.9 – Report on Advanced Reference Zones platform deployment and operational plan». SynchroniCity, 2019.
6. SynchroniCity Consortium, «D2.3 – Catalogue of OASC Shared Data Models for Smart City domains». SynchroniCity, 2019.
7. SynchroniCity-iot GitLab. synchronicity-data-models project. [Online]. Available: <https://gitlab.com/synchronicity-iot/synchronicity-data-models>
8. General Transit Feed Specification. GTFS Community. [Online]. Available: <https://gtfs.org>
9. SynchroniCity APIary. SynchroniCity. [Online]. Available: <https://synchronicityiot.docs.apiary.io/#>
10. FIWARE NGSI APIv2 Walkthrough [Online]. Available: [https://fiware-orion.readthedocs.io/en/master/user/walkthrough\\_apiv2/index.html](https://fiware-orion.readthedocs.io/en/master/user/walkthrough_apiv2/index.html)
11. ETSI GS CIM 004 V1.1.1 (2018-04) «Context Information Management (CIM); Application Programming Interface (API)». ETSI, 2018. [Online]. Available: [https://www.etsi.org/deliver/etsi\\_gs/CIM/001\\_099/004/01.01.01\\_60/gs\\_CIM004v010101p.pdf](https://www.etsi.org/deliver/etsi_gs/CIM/001_099/004/01.01.01_60/gs_CIM004v010101p.pdf)
12. IETF OAuth Working Group, «The OAuth 2.0 Authorization Framework, RFC6749» [Online]. Available: <https://tools.ietf.org/html/rfc6749>
13. FIWARE Identity Manager (IdM) KeyRock enabler [Online]. Available: <https://catalogue-server.fiware.org/enablers/identity-management-keyrock>
14. SynchroniCity Consortium, «D3.4 – Common methodology and toolset for city service customization». SynchroniCity, 2018.
15. SynchroniCity-iot GitLab. SynchroniCity project. [Online]. Available: <https://gitlab.com/synchronicity-iot>
16. OpenTripPlanner: Multimodal Trip Planning. [Online]. Available: <https://www.opentripplanner.org/>
17. Grafana, The open platform for beautiful analytics and monitoring. Grafana Labs ©, 2019. [Online]. Available: <https://grafana.com/>
18. SynchroniCity MIM Validator tool. SynchroniCity Consortium. [Online]. Available: <https://gitlab.com/synchronicity-iot/rz-instance-validator/tree/master#synchronicity-mim-validator>
19. SynchroniCity MIM Validator API Swagger. SynchroniCity Consortium. [Online]. Available: <https://framework-validator.synchronicity-iot.smartsantander.eu/api-docs/>
20. Docker: Enterprise Container Platform for High-Velocity Innovation. Docker Inc. ©, 2019 [Online]. Available: <https://www.docker.com/>
21. SynchroniCity Consortium, «D3.1 – Documentation of service designs (Specification and design of initial IoT applications)». SynchroniCity, 2018.
22. SynchroniCity Consortium, «D3.2 – Suite of baseline implementations - basic». SynchroniCity, 2018.
23. GTFS Realtime Overview. Google developers, 2108. [Online]. Available:

- <https://developers.google.com/transit/gtfs-realtime/>
24. SynchroniCity Data models. GTFSTransitFeedFile. [Online]. Available: <https://gitlab.com/synchronicity-iot/synchronicity-data-models/blob/master/UrbanMobility/GtfsTransitFeedFile/doc/spec.md>
  25. SynchroniCity Data models. UrbanMobility dataset. [Online]. Available: <https://gitlab.com/synchronicity-iot/synchronicity-data-models/tree/master/UrbanMobility>
  26. SynchroniCity Data models. TrafficFlowObserved. [Online]. Available: <https://github.com/Fiware/dataModels/blob/master/specs/Transportation/TrafficFlowObserved/doc/spec.md>
  27. SynchroniCity Data models. ParkingSpot. [Online]. Available: <https://github.com/Fiware/dataModels/blob/master/specs/Parking/ParkingSpot/doc/spec.md>
  28. SynchroniCity Data models. OnStreetParking. [Online]. Available: <https://github.com/Fiware/dataModels/blob/master/specs/Parking/OnStreetParking/doc/spec.md>
  29. SynchroniCity Consortium, Data Storage API - Historical data access [Online]. Available: <https://synchronicityiot.docs.apiary.io/#reference/data-storage-api-historical/get-historical-data>
  30. FIWARE Sort Term Historic (STH) Comet component [Online]. Available: <https://fiware-sth-comet.readthedocs.io/en/latest/>
  31. Google Maps Platform Documentation. Google Maps APIs. Google Developers, 2019. [Online]. Available: <https://developers.google.com/maps/documentation/?hl=en>
  32. SynchroniCity live validation table (RZ status) [Online]. Available: <https://validation.services.synchronicity-iot.eu/table/>
  33. Santander Ferry service. Los Reginas © 2019. [Online]. Available: <https://www.losreginas.com/>
  34. Keras: The Python Deep Learning library [Online]. Available: <https://keras.io/>
  35. TensorFlow: An end-to-end open source machine learning platform. [Online]. Available: <https://www.tensorflow.org/>
  36. SynchroniCity Parking Estimator API. SynchroniCity, 2019. [Online]. Available: <https://synchronicityparkingestimator.docs.apiary.io/#>
  37. SynchroniCity Traffic Flow Estimator API. SynchroniCity, 2019. [Online]. Available: <https://synchronicitytrafficflowestimator.docs.apiary.io/#>
  38. SynchroniCity dockerhub repository. SynchroniCity, 2019. [Online]. Available: <https://hub.docker.com/u/synchronicityiot/>
  39. FIWARE quantum leap. NGSIv2 time series. [Online]. Available: <https://smartsdk.github.io/ngsi-timeseries-api/>
  40. CrateDB: the distributed SQL database for machine data. Crate.io, © 2019. [Online]. Available: <https://crate.io/products/cratedb/>