

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:	D3.6
Deliverable title:	Customized IoT service prototypes for lead ref. zones – advanced
Document version:	Ver1.0
WP number:	WP3
Lead beneficiary:	5-ENG
Main author(s):	Giuseppe Ciulla (ENG), António Nunes, Filipe Aranda de Sá, Jaime Ventura, Sofia Peres (POR), Michael Vinall (BSL), Amir Khorasani (MMU), Adrian Slatcher (MAN), InSong Lee, Chekka Ramnath Teja, SeungMyeong Jeong (KETI), Ömer Ozdemir, Jan-Joost van Kan, Clara Pezuela, Said Rahma (ATOS), Johan van Rosmalen, Vincent de Waal (HW), Niels Wiersma (EIN), Thibault Urien, Cédric Crettaz (MI), Francesco De Rose, Matteo Filipponi, Eunah Kim (UDG), Ignacio EliceGUI (UC), Arturo Medela, Aranzazu Sanz, Pablo Pelayo (TST), Paolo Foschi (MIL), Antti Nurminen (AALTO), Veli Airikkala (FVH), Bouwe Ceunen (Rombit)
Internal reviewers:	Reza Akhavan (FCC), Daniel Puschmann (DigiCat), Martin Brynskov (AU)
Type of deliverable:	Report
Dissemination level:	Public
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

SynchroniCity project aims to open up a global IoT market to help both cities and businesses developing shared digital services to improve the quality of life for citizens and the development of local economies.

The aim of this deliverable is to report the final design of the applications belonging to the three application themes identified by SynchroniCity project: Human Centric Traffic Management, Multimodal Transportation and Community Policy Suite. The initial version of these applications is reported in SynchroniCity deliverable “D3.5 Customized IoT service prototypes for lead ref. zones - basic” [1].

In particular, this document intends to give updates about each application, giving insights about any changes between the first and the final version about scenarios, functional and non-functional requirements, available data sources and architecture. Moreover, this document enriches the description reported into D3.5 providing sequence diagrams describing the main interactions among users and the application and the user interfaces of the applications.

Finally, each RZ describes its experiences in building each application leveraging not only the SynchroniCity platform, but also the off-the-shelf Atomic Services that were introduced in Deliverable 3.3 “Suite of baseline implementations - advanced “ [2].

Abbreviations

AMAT	Agenzia Mobilità Ambiente e Territorio
AUDP	Atos Urban Data Platform
BLE	Bluetooth Low Energy
CA	Client Application
CB	Context Broker
CD	Camera Device
CPS	Community Policy Suite
D	Deliverable
EC	European Commission
ETA	Estimated Time to Arrival
ETL	Extract Transform Load
FOI	Freedom of Information
GDPR	General Data Protection Regulation
GTFS	General Transit Feed Specification
GTFS-RT	GTFS Realtime
GUI	Graphical User Interface
HCTM	Human-Centric Traffic Management
HSL	Helsingin Seudun Liikenne
IdM	Identity Management
iTLC	intelligent Traffic Light Controller
LCO	Local Care Organization
MQTT	Message Queue Telemetry Transport
NFC	Near-Field Communication
NGSI	Next Generation Service Interfaces
OTP	OpenTripPlanner
POI	Point Of Interest
RZ	Reference Zone
TCA	Thermal Camera Adapter
TLEX	Traffic Light EXchange
WP	Work Package
WT	Work Task

Contents

Executive Summary	3
List of Figures	8
List of Tables	11
1 Introduction	14
2 Application Theme: Human-centric traffic management	15
2.1 Eindhoven's Application "Data-driven bicycle mobility"	15
2.1.1 Scenarios	16
2.1.2 Requirements analysis	18
2.1.3 Available data sources	19
2.1.4 Application architecture	19
2.1.5 Sequence diagrams	24
2.1.6 User Interface	26
2.1.7 Data Management.....	29
2.1.8 Conclusion	29
2.2 Milan's Application "Decision support system for cycle path planning"	29
2.2.1 Scenarios	29
2.2.2 Requirements analysis	30
2.2.3 Available data sources	30
2.2.4 Application architecture	31
2.2.5 Sequence diagrams	33
2.2.6 User Interface	35
2.2.7 Data Management.....	37
2.2.8 Conclusion	38
2.3 Antwerp's Application "Bicycle patterns in the city of Antwerp"	38
2.3.1 Scenarios	39
2.3.2 Requirements analysis	39
2.3.3 Available data sources	40
2.3.4 Application architecture	40
2.3.5 Sequence diagrams	41
2.3.6 User Interface	42
2.3.7 Data Management.....	44
2.3.8 Conclusion	44
3 Application Theme: Multimodal transportation.....	45
3.1 Porto's Application "Porto Multimodal Assistant"	45
3.1.1 Scenarios	46

3.1.2	Requirements analysis	49
3.1.3	Available data sources	50
3.1.4	Application architecture	51
3.1.5	Sequence diagrams	54
3.1.6	User Interface	55
3.1.7	Data Management.....	58
3.1.8	Conclusion	58
3.2	Santander's Application "Park & Move"	59
3.2.1	Scenarios	59
3.2.2	Requirements Analysis.....	61
3.2.3	Available data sources	62
3.2.4	Application architecture	63
3.2.5	Sequence diagrams	66
3.2.6	User Interface	66
3.2.7	Data Management.....	68
3.2.8	Conclusion	68
3.3	Helsinki's Application " Clean Air Routing Service"	69
3.3.1	Scenarios	69
3.3.2	Requirements analysis	69
3.3.3	Available data sources	70
3.3.4	Application architecture	71
3.3.5	Sequence diagrams	73
3.3.6	User Interface	76
3.3.7	Data Management.....	77
3.3.8	Conclusion	77
3.4	Milan's Application "Multimodal-navigator for disabled people"	77
3.4.1	Scenarios	78
3.4.2	Requirements Analysis.....	80
3.4.3	Available data sources	82
3.4.4	Application architecture	83
3.4.5	Sequence diagrams	86
3.4.6	User Interface	89
3.4.7	Data Management.....	91
3.4.8	Conclusion	91
3.5	Carouge's Application "Smart Parking"	91
3.5.1	Scenarios	92

3.5.2	Requirements analysis	92
3.5.3	Available data sources	92
3.5.4	Application architecture	93
3.5.5	Sequence diagrams	93
3.5.6	User Interface	94
3.5.7	Data Management.....	95
3.5.8	Conclusion	95
3.6	Seongnam's Application "Commuter Parking Service"	96
3.6.1	Scenarios	96
3.6.2	Requirements analysis	97
3.6.3	Available data sources	98
3.6.4	Application architecture	98
3.6.5	Sequence diagrams	100
3.6.6	User Interface	103
3.6.7	Data Management.....	105
3.6.8	Conclusion	105
4	Application Theme: Community policy suite	106
4.1	Manchester's Application "Agile Governance"	106
4.1.1	Scenarios	108
4.1.2	Requirements analysis	111
4.1.3	Available data sources	111
4.1.4	Application architecture	111
4.1.5	Sequence diagrams	112
4.1.6	User Interface	114
4.1.7	Data Management.....	117
4.1.8	Conclusion	118
4.2	Porto's Application "Porto Open Interactive Map"	118
4.2.1	Scenarios	119
4.2.2	Requirements analysis	121
4.2.3	Available data sources	122
4.2.4	Application architecture	122
4.2.5	Sequence diagrams	123
4.2.6	User Interface	124
4.2.7	Data Management.....	125
4.2.8	Conclusion	125
4.3	Carouge's Application "Noise monitoring near bars"	127

4.3.1	Scenarios	127
4.3.2	Requirements analysis	127
4.3.3	Available data sources	128
4.3.4	Application architecture	128
4.3.5	Sequence diagrams	129
4.3.6	User Interface	130
4.3.7	Data Management.....	130
4.3.8	Conclusion	130
5	Conclusions	131
6	References	133

List of Figures

Figure 1: AUDP internal modules.....	21
Figure 2: AUDP – Architectural design and integration	23
Figure 3 Sequence diagram overview - Thermal camera events/data adaptation to NGSI/Context Broker	24
Figure 4 Sequence flow of the user interaction with AUDP platform.....	26
Figure 5: AUDP – Home and Map Dashboard	26
Figure 6: AUDP – Examples of charts provided by the Data Visualization Dashboard	27
Figure 7: AUDP – Data Visualization Dashboard	28
Figure 8: Milan - component diagram of Decision support system for cycle path planning application	33
Figure 9: Milan - Propose new layers diagram of Decision support system for cycle path planning application.....	34
Figure 10: Milan – Select Map layer diagram of Decision support system for cycle path planning application.....	35
Figure 11: General web interface	36
Figure 12: Web Interface: area selection.....	37
Figure 13: Overview of Antwerp application architecture for Bicycle patterns in the city of Antwerp	40
Figure 14: Antwerp - DateTimeFilter lookup sequence diagram of Bicycle patterns in the city of Antwerp application.....	42
Figure 15: Antwerp - DateTimeFilter Vehicle data	43
Figure 16: Antwerp - DateTimeFilter TrafficFlowObserved.....	43
Figure 17: Antwerp - DateTimeFilter BikeHireDockingStation	44
Figure 18: Matching layers between the overall architecture of Porto's SynchroniCity framework (left) and the MMT application architecture (right, © digitransit).....	51
Figure 19: Application component diagram of Porto Multimodal Assistant application.....	53

Figure 20 User Login and route search of Porto Multimodal Assistant application.....	54
Figure 21 Porto Multimodal Assistant application (mobile version)	55
Figure 22 Porto Multimodal Assistant application (desktop version)	56
Figure 23 User account history and recent searches screen of the Porto Multimodal Assistant application (mobile version)	57
Figure 24 User account history and recent searches screen of the Porto Multimodal Assistant application (desktop version)	57
Figure 25: Santander “Park & Move” functional architecture	64
Figure 26: Santander- component diagram of “Park & Move” application.....	65
Figure 27: Santander- sequence diagram of searching a route with “Park & Move” application	66
Figure 28: GUI Multimodal Transportation in Santander –found routes (in Spanish)	67
Figure 29: GUI Multimodal Transportation in Santander – details (in Spanish).....	67
Figure 30: Helsinki - component diagram of Clean Air Journey Planner application	71
Figure 31: Helsinki - Clean air route search component interaction sequence diagram of Clean Air Journey Planner.....	75
Figure 32: Helsinki - screenshots of Clean Air Journey Planner application	76
Figure 33: Milan - component diagram of Multimodal-navigator for disabled people application....	84
Figure 34: Milan - Explore Map and save Bookmarks sequence diagram of Multimodal-navigator for disabled people application	87
Figure 35: Milan - Select and show parking and stations sequence diagram of Multimodal-navigator for disabled people application.....	88
Figure 36: Mock-up of the Milan Multimodal navigator for disabled application - Home Page	89
Figure 37: Mock-up of the Milan Multimodal navigator for disabled application – Feature Detail ...	89
Figure 38: Mock-up of the Milan Multimodal navigator for disabled application – Preferences	90
Figure 39: Mock-up of the Milan Multimodal navigator for disabled application – Calculate Trip....	90
Figure 40: Mock-up of the Milan Multimodal navigator for disabled application – Final Path	91
Figure 41: Carouge - component diagram of Smart Parking application.....	93
Figure 42: Carouge - Smart Parking sequence diagram of Smart Parking application	94
Figure 43: Carouge - a screenshot of Smart Parking application.....	94
Figure 44: Carouge - a detail of Smart Parking application	95
Figure 45: Seongnam SynchroniCity Platform Architecture.....	98
Figure 46: Application Architecture	100
Figure 47: Sequence Diagram with City Parking	101
Figure 48: Sequence Diagram with Smart Parking.....	102
Figure 49: Sequence Diagram with CKAN	103
Figure 50: First Screen of Smart Parking	103
Figure 51: Main Screen of Smart Parking	103
Figure 52: Parking Lot Check in City Parking	104

Figure 53: Transportation information in City Parking.....	104
Figure 54: City Parking Tablet Screen Capture	104
Figure 55: Manchester - component diagram of Agile Governance application	111
Figure 56: Manchester - Subscription Creation sequence diagram of Agile Governance application	113
Figure 57: Manchester - Workflows sequence diagram of Agile Governance application.....	113
Figure 58: Manchester - High-level view of “On Street Parking” data within the dashboard.....	114
Figure 59: Manchester - New policy configuration.....	115
Figure 60: Manchester - New subscription configuration (choose an entity type, and select from available sensors).....	116
Figure 61: Manchester - Workflow configuration	117
Figure 62: Porto - component diagram of Porto Open Interactive Map application.....	123
Figure 63: Porto - Public dashboard sequence diagram of Open Interactive Map application	124
Figure 64: Porto - Public citizen map with event calendar and sensory data	125
Figure 65: Carouge - component diagram of Noise monitoring near bars application.....	128
Figure 66: Carouge - Noise monitoring near bars sequence diagram of Noise monitoring near bars application.....	129
Figure 67: Carouge - a screenshot of Noise monitoring near bars application.....	130

List of Tables

Table 1: User profiles of Data-driven bicycle mobility application	16
Table 2: Scenario "Basic01 - Counter for Cyclist" of Data-driven bicycle mobility application	16
Table 3: Scenario "Basic02 - Dashboard for professionals" of Data-driven bicycle mobility application	17
Table 4: Scenario "Advance 01 – Clean bicycle routes" of Data-driven bicycle mobility application	17
Table 5: Scenario "Advance 02 – Avoid parking in polluted areas" of Data-driven bicycle mobility application.....	17
Table 6: Functional requirements of Data-driven bicycle mobility application	18
Table 7: Non-functional requirements of Data-driven bicycle mobility application.....	18
Table 8: Data sources used in Data-driven bicycle mobility.....	19
Table 9: Data sources used in Decision support system for cycle path planning application	30
Table 10: Atomic Services adopted in Milan's Decision support system for cycle path planning application.....	33
Table 11: User profiles of Bicycle patterns in the city of Antwerp application.....	39
Table 12: Functional requirements of Bicycle patterns in the city of Antwerp application	39
Table 13: User profiles of Porto Multimodal Assistant application	46
Table 14: Scenario "Estimated time of arrival (ETA) and schedules" of Porto Multimodal Assistant application.....	47
Table 15: Scenario "Real-time mobility information" of Porto Multimodal Assistant application	48
Table 16: Scenario "Route planning by departure/arrival locations" of Porto Multimodal Assistant application.....	48
Table 17: Scenario "Route planning by POI" of Porto Multimodal Assistant application	49
Table 18: Functional requirements of Porto Multimodal Assistant application	50
Table 19: Non-functional requirements of Porto Multimodal Assistant application	50
Table 20: Data sources used in Porto Multimodal Assistant application	51
Table 21: Atomic Services adopted in Porto Multimodal Assistant application	53
Table 22: User profiles of "Park & Move" application.....	60
Table 23: Functional requirements of "Park & Move" application.....	61
Table 24: Non-functional requirements of "Park & Move" application	62
Table 25: Data sources used in "Park & Move" application	62
Table 26: Atomic Services adopted in Santander "Park & Move" application	65
Table 27: Data Management - Multimodal Transportation in Santander.....	68
Table 28: Non-functional requirements of Clean Air Routing Service application	70
Table 29: Data sources used in Clean Air Routing Service application	70
Table 30: User profiles of Milan Multimodal-navigator for disabled people application	78

Table 31: Scenario "Sc.1 - Saving Preferences" of Milan Multimodal-navigator for disabled people application.....	78
Table 32: Scenario "Sc.2 - Changing preferences" of Milan Multimodal-navigator for disabled people application.....	79
Table 33: Scenario "Sc.3 - Adding bookmark from the map" of Milan Multimodal-navigator for disabled people application	79
Table 34: Scenario "Sc.4 - Finding free parking lots " of Milan Multimodal-navigator for disabled people application	79
Table 35: Scenario "Sc.5 - Calculating routes plan between two stations" of Milan Multimodal-navigator for disabled people application	80
Table 36: Functional requirements of Multimodal-navigator for disabled people application.....	81
Table 37: Non-functional requirements of Multimodal-navigator for disabled people application ...	81
Table 38: Data sources used in Milan Multimodal-navigator for disabled people application.....	82
Table 39: Atomic Services adopted in Milan's Multimodal-navigator for disabled people application	86
Table 40: Non-functional requirements of Smart Parking application	92
Table 41: Data sources used in Smart Parking application	92
Table 42: User profiles of Smart Parking application.....	96
Table 43: Scenario "Find parking place Solutions" of Smart Parking application	96
Table 44: Scenario "My Route Solutions" of Smart Parking application.....	97
Table 45: Scenario "Transfer parking lot information " of Smart Parking application	97
Table 46: Functional requirements of Smart Parking application.....	97
Table 47: Non-functional requirements of Smart Parking application	98
Table 48: Data sources used in Smart Parking application	98
Table 49: Application Components in Seongnam SynchroniCity platform	100
Table 50: User profiles of Agile Governance application	109
Table 51: Scenario "Monitoring the City / Real-Time Response" of Agile Governance application	109
Table 52: Scenario "Management Information" of Agile Governance application	110
Table 53: Scenario "Dynamic Strategy" of Agile Governance application.....	110
Table 54: User profiles of Porto Open Interactive Map application	119
Table 55: Scenario "Noise monitoring" of Porto Open Interactive Map application.....	119
Table 56: Scenario "Policy assessment" of Porto Open Interactive Map application	120
Table 57: Scenario "Preventive action" of Porto Open Interactive Map application	120
Table 58: Functional requirements of Porto Open Interactive Map application	121
Table 59: Non-functional requirements of Porto Open Interactive Map application.....	121
Table 60: Data sources used in Porto Open Interactive Map application.....	122
Table 61: Non-functional requirements of Noise monitoring near bars application	127

Table 62: Data sources used in Noise monitoring near bars application 128

Table 63: Atomic Services adopted in Carouge Noise monitoring near bars application 129

Table 64: Adoption map of Atomic Services in applications designed by RZs 132

1 Introduction

This deliverable presents the final results of Task 3.3, which aims to guide the design and implementation of the different Reference Zones (RZs) in the three application themes of the SynchroniCity project: Human Centric Traffic Management, Multimodal Transportation and Community Policy Suite.

It describes the final design of the applications starting from their initial description reported in deliverable “D3.5 Customized IoT service prototypes for lead ref. zones - basic” [1].

Like its predecessor, this deliverable is organized in three main sections, one for each application theme: Human Centric Traffic Management (Section 2), Multimodal Transportation (Section 3) and Community Policy Suite (Section 4); finally, conclusions are reported in section 5.

For each application, the main differences between its initial and final version are reported; these mainly focus on: the reference scenarios, the functional and non-functional requirements, the available data sources and the application architecture.

Furthermore, new sections are provided for each application in order to deeply investigate its design and how it is taking advantage of SynchroniCity platform. In particular, this deliverable includes the following new sections:

- **Sequence diagrams:** in this section each RZ describes the main interactions among the users and the application or between the application and SynchroniCity Platform.
- **User interface:** in this section the RZ illustrates the Graphical User Interface (GUI) of each application showing the more relevant parts of the user interface.
- **Data management:** this section reports how the application deals with data (e.g.: privacy aspects).
- **Conclusion:** in this section, the RZ describes the main experiences and challenges faced during the design of the application.

It is important to underline that each RZ is not involved in every application theme, because their participation is driven by their specific aims and needs; in particular, participation of RZ is distributed as shown below (for total twelve applications):

- Human Centric Traffic Management
 - o Antwerp
 - o Eindhoven
 - o Milan
- Multimodal Transportation
 - o Carouge
 - o Helsinki
 - o Milan
 - o Porto
 - o Santander
 - o Seongnam
- Community Policy Suite
 - o Carouge
 - o Manchester
 - o Porto

This document reports also an additional experience with the SynchroniCity platform in the context of Multimodal transportation. In fact, a new application is reported and its design is described in Section 3.6.

2 Application Theme: Human-centric traffic management

Objective of applications belonging to this theme is to improve bicycle mobility in cities considering different factors that can influence this means of transportation and users' behaviours (e.g.: traffic generated by cars, air pollution, etc.); the final aim is to setup a "data-driven bicycle mobility" (described in SynchroniCity deliverable "D3.1 -Documentation of service designs" [3]). Human Centric Traffic Management applications leverage different kinds of data from the city; these are analysed and combined to provide useful insights and to improve different aspects of bicycle mobility, such as the cycling experience, safety of people and infrastructure planning.

As reported in deliverable "D3.5 Customized IoT service prototypes for lead ref. zones - basic" [1], there are three RZs participating in this application theme: Eindhoven, Milan and Antwerp. In particular:

- **Eindhoven** focuses on improving experience and management of bicycles in the city; these needs originates from the the cohabitation of the bicycles with other transportation means (such as cars) that generate pollution; the aim is to improve the experience of bicycle users. (Section 2.1)
- **Milan** aims to better manage cycle lanes; in particular, the objective is to support decision makers of the Municipality in planning new cycle lanes considering different factors that can influence their choices (such as location of potential obstacles), in order to facilitate and encourage the use of bicycles for personal transportation. (Section 2.2)
- **Antwerp** aims to acquire information about the use of bicycles in the city area in order to gain useful insight (for instance to identify congestion points); which will be then useful to adjust and manage mobility policies, to take the most appropriate decisions and improve the infrastructure for bicycle users. (Section 2.3)

2.1 Eindhoven's Application "Data-driven bicycle mobility"

This section is an update of previous information about reference zone provided in Deliverable D3.5. This includes alterations and adaptation to the zone needs and constraints since the submission of Deliverable D3.5. Thus, it will not repeat information but just provide a summary of the overall goal of the pilot and the updates of the different use cases which are included.

Eindhoven is a green city which is worried about sustainability and health of population, thus the bicycle is one of the preferred transportation means in the city. The cohabitation of the bicycles with other transportation means and the improvement of the experience of bicycle user are the main target of this pilot. The propose use cases are then contributing to this better experience and management of bicycles in the city.

All the use cases imply to collect data from diverse sources, which are detailed in each use case, normalize that data, analyse and in some cases aggregate some data sources, and visualize the useful data to the relevant municipality services (Traffic Management service for example). All these functionalities will be carried out by using the Atos Urban Data Platform (AUDP), the smart cities solution of Atos powered by FIWARE, which integrates the data sources and provide the useful views over the data.

AUDP is an open source components framework to access and manage heterogeneous context information through APIs. It relies on NGSI standard (Next Generation Service Interface) for the exchange of context information.

The main functionalities of AUDP [4] are:

- Collect information from the environment, through sensors or open data sources, mobile devices or other sources fulfilling the security and privacy requirements
- Real time data analysis, to support the decision making
- Visualization of data in a graphical and visual way through maps, graphics, alarms, etc.

- Data provisioning for developing on services from them
- Triggering of actions or actuations according to some defined rules and analyzed data
- Secured access control to devices, data and services

Licence

The application will be released under the terms of an Open Source licence (e.g.: MIT, GPL, Apache, etc.); the concrete licence will be decided in the future.

2.1.1 Scenarios

2.1.1.1 Users profiles:

Here the list of the updated users' profiles for Eindhoven scenarios.

Table 1: User profiles of Data-driven bicycle mobility application

User profile ID	User description
User01: Martha	Cyclist. A cyclist who wants to use bicycle routes which are less polluted and crowded.
User02: Marco	Traffic Management Professional. Traffic Management Professional is responsible for efficient use of Traffic Management Installations such as traffic lights to optimize flow of traffic for all modalities (bike, car, pedestrian, emergency, busses, etc.)
User03: Maria	Mobility and Sustainability Policy Advisor. She is responsible for developing and monitoring policies for the municipality with respect to transportation (for instance better cycling corridors, or optimal access of the city centre) and sustainability (for instance, clean air and /or noise reduction).

2.1.1.2 Scenarios

This section lists the final use cases defined for Eindhoven reference zone.

Table 2: Scenario "Basic01 - Counter for Cyclist" of Data-driven bicycle mobility application

Scenario title	Basic01 - Counter for Cyclist
User profile	User01 Martha – Cyclist
Background	Martha is a cyclist and she is using the service road parallel to the ring road. She intends to follow a course along the ring road for one or more intersections. Thermal cameras are installed in the indicated road to detect the passing bicycles.
Objective	Martha wants to stay in shape and have a positive contribution to the sustainability goals.
Storyline	Martha passes a RZ thermal camera installed at circa 125 metres before an intersection, possibly together with other cyclists. The RZ application picks up the RZ thermal camera traffic user data, correlates this data with other relevant data sets in the SynchroniCity platform, calculates the number of daily cyclists passing this intersection. This number may be shown on different type of displays

Table 3: Scenario "Basic02 - Dashboard for professionals" of Data-driven bicycle mobility application

Scenario title	Basic02 – Camera info for visualization (dashboard for professionals)
User profile	User 03 Maria – Policy advisor
Background	Maria is a policy advisor who is responsible for mobility and sustainability. She is looking for data that can help her make informed data driven decisions.
Objective	Maria wants to know how many cyclists cross the intersection in our reference zone at any given time and how this data evolves over time.
Storyline	Maria opens the dashboard on AUDP. In the application, she can select information from the thermal camera bicycle counter and display it on the map of the city. She can also make a report of historical count / number of cyclists over a period of months, weeks, days, hours and 15 min intervals. In this way, she gets insights that help her to report the success of bicycle stimulation measures that are taken.

Table 4: Scenario "Advance 01 – Clean bicycle routes" of Data-driven bicycle mobility application

Scenario title	Advance 01 – Clean bicycle routes (dashboard for citizens)
User profile	User01 Martha – Cyclist
Background	Martha is worried about her health and prefers to use her bicycle by using routes which are the cleanest as possible of pollution. Eindhoven city has real time sensors for air quality across the city and bicycles routes publicly available.
Objective	To plan her bicycle trip in advance by using the least polluted route.
Storyline	Martha looks at the AUDP dashboard (view for users) to check before her trip which route is less polluted. A map is showing the city bicycles routes and the air quality value in each route (when available), thus Martha may choose a less polluted route according to her preference, or at least be aware of the level of pollution of the route she will use.

Table 5: Scenario "Advance 02 – Avoid parking in polluted areas" of Data-driven bicycle mobility application

Scenario title	Advance 02 – Avoid parking in polluted areas (dashboard for professionals)
User profile	User02 Marco – Traffic Management Professional
Background	Marco oversees managing the traffic by looking at the data collected from several city sensors. Among his responsibilities is to close areas of the city to the traffic when pollution scenario is above a threshold level, so cars cannot park in some areas of the city. Eindhoven city has real time sensors for air quality across the city and public parking areas information publicly available.
Objective	To restrict the parking in some areas when pollution in those areas is too high.
Storyline	Marco looks at the AUDP dashboard (view for professionals) to check the air quality in different parking areas of the city. A map is showing the parking areas and the air quality value in each of them (when available), thus Marco may close the traffic access to certain area. On the other hand, any citizen may look at the AUDP dashboard (view for citizens) which areas have a high pollution and avoid them since they will be closed to traffic.

2.1.2 Requirements analysis

During previous project phase the RZ team was analysing the requirements of the previous scenarios and some of them were not reachable since some of the required data sources from providers which were not involved in the project and their data was not available. Thus, a new analysis was needed to find feasible scenarios with available data in the scope of the project and still under the interest of Eindhoven reference zone. Thus, here below the data sources and providers which has provided the data used in the use cases that are described in the previous section:

- Thermal cameras in roads, installed by Heijmans (manufactured by FLIR). They allow counting the number of cyclists passing by a road.
- From Eindhoven open data portal:
 - o Air quality¹
 - o Parking zones²
 - o Bicycle routes³

2.1.2.1 Functional requirements

Table 6: Functional requirements of Data-driven bicycle mobility application

ID	Description
FR-1	Real time & historic sensing data about the location of a cyclist (based on Thermal Camera data (B)) MUST be provided. (1) (B)
FR-2	The application MUST process data from real-time (when applicable combined with historic) sources in (near) real-time. (1) (A)
FR-3	The application MUST identify certain cyclist presence and classify these events in (near) real-time based on real-time (when applicable combined with historic) sources. (1) (B)
FR-7	The application MUST provide user identification and access control at least by a username/password basic security. (2) (B)
FR-8	The application SHOULD provide user admin features to control settings via admin panels. (2) (B)
FR-9	The application MUST provide flexibility in terms of viewing, or having access to, certain data sets (e.g. through dashboard widgets). (2) (B)
FR-10	The application SHOULD provide a map as the main way of interaction to superimpose data sets. (2) (B)
FR-11	The application MAY provide different manners of visualizations. (2) (B)
FR-12	The application SHOULD process information on real-time bicycle presence information and make it available on the dashboard. (2) (B)
FR-13	The application SHOULD allow selection of specified time window and display corresponding aggregated data over the selected time window on the dashboard. (2) (B)
FR-14	The application SHOULD provide advice/prediction based on aggregated historical data. (2) (B)

2.1.2.2 Non-functional requirements

Table 7: Non-functional requirements of Data-driven bicycle mobility application

ID	Description
----	-------------

¹ <https://data.eindhoven.nl/explore/dataset/locaties-airboxen/information/>

² https://data.eindhoven.nl/explore/dataset/parkeerzones/information/?disjunctive.automaatnummer&disjunctive.dagtarief&disjunctive.straat&disjunctive.type_en_merk

³ <https://data.eindhoven.nl/explore/dataset/strooiroutes-voor-de-fiets/information/?disjunctive.route>

NFR-1	The application MUST adhere to privacy laws (e.g. when it comes to tracking cyclists). This requirement is related to Privacy aspects.
NFR-2	The application MUST demonstrate a high level of security and include latest data encryption methods. This requirement is related to Security aspects.
NFR-3	The application MUST have a response time that is below 500 ms from the moment of sensing until actuation, so the cyclist has travelled <3.5 m (at 25 km/h). This requirement is related to Performance aspects.
NFR-4	The application MUST be backed up, data-loss MUST be prevented. This requirement is related to Availability aspects.
NFR-5	The application SHOULD have up-time higher than 99.5% per year. This requirement is related to Availability aspects.
NFR-6	The application SHOULD be deployed through a cloud-based service. This requirement is related to Deployment aspects.
NFR-7	The application SHOULD consume the minimum amount of resources as reasonably possible. This requirement is related to Efficiency aspects.
NFR-8	The application SHOULD provide enjoyable and intuitive experience to user. This requirement is related to Usability aspects
NFR-9	The application SHOULD use the latest standards in browser technology (e.g. be responsive independent of device-type and browser). This requirement is related to Accessibility aspects.
NFR-10	The application software & code base SHOULD comply with the most recent, de facto industry standards at time of development. This requirement is related to Compliance aspects.

2.1.3 Available data sources

Table 8: Data sources used in Data-driven bicycle mobility

Title	Description	Open Data / API	Adopted SynchroniCity Data Model	Notes
Thermal camera data endpoint	FLIR ThermiCam V2X ⁴ (thermal camera) providing following raw data: type of Modality, Count of cyclists	API/SDK	TrafficFlowObserved	Private
Air quality	AiREAS open air quality data	API	AirQualityObserved	Public
Parking areas	Parking zones in Eindhoven	API	OnStreetParking	Public
Bicycles routes	Bicycle routes in Eindhoven	API	RoadSegment	Public

2.1.4 Application architecture

The aim of this section is to summarize the overall architecture that provides the basis for the provision of services and the associated infrastructure within the scope of the SynchroniCity platform and scenarios, in terms of components, services, data and information management, security management, among others.

The overall architecture and infrastructure environment overview consists of three main groups of elements:

⁴ http://www.flirmedia.com/MMC/CSV/Traffic/IT_0013_EN.pdf

- The SynchroniCity infrastructure mainly represented by the hardware components based on physical server and underlying virtual machine, the network of services and IoT device, among others, which conforms the SynchroniCity cloud infrastructure.
- The SynchroniCity system and software components designed by the way of building blocks which represents these modules and elements involved in the SynchroniCity platform.
- The external entities as the end-users, service providers, application providers, external stakeholders, city representatives and third-party application developers who interact with the SynchroniCity platform by way of the API available to access the components, services and functionalities of the platform.

The SynchroniCity platform architecture enables the data exchange between the SynchroniCity components. It also manages the reception of information (IoT device events and data), services requests coming from the application providers or third-party stakeholders who can be also the cities backend service providers, send and exchange information through the northbound and southbound APIs, and vice versa by the way of the SynchroniCity API and gateway adapter components, such as the Thermal Camera Adapter (TCA), etc.

2.1.4.1 General architectural design

The general architectural design defines elements that deliver and provide services interconnected forming or composing a network of services dedicated to providing services for cities and client application(s) or third party backends, or to the applications or services located on client and/or external server side that allow the necessary interactivities related to the scenarios between the applications and services, in the cities clients or cities legacy systems, and the SynchroniCity platform and associated service network systems. This architecture and infrastructure environment involves different elements and actors interacting with each other following the principle based on consistent standards and specifications supporting that environment.

These architectural elements [5] could be deployed by configuring the SynchroniCity platform elements (set of services and components) as a private service network or local cloud services in the scope of an adapted local platform instance to each reference zone or as a global centralized instance which could be offered to the third-party service and application providers and stakeholders involved in the context of the SynchroniCity project, in order to take advantage of the SynchroniCity platform and their service network capabilities from the point of view of interoperability, entities interaction and data model shared, improving their smart cities services.

Finally, it is possible to use different typology of infrastructure, service and database servers, according to the type of services and data that are hosted and stored (standard user access information, IoT device data) for security and identity management (FIWARE PEP Proxy, Wilma, etc.), security protocols (OAuth 2.0, etc.) and adapted cloud services.

Application Components

In the context of the Human centric traffic management and service domain related to the bicycle target pilot and corresponding scenario, the main components are based on:

- **Atos Urban Data Platform (AUDP)** [4]: which is the smart cities solution of Atos powered by FIWARE, the final application to be used to demonstrate/implement use cases. The application integrates different data sources and provides useful views over the data.
- **SynchroniCity Framework** (mainly Orion Context Broker)
- **The Thermal Camera Adapter (TCA)**, explained later in the component's section). Thermal cameras provide events and data related to bicycle information.

Following sections describe the different components of the application:

Atos Urban Data Platform:

Notice that as the AUDP is powered by FIWARE, so, it is just directly plugged into Eindhoven SynchroniCity Framework. All the main FIWARE components are integrated in the AUDP meaning that the Orion Context Broker is also now part of the AUDP.

AUDP is a complete smart city platform that is composed of several generic software modules as shown below in Figure 1.

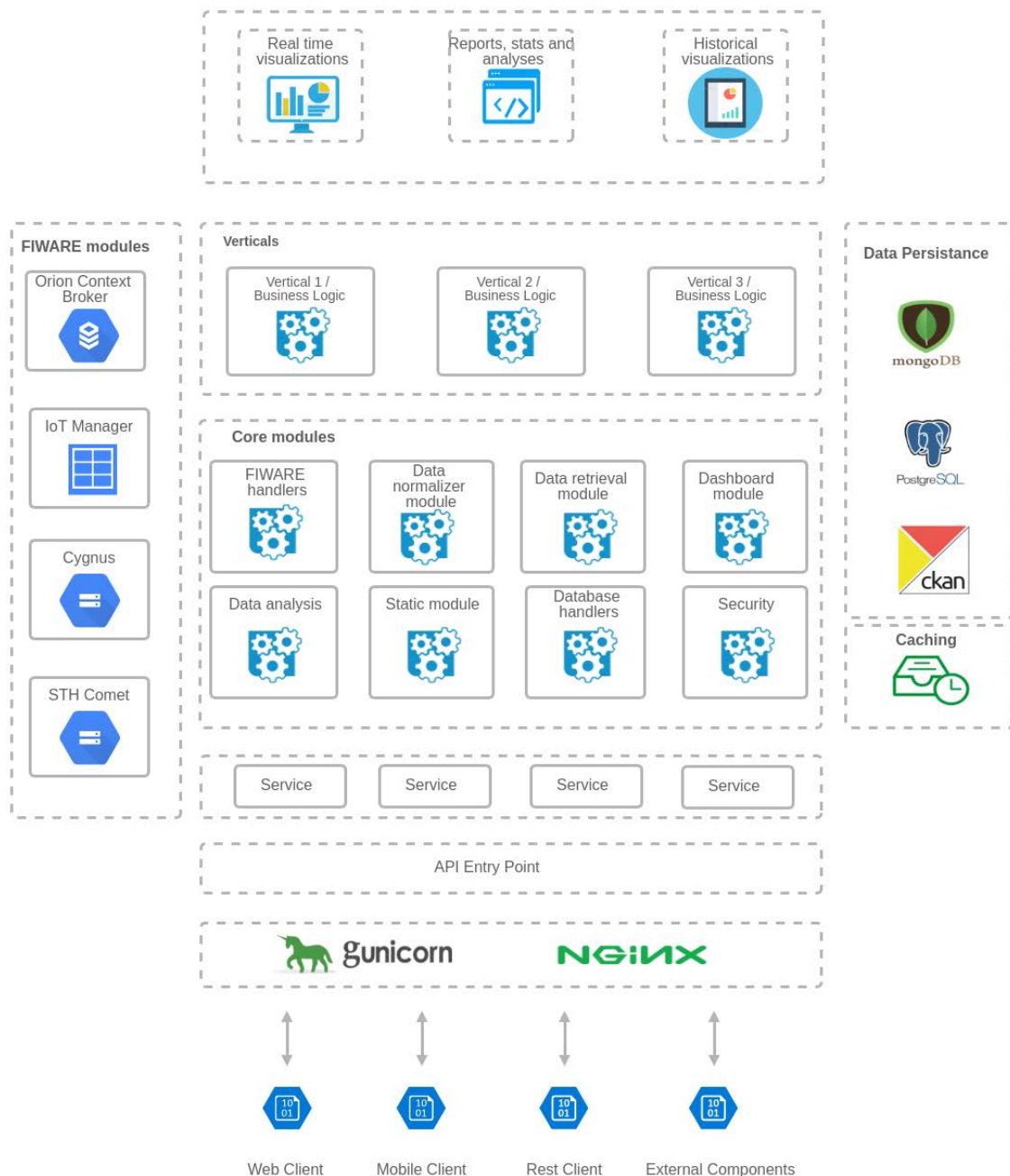


Figure 1: AUDP internal modules

Notice that the left side of the Figure 1 (FIWARE modules) shows the linking and interoperable points with the current existing SynchroniCity Platform implementation in Eindhoven. The AUDP is connected through these interoperability points (supporting the OASC principles).

These are custom software components that are a part of the overall AUDP platform by providing different logic. The most relevant component (about implementing the desired functionality explained in the use case's description), is the **Dashboard module**: Dashboard module manages front-end interactions with the AUDP back-end modules, typically by exposing/consuming data. More details about this module in the Section 2.1.6.

We are describing the other components that in the AUDP that form the Dashboard module, and allow the integration and interoperability with the existing SynchroniCity Platform:

- **FIWARE Handlers**: This module is responsible for managing FIWARE integrations and NGSI specific operations.
- **Data Normalizer module**: In charge of converting external data sources/types to NGSI based FIWARE data models.
- **Data Retrieval module**: This module manages HTTP connections with the external systems and data sources through their AUDP life-cycle
- **Data analysis**: This module is responsible for the data analytics operations based on the Pandas⁵ Python library
- **Static module**: Serves static project assets (images, documents, icons etc.)
- **Database Handlers**: Manages database related operations.
- **Security**: Security module supports the user authentication and authorization flows, it is used for resource restriction used in both front-end and back-end modules
- **Service layer**: AUDP environment supports the communication with external components (if needed) via this service layer which exposes and consumes several different resources and services.
- **Deployment layer**: AUDP environment is hosted on NGINX and Gunicorn

Figure 2 shows the architectural design and integration between the Atos Urban Data Platform and the existing SynchroniCity Platform for Eindhoven. The integration is straightforward thanks to the OASC principles and the compatibility between platforms.

⁵ <https://pandas.pydata.org/>

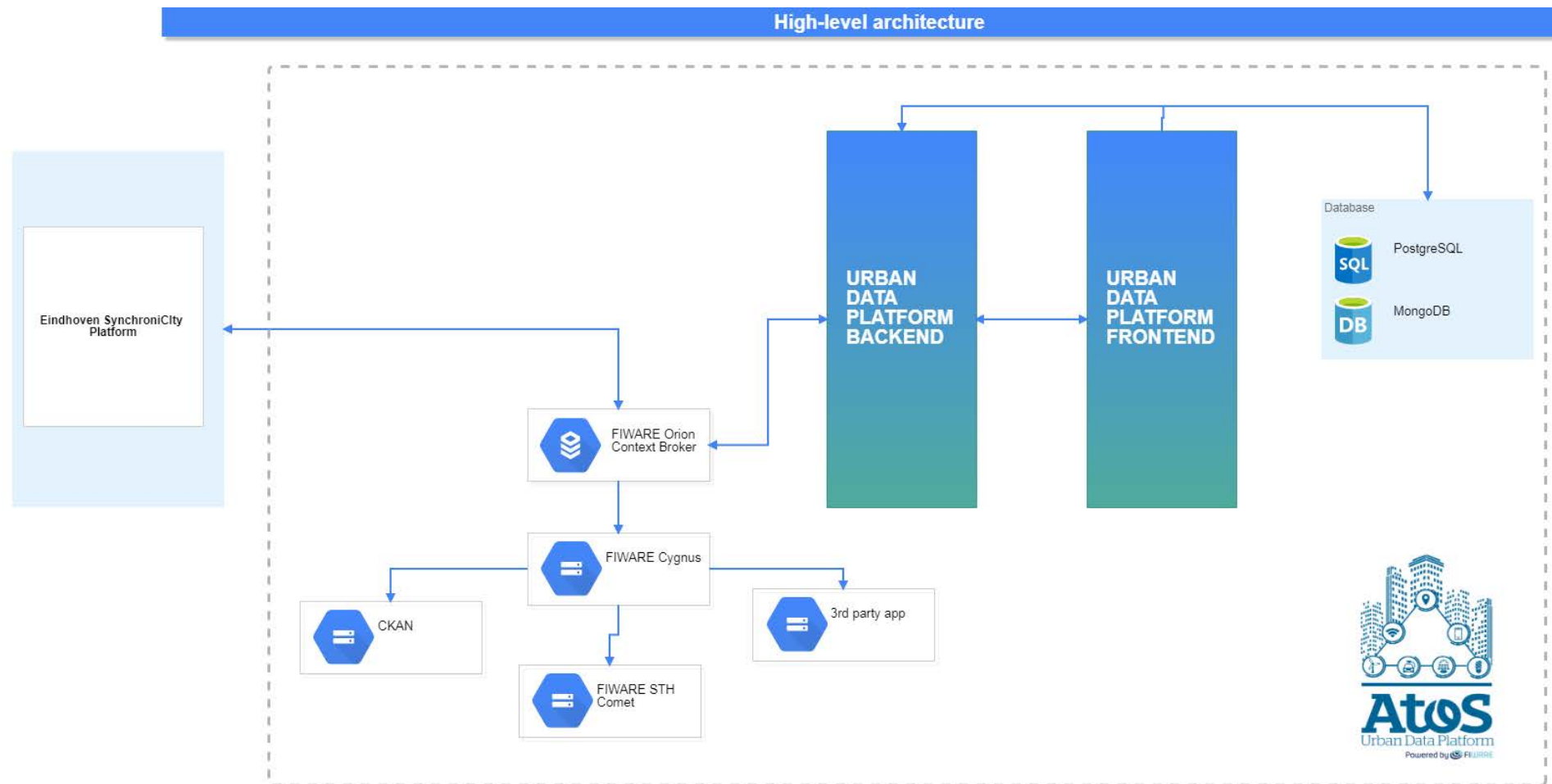


Figure 2: AUDP – Architectural design and integration

Thermal Camera Adaptor

The Thermal Camera Adaptor (TCA) runs inside the Eindhoven SynchroniCity Platform. The module is in charge of sending and receiving the bicycle count and bicycle presence events generated by the physical thermal camera deployed in the Eindhoven infrastructure. The module has been implemented using WebSocket technology[6], subscribing to the thermal camera's API. After the subscription, it starts to receive specific bicycle events from the thermal camera. TCA converts and normalizes the thermal camera data and transforms into proper NGSI formats to be sent to Orion Context. This information is stored into the AUDP database (by using the Cygnus connector) and showed in the Platform Dashboard.

2.1.5 Sequence diagrams

This section aims to presents the main sequence diagram related to the TCA component, that is customized to manage events and data coming from the thermal camera, adapting the specific camera data to a standardized NGSI data associated with a well-defined data model and corresponding NGSI entities aligned to the scenarios and focused on the needs of the Eindhoven city.

Figure 3 presents a common view of the interaction sequence for all the scenarios related to bicycle count and bicycle events. Basically, it shows the internal process to extract and convert the data from Thermal Cameras, into a normalized NGSI format to be stored and displayed (AUDP). The Client Application could be considered as the AUDP dashboard to show the information to implement use cases described in Sec 2.1.1.2.

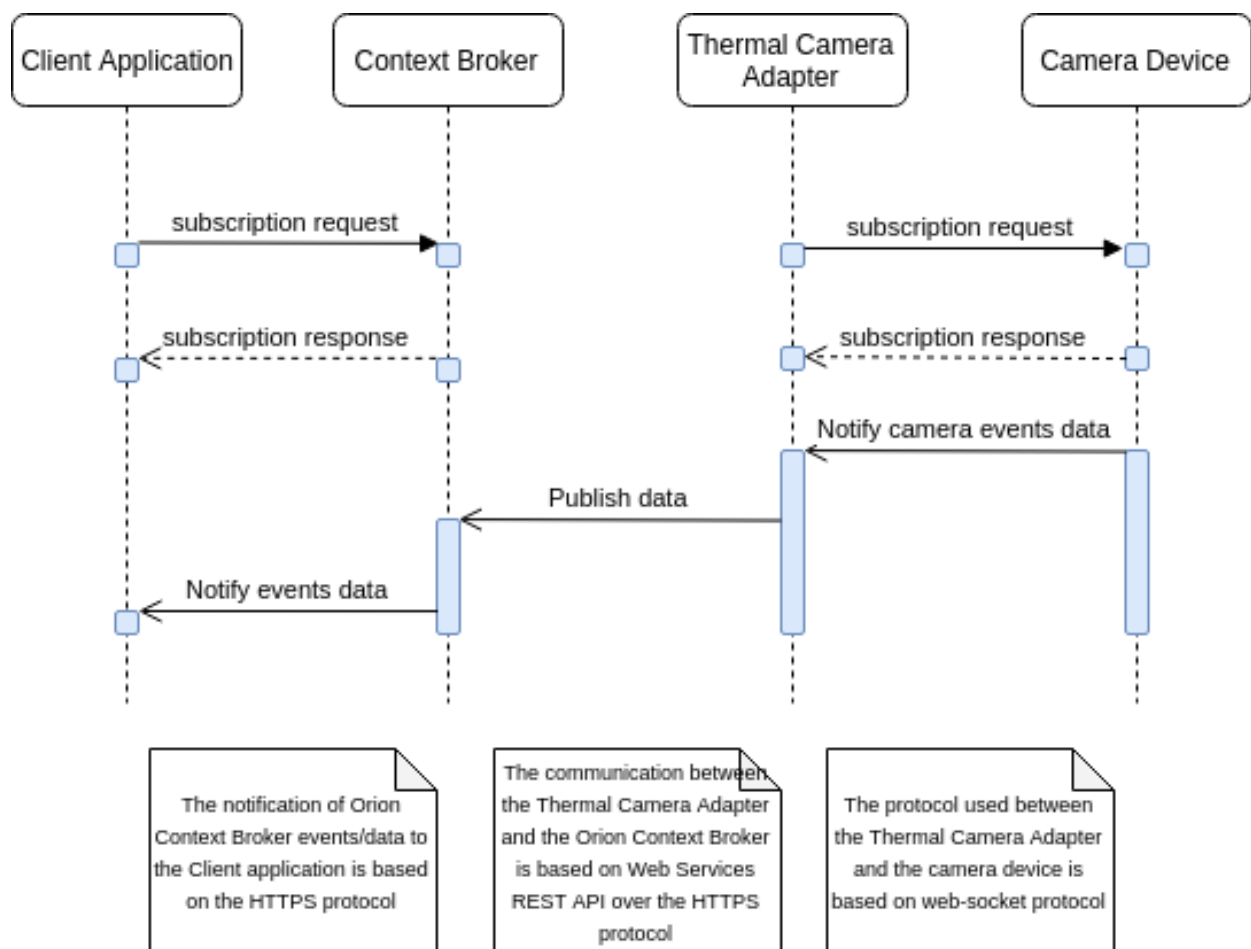


Figure 3 Sequence diagram overview - Thermal camera events/data adaptation to NGSI/Context Broker

The basic initial state of the system must be the following:

- The SynchroniCity platform must be installed, deployed, configured, ready and running.
- The thermal Camera Device (CD) must be running and sending events and data related to Bicycle.
- The component Thermal Camera Adapter (TCA) must be running and ready to receive events and data related to bicycle service domain from the thermal camera.
- Client Application (CA) must be subscribed to the Context Broker (CB)

The process and protocol related to this sequence diagram are the following:

- Client Application subscribes to the Orion Context Broker to receive NGSI notification from the CB related to bicycle count and bicycle presence data. To subscribe to the CB, the CA uses the Context Broker REST API over HTTP(S) protocol following the NGSI specification. The CA should receive a valid subscribe response without error.
- The Thermal Camera Adapter (TCA) subscribes to the Thermal Camera Device (CD) to receive events with data related to bicycle purpose. The Thermal Camera Device sends to the TCA a valid subscribe response. The communication between the TCA and the camera device is based on the WebSocket protocol [6]. Events and data received from the camera device are proprietary, i.e. not a standardized structure of the message. When the TCA is correctly subscribed, i.e. without error response returned by the camera device, the communication is established, and the camera starts to send traffic events / data to the TCA. These events are filtered to receive only bicycle events and corresponding data based on the camera device API specification.
- The TCA adapts and converts the camera device messages to a formal NGSI message and corresponding data model; therefore, the TCA publishes the adapted message to CB.
- The client application subscribed to the Context Broker for the corresponding NGSI entities related to bicycle message will be notified. The client application is responsible for maintaining and managing these notified data which could be stateless or stateful depending on the type of application that the CA wants to do.

Figure 4 represents a user accessing the AUDP dashboard to get the data and to act according to use cases Sec 2.1.1.2. Notice the data base is the integration point between the previous (Figure 3) diagram and the AUDP dashboard used to interact with users. From a high-level point of view, internally the infrastructure and the SynchroniCity Platform collects data into the database. The AUDP, and its different components, uses this data to provide functionality to final uses (use cases to implement)

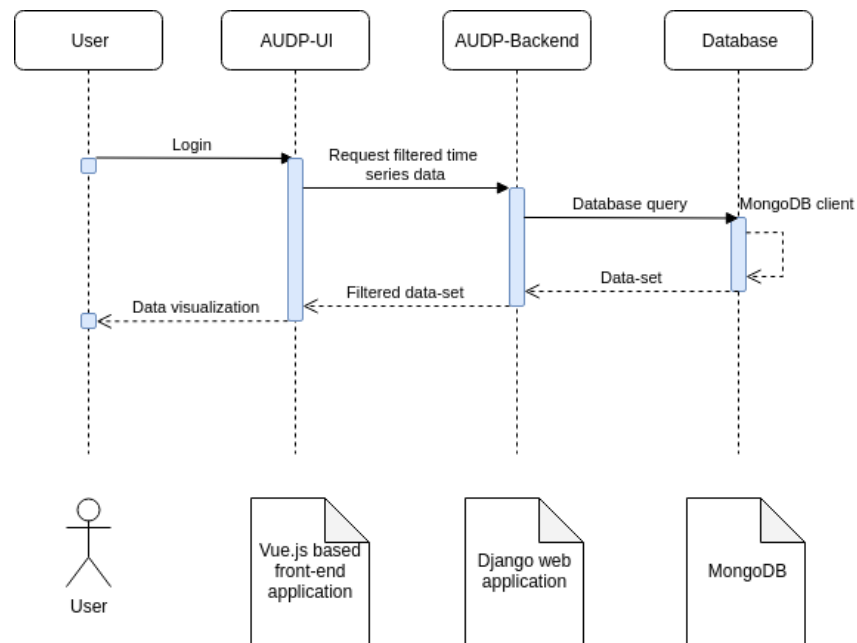


Figure 4 Sequence flow of the user interaction with AUDP platform

2.1.6 User Interface

As described above, AUDP platform is built up by different software modules and one of them is the front-end module which manages all kinds of user interaction with the generic smart city data. It offers display, list, change, download of the data-sets and provides several admin functions which eases the work for the end user.

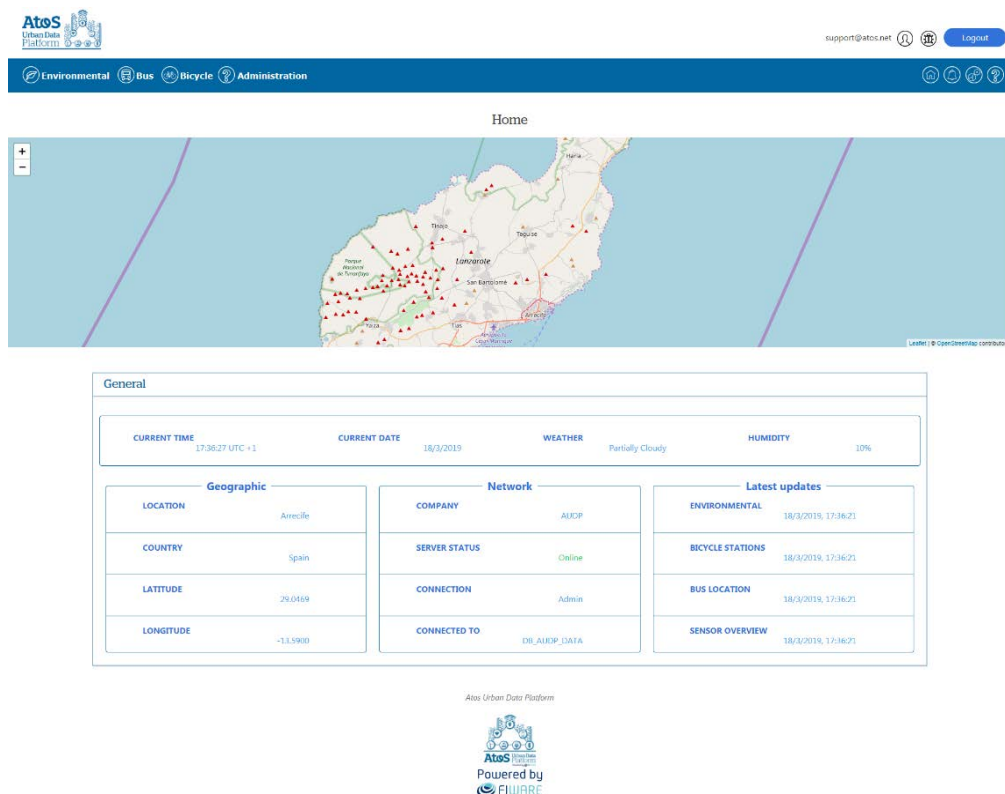


Figure 5: AUDP – Home and Map Dashboard

Figure 5 shows the landing page of the AUDP-UI component, where we display several useful information from the city itself. There are three main categories in this section:

- Geographical information of the city
- Sanity check information about the AUDP services
- Latest updates section which displays the status of the data-sets within the platform

Users can browse different sections from the navigation bar for more specific information on the collected data-sets. Platform itself offers a feedback mechanism for bug or idea reporting and a settings page for managing different kind of user & platform specific configurations.

From this page, users will have access to a map of the city with information related to the: bicycles (numbers of users by areas monitored by Thermal Cameras), air quality, routes, and parking areas.

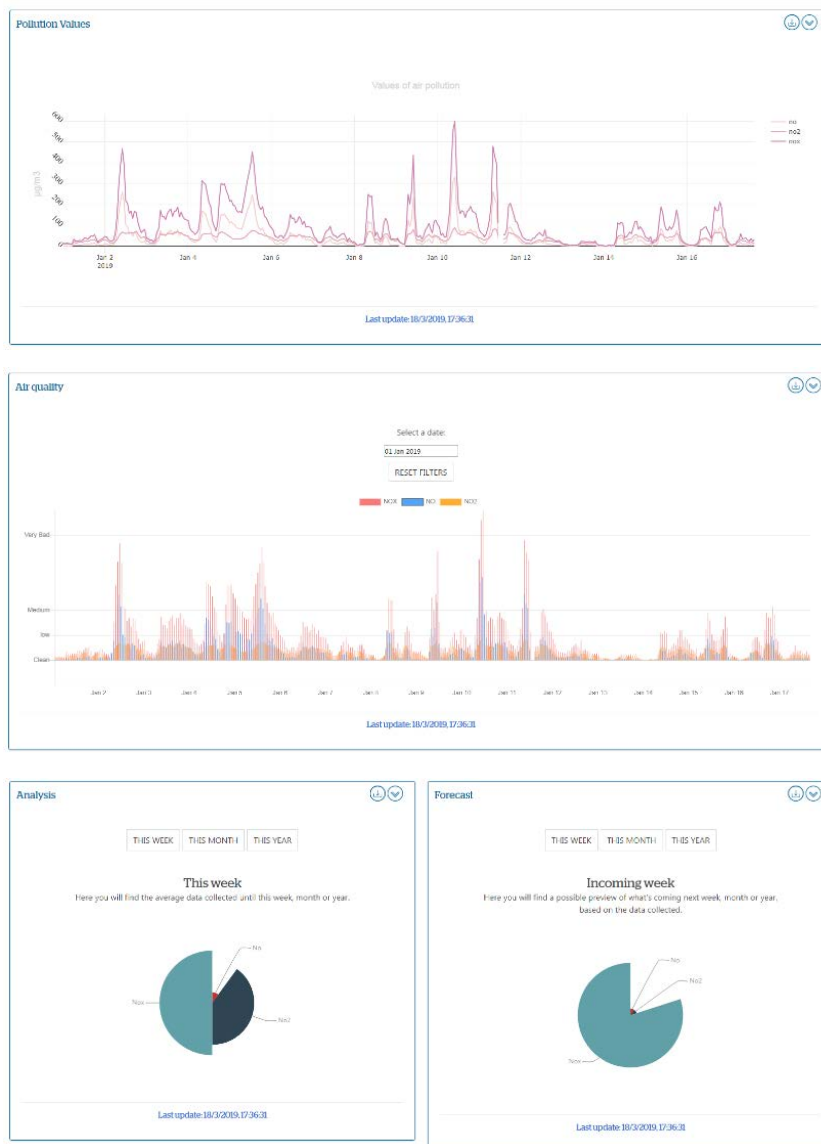


Figure 6: AUDP – Examples of charts provided by the Data Visualization Dashboard

Figure 6 provides examples of the main charts made available by the AUDP platform; in detail, the platform provides:

- Map view of real-time or historical data of the connected sensors or data sources.
- Historical data graphs over a time span
- Frequency of the data that is collected in real time
- Other interactive graphs for more information
- Ability to download the data in CSV or PDF formats
- Eindhoven city bicycle paths visualization including the air quality information to be displayed to the platform user.

Thanks to these functionalities, a user will have a more detailed view of the information provided by the platform in order to be informed or take different decisions: routes less polluted, areas with a high amount of cyclists passing by, etc., according to the objectives provided by the use cases and depending if the user is a regular cyclist or a municipality professional.



Figure 7: AUDP – Data Visualization Dashboard

Figure 7 shows the status section within the AUDP platform where we list and display the accessibility conditions of the data sources. This area is important since it gives insights about the maintenance of infrastructure and the current status of the platform.

2.1.7 Data Management

For the open data, there is not any privacy concern in using it, since it has been published by the city. In case of thermal camera, as only events are transferred and stored, no sensitive data is managed.

2.1.8 Conclusion

The presented use cases support the mindset and strategy of Eindhoven city to become a greener and healthier place to live. They are implemented by using Atos Urban Data Platform which complies perfectly with SynchroniCity approach of cross city services and FIWARE philosophy. They combine closed and open data from different city sources providing a typical situation in real life when data from diverse sources must be aggregated to offer an added value. The adoption of the use cases by Eindhoven will provide additional services which they already give to their citizens to improve their life quality and style.

2.2 Milan's Application "Decision support system for cycle path planning"

In the recent years, the city of Milan has experienced a huge increase in bike usage thanks to various bike sharing services that started to operate. In order to support bike usage, the Municipality of Milan has built reserved cycle paths widespread in the city and new ones are planned. The aim of the application is to support the Municipality of Milan in taking the right choices concerning new cycle paths to be offered to its citizens. The application will integrate GeoViewer [7], a geographical tool provided by the GeoPortal of Milan Municipality, enabling the user to visualize and interact with several layers available in the map, as reported in [1]. It will provide additional layers derived from the analysis of data retrieved through the SynchroniCity framework. These layers will graphically represent the path of new bicycle lines for the Municipality of Milan. In order to provide useful information that can be exploited in the proposal of new bicycle lines, an analysis of the available data was made, focusing on the historical bike-sharing usage and the position of existing bike-sharing stations. Therefore, the application will translate these data in newly proposed layers to be visualized through GeoViewer. In addition, those layers will be enriched with geo-referenced information about several points of interests, such as Restricted Traffic Areas, Parks, Pharmacies, public toilets and drinking fountains and so on. The user will be able to easily mix and compare several informational layers, including the ones representing the new cycle lines proposed by the application itself, facilitating the task of designing new cycle paths for the City of Milan.

Licence

The application is composed of two parts: a server back-end and a front-end: GeoViewer. The back-end is released under the terms of the GNU Affero General Public Licence (version 3). However, GeoViewer, the front-end, is based on a legacy software released by ESRI [8], and therefore cannot be licenced out the Municipality.

2.2.1 Scenarios

2.2.1.1 Users profiles:

There are no updates, since the user profiles were consolidated in D3.5 and are still valid.

2.2.1.2 Scenarios

There are no updates, since the scenarios for the application were consolidated in D3.5 and are still valid.

2.2.2 Requirements analysis

The requirement analysis, made after the updates in the application reported in this document, led to no updates in functional and non-functional requirements.

2.2.3 Available data sources

Table 9 reports the updated list of the available data sources for the application. Below is reported the rationale about these updates:

- Existing cycle network has been mapped to the Road data model, since this better fits shapes and metadata of the specific cycle paths.
- Points of Interest have been placed in three categories, considering the cyclists point of view: obstacles, facilities and generic interesting places. They have been retrieved from the Milan Open Data Portal [9] and no more from APIs, as reported in D3.5; then they have been mapped to the SynchroniCity data model (PointOfInterest) and finally imported in the SynchroniCity Platform as NGSI entities.
- Since the application now uses the street names that are provided natively by GeoViewer on the map, the entry for “Place/StreetNames”, reported in D3.5, was removed from the table.
- Traffic restricted and pedestrian areas have been mapped to the dedicated data model (RestrictedTrafficArea).
- The public transport network information has been mapped to their own dedicated data models (Urban Mobility). The free-floating bike trajectories, as well as the station-based ones, now refers to the historical data about the bike sharing usage, issued by the Milan’s mobility agency (Agenzia Mobilità Ambiente e Territorio - AMAT) [10] as static files and no more by APIs.

Table 9: Data sources used in Decision support system for cycle path planning application

Title	Description	Open Data / API	Adopted SynchroniCity Data Model	Notes
Cycle network	GeoJson description of current cycle-network in the city	Open data	Road	Data available in GeoJson format
Bike sharing Stations	Position and description of the stations where to pick bikes up	Open data	BikeHireDockingStation	Data available in GeoJson and CSV format
Parks	GeoJson description of parks in the city	Open data	Garden	Data available in GeoJson format
Restricted traffic and pedestrian areas	GeoJson description of traffic restricted areas and pedestrian only	Open data	RestrictedTrafficArea	Data available in GeoJson format
Point of interests	Descriptions and location of valuable places in an area as obstacles for cyclists (e.g. on street shops), or facilities for cyclists (eg. public toilets or drinking fountains), or useful places.	Open data	PointOfInterest	Data available in GeoJson format
Tramway and Railway network	GeoJson description of tramways and railways network	Open data	Urban Mobility	Data available in GeoJson format

Bike trajectories	Historical GPS position of free-floating bikes trajectories.	Static files (AMAT)	Data not available through SynchroniCity platform	Historical data regarding the free-floating bike sharing usage (updated to January 2019)
Bike Origin-destination	Historical data about the station-based bike sharing usage	Static files (AMAT)	Data not available through SynchroniCity platform	Historical data regarding the station-based bike sharing usage. (updated to January 2019)

2.2.4 Application architecture

This section describes the updated architecture of the Decision support system for the cycle path planning application. The application has been developed in NodeJS [11] and designed following a microservice approach. In particular, the Express [12] framework was used, to easily develop a reliable NodeJs Server application, composed by several NodeJs modules representing the concrete implementation of the components introduced below.

Overview

The application retrieves NGSI data from the SynchroniCity Context Management API and uses this information as input for its core components. Access to this API is secured by the IdM currently working on the WSO2 platform of Milan Municipality, which is compliant with OAuth2 standard, according to the SynchroniCity Security layer specification.

The architecture has been updated with minor changes in the components already defined in D3.5; in particular, the Additional Info Adapter was defined in order to manage all the other NGSI entities in a general way, as described below. In addition, the internal Scheduler was added within the Layer Manager component, since the application needs to periodically generate new layers to be ones provided by GeoViewer.

Figure 8 depicts the updated architecture of the application, reporting how its internal components communicate with each other, with the SynchroniCity framework and with the external component.

Application Components

The application is composed of the following set of internal modules, which are an updated version of the ones introduced in D3.5:

- **Adapters: Bike Sharing Station, Existing Cycle Network and Additional Info Adapter:** Each adapter converts the specific NGSI Context information, coming from the NGSI data sources, mapped to SynchroniCity data models and reported in Table 9, to the intermediate representation in GeoJson. Then, the GeoJson will be converted by the Information Layer Producer to the appropriate format, in order to be used by GeoViewer to show information to the end user of the application. The first two adapters convert existing NGSI entities for bike sharing stations and existing cycle paths. The last adapte, Additional Info Adapter, converts all the other generic NGSI entities, such as parks, restricted traffic Areas and POIs.
- **New Cycle Network Proposer:** This component is in charge of proposing new cycle routes as new informational layers. It will use and convert the data about the usage and the traffic of bike sharing network, retrieved from the external Origin–Destination Storage component. The Proposer will also exploit the information about the Bike Sharing Stations referenced from the Origin-Destination data. These Stations are provided by the Bike Sharing Station Adapter, which retrieves the corresponding NGSI entities from the Context Broker and

translates them to the specific format (GeoJSON) needed by the Proposer. The data processed by the Proposer is sent to the Routing Service, which will use this data to calculate a new possible cycle route that will be packed and issued by the Proposer as GeoJson to the Information Layer Producer.

- **Information Layer Producer:** This component is in charge of creating the appropriate format for the layers that will be issued to the Layer Manager, which will save them in its own Storage. This data will be used by the application to visualize the graphical information layers in the GeoViewer map. The component will take in input the GeoJson representation produced by either the particular adapter or the New Cycle Network Proposer and converts it to the Json format defined by Esri [8], as required by the GeoViewer.
It is important to underline that at first, as reported in D3.5, the chosen format for importing the informational layers was Shapefile⁶; further technical investigations have shown that the Esri Json format, equally supported by the GeoViewer, makes the conversion from the GeoJson representation simpler and more efficient. In addition, a layer in the ShapeFile format, once imported into GeoViewer, presented truncated attributes for the metadata. As anticipated, this component will receive the data produced by the New Cycle Network Proposer, containing proposals for new cycle lines/routes, which will be also represented as new informational layers.
- **Storage:** It will keep the Esri Json files representing the produced informational layers.
- **Layer Manager:** this component is in charge of managing the lifecycle of the stored layer representations and exposing them to the User Interface. The component also has an internal Scheduler that will launch periodically the process to generate and store newly proposed layers.
- **User Interface:** This component relies on GeoViewer. It enables the user to visualize a map, by toggling several information layers that can be overlapped and mixed together. GeoViewer will query the Layer Manager to read the layers to be loaded in the map; in addition, the user can draw and edit directly on the map, both textual labels and new geometric shapes. Therefore, this component is the user interface that enables the user to visualize new layers produced by the core of the application.

External Systems

The application interacts with the external Origin – Destination Storage component, which provides several datasets representing the historical data about the usage and the traffic of existing Bike Sharing networks. In detail, these datasets report all the Origin – Destination locations used by Bike Sharing users during the past years. The New Cycle Proposer will use and process this information, aggregated by location pairs and then sorted by the number of occurrences. The pairs with the greatest number of occurrences will highlight the need for new cycle paths in different areas.

⁶ <https://wiki.openstreetmap.org/wiki/Shapefiles>

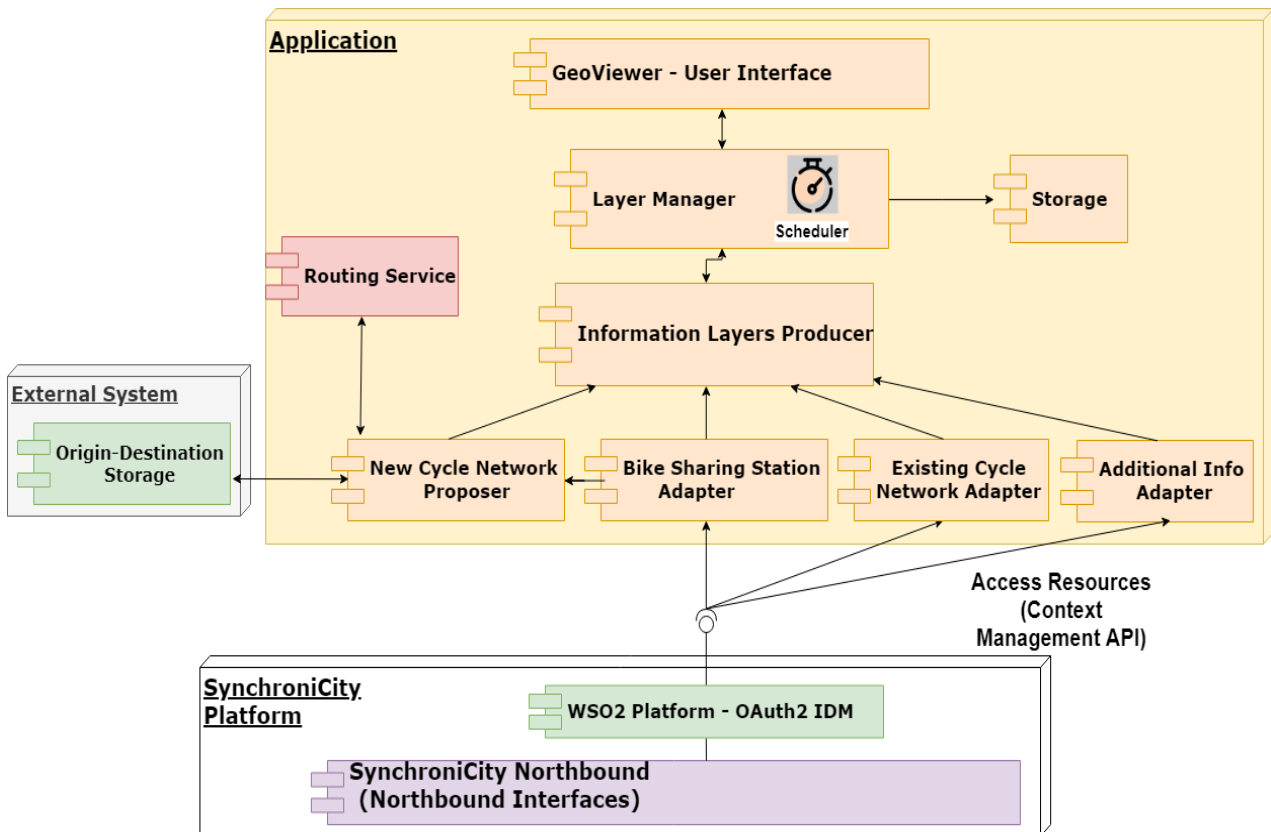


Figure 8: Milan - component diagram of Decision support system for cycle path planning application

Reusable Atomic Services

Table 10: Atomic Services adopted in Milan's Decision support system for cycle path planning application

Atomic Service	Rationale
Routing Service	The integration of this atomic service will provide the possibility to calculate new possible cycle paths within the city; in particular it will be tailored in order to provide as accurate as possible estimation of new cycle path according to constraints already existing, such as pedestrian areas in the city and public transport routes that could cross the proposed cycle paths.

2.2.5 Sequence diagrams

Among the functionalities offered by this application, the most relevant ones are represented by:

- Generate in a time-scheduled way, the new layers regarding newly proposed cycle paths.
- Select map layers in the GeoViewer.

The sequence diagrams that depict how the application provides this functionality are reported and described in this section:

- **Generate proposed cycle paths layers:** this sequence diagram depicts interactions among the components of the application, in which a scheduled job is launched in order to generate and store new layers representing the proposed cycle paths, which will be later loaded and visualized in the GeoViewer map (as described in the next point).
- **Select map layers:** this sequence diagram depicts interactions among the components of the application in order to enable the user to select and toggle among the available layers. The layers list can include the ones already existing and provided natively by the GeoViewer

and the ones proposed by the application itself, which are loaded in the GeoViewer by the means of the Layer Manager.

Generate proposed cycle path layers – Sequence Diagram

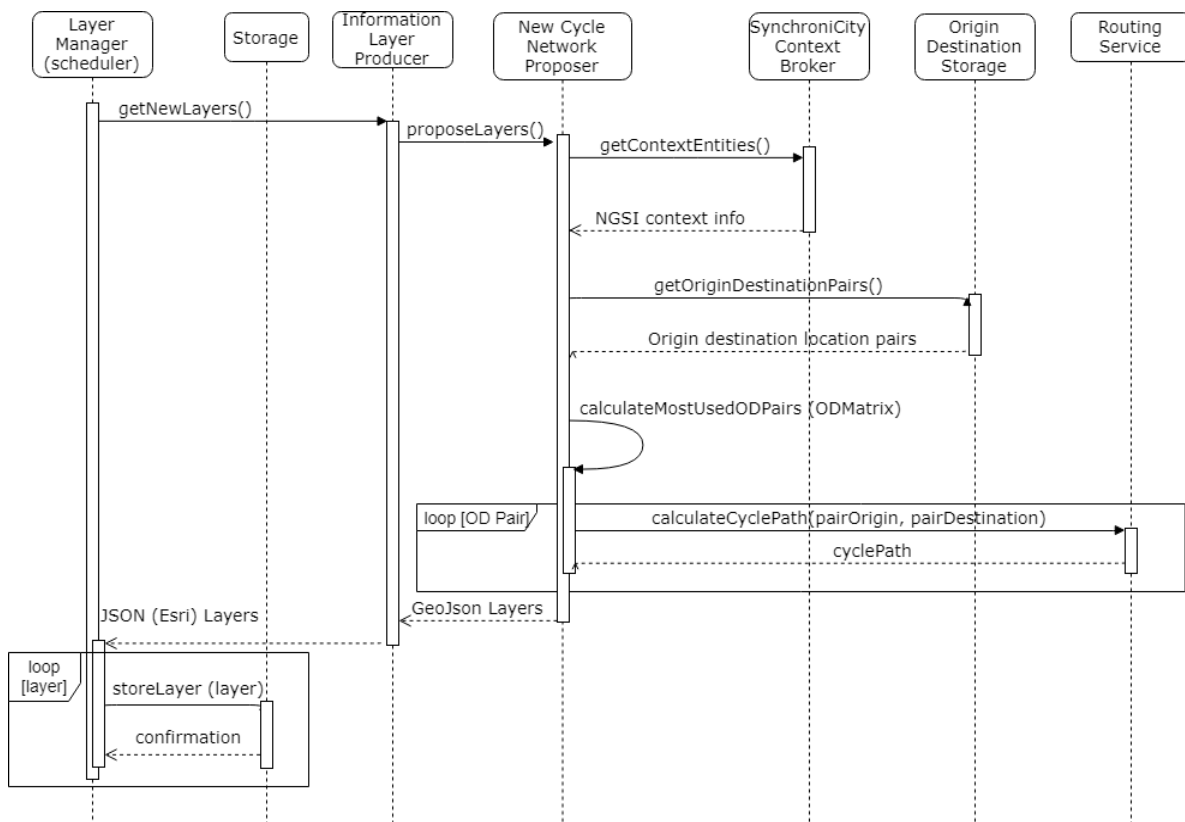


Figure 9: Milan - Propose new layers diagram of Decision support system for cycle path planning application

The Scheduler included in the Layer Manager starts the process to generate new proposed layers at a fixed periodic rate, sending a request to the Information Layer Producer, which in turn, sends the request to the New Cycle Network Proposer, in order to calculate new cycle paths. The Proposer sends a request to the SynchroniCity Framework's Context Broker, through the Context Management API, in order to retrieve all the needed NGSI Entities, such as existing cycle and public transport networks. Then, the Proposer sends a request to the Origin – Destination Storage, in order to retrieve the historical data needed for calculating the location pairs most used by Bike Sharing users. Then, it sends these origin-destination pairs (one request per pair) to the Routing Service, in order to calculate the geometric paths for each pair. These paths, representing the newly proposed cycle routes in GeoJson, are returned to the Proposer and then to the Producer. This translates the GeoJson geometries to the JSON layers representation, according to the Esri specification. The Producer returns the resulting collected JSON/Esri layers to the Layer Manager, which will send them to the Storage component, in order to be saved. Finally, the Storage component returns the success confirmation to the Layer Manager.

Select map layers – Sequence Diagram

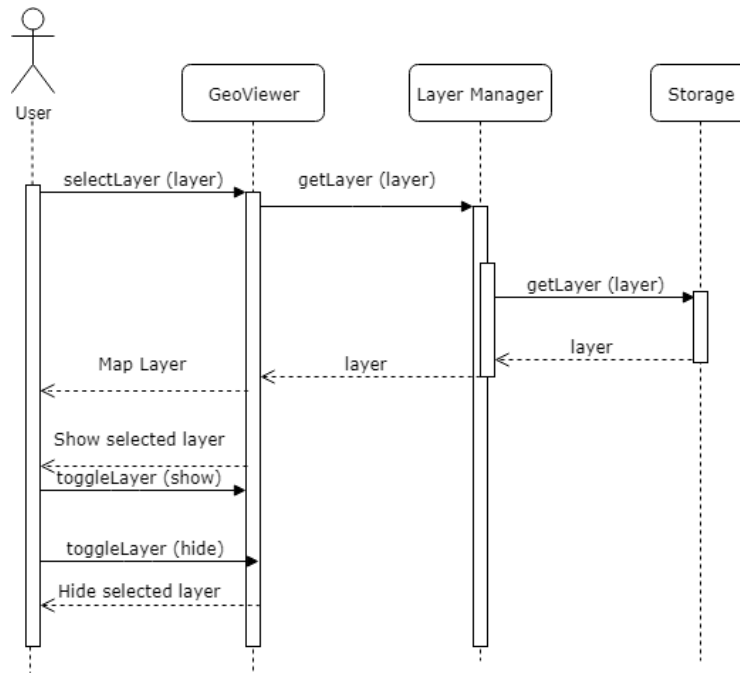


Figure 10: Milan – Select Map layer diagram of Decision support system for cycle path planning application

The User logged in GeoViewer selects a layer from the displayed list of the available ones. The GeoViewer sends the request to the Layer Manager. The Layer Manager forwards the request to the Storage component, which reads the requested layer. The returned layer is received by the Layer Manager, which returns it to the GeoViewer. GeoViewer shows the map with the selected layer. The user can show or hide a layer using the specific toggle button.

2.2.6 User Interface

The User interface of the application, as already described, consists of the GeoViewer map tools. It is intended to be used only by Milano Municipal's bike planners and is accessible only after user authentication.

Figure 11 shows the main web interface of the GeoViewer after the successful user authentication. The interface shows a map that the users can scroll, pan and zoom as they wish. Besides, the users have several buttons, widgets and pop-up menus from which they can access all the available functionalities such as: activate or deactivate information layers, explore the displayed map markers, select one or more of them and display their details, and so on. The users can also select the available layers from the list on the right and then can use widgets (the pop-up menu on the left) to draw different kinds of areas, and in case to record a note, placed on the map.

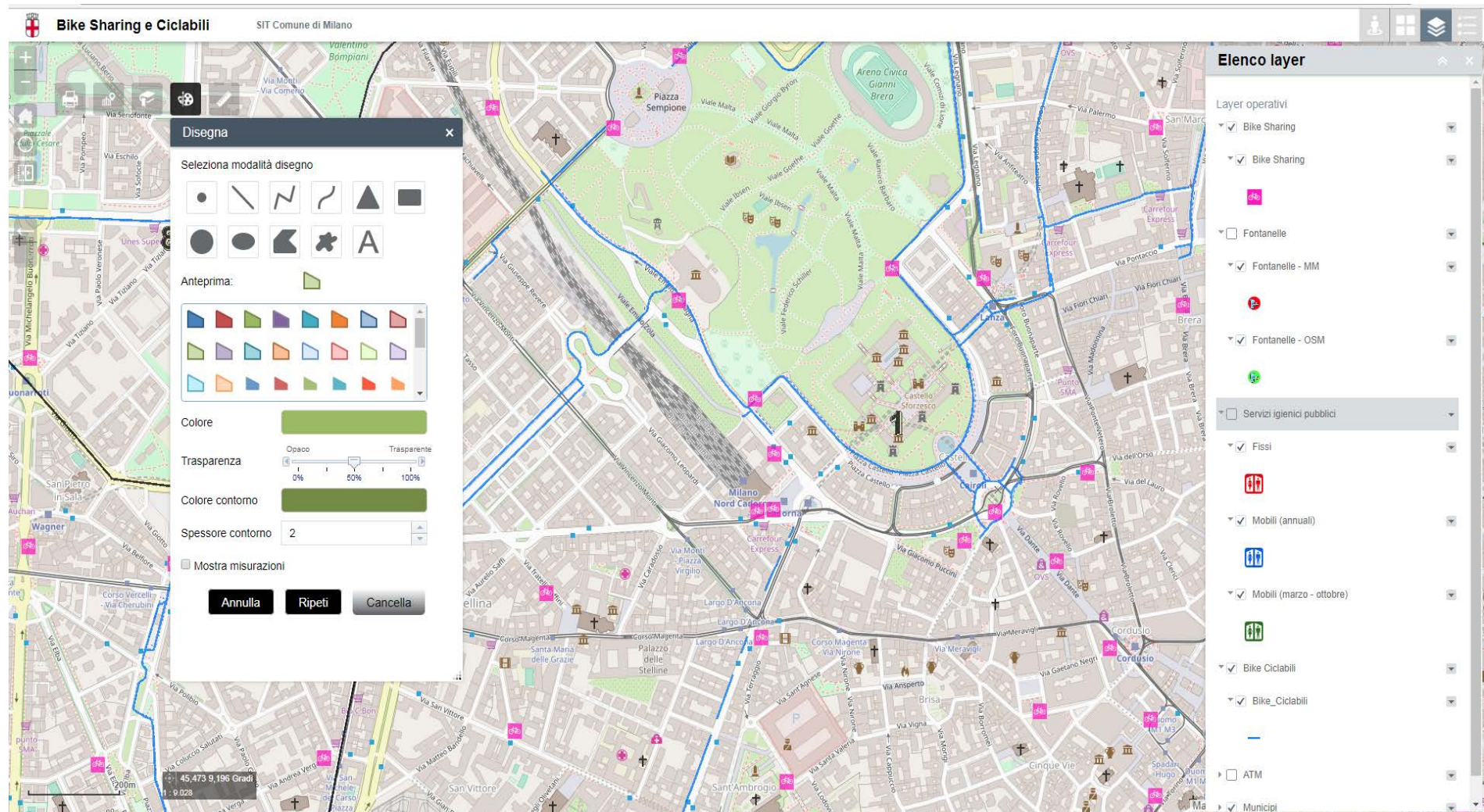


Figure 11: General web interface

Figure 12 shows an area selected by the user and the main attributes coming from the entities displayed in the area coming from the selected information layers.

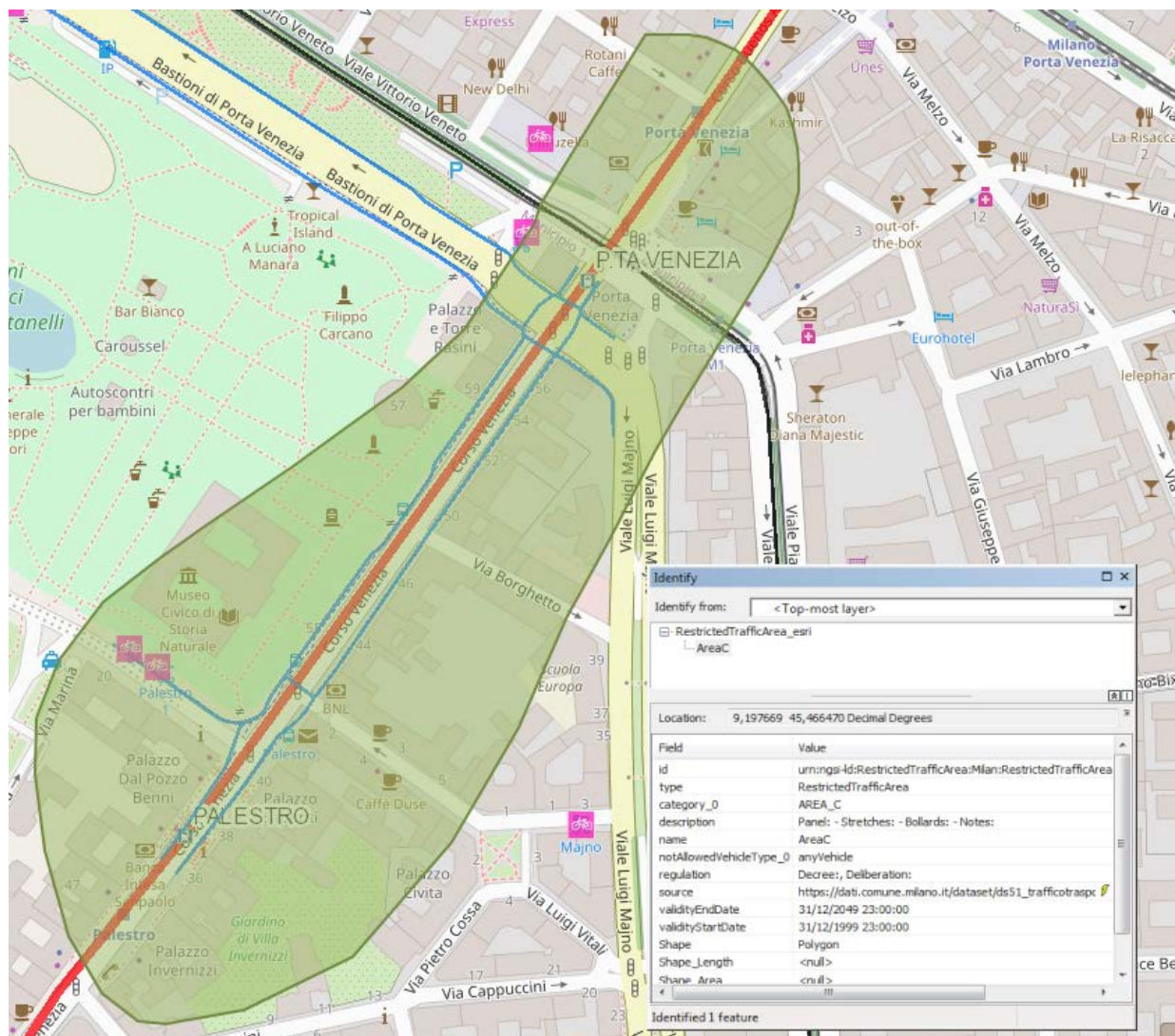


Figure 12: Web Interface: area selection

2.2.7 Data Management

All data involved in the application is anonymous, since it does not involve personal data or data that can be used to identify a person. The application is intended to be used for internal Milan Municipality purposes and since all data consumed and generated will remain internally there will be no other licensing issues. All data coming from Open Data are released under CC BYs⁷ or IODL⁸ licence. Bike Origin-destination information is closed data and it has been provided by AMAT only for this application. All data generated are intended for Municipality internal usage.

⁷ <https://creativecommons.org/licenses/by-sa/3.0/>

⁸ <https://www.dati.gov.it/content/italian-open-data-license-v20>

2.2.8 Conclusion

The Decision support system for cycle path planning application represents a concrete example of exploiting the SynchroniCity framework, by means of the provided NGSI context information and the use of the atomic services. Indeed, the legacy data sources issued by the Municipality of Milan, have been mapped to SynchroniCity Data Models and then loaded in the SynchroniCity Platform, enabling their full exploitation by providing a useful basic knowledge for the application. As described in the introduction, this knowledge has also supported and enabled the development of the application components, which are able to use in an integrated way both the NGSI information coming from the SynchroniCity platform and the historical data retrieved from external systems, for proposing new bicycle lines (e.g. historical data about the usage of bike sharing).

The user can easily compare and overlap several informational layers, including the ones representing the new cycle lines proposed by the application itself but also the enriched layers with additional information (e.g. POIs, Restricted Traffic Areas, etc.). The interactive tools and the intuitive graphical interface provided by the GeoViewer will support and facilitate the task of designing new cycle paths for the City of Milan.

2.3 Antwerp's Application "Bicycle patterns in the city of Antwerp"

There is a lot of traffic in cities these days. A lot of people ride their bikes everyday to work, congestion points arise, and it is important to pinpoint those to further streamline the in and outflow of the city. A lot of bicycle data can be gathered but it is hard to see and use it to produce useful information. This can be done by transforming plain bicycle data lines into a common NGSI data format that provides machine-readable information in a unified manner (e.g. the TrafficFlowObserved⁹ format). The objectives of this application are gaining insight into the traffic of bicycles on any given day and hour. This can be used to see congestion points and adjust the city infrastructure based on that. The background conditions needed for this application, are useful and consistent bicycle data which is delivered by tracking Ring-Ring¹⁰ bicycles. Ring-Ring is a company that allows you to transfer kilometers ridden on your bike into discounts at local stores. 22 million data lines were gathered which can produce valuable insights into traffic flow of the city. These data lines need to be transformed into TrafficFlowObserved and gathered into new data lines. Each trip was recorded and average speed on any given street segment can be calculated. The amount of vehicles at a given time and place is also taken into account to see the busy places and gain insight into routes people take when they go to work or when they go to do leisure activities in the weekend. Difficulties which will be faced are the sheer amount of data which needs to be transformed. The 22 million lines need to be processed and stored into buckets with a time interval of 15 minutes along with each street segment gathered by OpenStreetMap. This application will have a frontend which will visualise the data and insights can be gathered from there on. Separate vehicle data will also be visualised, this will represent the 22 million Ring-Ring lines, when plotted as a heatmap it will become clear where people like to ride their bikes and which places they try to avoid. The city can plan new infrastructure or infrastructure changes based on their findings of this data. This application does not make use of any Atomic Services.

Licence Closed Source

⁹ <https://fiware-datamodels.readthedocs.io/en/latest/Transportation/TrafficFlowObserved/doc/spec/index.html>

¹⁰ <https://ring-ring.nu/>

2.3.1 Scenarios

2.3.1.1 Users profiles:

Table 11: User profiles of Bicycle patterns in the city of Antwerp application

User profile ID	User description
Public user	Every user can use the web application to view data regarding bicycle usage in the city. Citizens can view information on bicycle count points, accidents, bicycle speed in the city, etc.
Mobility department User	All personnel from the mobility department have access to the IBM Cognos BI ¹¹ tool to have access to reports of the data. The user can gain insight in the data and use it to prove the needs of cyclists in the city, e.g. the influence of construction sites on the average cycling speed or number of accidents. Data for the mobility department will be available in the Cognos BI tool where only these users will have access to it.

2.3.1.2 Scenarios

There are no updates, since the users of the user profiles were consolidated in D3.5 and are still valid.

2.3.2 Requirements analysis

2.3.2.1 Functional requirements

In comparison to the original list of D3.5, 2 functional requirements have been left out, i.e.:

- FR-3: The application SHOULD have additional data sources, e.g., construction sites in the city, bicycle accidents in the city.
- FR-11: The application SHOULD visualise different bicycle accidents in the city.

The new list of functional requirements is listed in Table 12.

Table 12: Functional requirements of Bicycle patterns in the city of Antwerp application

ID	Description
FR-1	The application MAY be open for the public.
FR-2	Data regarding bicycle docking stations MUST be provided in real time.
FR-3	The application MUST provide improved average speed for road segments in the city.
FR-4	The application MUST provide improved average speed on street level.
FR-5	The application MUST be integrated with the SynchroniCity framework.
FR-6	The application SHOULD provide different overviews of the data sources.

¹¹

https://www.ibm.com/support/knowledgecenter/en/SSEP7J_10.1.1/com.ibm.swg.ba.cognos.wig_cr.10.1.1.doc/c_gtstd_c8_bi.html

FR-7	Users SHOULD be able to select a street segment.
FR-8	The application MUST visualise different data sources.
FR-9	The application SHOULD provide extended reporting.
FR-10	The data MUST be transformed and loaded in the marketplace.

2.3.2.2 Non-functional requirements

There are no updates, since the non-functional requirements were consolidated in D3.5 and are still valid.

2.3.3 Available data sources

There are no updates, since the data sources were consolidated in D3.5 and are still valid.

2.3.4 Application architecture

Overview

The application is an open web application for the public to gain insight of bicycle use in the city of Antwerp. The different components of the application ensure that a NGSI compliant data model is used and improved estimations on street segment level can be provided for external route planners. The data is open for the public and can be improved in the future with live coupling of GPS tracking data. A general overview of the architecture is reported in Figure 13.

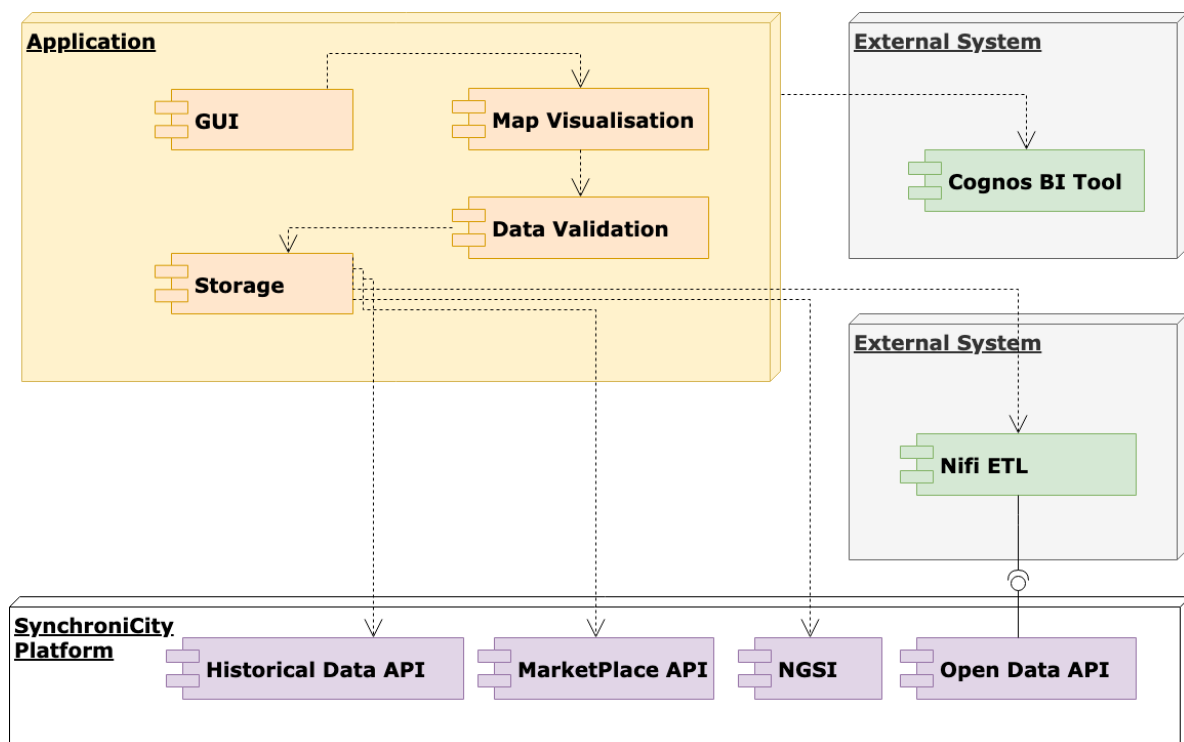


Figure 13: Overview of Antwerp application architecture for Bicycle patterns in the city of Antwerp

Application Components

- **ETL:** Extraction transform and load of the different data sources, which need to be transformed to NGSI models. This is done by using NiFi ETL¹². The GPS data will be transformed to Vehicle data and bike docking station will be coupled to the BikeHireDockingStation. After the data is transformed to the correct format it is stored and distributed through the market-place.
- **Data validation:** Data is validated on correctness, validated and filtered on what data can be shown to the public. After the validation the GPS data is transformed to the NGSI model TrafficFlowObserved, here the GPS data will be coupled to street segments. For every street segment the average speed is calculated on the basis of GPS data.
- **Coupling to street segment:** The data on street segment will give insight on the average speed of bicycles over time. This can be used to assess the impact of city policies or major construction sites in the city of Antwerp.
- **Map visualisation:** A map component is used to visualise the different data sources. Different views are available on the basis of the data. In addition, a GUI is setup to ensure that users can use functionalities like date selection in the web application. The main goal of the application is improved average speed on street segment level and visualisation of data for the mobility department. There were no tools for the mobility department available to visualise their data on bicycle count points or mobility applications.

External Systems

Cognos BI tool: The transformed and enriched data is coupled to the Cognos BI tool of the city of Antwerp. Here extended reporting can be done on the basis of the gained data and other datasets of the city of Antwerp.

Reusable Atomic Services

Currently the application does not make use of Atomic Services; despite this, we are investigating if some of its modules could become an Atomic Service; in particular, the "Map visualisation" component is a potential candidate Atomic Service.

2.3.5 Sequence diagrams

All visualisation is done using the GUI. It searches for data according to the filters set. The Orion database is called through the MarketPlace API and results are returned accordingly. These results are shown in the application so further analysis and investigation can be done.

User initiates the filtered search on the Datetimefilter GUI application, it then proceeds to do a filtered search to the underlying Orion database to get the filtered data and returns this to the application to be visualised like the images below. TrafficFlowObserved, Vehicle and BikeHireDockingStation data is visualised. See following sequence diagram.

¹² <https://nifi.apache.org/>

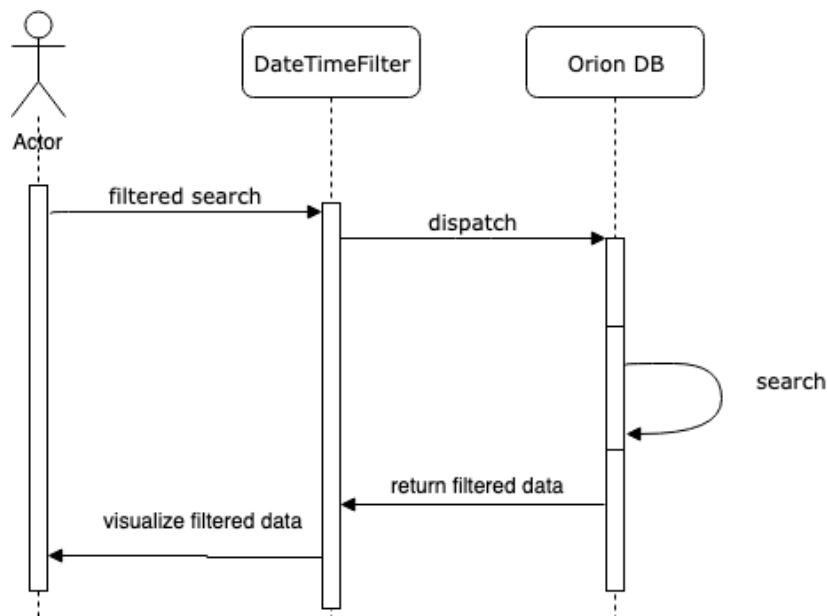


Figure 14: Antwerp - DateTimeFilter lookup sequence diagram of Bicycle patterns in the city of Antwerp application

2.3.6 User Interface

The UI has a filter where users can specify the start and end date they want to search on, along with a day filter so they can select the day and a quarter hour filter which enables them to specify up to 15 minutes to filter the data on. It is also possible to search for Vehicle, TrafficFlowObserved and/or BikeHireDockingStation data (Figure 15, Figure 16 and Figure 17 respectively)..

The Vehicle data is plotted as a heatmap to visualize the busy spots at a specified time and date. The TrafficFlowObserved data is plotted as lines which has visual indicators to represent the data on each line. For example, the colour helps visualize the average speed on a particular road segment and the thickness of the lines represents the intensity at that given time and date. The BikeHireDockingStations are visualized through icons dispersed throughout Antwerp.

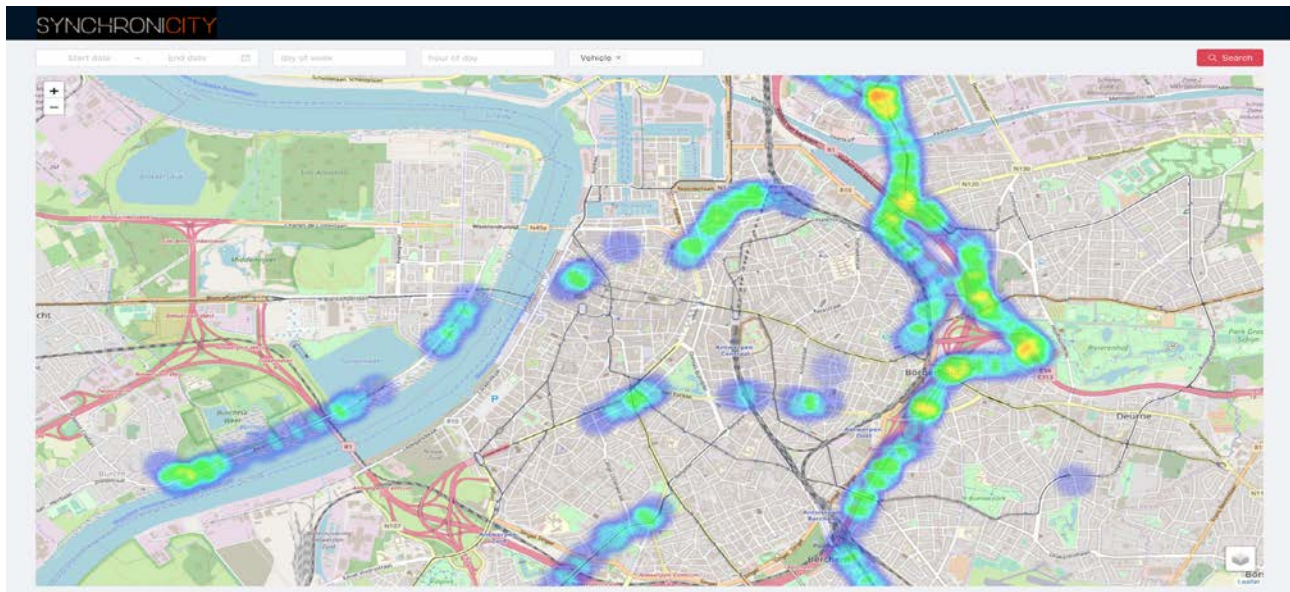


Figure 15: Antwerp - DateTimeFilter Vehicle data

Figure 15 displays the heatmap of the Vehicle data. It shows where congestion points are and how Vehicles flow throughout the city. It is clearly visible that the edge of the city is most affected. Note that this is done on a small subset of the data.

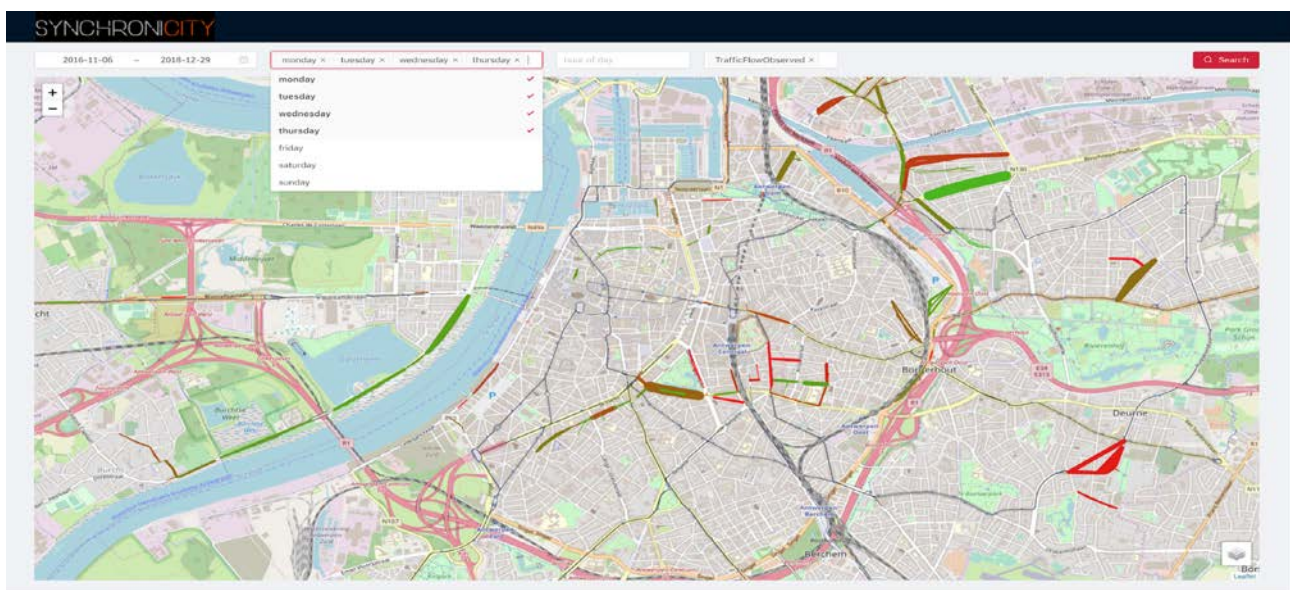


Figure 16: Antwerp - DateTimeFilter TrafficFlowObserved

TrafficFlowObserved is visualized in Figure 16. Thickness represents the amount of vehicles passing and the colour resembles the average speed. Note that this is done on a small subset of the data.

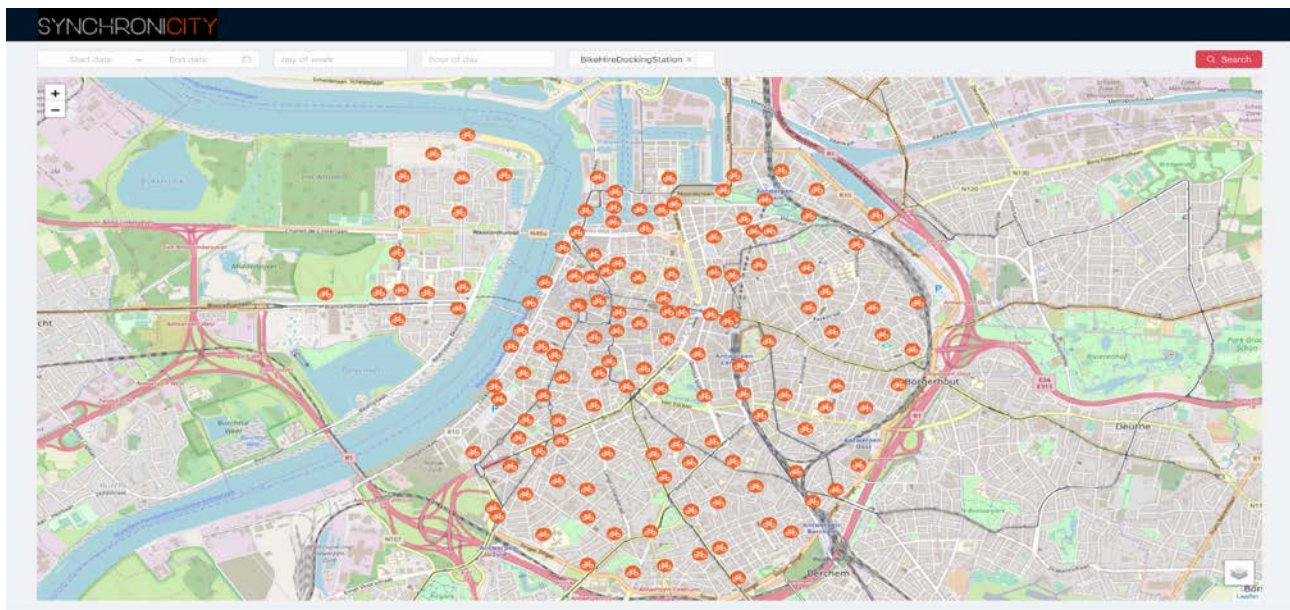


Figure 17: Antwerp - DateTimeFilter BikeHireDockingStation

All BikeHireDockingStation entities are visualised in Figure 17. It shows all available locations for hiring and depositing Velo bicycles.

2.3.7 Data Management

Ring-Ring data is collected and has a tripId. In no way user information can be linked to this tripId. This is compliant with GDPR in any way.

2.3.8 Conclusion

The transformation from Ring-Ring data to TrafficFlowObserved was more computationally expensive than initially anticipated. It needed special care to be run in a decent amount of time. Ingestion into Nifi of this data was not as smooth as we'd like. Filtering on a small subset of the final TrafficFlowObserved data revealed interesting insights into traffic at specific time intervals. The Vehicle data showed density of vehicles and BikeHireDockingStation revealed all possible bike docking stations. Development slowed down due to Nifi issues but nevertheless we were able to gain insight into the 22 million Ring-Ring lines. We used the Marketplace API to contact the database for filtering which uses NGSI models of the Ring-Ring data.

3 Application Theme: Multimodal transportation

One of the main challenges of modern cities is urban mobility. The objective of this application theme is to find solutions and approaches to improve mobility in the cities taking into account the different needs and behaviours of people, such as citizens and visitors. Appropriate tools could encourage the use of public transportation instead of the private one and to make urban transportation in general more sustainable and easier to use. These tools will support citizens and visitors to take advantages of different means of transportation and of the facilities offered by a city.

Six RZs participate in this application theme; in addition to the five reported in D3.5 (Porto, Santander, Helsinki, Milan and Carouge) and Seongnam joined this application theme. All these RZs are facing problems related to mobility in the city area and they are looking for solutions to solve their specific needs. In particular:

- **Porto** is dealing with the development of a comprehensive journey planner that leverage different means of transportations (e.g.: public transportation, vehicle-sharing), information and event (e.g.: noise, air pollution, weather forecasts, traffic jams) that can impact on users' behaviours and on how people use them. (Section 3.1).
- **Santander** aims to improve the sustainable mobility within the City, taking advantage of new mobility premises (such as escalators and lifts that have been deployed) and parking information, in order to encourage the use of public transportation instead of private cars. (Section 3.2)
- **Helsinki's** goal is to provide a lower threshold for choosing "lighter" traffic alternatives for moving around the city. Provision of walking and cycling route with good air quality for a more enjoyable cycling experience in the city is the aim of Helsinki's application. (Section 3.3)
- **Milan** is aiming to realize a multimodal navigator to help disabled people moving around the city providing useful information currently available in other navigators, such as restricted traffic areas in the city that disabled people in Milan are allowed to traverse. (Section 3.4)
- **Carouge** is dealing with limited number of parking spots in the city area, and its aim is to reduce the time for finding free parking spots. To reach this result the application will provide adequate information to drivers such as real-time information about the status of parking spots and short-term predictions about their availability. (Section 3.5)
- **Seongnam** aims to support commuters to easily find a parking space and the commute route in order to minimizing the transfer time. Drivers will be able to check availability of parking spots in real-time and to check information about connected public transportation; such as the estimated arrival time of metro line, the time schedule of intercity and express bus and intracity bus. (Section 3.6)

3.1 Porto's Application "Porto Multimodal Assistant"

Over the last few years, Porto has been affirmed as an excellent touristic destination. This has a huge impact on the way people move around the city. Upfront, more people means more traffic, more pollution, more incidents and accidents, more places requiring interventions, etc.

These problems are being solved with different approaches, and following the SynchroniCity's goals, Porto decided to make available a Multimodal Assistant application based on its Urban Platform (FIWARE based platform).

Although Porto has already a multimodal navigator¹³ which provides mobility information about all the public transportation network of the Metropolitan Area of Porto (AMP), it lacks several features and is not very user friendly.

¹³ MOVE-ME.AMP, available as a web app (<http://move-me.mobi>), and as an app for Android OS and iOS.

The **Porto Multimodal Assistant** application will gather city and user data, and thus provide a personalised and customised multimodal assistant, based on the user's preferences, requirements and choices.

The main city data sources that are expected to be handled by this application are: public transportation network (schedules, stops and stations, routes and lines, etc.); POIs and geographical data (highway, city roads and streets, sidewalk, etc.). Some of this data will be static (from a database), other will be provided in real-time (from sensors).

The route planner will enable searches by different and complementary parameters, namely, single or multi-point destinations, POIs names, street names, different transportation modes and also based on personal choice criteria (such as fast route, avoid transfers and avoid walking), etc.

The routes and complementary information (such as POIs location) will be displayed on a map.

Apart from the route planner features, the application will also provide other transportation information, such as, timetables at stops and stations, and nearby stops and stations. Besides, the user will also have access to complementary information (not related to the transportation needs), such as POIs and geographic information.

The use of this app will also provide valuable mobility data to the city authorities and public transportation operators (anonymized data), which will enable all the mobility stakeholders, in particular the City Council, to have better informed decisions and policy making in the areas of mobility, environment, urbanism and tourism.

Licence

Code application will be made available and released under the European Union Public Licence (EUPL), as stated in the application's source code¹⁴. At this time, it is not defined if the code implemented by Porto (which will add extra functionalities) will be merged with Digitransit's code base/master branch¹⁵. In either case, Porto will make the code available under the same licence.

3.1.1 Scenarios

Compared to D3.5 user profiles and scenarios of **Porto Multimodal Assistant** application have not been modified; they are reported in this section for reasons of clarity and completeness.

3.1.1.1 Users profiles:

Table 13: User profiles of Porto Multimodal Assistant application

User profile ID	User description
User Porto 1 (Commuter)	This user is a local inhabitant (lives and works in the Metropolitan Area of Porto) and daily commutes between home and job locations twice a day (at the beginning of the day in one direction and at the end of the day in the opposite direction), using the same schedules every day. This user has very limited time flexibility and is mainly familiar with one route (same departure and arrival destinations) and a few schedules. He/she uses the same transportation mode(s) every day, checks-in on the transportation network twice a day, and uses a monthly subscription ticket. This user is mainly interested in speed/total time of transportation and on-time arrivals and departures. The more relevant information he/she requires is real-time information about traffic constraints, estimated time to arrival (ETA) and departure time.
User Porto 2 (Expert)	This user is a local inhabitant (lives, works and has other regular activities in the Metropolitan Area of Porto), daily commutes between home and job locations,

¹⁴ <https://github.com/HSLdevcom/digitransit-ui/blob/master/LICENSE-EUPL.txt>

¹⁵ <https://github.com/HSLdevcom/digitransit-ui>

	and also daily uses public transportation for other activities (before, after or in between the two locations). He/she uses different schedules every day and along the day, has semi-limited time flexibility and uses a monthly subscription ticket. This user is totally familiar with the global multimodal transportation network: uses different routes, schedules and transportation modes; uses different departure and arrival destinations along the day and week; and checks-in on the transportation network several times a day. This user is mainly interested in flexibility (multimodal system) and on-time arrivals and departures. The more relevant information he/she requires is real-time information about traffic constraints, estimated time to arrival (ETA) and departure time.
User Porto 3 (<i>Sporadic</i>)	This user is a local inhabitant (lives and works in the Metropolitan Area of Porto) and has a sporadic use of the public transportation network. He/she uses a pre-charged occasional ticket to travel along non-frequent or planned schedules, and has semi-limited time flexibility. He/she uses the public transportation system for a particular need or due to a temporary restriction (for example, the car is broken); uses specific routes and transportation modes (for each use); and is not very familiar with the ticketing and transportation modes options. This user is mainly interested in transportation availability and clear information. The more relevant information he/she requires is bus/station location, route map and route options, and real-time information about estimated time to arrival (ETA) and departure time.
User Porto 4 (<i>Outsider</i>)	This user lives and works outside the Metropolitan Area of Porto (visitor) or abroad (tourist). He/she has a very sporadic use, and uses the public transportation network to travel to predefined locations (hotel, main POIs, airport, etc.). He/she uses non-frequent or planned schedules and has significant time flexibility, and uses a tourist ticket or a pre-charged occasional ticket. This user is not familiar with ticketing options, transportation modes and the global multimodal transportation network; uses different routes, schedules and transportation modes; uses different departure and arrival destinations along the day and stay period; checks-in on the transportation network several times a day. This user is mainly interested in transportation availability, and clear and multilingual information (in particular, in English language). The more relevant information he/she requires is stop/station location, route map and real-time information about departure time.

3.1.1.2 Scenarios:

Table 14: Scenario "Estimated time of arrival (ETA) and schedules" of Porto Multimodal Assistant application

Scenario title	Estimated time of arrival (ETA) and schedules
User profile	User Porto 1
Background	Maria lives and works in the Metropolitan Area of Porto and she daily commutes between home and job locations. Usually, apart from the bus schedules tables available at each bus stop, she doesn't have real-time information regarding the ETA and/or departure time.
Objective	Maria wants to know if the bus she needs to take is on-time and/or has already departed. In the latter case, she wants to know when she can pick up the next one.
Storyline	As a daily commuter, Maria has routine schedules. In particular, she normally picks up the same 999 bus that will take her back home at the end of the day: she arrives at the bus stop around 18:20, and the bus leaves around 18:30. Sometimes, she arrives later to the bus stop and/or the bus departs sooner than expected. When she arrives at the bus stop some minutes later than usual, Maria is not sure if the bus has already departed or not. Then, Maria uses the Multimodal Assistant app to check the ETA of the next 999 bus. She opens the

	app, which automatically detects her location by GPS (in particular, the bus stop). The app provides her information about the available buses at that particular bus stop, and then she selects the 999 route. The app provides her information about the ETA (real-time data) of the next 999 bus, and the scheduled departure times of the next ones. Maria finds out she has just missed the bus by a few minutes, and the ETA of the next 999 bus is 25 minutes. Instead of waiting at the bus stop for almost half an hour, she takes the opportunity to go the nearby grocery and buy a few supplies.
--	--

Table 15: Scenario "Real-time mobility information" of Porto Multimodal Assistant application

Scenario title	Real-time mobility information
User profile	User Porto 2
Background	João lives, works and has other regular activities in the Metropolitan Area of Porto. He commutes daily between home and job locations, and also daily uses public transportation for other non-routine activities (for example, go to the supermarket, pick-up the kids, meet with friends downtown or play football). Once João uses different schedules every day and along the day, he needs to have real-time information regarding the ETA and/or departure time of different transportation modes, at different locations and different times.
Objective	João wants to know when and where he can get the most adequate transportation mode(s) that allow him to reach the desired destinations.
Storyline	João works in the city centre, but is going to have lunch with a few friends close to the beach. Before leaving the office, he uses the Multimodal Assistant app to plan his route between the office and the restaurant: he opens the app, which automatically detects his location by GPS, and then João inserts the desired arrival location. The app provides him optimized route suggestions, based on the departure/arrival destinations, his actual location, and informs him: where he must pick up the bus (bus stop location) and the ETA (real-time data) of the next bus. After lunch, he decides to use the app to plan his return to the office. Instead, he chooses another route, which will make him walk a little more, but will take him to a Metro station. The app provides him ETA of the next Metro train. He picks up the train, and reaches his destination without any delays, because the Metro train is not affected by traffic.

Table 16: Scenario "Route planning by departure/arrival locations" of Porto Multimodal Assistant application

Scenario title	Route planning by departure/arrival locations
User profile	User Porto 3
Background	Ana lives and works in the Metropolitan Area of Porto but has a sporadic use of the public transportation network, because she only uses it for a particular need or due to a temporary constraint. Usually, apart from the bus numbers and bus schedules tables available at each bus stop, she doesn't have real-time information regarding the ETA and/or departure time.
Objective	Ana wants to find out where she can pick up the nearest bus and if the bus she needs to take is on-time and/or has already departed. In the latter case, she wants to know when and where she can pick up the next one.
Storyline	Ana is driving to the work office, but she suddenly hears a weird noise in her car. She decides to go to a nearby auto repair shop to check the problem, which detects a major break failure. Accordingly, she is strongly advised to leave the car to be repaired, which she does. This is the first time she has been at this shop, so she has no idea how to go from here to the office. Before leaving the shop, she uses the Multimodal Assistant app to plan her route between the shop and the office: she opens the app, which automatically detects her location by GPS, and then Ana inserts the desired arrival location. The app provides her the

	only available route, based on the departure/arrival destinations, and informs her: where she must pick up the bus (bus stop location) and the ETA (real-time data) of the next bus; where she must leave out of the bus and pick-up a Metro train and the ETA (real-time data) of the next train. She picks up the bus, then leaves it and takes the Metro. She finally leaves the Metro station and then walks to the office.
--	---

Table 17: Scenario "Route planning by POI" of Porto Multimodal Assistant application

Scenario title	Route planning by POI
User profile	User Porto 4
Background	Alex is a British tourist visiting Porto for the first time and for a few days. He only knows the name and address of the hotel where he is staying, and he has just arrived at the city's airport.
Objective	Alex wants to know how to reach his hotel and after that, how to go from the hotel to the city's Cathedral, which are both located in the city and historical centre, respectively.
Storyline	Alex has just arrived at Porto's airport, which is actually located in the nearby city of Maia. He wants to know how to reach his hotel, which is located in Porto's city centre. Alex uses the Multimodal Assistant app (which is also available in English language) in order to plan his route: he opens the app, which automatically detects his location by GPS, and then Alex inserts the desired arrival location: "Allies Hotel". The app database comprises several POIs, including hotels, and provides him with the most appropriate route, based on the departure/arrival destinations, and informs him: where he must pick-up the Metro train and the ETA (real-time data) of the next train. He picks up the Metro and leaves at the final Metro station ("Aliados"). He then leaves and walks to his hotel, where he checks-in and leaves his bag in the room. After that, he wants to find out how to reach the city's Cathedral. He uses the Multimodal Assistant app in order to plan his route: he opens the app, which automatically detects his location by GPS, but since he is in the room (indoor), he has to provide his departure location ("Allies Hotel"); then he provides the desired arrival location ("Cathedral Porto"). He then realizes that he is very close to the Cathedral, because the two route options provided by the app are to take the Metro for a 5 minute travel distance or to walk for 15 minutes. He decides to walk to the Cathedral, and the app provides him the walking route and his real-time location on a map. Finally, he reaches the Cathedral, and feels amazed by its beauty and by the unique view of the city from its top balcony!

3.1.2 Requirements analysis

3.1.2.1 Functional requirements

This section report the final requirements of **Porto Multimodal Assistant** application. Compared to D3.5 an important revision has been performed; this revision led to a more realistic and practical identification of requirements, both functional and non-functional.

Table 18: Functional requirements of Porto Multimodal Assistant application

ID	Description
FR-1	The application MUST enable the creation of a user account
FR-2	The application MUST enable the user to set mobility preferences and requirements
FR-3	The application MUST enable saving user's mobility historical
FR-4	The application MUST provide user's behaviour indicators (based on mobility historical) and comparison with 'average' user
FR-5	The application MUST provide POIs for a given location, in particular, near the start and end locations
FR-6	The application MUST provide geographical info (street name) and mobility info (nearby lines, stops and stations) for a given location
FR-7	The application MUST provide bus, metro/tram and train timetables at the nearby stops and stations
FR-8	The application MUST enable single or multi-point routes
FR-9	The application MUST enable route search and planning by street name
FR-10	The application MUST enable route search and planning by POIs name
FR-11	The application MUST enable route choice based on the use of different transportation modes, such as, buses, trains, metro/tram, taxis, etc. (when available)
FR-12	The application MUST enable route choice based on several criteria, such as, faster, minimum distance, minimum number of transfers, etc.
FR-13	The application SHOULD provide customized route suggestions based on user's preferences and requirements
FR-14	The application MUST provide a map visualization for all the route suggestions
FR-15	The application MUST take into consideration scheduled timetables of public transportation modes on the route search and planning
FR-16	The application MUST display the user's real-time location on a map (if user's GPS is enabled)
FR-17	The application MUST provide estimated time of arrival (ETA) based on the route and real-time location
FR-18	The application MUST provide a trip summary (distance travelled, time, calories burned, CO2 emissions, cost, etc.) at the end of the trip
FR-19	The application MUST open an info bubble when a marker is clicked

3.1.2.2 Non-functional requirements

Table 19: Non-functional requirements of Porto Multimodal Assistant application

ID	Description
NFR-1	The application MUST support Portuguese (Portugal) language
NFR-2	The application MUST support English (UK) language
NFR-3	The application MUST provide the terms and conditions of use
NFR-4	The application MUST provide the data privacy policy
NFR-5	The application MUST have a web app version
NFR-6	The web app version of the application MUST be responsive
NFR-7	The application MUST provide a user's assistance service by email or contact form

3.1.3 Available data sources

This section report the final data sources that will be used in the **Porto Multimodal Assistant** application. Compared to D3.5, a huge revision of data sources has been performed; some of them have been removed because no more available, other because no longer needed to implement functionalities it eh application.

Table 20: Data sources used in Porto Multimodal Assistant application

Title	Description	Open Data/API	Adopted SynchroniCity Data Model	Notes
POIs	Data about Points of interest (POIs)	Open data	PointOfInterest	
Public transportation	Public transportation data (schedules, routes, lines, stations/stops, etc.)	Open data	GTFSTransitFeedFile	

3.1.4 Application architecture

Porto Multimodal Assistant is an application built with Digitransit platform¹⁶. Digitransit is an open source journey planning solution that combines several open source components into a modern, highly available route planning service. Route planning algorithms and APIs are provided by OpenTripPlanner¹⁷ (OTP).

Overview

OTP is a great solution for general route planning, but in order to provide top-notch journey planning other components such as mobile friendly user interface, Map tile serving, Geocoding, and various data conversion tools are needed. Digitransit platform provides these tools, as illustrated in the following figure, which also depicts the matching layers between the overall architecture of Porto's SynchroniCity infrastructure and the multimodal transportation application one.

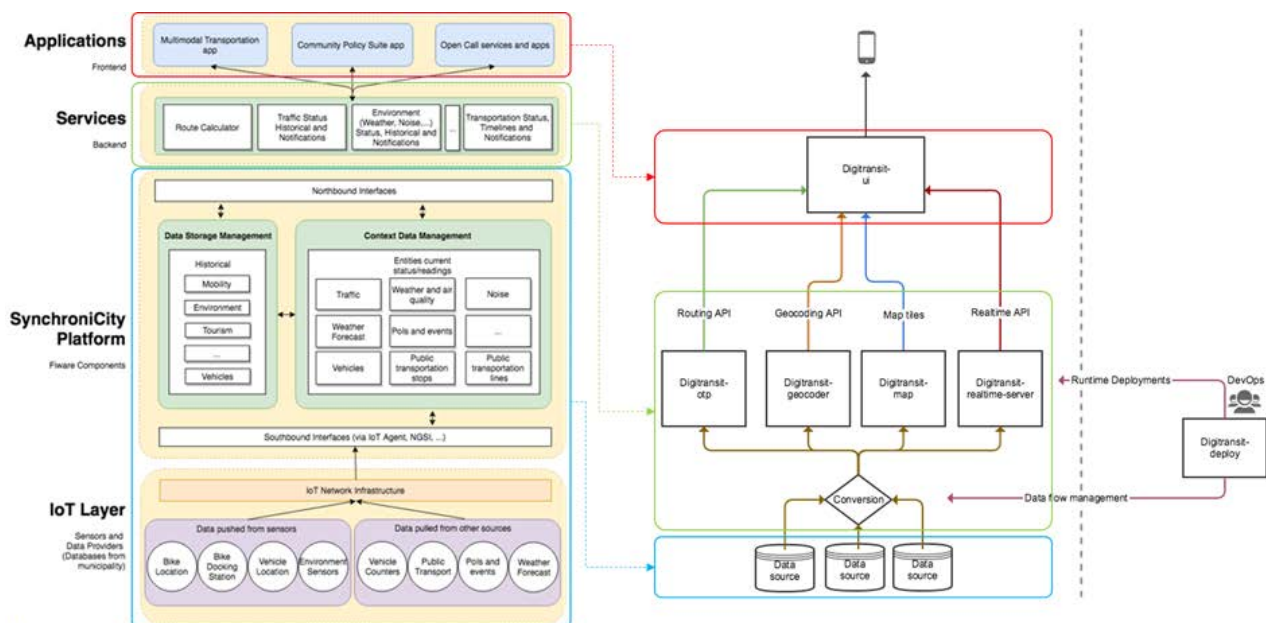


Figure 18: Matching layers between the overall architecture of Porto's SynchroniCity framework (left) and the MMT application architecture (right, © digitransit)

From bottom-up, the layers are the following:

- Bottom (blue rectangle): Data sources for the services supporting the application, and Porto's SynchroniCity infrastructure (namely all the FIWARE components, sensors and other city's databases).

¹⁶ <https://digitransit.fi/en>

¹⁷ <http://www.opentripplanner.org>

- Intermediate (green rectangle): Support services. Note the “conversion” process between this and the layer below. Given that these services are not really consuming data directly from a NGSI interface, this data conversion is required. The reason why this process is included in the green rectangle is because, most probably, it will be handled by a service (although it is not an Atomic Service).
- Top (red rectangle): Application layer, where the multimodal transportation application will sit.

Application Components

The application is basically composed by one component handling the User interface, and a set of underlying services, some of which are atomic services, described later in this chapter.

- **Digitransit (user interface)**

As stated above, and as described in D3.5, the application itself is only one component built with React¹⁸, a JavaScript library for building user interfaces; following an application architecture (for building client-side web applications) named Flux¹⁹ React components (responsible for UI and I/O with the user) query underlying services as follows:

- using Relay²⁰ (a JavaScript framework for building data-driven React applications) for querying the OpenTripPlanner service, through the Routing API²¹ (which supports GraphQL)
- with Fluxibe²² (a pluggable container for universal Flux applications) for querying other service's API (Realtime HSL²³, Map²⁴, Geocoding²⁵ and eventually²⁶ the Traffic Flow Estimator).

As mentioned, querying servers through APIs other than GraphQL, is done through a Flux model, implemented with Fluxibe. In this model, the view component (React components), using an action creator method, invokes the dispatcher component (Fluxibe Dispatcher), which will then trigger an action. In Porto Multimodal Assistant application, this action is responsible for querying the service and send the result in a store component. The result of the queries is delivered back to the React components through 'notifications'.

- **Geocoder**

This service, based on Pelias²⁷, provides a way to perform address searches and address lookups (also known as reverse geocoding). Internally, Pelias uses an Elasticsearch²⁸ instance, for fast index and search data in a nosql structure.

- **User Management Handler**

This service is based on Firebase²⁹ (a Google Mobile platform that helps to quickly develop high-quality apps), and is used for managing and handling the user authentication process, and storage user's data. Besides its also used to do some of the computational work, needed to calculate averages and other measurements.

¹⁸ <https://reactjs.org>

¹⁹ <https://facebook.github.io/flux>

²⁰ <https://github.com/facebook/relay>

²¹ <https://digitransit.fi/en/developers/apis/1-routing-api>

²² <https://github.com/yahoo/fluxible>

²³ <https://digitransit.fi/en/developers/apis/4-realtime-api>

²⁴ <https://digitransit.fi/en/developers/apis/3-map-api>

²⁵ <https://digitransit.fi/en/developers/apis/2-geocoding-api>

²⁶ The usage for this service is still under discussion.

²⁷ <https://pelias.io>

²⁸ <https://www.elastic.co/>

²⁹ <https://firebase.google.com/>

External Systems

The only external system that the multimodal application relies on, is the one responsible for map visualization, based on OpenStreetMap³⁰.

Figure 19 shows the interaction between all the components (internal and external), including the atomic services, described in the next section.

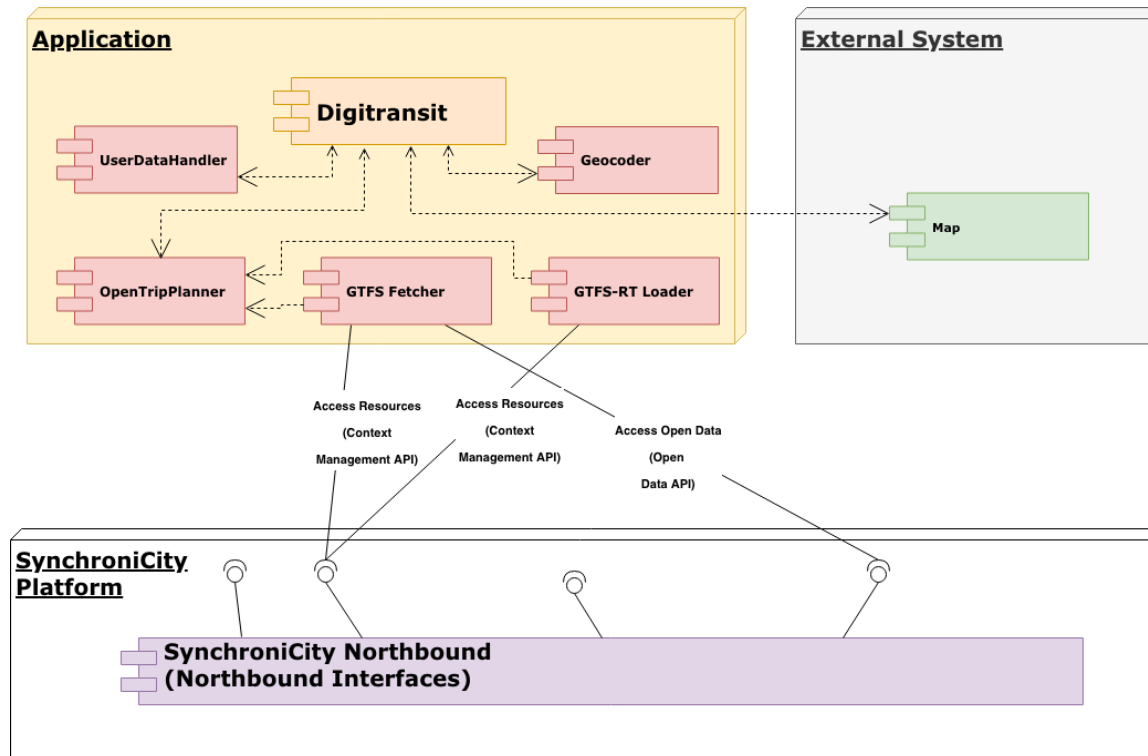


Figure 19: Application component diagram of Porto Multimodal Assistant application

Reusable Atomic services

Table 21 lists and provides the rationale for the adoption of the Atomic Services in Porto Multimodal Assistant application. The reader might refer to [2] for a deeper description of each of them.

Table 21: Atomic Services adopted in Porto Multimodal Assistant application

Atomic Service	Rationale
Routing Service	Based on OpenTripPlanner it finds suitable routes combining taxi stops, buses info and bicycle routes (among other available possibilities) between two points and according to user's preferences.
GTFS Fetcher	Extracts data from GTFSTransitFeedFile entities ³¹ and imports GTFS uploaded files (static timetables and related info) into the OpenTripPlanner platform, making this available for routing calculation (within the Routing Service).
GTFS-RT loader from NGSi	This service consumes Real Time Urban Transport entities (so far, ArrivalEstimation) and generates GTFS-RT feeds from it. These

³⁰ <https://www.openstreetmap.org>

³¹ See deliverable D2.3 (Catalogue of OASC Shared Data Models for Smart City domains).

	feeds will be later consumed by the Routing Service to calculate/update the requested routes.
--	---

3.1.5 Sequence diagrams

Among the functionalities offered by this application, the more relevant one is represented by the possibility to search for a route in the city, with different means of transportation. The sequence diagram that depicts how the application provides this functionality is reported and described in this section.

- **User Login and route search:** this sequence diagram depicts interactions among the components of the application in order to calculate and provide the end user with the shortest path between start and end points

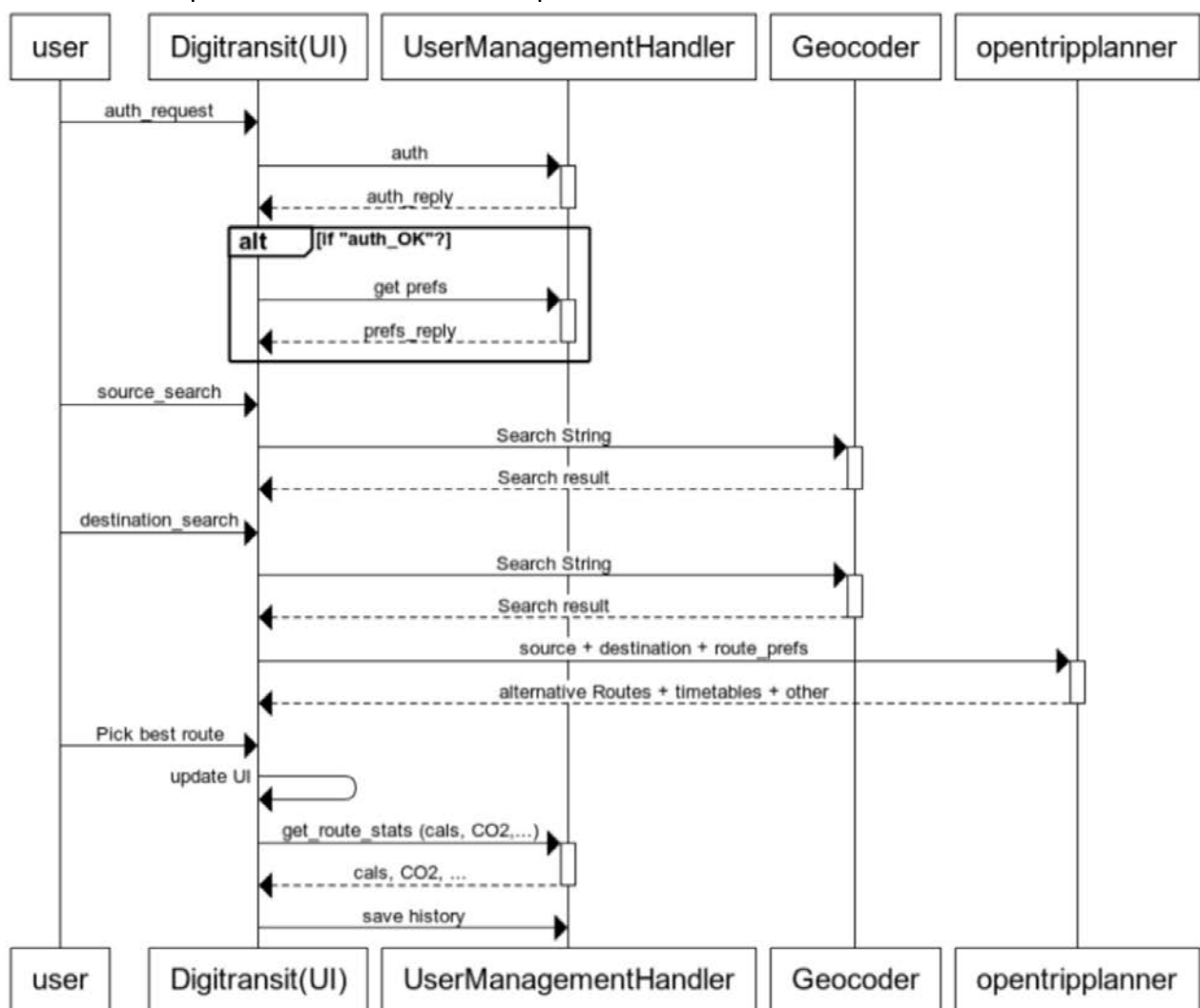


Figure 20 User Login and route search of Porto Multimodal Assistant application

This is the common use case, where the user logs in, so the application loads the user's preferences (via **UserManagentHandler**). After a successful authentication, the user picks the source and destination points. For this, the **Geocoder** component is used, to find a location based on the user input data and reply with a LAT/LONG location to be later delivered to **OpenTripPlanner**. **Opentripplanner** will then calculate the best possible routes.

The user will then pick the desired route. The app will show several stats related to the picked route, based distance traveled walking, using public transportation, etc.

Finally, the stats are saved in users profile.

3.1.6 User Interface

The Porto Multimodal Assistant application is a responsive web app, which means that it automatically adjusts to the screen/browser dimension, layout and resolution, and can thus both be used as a desktop or a mobile version.

Figure 21 and Figure 22 illustrate the app homepage for the mobile and desktop versions, respectively. The homepage comprises the following main features and tools: menu button, login/sign in button, map, destination search bar, “near you” and “favorites” buttons and origin selection tool.

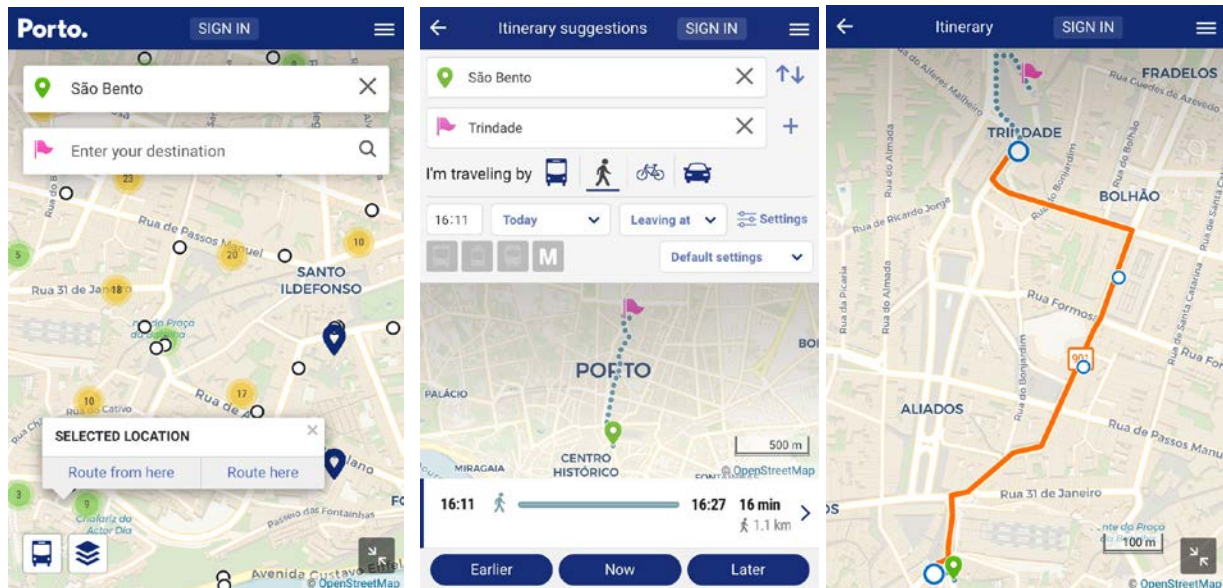


Figure 21 Porto Multimodal Assistant application (mobile version)

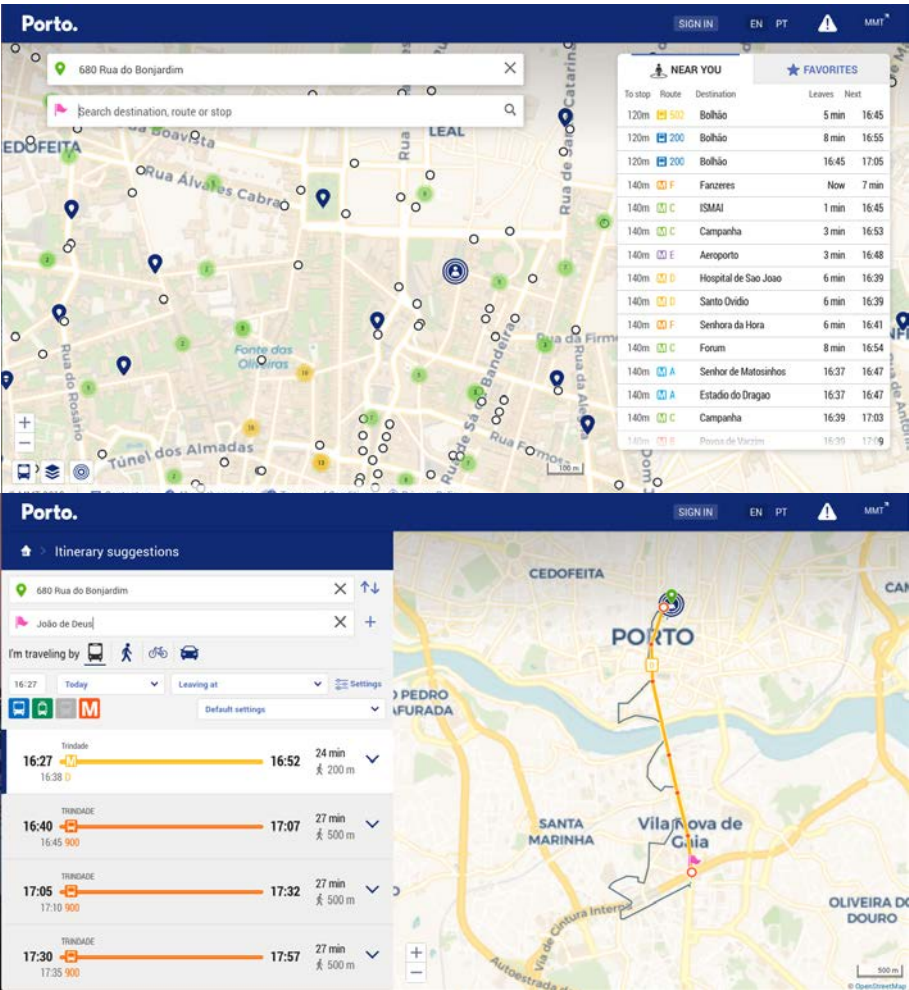


Figure 22 Porto Multimodal Assistant application (desktop version)

Figure 23 and Figure 24 illustrate the user account history and recent searches screen for the mobile and desktop versions, respectively. This screen comprises the following main features and tools: menu button, login/sign in button, “usage metrics” results and “recent searches” results.

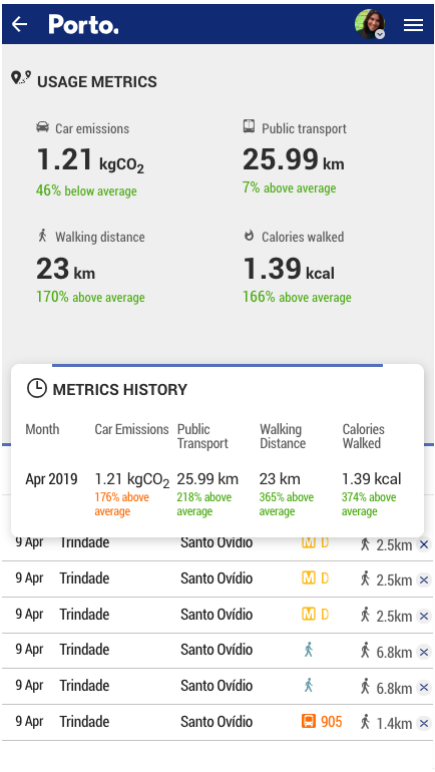


Figure 23 User account history and recent searches screen of the Porto Multimodal Assistant application (mobile version)

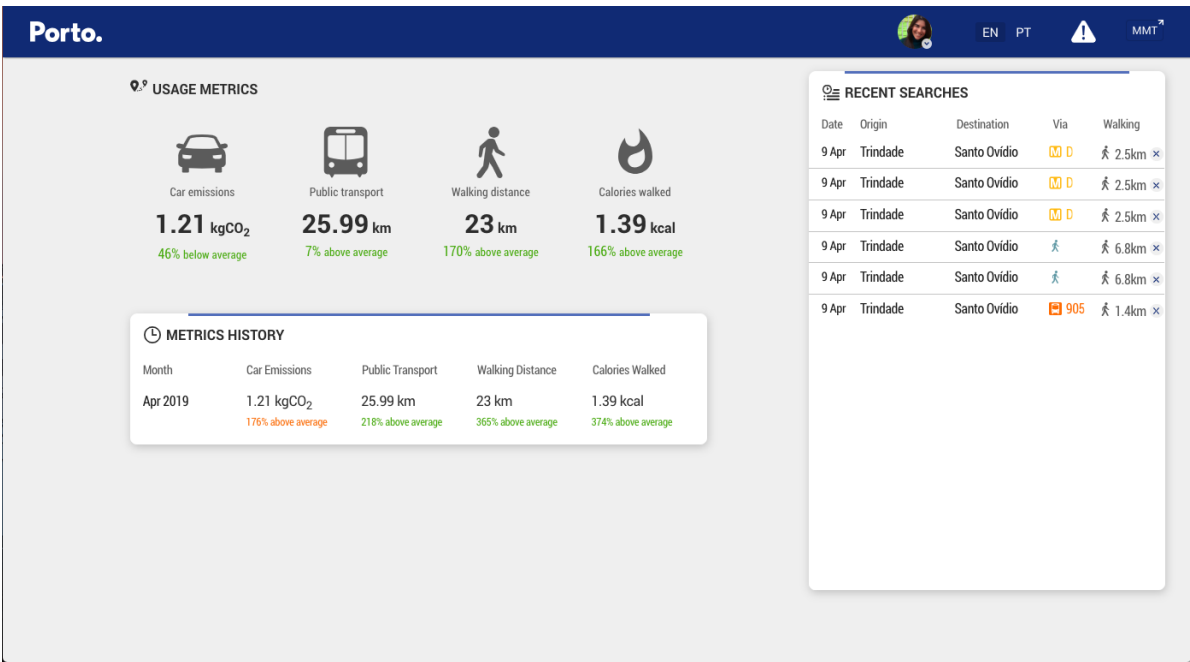


Figure 24 User account history and recent searches screen of the Porto Multimodal Assistant application (desktop version)

3.1.7 Data Management

As stated in the Functional Requirements, the application will allow the user to login (although it is not required for him/her to use it). That will allow the user to store its preferences, preferred locations (favourites), etc.

The application is not storing credentials, since it only allows authentication to be made via OAUTH to external providers (like Google or Facebook).

The following user's information is stored:

- email;
- Avatar;
- navigation data (origin, destination, CO₂, distance walked, calories burned, transport used, price, route);
- user settings.

Concerning Privacy and GDPR compliance, the application stores the above information after, and only after, the user accepts the Terms and Conditions of usage. Also, a registered user is allowed to download all the stored data without explicit request, and is able to cancel the registration, having its data completely removed.

3.1.8 Conclusion

We are still in a pre-release phase, where we presented the application to a small group of users. Nevertheless, the feedback is very positive. At first sight, most common reaction was on how quick and agile it was to implement and release an application such as this one, not only because of its functional requirements, but also because of data supporting them³². In fact, because the application (Digitransit) is based on open source code, it was simple to have it up and running, but also to implement and extend user driven functionalities. Besides, on the data access perspective, the atomic services provided by SynchroniCity, played a primary role.

Also, the application relies on open-source based services, which handle the procedures to access data from its sources. Although they are open-source, it didn't make sense to make specific changes or adaptations to the code, so that services would handle NGSI sources (it would add an extra layer of complexity, and force the city to maintain a fork of it). Also, we wouldn't need it, since reusable Atomic Services such as the GTFS Fetcher handles the hard work.

From the usage perspective, the test group revealed (in contrast with similar application available) as beneficial the chance they have to have a more personalized interface, with information locally gathered (such as Points of interest) to see their mobility history and understand the impact of its day by day mobility needs.

As it was meant, this application is a great example on how to effortlessly use a single source of information, and so having focus on the application functionalities, and not on the data itself or the infrastructure supporting that data.

For the city, having implemented and reusable components (such as the Atomic Services mentioned above) made the connection between the application and the data source seamlessly simple.

Regarding the maintenance of the application, personnel responsible for that task need only to concern about functional aspects, like high availability, performance, etc. Although it is extremely important to have high quality and updated data feeding the application, these are not concerns for the application maintainers, as it is delegated to the SynchroniCity framework and the data providers. This is the most noticeable benefit of the SynchroniCity framework.

³² Further details about users' perception, experience and feedback will be provided at deliverable D3.8 (Report on reference zones IoT service deployment and operations), in particular because we are now starting the deployment phase, and we are still working on the user validation within the scope of WP4 / T4.3 (User, Stakeholder and market validation).

On another perspective, the city will benefit from the application, in the sense that it will promote and ease the usage of the public transportation, learn about the mobility needs of its users (demand), increase awareness to the positive impact of those habits in terms of pollution, energy and traffic reduction, and enable adjusting the public transportation offer.

3.2 Santander's Application "Park & Move"

Thanks in part to its international airport and its connection by weekly ferries with the UK, Santander can be considered as an outstanding tourist destination in the north coast of Spain. This has a huge impact on the way people move around the city. Thus, as introduced in D3.5, Santander RZ selected the Multimodal Transportation use case to exploit new urban premises and enhance sustainable mobility. Accordingly, the "Park & Move" application will be not only a route journey planner but can be considered as a mobility solution to provide useful information to citizens and visitors, easing their mobility experience in Santander.

The core of the final application deployed in Santander adheres to the original Park & Move specifications given in D3.5: it combines two related use cases represented by two applications, the **Smart Parking** app, that exploits the parking estimations; and the **Multimodal Navigator**, that enriches the SynchroniCity Routing service with context information and includes new elements such as the Santander deployed mobility premises. These two apps can be merged into a single one that reuses the Smart Parking outcomes as starting points for the Multimodal Navigator, providing the user with a tool that also relates parking zones and city routes.

This way, the data sources to be potentially consumed by this application are: public transportation network, namely bus and local ferries (schedules, stops and stations, routes and lines, ticketing and pricing, etc.) and bike-sharing systems (docking and parking locations and status, vehicle availability), as well as urban premises to help pedestrians reduce fatigue (escalators, lifts); POIs (Point Of Interest); environment (noise, air quality and meteorological parameters); weather forecasts; traffic flow and constraints; and geographical data (highway, city roads and streets, bicycle path, sidewalk, public transport lines, etc.). Some of this data will be static (from a database or GTFS files), other will be provided in real-time (from sensors and/or GTFS-RT feeds).

Based on this context information and the user's requirements, the route planner will display on a map the routes, user's real-time location and complementary information (such as POIs).

In addition, it will also provide other transportation information, such as, timetables at stops and stations, nearby stops and stations, parking availability, and price and ticketing details. In addition, the user will also have access to complementary data such as POIs in the route. Other features like the display of the weather forecast may be implemented at a later stage.

Further features to be analysed consists in providing users with a mean to pass on valuable information to the city and transportation operators, in particular, by reporting traffic events, road conditions, occurrences and malfunctions (crowdsourcing), and also to rate and add comment about the journey experience. These features may be implemented at a later development stage as well.

All in all, the use of "Park & Move" will provide valuable mobility anonymized data to the city authorities and public transportation operators, which will help the City Council and other mobility stakeholders to make decisions in the areas of mobility, environment, urbanism and tourism.

Licence

The application will present an open source licence and it will be made available and released under a General Public Licence (GPL).

3.2.1 Scenarios

Park & Go application keeps the original approach and so the same scenarios identified in D3.5 with some minor upgrades.

3.2.1.1 Users profiles:

User's profiles have been refined since the initial description provided in D3.5. Current IDs and descriptions are provided in Table 22.

Table 22: User profiles of "Park & Move" application

User profile ID	User description
Car driver	End user that comes to the city by car (driver). In order to save time and fuel (money) he/she wants to know what the best area of the city would be to quickly park his/her car, including the best route to get to this area, depending, among others, on traffic/weather conditions, time and date.
Pedestrian Expert	A user that is a local / regional inhabitant and moves within the city on foot and/or using public transport on a daily basis. Combining bus lines, stops, time estimations and availability of mobility infrastructures, he/she wants to find out the best/adapted route to move from one point to another. This user is mainly interested in speed/total time of transportation and on-time arrivals and departures. The more relevant information he/she requires is real-time information about traffic constraints, estimated time to arrival (ETA) and departure time.
Pedestrian Amateur	A user that is a local / regional inhabitant and moves within the city on foot and/or using public transport on a sporadic way. He/she uses the public transportation system for a particular need or due to a temporary restriction (for example, the car is broken); uses specific routes and transportation modes (for each use); and is not very familiar with the ticketing and transportation modes options. This user is mainly interested in transportation availability and clear information. The more relevant information he/she requires is bus stop/ferries/bike sharing location, route map and route options, and real-time information about estimated time to arrival (ETA) and departure time.
Tourist	A user that arrives (for the first time) to the city and wants to move within the city on foot and/or using public transport. Combining bus lines, stops, time estimations and availability of mobility infrastructures, he/she wants to find out the quickest/adapted/safest/nicest route to move from one point to another as well as the estimated time to arrival (ETA).

3.2.1.2 Scenarios

Current Park & Go version maintains the same set of scenarios defined in D3.5. These are:

1. **Where do I park?** Exposes the parking probability in different areas of the city according to user's requirements and provides with a car route to get the selected parking area.
2. **Transport Me!** Provides a multimodal route within the city, exploiting different context information.
3. **Come & Go** Combines the two initial scenarios to provide a route by car to a park & ride premise (or to the best area to park) plus a multimodal route within the city to get to the final destination.
4. **Visit my City!** Joins public transport feeds to provide a multimodal route from outside the city to a final destination within the city.

3.2.2 Requirements Analysis

3.2.2.1 Functional requirements

According the progress on defining the final Park & Move app, the lists of functional and Non-functional requirements covered have been expanded. These lists are shown in Table 23 and Table 24.

Table 23: Functional requirements of “Park & Move” application

ID	Description
FR-1	Sensed parking lots: real time info about parking lots status, including location MUST be provided.
FR-2	Predefined city areas: that group referred parking lots MUST exist. These SHOULD be defined / provided by the city.
FR-3	Historical data referred to Parking in defined city areas SHOULD be provided
FR-4	The system MAY provide Parking Prob. Estimation info, referred to a city area and a specified time slot.
FR-5	A Routing Service MUST exist: This routing service MUST provide a valid route between point A and point B. This service SHOULD be able to consider different routes profiles (trekking, bicycle, car, etc.) and support mobility premises consideration (lifts, escalators, etc.).
FR-6	The application MUST provide geographical info (street name) and mobility info (nearby lines, stops and stations) for a given location
FR-7	The application MUST provide bus and ferries timetables at the nearby stops and station
FR-8	The application MUST enable route search and planning by street name
FR-9	The application MUST provide a map visualization for all the route suggestions
FR-10	The application MIGHT provide price and ticket information for each route
FR-11	The application MIGHT take into consideration real-time meteorological, noise and air quality data on the route search and planning
FR-12	The application MIGHT take into consideration weather forecast on the route search and planning
FR-13	The application MUST take into consideration scheduled timetables of public transportation modes on the route search and planning
FR-14	The application MUST display the user's real-time location on a map (if user's GPS is enabled)
FR-15	The application MUST provide estimated time of arrival and check-out warnings, based on the route and real-time location
FR-16	The application MUST provide a trip summary (distance travelled, time, etc.) at the end of the trip
FR-17	The application MIGHT enable the user to provide feedback of the journey experience (review, rating, comment, etc.)

3.2.2.2 Non-functional requirements

Table 24: Non-functional requirements of “Park & Move” application

ID	Description
NFR-1	Traffic info: real time & historical data about traffic status MAY be considered within the route calculation process.
NFR-2	Weather forecast: real time & historical data about weather status MAY be considered within the route calculation process.
NFR-3	Required data MUST be accessible through a NGSI interface.
NFR-4	Historical data available referred to parking lots and city areas MUST be accessible through SynchroniCity Historical data access API.
NFR-5	The application MUST have a web app version
NFR-6	The application MUST provide a user's assistance service by e-mail or contact form
NFR-7	The application SHOULD provide a FAQs list
NFR-8	The application MUST provide the terms and conditions of use
NFR-9	The application MUST provide the data privacy policy
NFR-10	The application MUST support Spanish (Spain) language
NFR-11	The application SHOULD support English (UK) language

3.2.3 Available data sources

Table 25 represents the final list of datamodels and datasets to be considered within Park & Go app. Currently, some datamodels (BusArrivalEstimation, BusLine and BusStop) are moving to UrbanMobility datamodels (ArrivalEstimation, PublicTransportRoute and PublicTransportStop respectively). These changes will be reflected in the last update of the app.

Table 25: Data sources used in “Park & Move” application

Title	Description	Open Data / API	Adopted SynchroniCity Data Model	Notes
AirQualityObserved	An observation of air quality conditions (temperature, CO, NO, NO2, etc.) at a certain place (location) and time	Open Data. NGSIv2	Environment (Air Quality Observed)	To be considered for 2nd version of SAN Multimodal APP
BikeHireDockingStation	Describes a place where subscribed users can hire and return a bike	Open Data. NGSIv2	Transportation (Bike Hire Docking Station)	Included in first SAN Multimodal APP version
BusArrivalEstimation (GTFS-RT)	Provides arrival times estimations for the bus lines linked to a bus stop	Open Data. NGSIv2. GTFS-RT	UC-Transportation (Bus Arrival Estimation) → ArrivalEstimation	Included in first SAN Multimodal APP version
BusStop	Describes a bus stop (location and linked bus lines)	Open Data. NGSIv2. GTFS	UC-Transportation (Bus Stop) → PublicTransportStop	Included in first SAN Multimodal APP version
NoiseLevelObserved	It represents an observation of those acoustic parameters that estimate noise pressure levels at a certain place and time	Open Data. NGSIv2	Environment (Noise Level Observed)	To be considered for 2nd version of SAN Multimodal APP

ParkingSpot	Well delimited area where one vehicle can be parked	Open Data. NGSiv2	Parking (Parking Spot)	Included in first SAN Multimodal APP version
OnStreetParking	A site, open space zone, on street, (metered or not) with direct access from a	Open Data. NGSiv2	Parking (On Street Parking)	Included in first SAN Multimodal APP version
PointOfInterest	Harmonised geographic description of a Point of Interest	Open Data. NGSiv2	Point of Interest (Point Of Interest)	Included in first SAN Multimodal APP version
TrafficFlowObserved	An observation of traffic flow conditions at a certain place and time	Open Data. NGSiv2	Transportation (Traffic Flow Observed)	To be considered for 2nd version of SAN Multimodal APP
WeatherObserved	An observation of weather conditions at a certain place and time	Open Data. NGSiv2	Weather (Weather Observed)	To be considered for 2nd version of SAN Multimodal APP
WeatherForecast	Harmonised description of a Weather Forecast, including different parameters, for current day and current week (7 days ahead)	NGSiv2	Weather (Weather Forecast)	To be considered for 2nd version of SAN Multimodal APP
busLine	Describes a public transport bus line, including the related bus stops	Open Data. NGSiv2	UC-Transportation (bus Line) → PublicTransportRoute	Included in first SAN Multimodal APP version
Programacion TUS (GTFS info)	Data related to Santander bus lines routes description and timetables	Open Data. GTFS	GTFS Transit Fee dFile	Included in first SAN Multimodal APP version
Ferries schedule	Data related to Santander ferries lines routes description and timetables	Open Data. NGSiv2. GTFS	GTFS Transit Fee dFile	Included in first SAN Multimodal APP version
Slopes, Funiculars & Escalators location	Data related to the location and direction of Santander slopes, funiculars and escalators	Open Data. NGSiv2	Transportation (Urban premises)	Included in first SAN Multimodal APP version

3.2.4 Application architecture

Santander Multimodal Assistant is an application that relies in the route planning algorithms and APIs provided by OpenTripPlanner (OTP) [13].

Overview

The way OTP provides passenger information and transportation network analysis services make it a useful tool to generate a general route planning service. The core server-side Java component finds itineraries combining transit, pedestrian, bicycle, and car segments through networks built from widely available, open standard OpenStreetMap [14] and GTFS data and can be adapted to the particularities of Santander city itself.

Application Components

Park & Move app is composed by the set of functional building blocks described in D3.5 and depicted in Figure 25:

- **Data Composer** which captures/request information from all the sources needed to build and present the routes, this is, info from the user, context information, routes data from the routing, and parking estimations.
- **Recommendations System**, that recommends best places and relative alternatives to park.
- **Itinerary Planner** elaborates possible itineraries according to multimodal options, user preferences and context information.
- **User Interface** as the communication and interaction channel between the end-user and the core of the app.

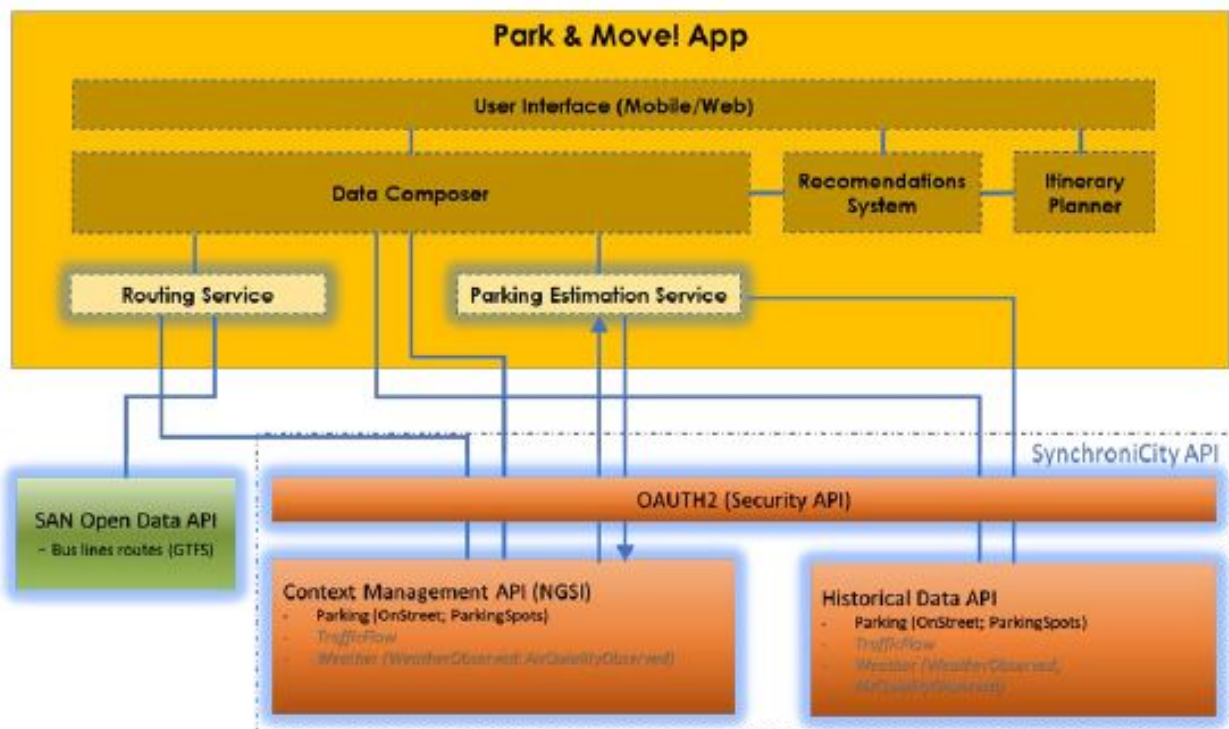


Figure 25: Santander “Park & Move” functional architecture

External Systems

The only external system that the Santander Multimodal Application relies on is the one responsible for map visualization, based on OpenStreetMap.

The following Figure 26 shows the interaction between all the components (internal and external), including the atomic services, described next.

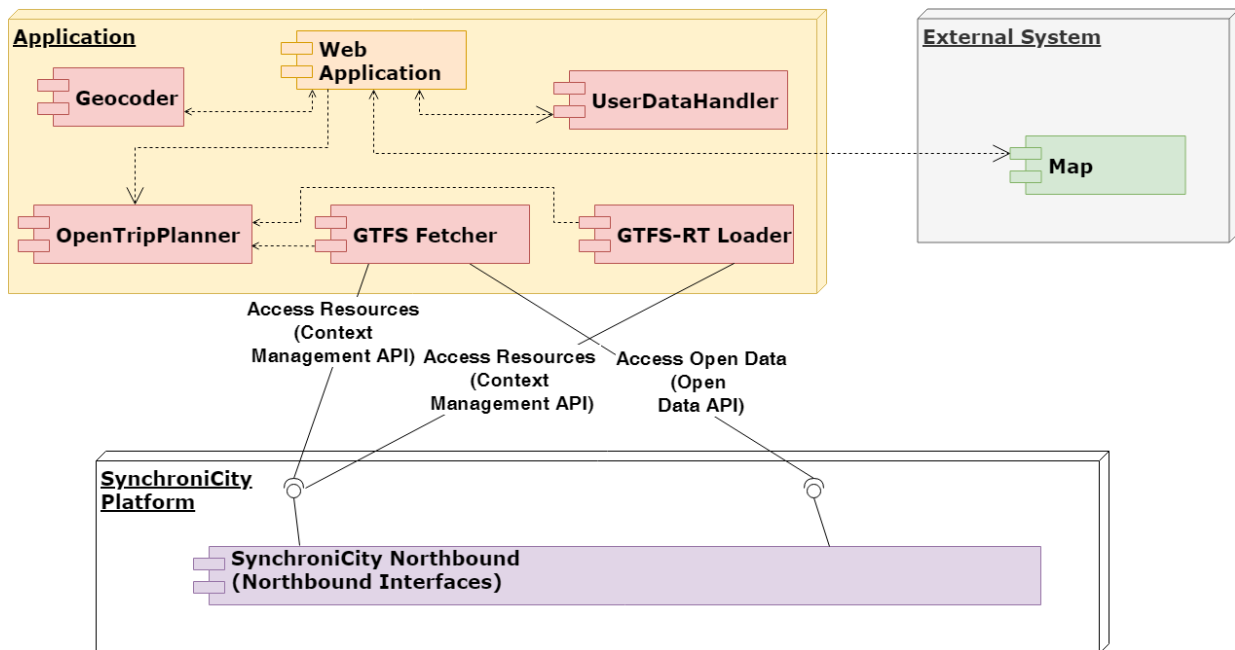


Figure 26: Santander- component diagram of “Park & Move” application

Reusable Atomic Services

Table 26 lists and provides the rationale for the adoption of the Atomic Services in Santander Multimodal Assistant application.

Table 26: Atomic Services adopted in Santander “Park & Move” application

Atomic Service	Rationale
Routing Service	Based on OpenTripPlanner it finds suitable routes combining ferries schedule, escalators location, buses info and bicycle routes (among other available possibilities) between two points and according to user's preferences.
GTFS Fetcher	Extracts data from GTFSTransitFeedFile entities and imports GTFS uploaded files (static timetables and related info) into the OpenTripPlanner platform, making this available for routing calculation (within the Routing Service).
GTFS-RT loader from NGSi	This service consumes Real Time Urban Transport entities (so far, ArrivalEstimation) and generates GTFS-RT feeds from it. These feeds will be later consumed by the Routing Service to calculate/update the requested routes.

In addition, there is an interest on incorporating the Parking Estimator atomic service, which would enrich the “Park & Move” application notably offering users information related to the probable occupancy of parking slots in certain city areas at a specified time slot. This way, the application would incorporate an additional feature useful to those users who plan their trips using their own car or even a rented one, if they are visitors.

3.2.5 Sequence diagrams

The main functionality offered by the “Park & Move” application is represented by the possibility to search for a route in the city of Santander, involving different means of transportation and urban premises. The corresponding sequence diagram that depicts how the application provides this functionality is reported and described in Figure 27 and the text below.

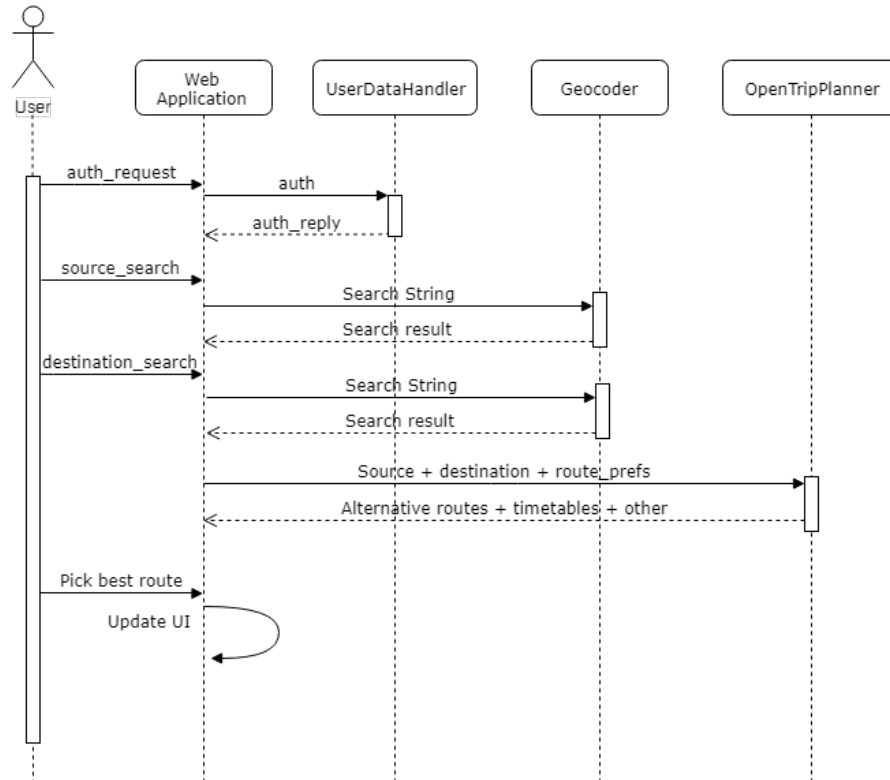


Figure 27: Santander- sequence diagram of searching a route with “Park & Move” application

This could be considered as the core use case, where the user logs in and in turn the application loads that user’s preferences (via **UserDataHandler**). After a successful authentication, the user picks the source and destination points, employing the **Geocoder** component, to find a location based on the user input data, and reply with a LAT/LONG location to be later delivered to **OpenTripPlanner**, that will then calculate the best possible routes.

3.2.6 User Interface

The Santander “Park & Move” Multimodal Assistant application is a responsive web app, which means that it automatically adjusts to the screen/browser dimension, layout and resolution, and can thus both be used as a desktop or a mobile (smartphone, tablet) version.

The user interface in this particular application relies on a map provided by Open Street Maps, which represents a kind of tool well-known nowadays by users around the globe, thus being intuitive and easy to interact with.

The information related to the routes offered and the related means of transportation will appear directly over these maps, not being necessary to perform any change or movement among various screens. As an example, Figure 28 depicts the case when users pick certain options for their trip and the app offers as an answer a collection of potential routes that fit this objective, going into details about the time it would take to follow each option, which means of transportation would include and finally wraps the information snippet with a summary.

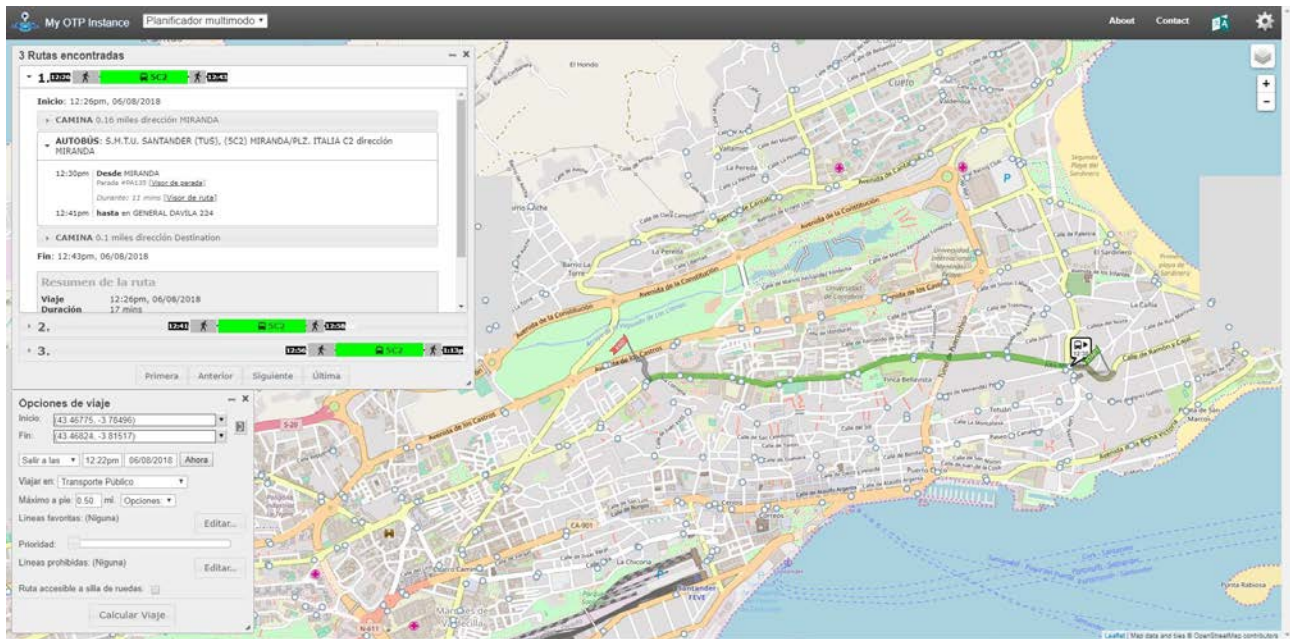


Figure 28: GUI Multimodal Transportation in Santander –found routes (in Spanish)

On the other hand, and thanks to the data handled by the app, users will enjoy the possibility of getting detailed information related to certain points in their routes (see Figure 29) such as precise names of the bus stops and expected timetable of the buses or even ferries suggested.

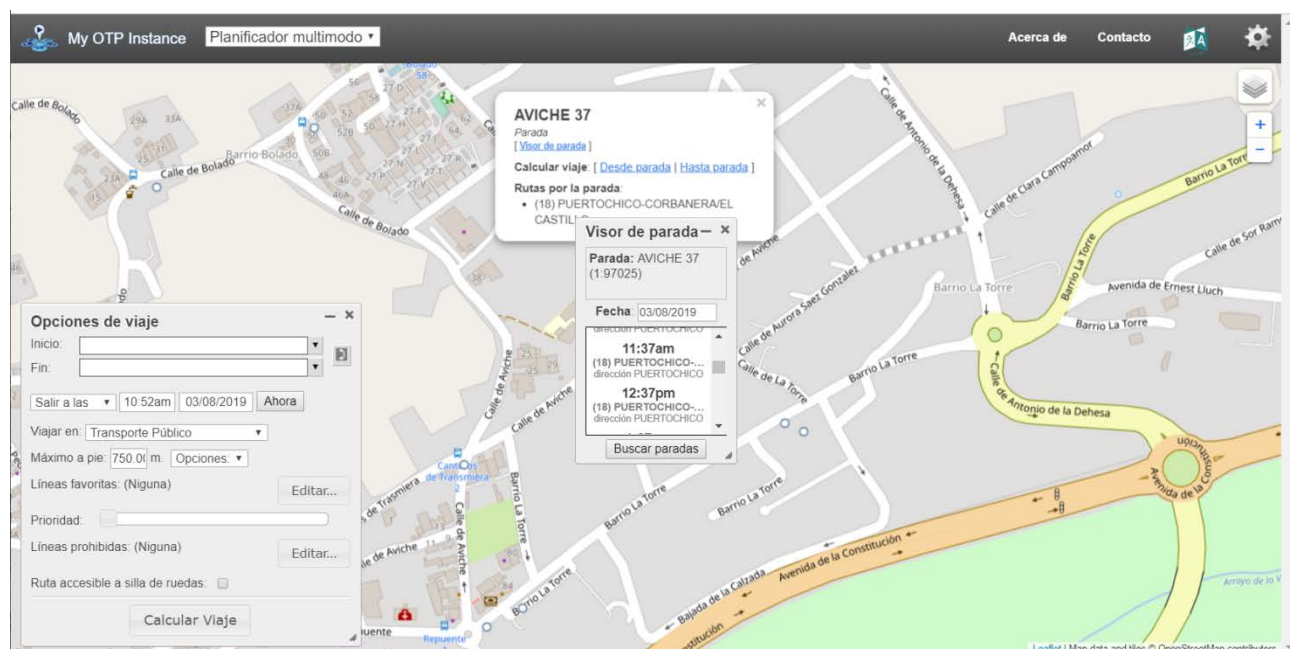


Figure 29: GUI Multimodal Transportation in Santander – details (in Spanish)

3.2.7 Data Management

In this pilot, participants self-register themselves using a web app where an informed consent is filled out. When they do self-registration, they are notified about the data management principles as detailed in the following Table 27.

Table 27: Data Management - Multimodal Transportation in Santander

Type of data	• Raw data from the means of transportation included. • Routes followed by users.
Format of data	• Plain text (routes)
Data collection	• Over the course of the pilot, data will be generated from the app, and be collected in the format defined.
Data storage	• Over the course of the pilot, data will be collected and entered into a SQL database.
Data management	• All the data generated during the pilot will be deleted. Before their elimination, the pilot evaluation report will generate anonymous and aggregated views of the data to illustrate the pilot actions and results aligned to the defined KPIs.
Data Management Principles	• The anonymity and privacy of participants must be respected. Personal information must be kept confidential. Guarantees of confidentiality and anonymity given to the participants must be honoured, unless there are clear and overriding reasons to do otherwise. • In case that the participants must be registered at the SynchroniCity platform and the pilot clients, they must not be registered with their name. E.g. an ID-code will be applicable instead of it. • All researchers have the duty of confidence in regard to collected data. • The integrity of stored, processed and published data must be ensured by the researchers and the project consortium.

With regard to Privacy and GDPR compliance, participants will have a right to retract anytime from the application and when they do, all relative data to the participant will be deleted, except for those used to generate KPIs. In practice, all partners hosting data related to a participant will be notified about the willingness of the participants to withdraw and must then perform the deletion of the data.

By default, all data related to participants will be deleted at the end of the project. Each partner treats data with confidentiality in its information system with accountability for accessing and manipulating those data within their organisation.

3.2.8 Conclusion

The “Park & Move” application in Santander runs on top of already existing open source platforms, such as OTP, providing users with the ability to easily obtain routes in a map involving transportation means available in the city to get them from their starting point to the desired destination, along with some additional information that they may find interesting.

The SynchroniCity framework eased the full exploitation of the services and the related data provided by the Santander Municipality, and the atomic services provided by SynchroniCity played an outstanding role on what refers to the data access.

Nevertheless, at this stage the application development is still ongoing, and there is an interest in focusing efforts on improving the user experience when using this solution. Since we are currently in the pre-release phase, the focus is on beta testers and initial users, all of them Santander locals, to provide us with their valuable opinions and feedback in order for the consortium to know whether "Park & Move" satisfies its needs and represents a useful tool to perform their daily routines.

3.3 Helsinki's Application " Clean Air Routing Service"

With clean air being a large discusses and prominent theme for the last few years, the Helsinki RZ saw it beneficial to implement a clean air routing service for potential implementation to an already existing and widely used routing service (owned by HSL³³).

Even though air pollution does not compose a huge issue on a geographical comparison, it is something to take into consideration when designing services and products for citizens. People with breathing difficulties or severe allergies are disproportionately affected by changes in air quality, thus making this issue something that has to be addressed. During tentatively surveys cyclists and people with small children showed interest toward clean air optimisation services.

The goal for the application is to provide a lower threshold for choosing light traffic alternatives for moving around the city.

Finally, the title of the application has been modified; in D3.5 the tile was "Clean Air Journey Planner".

Licence

The application uses the same licence as Digitransit; the source code of the platform is dual-licensed under the EUPL v1.2 and AGPLv3 licences.

3.3.1 Scenarios

3.3.1.1 Users profiles:

There are no updates, since the user profiles were consolidated in D3.5 and are still valid.

3.3.1.2 Scenarios:

There are no updates, since the scenarios for the application were consolidated in D3.5 and are still valid.

3.3.2 Requirements analysis

3.3.2.1 Functional requirements

The requirement analysis, made after the updates in the application reported in this document, led to no updates in functional requirements.

³³ Helsingin Seudun Liikenne, <https://www.hsl.fi/en>

3.3.2.2 Non-functional requirements

In addition to non-functional requirements reported in D3.5, two new ones have been identified: NFR-6 and NFR-7.

Table 28: Non-functional requirements of Clean Air Routing Service application

ID	Description
NFR-1	The app MUST be available from the web for free without registration
NFR-2	The app MUST support at least one of the most common mobile platforms (i.e. Android OS 7) or be based on web pages (i.e. HTML5)
NFR-3	The app MUST be easy to use
NFR-4	The app MUST respond to queries swiftly in 4G or wifi networks
NFR-5	The application MUST be multilingual
NFR-6	The app SHOULD provide intuitive visual information on the quality of the suggested routes.
NFR-7	Able to integrate new relevant data sources to the application UI.

3.3.3 Available data sources

Compared to D3.5 data sources ‘events and accidents’, ‘Public transport tracking’ & ‘weather’ have been removed; in particular “Public transport tracking” has been removed because the feature is already incorporated in the HSL³⁴ “mother application” ‘Reittiopas’³⁵.

Table 29: Data sources used in Clean Air Routing Service application

Title	Description	Open Data / API	Adopted SynchroniCity Data Model	Notes
ENFUSER Air Quality data(Johansson, Karppinen, & Dong, 2016)	Air pollutants such as: PM2.5 particles, PM10 particles, Ozone (O3), Nitrogen dioxide (NO2), and generalized Air Quality Index.	Open since Sep 2018	WFS2.0(Inc., 2010-11-02) (Web Feature Service) NetCDF (Network Common Data Form) ³⁶	ENFUSER data, 475 million data points in an hourly update.
Map data	OpenStreetMap ³⁷ map data to generate a graph for routing algorithm	Open Data	OSM data model	Pre-processed prior to launching the service
Public transport schedule data ³⁸	Schedule, line and stop data for public transport	Open Data	GTFS ³⁹	Pre-processed prior to launching the service
Noise	Noise levels	Open data	FIWARE: NoiseLevelObserved	3 sensors atm, possibly more by spring 2019

³⁴ Helsingin Seudun Liikenne, <https://www.hsl.fi/en>

³⁵ <https://reittiopas.hsl.fi/>

³⁶ <https://www.unidata.ucar.edu/software/netcdf>

³⁷ <https://www.openstreetmap.org/>

³⁸ <https://dev.hsl.fi/gtfs/>

³⁹ <https://gtfs.org/>

BikeHire	City bycycle rental data	Open data	FIWARE: BikeHireDockingStation	Wrapped from Digitransit GraphQL API ⁴⁰
----------	--------------------------	-----------	--------------------------------	--

3.3.4 Application architecture

Helsinki's Clean Air Routing Service is built on top of an existing open source platform called Digitransit by integrating city-wide air quality data into the optimisation function. The air quality data is provided by the Finnish Meteorological Institute's Environmental Information Fusion Service model (ENFUSER). Noise data is provided via the SynchroniCity platform, with potential support to add more FIWARE data model compatible data sets.

Overview

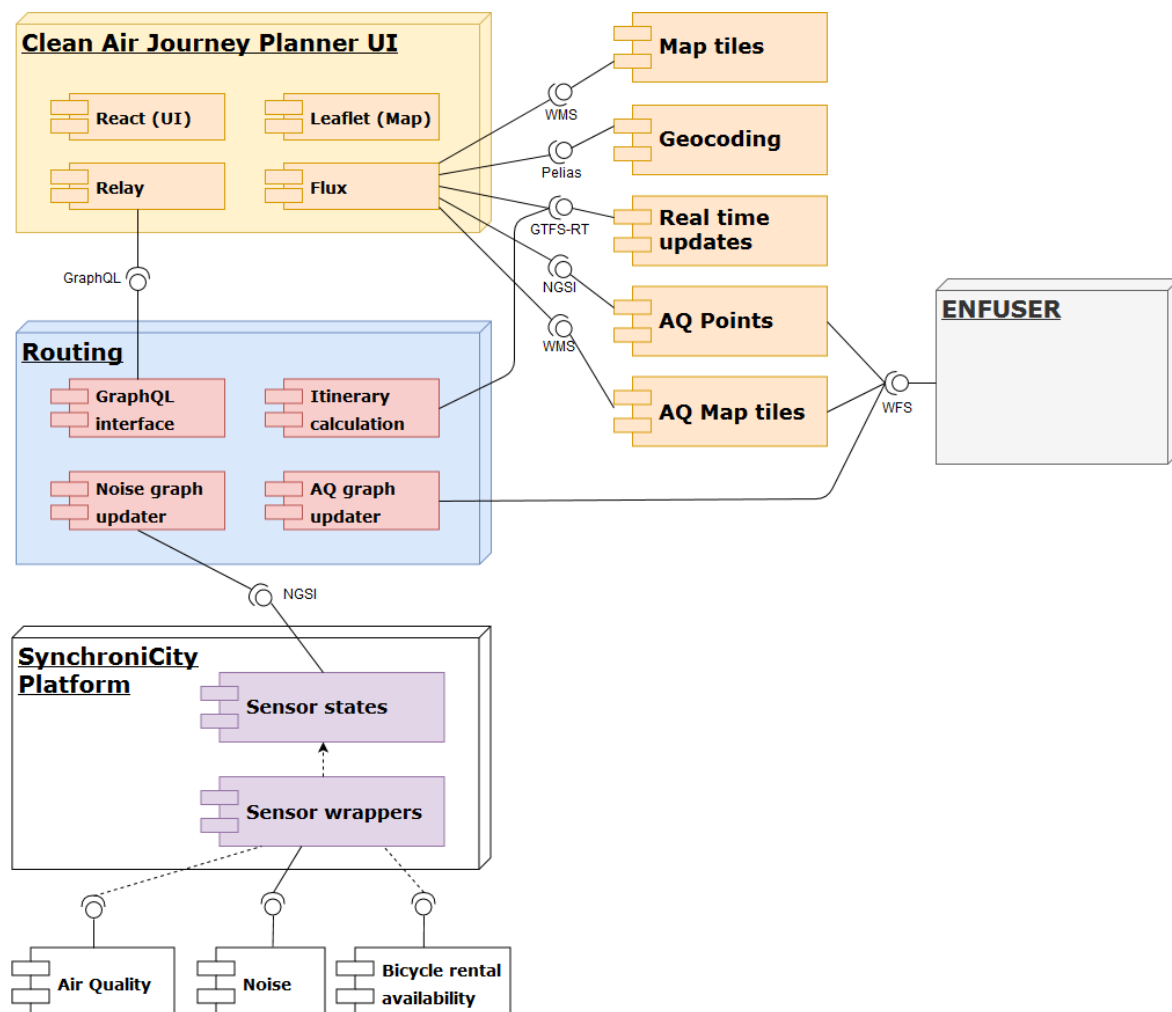


Figure 30: Helsinki - component diagram of Clean Air Journey Planner application

Digitransit is based on *micro service architecture* with four distinct services: *Routing*, *Maps*, *Geocoding* and *Real-time Updates*. New, additional micro services include *Air Quality Points* and *Air Quality Maps*.

⁴⁰ <https://graphql.org/>

Application Components

OpenTripPlanner

Routing functionality is provided by *OpenTripPlanner* (OTP), a SynchroniCity atomic service that has been modified to suit the needs of Helsinki's initial application. The modification entails a change in the the OTP routing core to allow external data sources to affect weights of the edges within the routing graph as an on-line process. Helsinki uses its own version of OTP as part of their DigiTransit Platform to support localized features. The Clean Air Journey Planner version relies on this version of OTP, although the graph altering code is implemented in the generic OTP's core code and hence compatible with it.

As a new modification, OTP has been extended to support also Noise data-based routing, where near real time Noise data is integrated via NGSI from ORION Context Broker. This modification reuses the work for the graph weight alteration in the routing core of OTP. The NGSI support was developed into OTP particularly for this purpose, although in principle other data sources could be integrated via NGSI. It should be noted that such raw sensor data does not suit well the purpose of routing - in order for this to be truly applicable, noise data should be available throughout the city, and in this case the scalability of NGSI and currently supported FIWARE data models might be insufficient. However, this implementation demonstrates the possibility and the core level technical feasibility of adding noise avoidance to a journey planner.

Air quality updates are done on an hourly basis while noise data is currently updated every 10 minutes.

Map Tiles

Map tiles for the app background are provided to the user interface via a Web Map Service interface. Helsinki's tiles are preprocessed to include Helsinki specific localization content and style.

Air Quality Map Tiles

A separate layer that provides a visualization of overall air quality over the entire map region fetches map tiles from a separate Web Map Service interface.

Geocoding

Pelias geocoding service maps addresses to coordinates and vice versa.

Real Time Updates

Real time updates deliver alerts and disruption data along with public transport tracking data via GTFS-RT.

Air Quality Points

Clean Air route points are individually coloured based on air quality at each leg. The data is fetched from a service that emulates NGSI with minimal functionality (i.e. supporting only coordinates).

DigiTransit User Interface

DigiTransit UI enables users to search for routes with their locally stored preferences (no personal information is stored in the platform). The Clean Air Journey Planner version allows users to also search for routes with cleaner air. The UI supports the architecture with two modules, *Flux*⁴¹ for conventional REST interfaces and *Relay*⁴² for communicating with the routing service via GraphQL⁴³. The UI utilizes *React*⁴⁴ components, *Leaflet* for map visualization and a number of other Open Source components⁴⁵.

⁴¹ <https://facebook.github.io/flux/docs/overview.html>

⁴² <https://facebook.github.io/relay/>

⁴³ <https://facebook.github.io/graphql/>

⁴⁴ <https://github.com/rackt/react-router>

⁴⁵ <https://digiTRANSIT.fi/en/developers/services/5-digiTRANSIT-ui/>

SynchroniCity Platform

Noise and air quality sensor data, along with bicycle rental station state data, are integrated to the SynchroniCity Platform, or the Orion Context Broker, via wrappers. Orion serves this data via NGSI. OTP currently fetches noise data from this interface. As mentioned, without large scale coverage, this feature is not yet very useful for end users. Air quality sensor data and bicycle rental station data can also be transmitted to OTP via NGSI. However, bicycle rental station data is already part of Digitransit and visible in the standard UI.

External Systems

ENFUSER

The Clean Air Journey planner utilizes large scale air quality data (475 million data points), which emerges from the ENFUSER model by Finnish Meteorological Institute hourly via a WFS 2.0 API in compressed binary NetCDF⁴⁶ format. Availability of this data is critical and forms the basis for the entire application. Integration of NetCDF formatted data into the routing core was relatively straightforward due to the good availability of tools and documentation. The format has been recently extended to provide floating point air quality values instead of bytes, leading to a 4-time increase in file size, which is currently approximately 1.7GB for each update.

Reusable Atomic Services

Table 30: Atomic Services adopted in Helsinki Clean Air Journey Planner application

Atomic Service		Rationale
Routing Service	Atomic	existing Journey Planner based on Digitransit platform, which based on Open Trip Planner (OTP)

3.3.5 Sequence diagrams

The main purpose of the Helsinki Clean Air Journey Planner is to provide an alternative route, where the importance of air quality is weighed in the route calculation. The Journey Planner is based on the existing Helsinki Digitransit Journey Planner, which provides various features that users can customize. These include the preferred transport mode, park & ride, amount of walking, walking speed, bicycle conditions (citybike use, elevation avoidal), number of transfers, transfer margin, route preferences, fare zones and accessibility requirements (wheelchair or no limits). These have not been modified in SynchroniCity and will not be presented here.

Figure 4 presents the component interaction sequence diagram of a clean air route search from the viewpoint of the user interface. The results of the search will be presented on the user's screen with a map interface, where the default path and clean air path are shown. An air quality map is overlaid over the basic map. Noise sensors are presented on the map as icons and coloured according to current noise values.

The activity starts with the user defining start and end points, possibly by entering addresses. These addresses are mapped to latitude and longitude coordinates in the Geocoder service. On the background at server side, the routing service has been updated with air quality data independently of users. This is an hourly loop where new air quality data is fetched from the ENFUSER service, and routing graph modified accordingly. The air quality data is also pre-rendered as a layer of map tiles and stored at a WMS service. It is also stored to an air quality service under a minimal NGSI interface to provide point-based air quality data. Similarly, noise sensor data and real time information data is continuously processed and made available in Noise and Real Time Updates services via NGSI and GTFS-RT, respectively.

Two route requests are sent to the routing service, one with default search options and one enhanced with air quality options. After both routes are received, the user interface side fetches all required

⁴⁶ <https://www.unidata.ucar.edu/software/netcdf/>

data sets asynchronously. Map data that covers both routes are fetched from the Map Tile and Air Quality Map Tile services and rendered on the screen, with air quality tiles modulating the colour of the underlying map. Route legs are sent as coordinate pairs to Air Quality Point service and received values used to colour the legs accordingly. Noise sensor values are requested from the Noise service via NGSI, and sensor icons rendered with colours depicting the measured noise. Traffic updates are requested from the Real Time Update service with GTFS-RT and selected updates represented on the map.

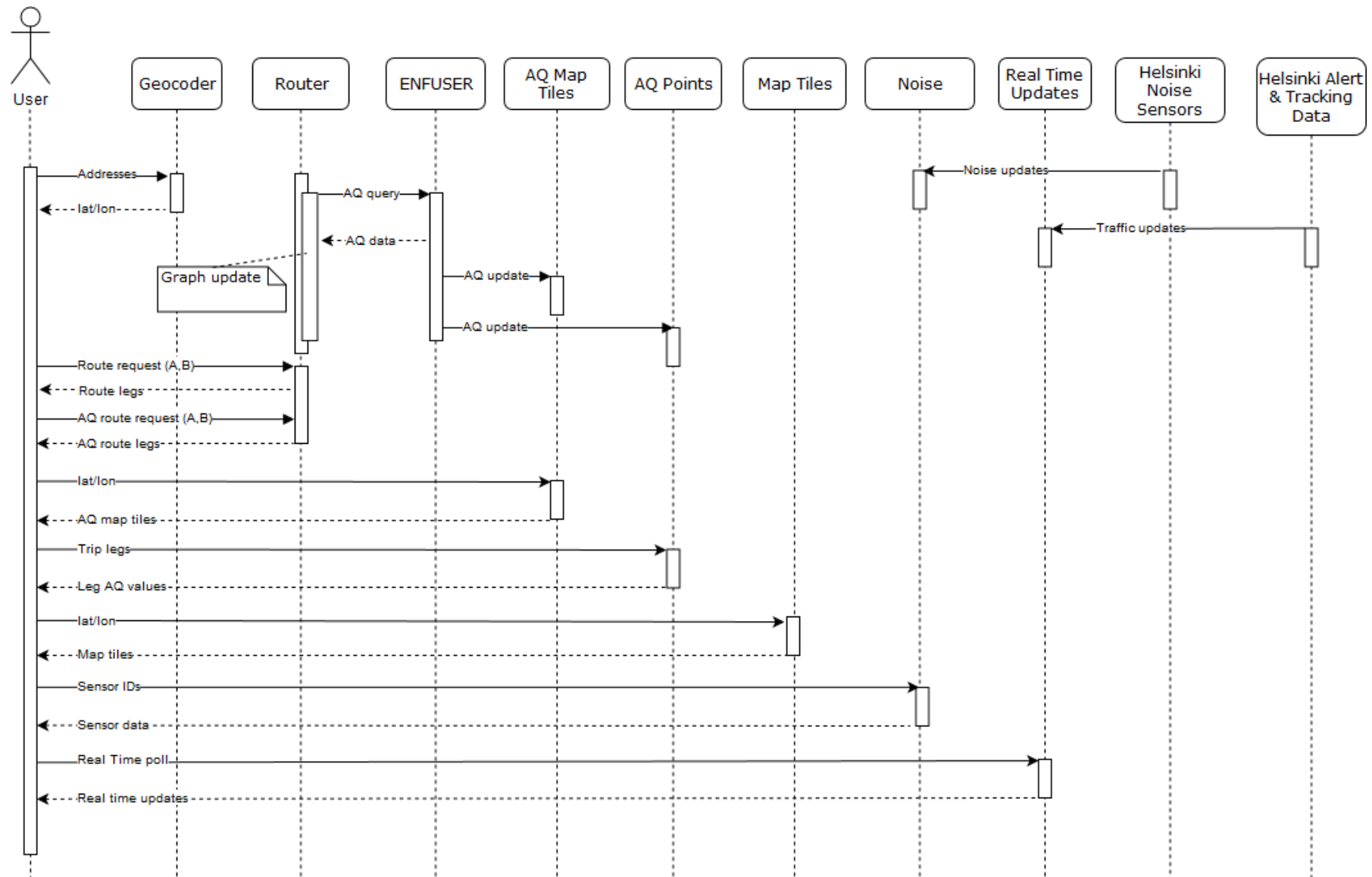


Figure 31: Helsinki - Clean air route search component interaction sequence diagram of Clean Air Journey Planner

3.3.6 User Interface

Digitransit User Interface (UI) is a React-based web application, written using modern ES2015⁴⁷ JavaScript. It supports older browsers too, because the code is compiled to ES5 using Babel.

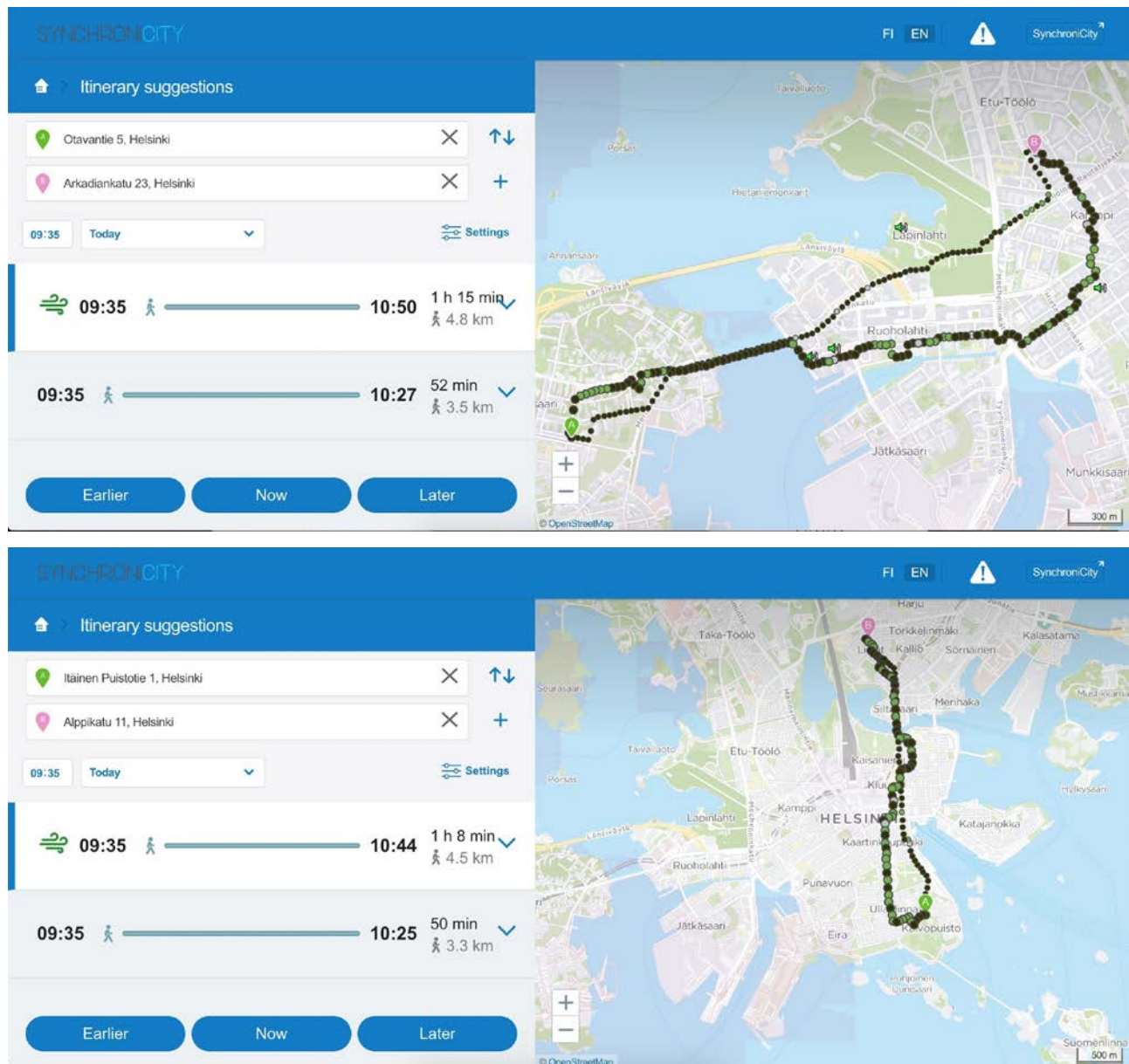


Figure 32: Helsinki - screenshots of Clean Air Journey Planner application

The user interface is based on an existing product⁴⁸. However, for a pedestrian route the interface will provide two possibilities: the default one, and one with cleaner air. To get the desired routing suggestions the user can type the origin and destination into the search field.

Using an Enfuser2NGSI -wrapper the application fetches point data from the model and colours the route accordingly. The Enfuser model creates an unmanageable amount of virtual points; the wrapper allows for NGSI formatted individual points to be created based on routing needs and tasks.

⁴⁷ <http://www.ecma-international.org/ecma-262/6.0/index.html>

⁴⁸ <https://reittiopas.hsl.fi/>

3.3.7 Data Management

No personal data is collected or stored. Locational information is provided to the routing algorithm through the browser only for the routing task; it is not either collected or stored.

3.3.8 Conclusion

The Initial App of Clean Air Journey Planner has been designed to be suitable for *Large Scale Piloting* -- both to be available to a large amount of end users and to utilize large scale data. This goal has been reached by modifying an existing and widely used Open Source high TRL platform, a city-wide journey planner. The new added data, air quality, was also of a very large scale, covering an entire metropolitan area.

The modified Routing Service performs nearly equally to the original one. During performance testing, we did not experience any stability issues. Hence we expect the new service to reach relatively high TRL levels and could potentially serve thousands of simultaneous users on a single physical server computer.

Due to the initial high TRL, the system architecture was already in place. Therefore we faced some difficulties in adopting the Synchronicity platform. All network interfaces and data formats existed, and integrate them with SynchroniCity components has been a challenge. In particular, due to the very large amount of data currently managed the system, we estimated that using a brand new data model for air quality would imply a very long time for data conversion. However, modifying the existing Open Source journey planner platform to integrate the available interfaces directly was fluent, with main bottleneck being the very large update data sets (approx. 1.7GB at the time of writing this deliverable), where the mere raw transmission time is a challenge. The new data formats and interfaces were available with good open source tools and decent documentation, which made the adaptation straightforward from an architectural viewpoint.

In order to introduce the Synchronicity platform to the system, we were able to create wrappers to transform noise level and bicycle rental availability data to FIWARE Orion, and serve that data via NGSI. The Routing Service was furthermore modified to incorporate this data into the routing algorithm. This implementation does not include any advanced dissipation model such as ENFUSER, and is hence quite limited. In other words there is no model to map out the noise in an environment; the effect of the individual noise inputs on the map routing algorithm is to be perfected and developed further. Only those road segments that have associated noise data available can influence routing.

In the future, the Clean Air Journey Planner will be improved by engaging end users, mainly focusing on the user interface. We expect further requirements to arise during this process and improvements will be implemented accordingly.

3.4 Milan's Application "Multimodal-navigator for disabled people"

The main focus of the application is to facilitate the mobility of disabled people across the public services available in Milan.

The Municipality of Milan allows disabled people to traverse the restricted traffic areas with their vehicles. Moreover, disabled people have reserved parking lots wide spread across the city and the Municipality installed parking sensors in order to monitor the free/occupied status of these reserved slots. Currently available navigators usually disregard these special features for disabled people that consider their special needs, permits and reserved parking lots.

Moreover, when disabled people are moving across the city, they may want to use both their vehicle and the public transport system, so they need information about the closest available parking lot for their vehicle and all the information to get and to access to public transport stop or station they want to reach with their wheelchair. The main challenge was harmonizing data about parking areas and the status of related slots, with the ones regarding the existing public transport lines, such as metro, train and bus, in order to offer an accessible and multimodal way of planning a trip around the city of Milan.

In order to provide to the citizens a better service that will consider all the information and disabled people's special needs, the application offers a multimodal navigator especially designed to be helpful for disabled people and their companions.

Licence

The application is released under the terms of the GNU Affero General Public Licence (version 3).

3.4.1 Scenarios

The following section describes the new scenario defined in addition to the ones already described in D3.5. The definition of a new scenario was needed after a deeper analysis of the use cases, where the explicit functionality of calculating and displaying a route on the map between two selected stations was missing. Moreover, the analysis led to the removal of the Registered Disabled Citizen from the Users Profiles, since the application will provide all of its functionalities without requesting user registration; in any case, the application, will allow users to set their preferences, as described in D3.5. Furthermore, the scenarios described in D3.5 are modified accordingly to the previous assumption. It is important to underline that every user's bookmarks or preferences will be managed locally in their mobile devices within the application and not sent to any server. Final scenarios of the application are reported in this document.

3.4.1.1 Users profiles:

Table 30: User profiles of Milan Multimodal-navigator for disabled people application

User profile ID	User description
Disabled Citizen	A wheelchair person. This person can be a commuter, a citizen or tourist and can drive or not a car, moreover, this person can have special permits to go through specific zones of the city while driving a car

3.4.1.2 Scenarios

Table 31: Scenario "Sc.1 - Saving Preferences" of Milan Multimodal-navigator for disabled people application

Scenario title	Sc.1 - Saving Preferences
User profile	Disabled Citizen
Background	Alice just downloaded the application on her mobile phone. Alice runs the application for the first time, and the application shows to Alice a menu where she can set her preferences in order to take advantages of application benefit. Preferences are saved on her mobile phone and data are not sent to a cloud server
Objective	Alice set the application preferences on the first run
Storyline	After downloading the application on her smartphone, Alice is asked to enter her preferences. The application shows to Alice the form to enter her preferences: • home location and work location; • option to find the best solutions for wheelchair people; • special permits for her vehicle to pass across restricted city zone. • add a new bookmarks by address and categorize it as she prefers (e.g. "dad", "mom", "favourite restaurant"); Alice sets the preferences and confirms them; alternatively, she can decide to postponed this activity or change it later.

Table 32: Scenario "Sc.2 - Changing preferences" of Milan Multimodal-navigator for disabled people application

Scenario title	Sc.2 - Changing preferences
User profile	Disabled Citizen
Background	Alice is normally running the application
Objective	Alice may change her preferences in the application
Storyline	Alice open the section of the application to manage preferences and the application shows to Jane the form to manage preferences with her previous settings: • home location and work location; • option to find the best solutions for wheelchaired people; • special permits for her vehicle to pass across restricted city zone; • bookmarks: add, modify or delete and categorize (e.g. "dad", "mom", "favourite restaurant"). Alice sets the preferences and confirms them.

Table 33: Scenario "Sc.3 - Adding bookmark from the map" of Milan Multimodal-navigator for disabled people application

Scenario title	Sc.3 - Adding bookmark from the map
User profile	Disabled Citizen
Background	Alice is running the application and the application shows her a map where she recognizes an important point for her
Objective	Alice wants to save a new bookmark
Storyline	Alice is running the application and looking at the map. Jane noticed an interesting point on the map and wants to save it as a bookmark. Alice selects the point on the map and then the application shows her a form where she is asked if she wants to save the selected location as a bookmark. Alice answers confirm the operation and the application shows her the coordinates and the address and a form to add a description of the location and to and categorize it (e.g.: "restaurant", "cinema"). Alice inserts a description and categorizes the bookmark and then she confirms the provided information. Finally, the application saves the new bookmark.

Table 34: Scenario "Sc.4 - Finding free parking lots " of Milan Multimodal-navigator for disabled people application

Scenario title	Sc.4 - Finding free parking
User profile	Disabled Citizen
Background	Alice is going to the city centre; she is planning to leave her car in a parking area and to take public transportation.
Objective	Alice wants to find free parking lots near a bus stop or a metro station accessible for wheelchaired people.
Storyline	Alice is running the application that shows her a map. Alice presses the button to visualize on the map parking areas; for each parking area, the application shows a marker on the map indicating; each marker has a different colour: green if the parking lots of the area are free, yellow if some parking lots are free, and red if the area has no free parking lots. Alice presses button to display on the map her bookmarks and the application show markers on the map indicating Alice's bookmarks.

	Alice presses button to display on the map the zones of the city she can traverse with her car and the application paints on the map those zones putting them in evidence. Alice presses the button to display on the map public transport stations (e.g.: metro, bus, trams, train). The application show markers on the map indicating public transport stations; each marker has different icons/signs for the different means of transportation. Alice alternatively scroll the map up/down/left/right or zoom/pan the map with her fingers until she finds the point she is interested into at the zoom level she prefers. Alice selects a specific transportation stations and the application opens a pop-up showing her the facilities for wheelchaired people of the selected station. Alice selects a marker representing a parking area and the application opens a pop-up providing information about the parking lots of the selected area.
--	--

Table 35: Scenario "Sc.5 - Calculating routes plan between two stations" of Milan Multimodal-navigator for disabled people application

Scenario title	Sc.5 – Calculating routes plan between two stations
User profile	Disabled Citizen
Background	Alice is going to the city centre and left her car in a parking area to take public transportation.
Objective	Alice wants to go from a near starting station accessible for wheelchaired people (e.g. bus stop or a metro station) to a destination. She wants the route to be followed to arrive at her desired destination.
Storyline	Alice is running the application that shows her a map. Alice presses the button to visualize on the map the transportation stations e.g.: metro, bus, tram, train) available near where she parked the car. Alice presses button to display on the map her bookmarks and the application show markers on the map indicating Jane's bookmarks. The application shows markers on the map indicating public transport stations; each marker has different icons/signs for the different means of transportation. Alice selects a specific starting transportation station and the application opens a pop-up showing her the facilities for wheelchaired people of the selected station. Alice scrolls the map up/down/left/right or zoom/pan the map with her fingers until she finds the destination station she is interested into at the zoom level she prefers. Alice selects the specific destination station and the application opens a pop-up showing her the facilities for wheelchaired people of the selected station. Alice confirms to visualize the planned route (on or more) between the selected starting and destination stations. The application shows the route paths traced on the map, reporting the possible alternative paths between the two selected stations.

3.4.2 Requirements Analysis

The following sections report updated and revised requirements, functional and non-functional, for the application. In particular, in addition to D3.5, has been added the FR-21, which led to the definition of the new scenario, as described in the section above. Moreover, as the result of a deeper

analysis on the application requirements, FR-1 (originally reported in D3.5) has been removed since the application does not need user registration.

3.4.2.1 Functional requirements

Table 36: Functional requirements of Multimodal-navigator for disabled people application

ID	Description
FR-2	Application MUST be able to manage user's preferences
FR-3	User MUST be able to set and edit home and work location among the preferences
FR-4	User MUST be able to set and edit permits for her vehicle to pass through restricted city zone among the preferences
FR-5	User MUST be able to add a new bookmark by address and to categorize it (e.g. "dad", "mom", "favourite restaurant")
FR-6	User MUST be able to edit an existing bookmark
FR-7	User MUST be able to delete an existing bookmark
FR-8	User MUST be able to enable an option to find the best solutions for wheelchaired people
FR-9	The application MUST provide a map to search for a destination.
FR-10	User MUST be able to navigate the map scrolling up/down/left/right or to zoom/pan the map
FR-11	User MAY be able to navigate the map with fingers.
FR-12	User MUST be able to add a new bookmark selecting a point on the map.
FR-13	The application MUST provide functionalities to display parking areas on the map
FR-14	The application SHOULD display parking areas on the map with different colours to indicate their status: green if the parking lots of the area are free, yellow if some parking lots are free, and red if the area has no free parking lots.
FR-15	When the user selects a parking area on the map, the application MUST provide its related information.
FR-16	The application SHOULD display public transport stations on the map.
FR-17	The application SHOULD display public transport stations on the map with different markers to indicate the different means of transportation (e.g., bus, metro, train).
FR-18	When the user selects a public transport station on the map, the application MUST provide its related information.
FR-19	The application MUST be able to display user's bookmarks on the map.
FR-20	The application MUST be able to display on the map the restricted zones of the city user can traverse driving a car.
FR-21	The application MUST be able to calculate and display on the map the planned routes between two stations selected by the user.

3.4.2.2 Non-functional requirements

Table 37: Non-functional requirements of Multimodal-navigator for disabled people application

ID	Description
NFR-1	The application MUST support Italian language.
NFR-2	The application MUST be accessible on mobile devices.
NFR-3	Required data MUST be accessible through a NGSI interface.

3.4.3 Available data sources

Table 38 reports the updated list of the available data sources for the application. It is important to underline that all the data reported have been mapped to a specific SynchroniCity data model and then loaded in the Context Broker provided by the SynchroniCity platform. Not all the data coming from E015 API [15] have been authorized for use in the application. Below is described the rationale about these updates:

- They have been retrieved from the Milan Open Data Portal and no more from APIs, as reported in D3.5, and then have been mapped into SynchroniCity data model (PointOfInterest) and imported into the SynchroniCity Platform as NGSI entities.
- The events are no longer available, since the Municipality stopped the source API.
- Car sharing areas have been mapped to its own dedicated CarSharingStation data model, and no more with the On/OffStreetParking one. The CarSharingStation data model was designed within SynchroniCity project and adopted to give a specific representation of car sharing areas.
- Traffic restricted areas have been mapped to the defined RestrictedTrafficArea data model.
- The public transport network information has been mapped to their own dedicated data models called Urban Mobility. These data models were defined within SynchroniCity project and then adopted as official FIWARE NGSI data models.
- The access to real time information about the status of metro lines, interchange parking and public transport waiting time (coming from E015 API), has been denied by the Milan Transport Agency (ATM - Agenzia Trasporti Milanese SPA).
- The access to the E015 APIs for the traffic status and events, provided by InfoBlu SPA, has been denied.
- The use of the data about parking lots status, coming from the sensors provided by Kiunsys Srl, has been authorized.
- The E015 Ev-Router API, which returns stations for charging electric vehicles, ordered by proximity to the address specified in input, has been authorized by Route220 SRL, and mapped to the EvChargingStation data model.

Table 38: Data sources used in Milan Multimodal-navigator for disabled people application

Title	Description	Open Data/API	Adopted SynchroniCity Data Model	Notes
Place/Street names	Official place and street naming system associated to geographic position.	API	Road	
Point of interests and touristic itineraries	Descriptions of valuable places in an area and the suggested touristic itinerary.	Open Data	PointOfInterest	POIs types differentiated by using different Id patterns.
Traffic Restricted areas	GeoJson description of traffic restricted areas.	Open Data	RestrictedTrafficArea	Data is available in GeoJson format
Public transport metro network	GeoJson description of metro network and stations.	Open Data	UrbanMobility	Data is available in GeoJson format
Public transport	GeoJson description of terrestrial bus/tram	Open Data	UrbanMobility	Data is available in

terrestrial network	network and relative stops.			GeoJson format
Car sharing parking areas	GeoJson description of car-sharing parking areas, with parking lots.	Open Data	CarSharingStation	Data is available in GeoJson format
Parking areas	GeoJson description of parking areas.	Open Data	On/OffStreetParking	Data is available in GeoJson format
Parks and gardens	GeoJson description of public parks and gardens.	Open Data	Garden	Data is available in GeoJson format
City facilities	GeoJson description of facilities such as pharmacie, school, congress centres.	Open Data	PointOfInterest	Data is available in GeoJson format
Airport parking availability	Real time status of interchange parking areas.	API	OffStreetParking	E015 API. It has been authorized.
Train time schedule	Real time status of urban train network.	API	UrbanMobility	E015 API. It has been authorized.
Parking lots status	Information about the status of parking lots, such as availability.	API	On/OffStreetParking	Parking sensors have been installed and their use has been authorized.
Electric car charging stations	Information about the location of electric cars charging stations	API	EVChargingStation	E015 API. It has been authorized.
Weather Stations and weather measurements	Information about weather stations and real-time measurements for each station.	API	WeatherObserved	Legacy LAMPPPOST APIs are available.

3.4.4 Application architecture

This section describes the updated architecture of the Multimodal-navigator for disabled people application.

Overview

Milan's Multimodal-navigator for disabled people application leverages on SynchroniCity APIs exposed by the Municipality of Milan itself; in particular, it will consume two of them: Context Management API and Historical Data API.

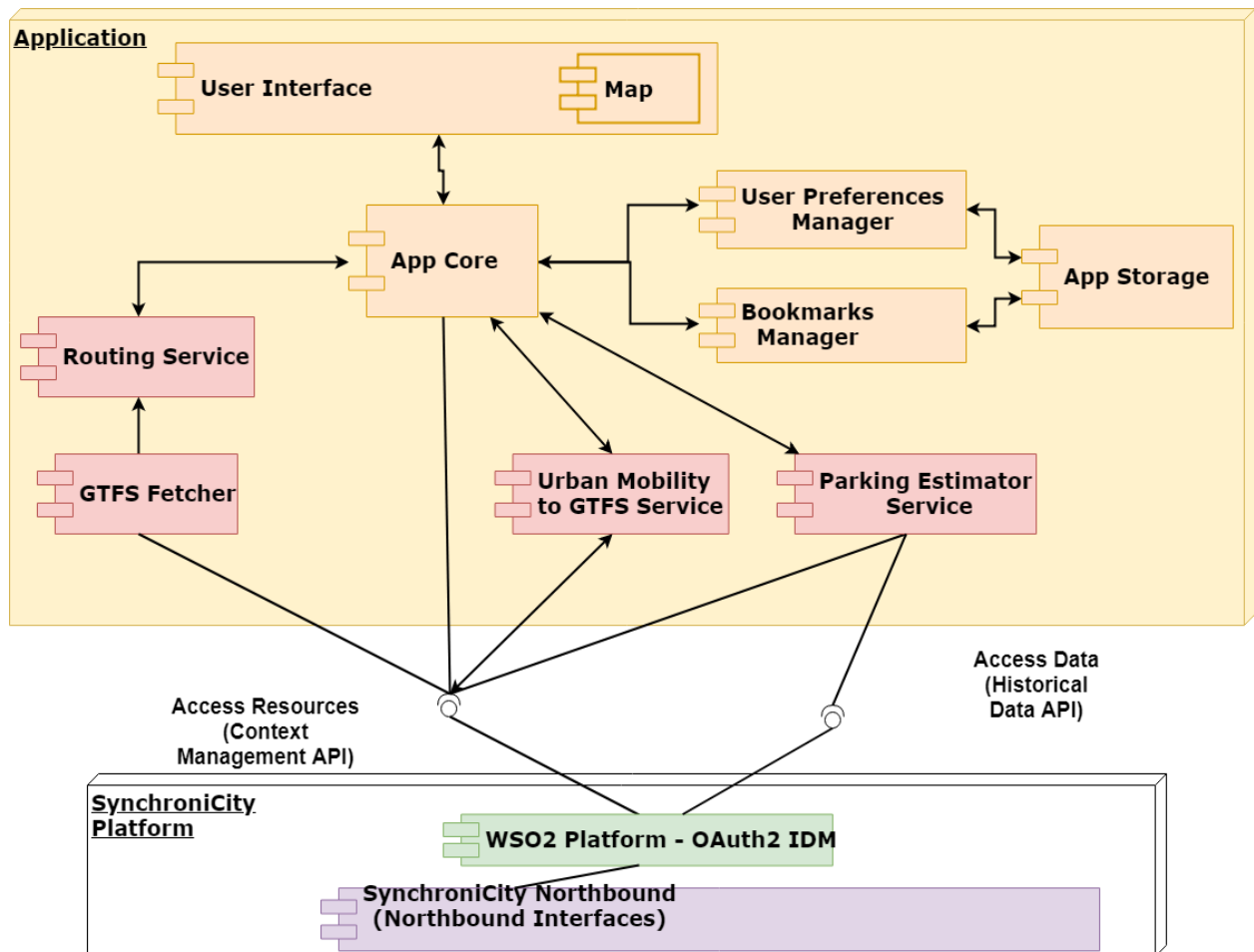


Figure 33: Milan - component diagram of Multimodal-navigator for disabled people application

The architecture has been updated with minor changes in the components already defined in D3.5; in particular, two new atomic services have been added, since the application uses the GTFS data, translated by the NGSI Urban Mobility to GTFS service and downloaded by the GTFS Fetcher service. Figure 33 depicts how the application is built internally, how the internal components interact among them and with SynchroniCity APIs. The application will use data retrieved from SynchroniCity API for its core functionalities, such as the GTFS data translated from the NGSI Urban Mobility entities. The access to this API will be secured by the IdM module/component currently working on the WSO2 platform (WSO2 Identity and Access Management, 2018) of Milan Municipality which is compliant with OAuth2 standard according to the SynchroniCity Security layer specification.

Application Components

The application is composed by a set of internal functional components:

- **User Interface:** this component provides the interaction point with the end users and the possibility to manage all the aspects of the application, such as preferences and bookmarks. It enables the user to visualize and navigate a map, to look for Point of Interests, parking, public transport stations and access Points, save them as Bookmarks and finally to visualize the routes between two selected public transport stations.
- **App Core and atomic services:** this component represents the logical controller of the application and it manages all the requests made by the end users through the User Interface, by coordinating the interactions of all the other components of the application. This component also interacts through a REST interface with the atomic services adopted by the application: the **Parking Estimator Service**, the **Routing Service**, the **Urban Mobility to GTFS Service** and indirectly with the **GTFS Fetcher**. The Parking Estimator will retrieve NGSI current and historical data, coming also from installed sensors and containing location

and availability of the parking slots, with particular attention to specific accessibility aspects (e.g. number of reserved slots, type of physical access, type of permit needed, etc) for disabled users. These data will be used to return either the current or the predicted availability of free parking slots. The Routing Service will be used to calculate the routes of a trip, taking into account the information regarding the existing public transport lines and restricted access areas. The Routing Service needs as input the GTFS feed files, which will be generated from the corresponding NGSI Urban Mobility entities present in the SynchroniCity platform, by the means of the **Urban Mobility to GTFS atomic service**. This service indeed, retrieves all the Urban Mobility NGSI entities representing the public transport services (as reported in Table 38), issued by one or more transportation agencies, and translates them to a GTFS feed archive, stored in a publicly exposed directory. The Service then creates the particular *GTFS TransitFeedFile* entity⁴⁹, which contains the URL reference from where the corresponding GTFS feed file will be downloaded. Thus, the GTFS Fetcher service, is in charge of downloading the GTFS feed archive, by the means of the URL reference contained in the *GTFS TransitFeedFile* entity, and then to send it in the Routing Service. Once the GTFS feeds have been loaded into the Routing Service, the App Core is in charge of querying the Service correctly, in order to get one or more routes between a starting and a destination station. Finally, the App Core also interacts directly with the SynchroniCity Context Management API, in order to retrieve specific NGSI entities, such as *PointOfInterests* or *Parks*, which will be used to enrich the map displayed in the User Interface.

- **User Preferences Manager:** this component manages the user preferences; in particular it is in charge of saving, retrieving, updating and deleting user preferences as requested by the App Core on the base of user's actions; for these goals it leverages on functionalities offered by App Internal Storage.
- **Bookmarks Manager:** this component manages the bookmarks of the user; in particular it is in charge of saving, retrieving, updating and deleting bookmarks as requested by the App Core on the base of user's actions; for these goals it leverages on functionalities offered by App Internal Storage.
- **App Internal Storage:** this component provides storage functionalities to manage both *user preferences* and *bookmarks*, as requested respectively by User Preferences Manager and Bookmarks Manager.

All these components will be packaged together in order to be deployed and executed in mobile devices, whereas the four atomic services (the Parking Estimator Service, the Routing Service, the Urban Mobility to GTFS service and the GTFS Fetcher service), even if, from a logical point of view they are part of the application, they will act as remote services.

External Systems

The application does not rely on any external component.

Reusable Atomic Services

As depicted in Figure 33, Milan's Multimodal-navigator for disabled people application leverages on four atomic services, which support the exploitation of the NGSI data provided by the SynchroniCity framework and the provision of the functionalities of the application itself. These four atomic services are: Parking Estimator Service, Routing Service, Urban Mobility to GTFS and GTFS Fetcher, reported with their rationale in Table 39.

⁴⁹ See deliverable D2.3 (Catalogue of OASC Shared Data Models for Smart City domains).

Table 39: Atomic Services adopted in Milan's Multimodal-navigator for disabled people application

Atomic Service	Rationale
Parking Estimator Service	This baseline service provides both the current and predicted free parking lots in a parking area in a specific time window. It is tailored in order to provide support to disabled people, for instance providing adequate information about parking lots dedicated to wheelchair people. Indeed, its functionalities will help the disabled users to plan and optimise their routes within the city.
Routing Service	Based on OpenTripPlanner, it finds suitable routes, combining several transport information such as tram, train and bus lines, between two points and according to user's preferences. It will be used to provide routes to be followed by users of the application in order to arrive at their desired destination; in addition, it will be customised in order to provide tailored routes for wheelchair people. It uses the GTFS Feed generated by the Urban Mobility to GTFS Service.
Urban Mobility to GTFS Service	It retrieves all the NGSI Urban Mobility entities from the Context Broker, representing the public transport lines (routes, trips, timetables, stations, stops, etc.) and translate them to the corresponding GTFS Feed archive, to be used by the Routing Service.
GTFS Fetcher	It extracts metadata from the GTFSTransitFeedFile entities, in particular the URL where to download the corresponding GTFS Feed; then, it imports the downloaded GTFS files (static timetables, route shapes and related info) into the Routing Service, making this available for routing calculation.

3.4.5 Sequence diagrams

Among the functionalities offered by this application, the more relevant ones are represented by the possibility to:

- Navigate the map, select one or more displayed elements (e.g. PointOfInterest) and optionally save them as bookmarks.
- Navigate the map, selecting parking and public station, requesting related information about slot availability and accessibility details.

The sequence diagrams that depict how the application provide this functionality are reported and described in this section:

- **Explore Map and save Bookmarks:** this sequence diagram (Figure 34) depicts interactions among the components of the application in order to navigate the map, explore and select displayed markers representing additional context information (e.g. pharmacies, parks, city facilities, etc.) and optionally save one or more of them as bookmarks.
- **Select parking, stations and plan a route:** this sequence diagram (Figure 35) depicts interactions among the components of the application in order to navigate and explore the map, selecting displayed parking and public transport stations that include additional information, such as the availability of free reserved parking slots and accessibility details. Thus, the user can select the desired starting and destination stations, in order to visualize the calculated routes using the available public transportation lines in the city.

Explore Map and save Bookmarks

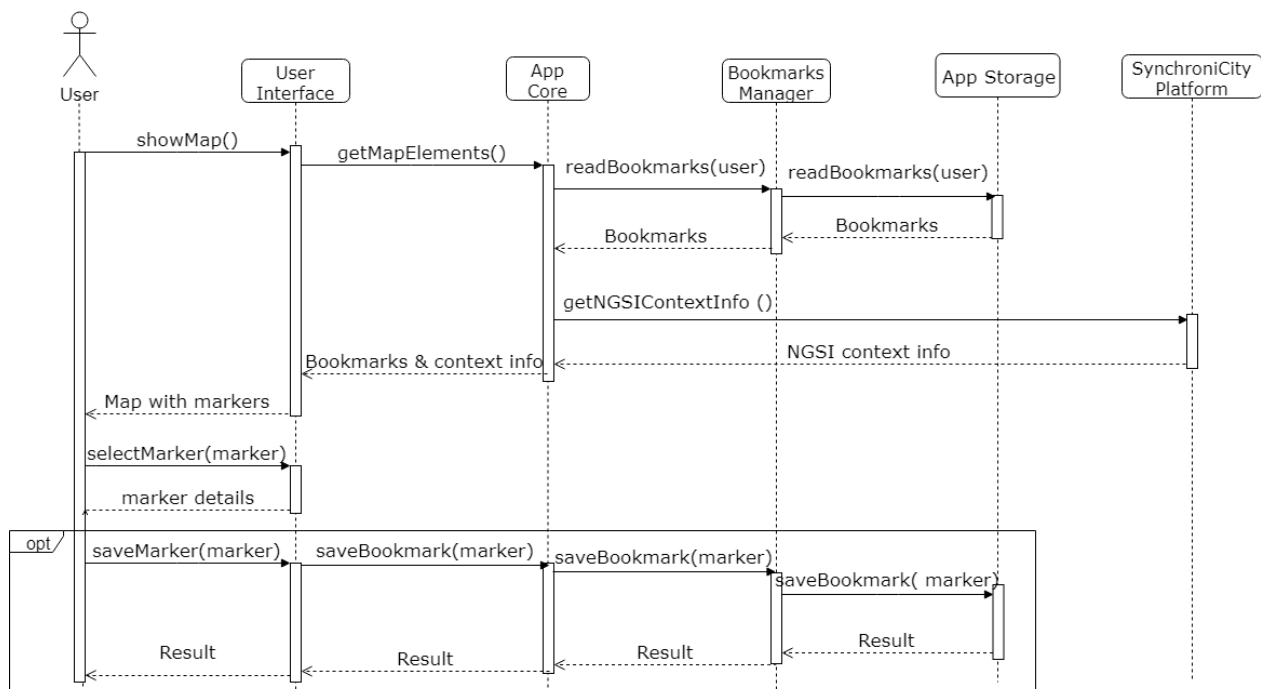


Figure 34: Milan - Explore Map and save Bookmarks sequence diagram of Multimodal-navigator for disabled people application

The user requests through the interaction with the User Interface to show the map reporting all his/her own bookmarks, if there are any, and all the additional context information as map markers. The User Interface sends the request to get all the map elements to the App Core. This, at first, sends the request to read all the user bookmarks to the Bookmarks Manager. This, in turn sends the request to the App Storage, which returns to the Bookmark Manager and then to the App Core the user bookmarks retrieved from its own internal storage. Then, the App Core sends a request to the SynchroniCity Platform, through the Context Management API, in order to retrieve all the NGSI entities regarding additional context information such as PointOfInterests, Parks, etc. The Context Broker (within SynchroniCity Platform) returns the NGSI Entities to the App Core; the Core processes and translates the retrieved NGSI entities and the previously gathered Bookmarks with the related details, to the appropriate format required to show them as markers. Then, it sends the markers to the User Interface, which shows to the user a map enriched with all the markers representing the user bookmarks and all the other interesting elements.

Then the user can select a specific marker (one or more), which details, previously returned by the App Core, are shown in the User Interface. The user can optionally request to save the selected markers as bookmarks. The User interface sends the request to save the selected markers to the AppCore, which sends the request to the Bookmarks Manager and this in turn to the App Storage components. The Storage saves the bookmark in its own internal storage and sends back a confirmation to the Bookmarks Manager. It returns the confirmation to the App Core, and this to the User Interface, which shows to the User the confirmation message.

Select and show parking and stations

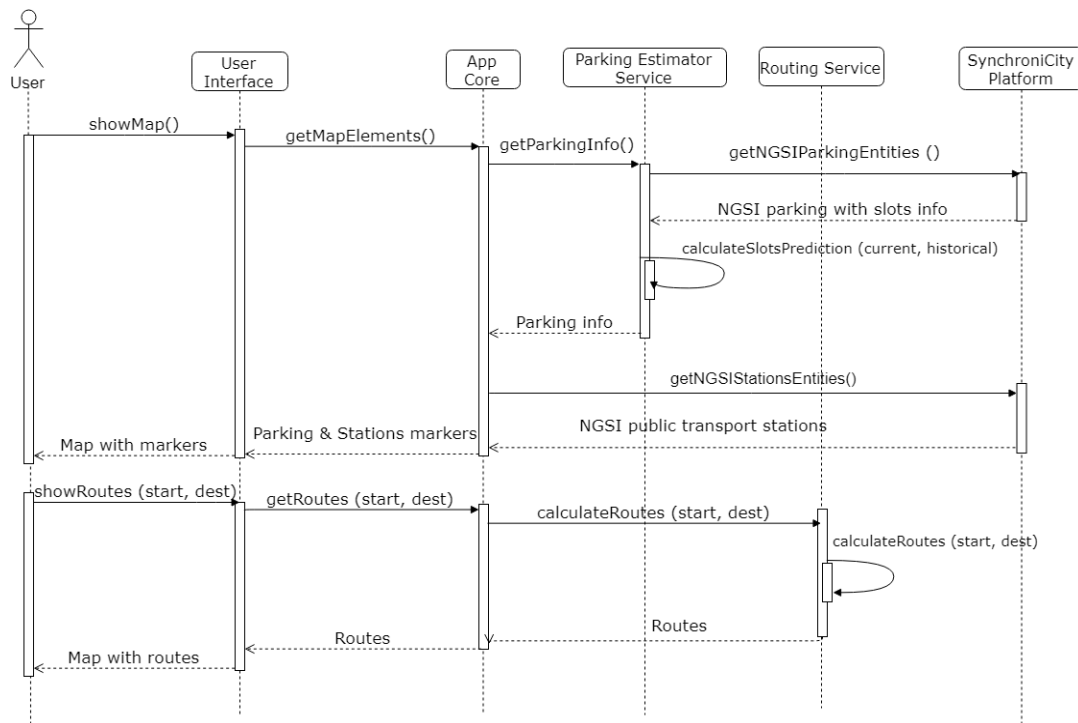


Figure 35: Milan - Select and show parking and stations sequence diagram of Multimodal-navigator for disabled people application

The user requests, through the interaction with the User Interface, to show the map reporting the markers for parking areas and public transport stations (e.g. metro, tram, bus). Displayed parking areas (e.g. Airport parking, CarSharingStations, etc.) will have additional information regarding the availability (current or estimated in a time slot specified in the user's preferences) of its own slots, in particular, the ones reserved for the disabled people. The User Interface sends the request to the App Core, in order to get all the map elements. The App Core, at first, requests to the Parking Estimator Service to get all the parking areas, enriched with the estimations about the occupancy of the related parking slots, made by the atomic service itself. The Parking Estimator Service sends the request to the SynchroniCity platform, in order to get current and historical values of the parking areas entities (On/OffStreetParking), including the information about their slots' occupancy. The Parking Estimator then calculates internally the prediction about the availability of the parking slots, relying on the retrieved current and historical slots info. It fills the parking area info with the calculated estimations and returns them to the App Core. Then, the Core sends directly a NGSI request to the SynchroniCity Platform, through the Context Management API, in order to get all the NGSI entities regarding public transport stations. Once all the parking and stations info have been retrieved, the Core translates all these data to the appropriate format, required to show them as markers. Then, it sends the markers to the User Interface, which shows to the user a map enriched with all the markers representing the parking areas, with different colours (red, yellow, green), based on the percentage of either current or estimated slots occupancy.

Then the user can select, through the User Interface, the start and destination stations, displayed as markers, for which he wants to show all the possible routes. He/She then sends the request to the User Interface, which forwards it to the App Core. The Core sends the request to the Routing Service, in order to calculate the routes for the input stations pair. The GTFS Feeds, reporting available public transportation services in the city, have been already loaded by the GTFS Fetcher into the Routing Service, which can now calculate all the possible routes between the two stations in input, and return the results to the App Core. The App Core translates the received routes in the appropriate format, required to show them as paths in the map. Finally, the App Core sends the results to the User Interface, which shows to the user the map with the calculated routes represented as tracked paths.

3.4.6 User Interface

The Multimodal Navigator for disabled people application is a responsive web app that adjusts its layout depending on the device where it runs. Figure 36 depicts the mock-up of the main page of the application user interface, whose development is ongoing. The user interface shows a navigable map, which let to scroll up/down/left/right or to zoom/pan with the fingers. The map reports all the markers representing parking or car sharing stations, public transportation stations and, if any, the user bookmarks. It enables also to select the markers in order to show their details or to save them as bookmarks.

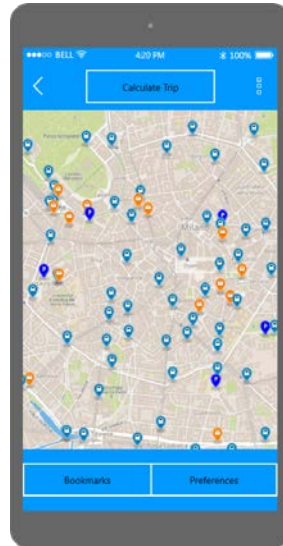


Figure 36: Mock-up of the Milan Multimodal navigator for disabled application - Home Page

By clicking on a marker, the user is able to access its detail and if it has specific services for disabled people, this information will be provided to the user, as depicted in Figure 37. Furthermore, the user can add the feature to his/her bookmarks by clicking on the plus button in the bottom-right corner of the popup.

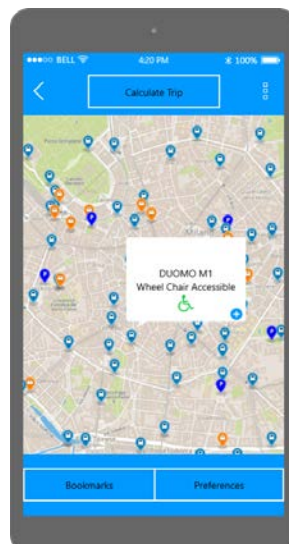


Figure 37: Mock-up of the Milan Multimodal navigator for disabled application – Feature Detail

The user can set his/her preferences by clicking on the Preferences button in the application. A specific form will appear where the user can choose the preferred means of transportation. Figure 38 shows the preferences form. It is important to underline that by checking or removing an element its features will be dynamically removed by the map. Moreover, by checking the “Disability Support Services” checkbox the map will show only the features (bus stations, metro stations, etc.) providing services for disabled people. This will also allow to provide the user with the best path to follow to reach his/her destination guaranteeing that all of the means of transportation used by the Routing Service will suite his/her needs.

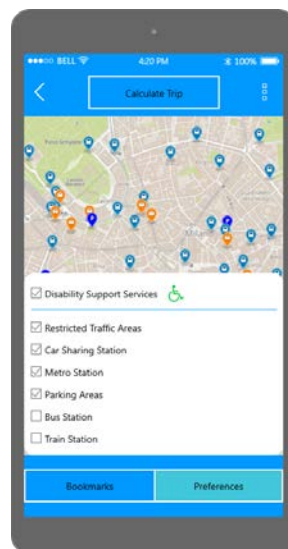


Figure 38: Mock-up of the Milan Multimodal navigator for disabled application – Preferences

By clicking on the “Calculate Trip” button the user will insert the origin and the destination of his/her path (Figure 39) and finally the application will provide the path to be followed in the map that satisfying user preferences, as depicted in Figure 40.

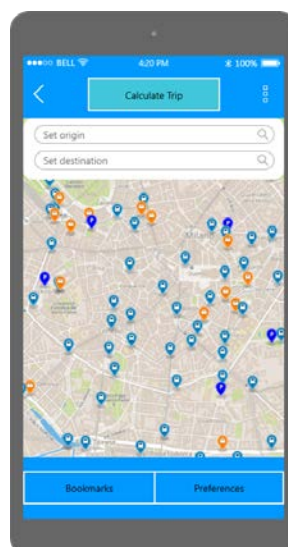


Figure 39: Mock-up of the Milan Multimodal navigator for disabled application – Calculate Trip

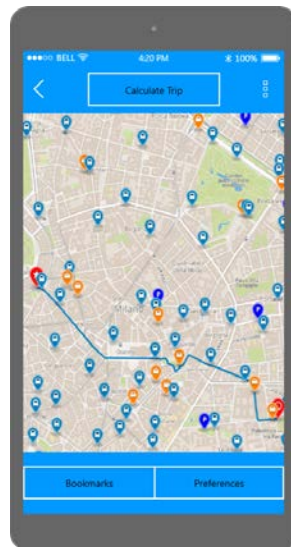


Figure 40: Mock-up of the Milan Multimodal navigator for disabled application – Final Path

3.4.7 Data Management

All the information used by the application comes Open Data provided by Municipality of Milan and from E015 APIs. Used Open Data are released under CC BYs (Creative Commons Attribution-ShareAlike) or IODL⁵⁰ (Italian Open Data Licence) licences. Instead, all the data coming from E015 APIs which use was authorized, are intended to be used only internally to the application. Furthermore, every user's preferences and/or bookmarks are stored internally within the installed application on the device and they are not processed for any purpose. Information about start and end points of routes requested by users is not collected or stored.

3.4.8 Conclusion

The main focus of the Multimodal Navigator for disabled people application is to ease the mobility and life in general of disabled people, who daily face many difficulties when moving around the Milan city, in particular when using several vehicles during their trips. The application running in their devices gives them the possibility to easily show a map with an intuitive overview of the transportation solutions and other interesting city facilities. Therefore, the SynchroniCity framework eased the full exploitation of the services and the related data provided by the Municipality of Milan, in order to build up the advanced logic at the base of the application. The development itself, in particular regarding the user interface, is still ongoing and will be completed very soon.

3.5 Carouge's Application "Smart Parking"

Carouge's application for the smart parking developed by SynchroniCity will help to reduce unnecessary traffic generated by drivers that spend a long time to find a free parking spot and will provide a convenient service to the citizens and visitors coming in Carouge by cars. Drivers will know in advance if there is an available parking spot or not; subsequently, they will be guided to a free parking spot near the destination. In particular a parking estimation service will provide drivers with the probability to find a free parking spot in a specific area of Carouge and in a certain time frame. The application will also provide real-time data of the parking availability. Thank to these functionalities, drivers can save time to find a free parking spot and the city can reduce unnecessary traffics

⁵⁰ <https://www.dati.gov.it/content/italian-open-data-license-v20>

Licence

No licence has not yet been defined in the context of this application.

3.5.1 Scenarios

3.5.1.1 Users profiles

There are no updates, since the user profiles were consolidated in D3.5 and are still valid.

3.5.1.2 Scenarios

There are no updates, since the scenarios for the application were consolidated in D3.5 and are still valid.

3.5.2 Requirements analysis

The requirements analysis has been made following the goals defined by the city of Carouge and the scenarios described in this document.

3.5.2.1 Functional requirements

The requirement analysis, made after the updates in the application reported in this document, led to no updates in functional requirements.

3.5.2.2 Non-functional requirements

Only the NFR-1 has been updated

Table 40: Non-functional requirements of Smart Parking application

ID	Description
NFR-1	The service SHOULD avoid to consume too many resources in the virtual machine where it is installed.
NFR-2	The service MUST be secure and reliable.
NFR-3	Required data MUST be accessible through a NGSI interface.
NFR-4	The service SHOULD have low latency.
NFR-5	The service SHOULD be intuitive and easy to use
NFR-6	The service SHOULD be easy to extend itself and easy to integrate into another city services.
NFR-7	The service SHOULD provide multiple languages used in Switzerland.
NFR-8	The service MUST be accessible in diverse mobile devices and web browsers.

3.5.3 Available data sources

Table 41: Data sources used in Smart Parking application

Title	Description	Open Data API	Adopted SynchroniCity Data Model	Notes
Status	Status of the parking spot from the point of view of occupancy	API	ParkingSpot	

Available Spot Number	The number of spots available	API	OffStreetParking	
Bus stop and vehicle	Real tracking of the vehicles on the public transportation stops	API	Vehicle	
Occupancy rate	An observation of traffic flow conditions at a certain place and time	API	TrafficFlowObserved	

3.5.4 Application architecture

There are no updates, since the architecture of the application was consolidated in D3.5 and is still valid. For sake of completeness Figure 41 depicts the component diagram of Smart Parking application, refer to D3.5 for any further detail.

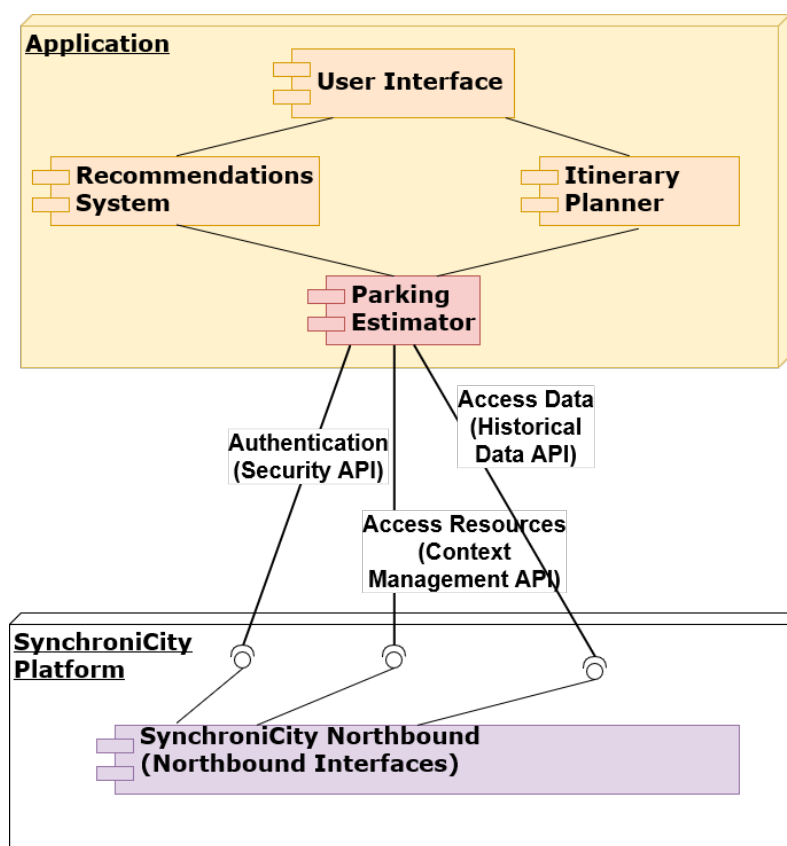


Figure 41: Carouge - component diagram of Smart Parking application

3.5.5 Sequence diagrams

This application offers several functionalities and the most interesting one is the possibility to obtain an estimation of the parking availability. This is combined with an itinerary planning and some recommendations to the users. The sequence diagram of this functionality is reported and described in this section. The following sequence diagram shows the basic transactions for the Smart Parking application involving all the components deployed in the Carouge reference zone.

Smart Parking sequence diagram

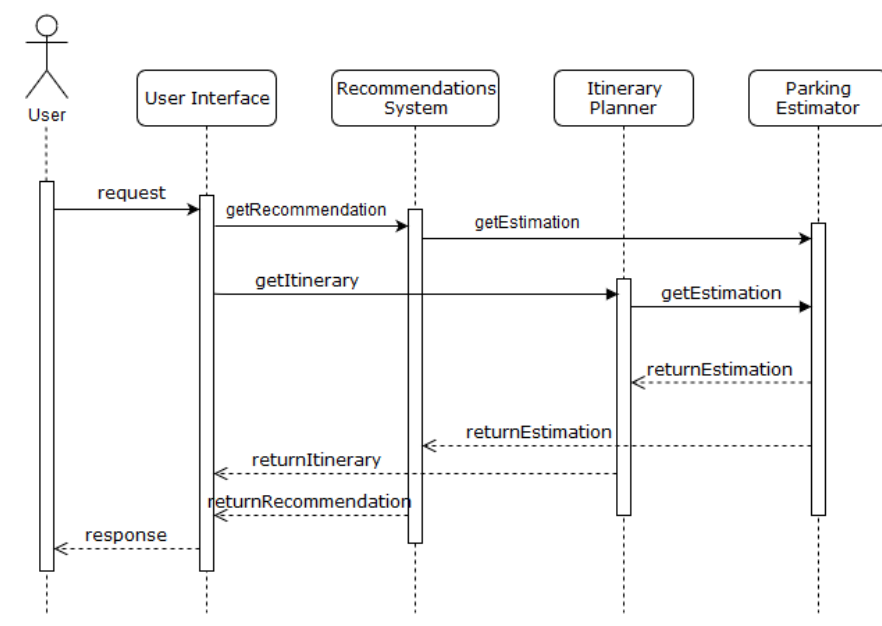


Figure 42: Carouge - Smart Parking sequence diagram of Smart Parking application

3.5.6 User Interface

The data concerning the Smart Parking application is available on a Google map reachable at the following link <https://map.cityreport.org:9443/>

The access to the map is limited with different access rights for the different users. The types of users are the citizens, the managers of the city of Carouge and the administrators. The map is in production, but a continuous improvement is currently done.

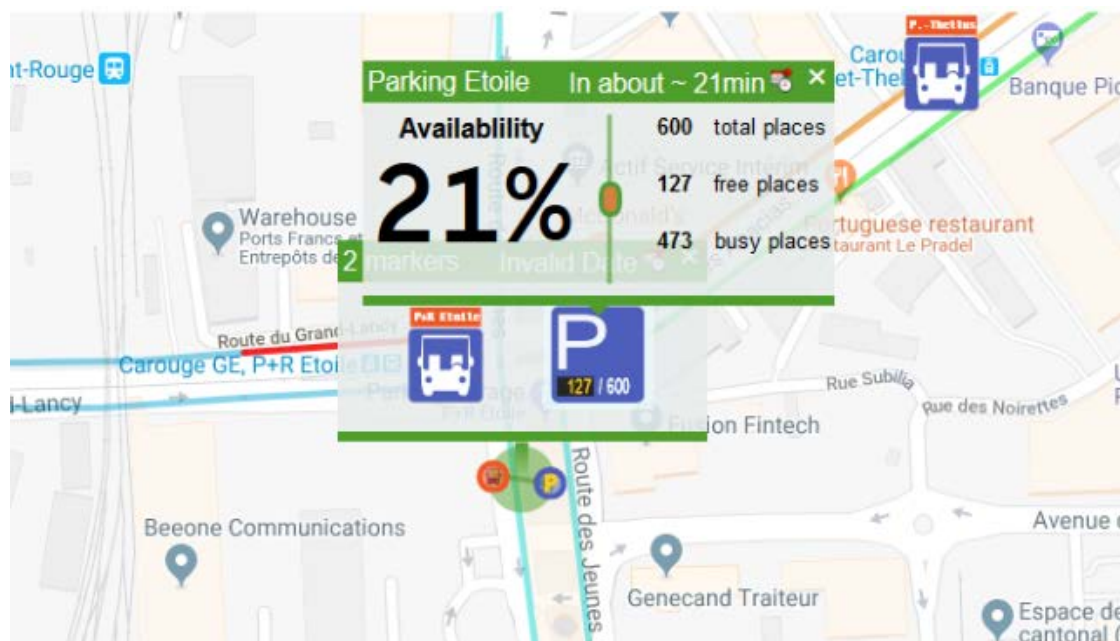


Figure 43: Carouge - a screenshot of Smart Parking application

The previous screenshot (Figure 43) shows the parking availability on a parking with 600 parking slots. The availability is displayed in percent as illustrated in the previous screenshot. To get the information for the parking availability, the user directly clicks on the desired parking icon on the map. If the user clicks on a parking icon, a popup will appear showing the parking name along with the time needed to arrive there.

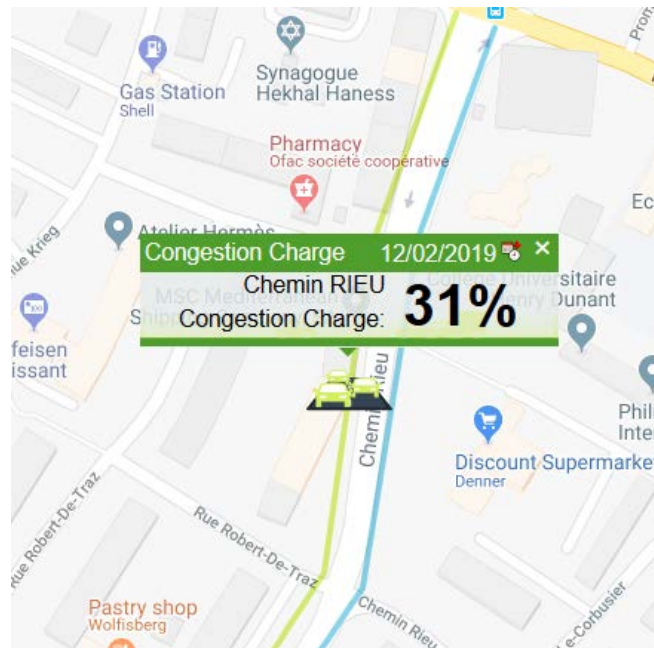


Figure 44: Carouge - a detail of Smart Parking application

On the same manner, the user can select a road, as displayed in Figure 44, and the percentage of traffic load is shown. A lot of roads crossing the Canton of Geneva are displayed on the map with the corresponding data concerning the traffic flow. The data model used is TrafficFlowObserved. This information is useful for the user when he/she drives across the city of Carouge.

3.5.7 Data Management

The access to the data is protected by KeyRock and Wilma. There is no personal or sensitive data stored in our SynchroniCity platform instantiation. So, our reference zone is fully compliant with the GDPR.

3.5.8 Conclusion

The application will help to find free parking slots and the schedules for the public transportation in the Carouge reference zone. The atomic service for the parking estimation written by ATOS facilitates the development of the application dedicated to the use case described in this document. Globally, the SynchroniCity platform and its atomic services allow to create applications like the smart parking application in a quick and standardised manner. Moreover, the reusability of the different atomic services between the smart cities involved in the SynchroniCity permits to exchange experience and knowledge, notably to the people working in the technical implementation. So, at the end, the Carouge reference zone has a very good experience with the SynchroniCity and its different components installed in Carouge. The map permits to the citizens or drivers to see in a single view all the information needed to drive or travel to and in the city of Carouge. This allows a good experience for the end-users in the Carouge reference zone.

3.6 Seongnam's Application "Commuter Parking Service"

Seongnam is designed as a satellite city to accommodate more people near Seoul, capital of the Republic of Korea. For this reason, many people who live in Seongnam commute to Seoul every day. Since Seoul is a very crowded city with 10 million people, it costs more and takes more time if commuters drive to Seoul with their own car. Yatap area has been chosen as the main service area in Seongnam because of the following reasons: Yatap station is one of the Bundang line stations, and it only takes 40 minutes by metro to go center of Seoul; People will also be able to use public parking lots near the Yatap station and more than 40 different bus lines. For this reason, Yatap district is a good area for commuters to transfer from their car to public transport. In addition, Seongnam intercity terminal and the express terminal is right next to the Yatap station.

By using the commuter parking service, people can easily find a parking space and the commute route such as minimizing the transfer time. Drivers can check the available parking space on public parking lots near the Yatap station in real-time. This service is provided through applications that enable the user to get information about connected public transportation; such as the estimated arrival time of metro line, the time schedule of intercity and express bus and intracity bus information.

Licence

All software developed except smart phone application is open sourced under BSD and NPM. Code will be made available on the KETI's GitHub webpage⁵¹.

3.6.1 Scenarios

3.6.1.1 Users profiles:

Table 42: User profiles of Smart Parking application

User profile ID	User description
User01	City mobility services: they manage the traffic in Seongnam and put in action the policy decisions about the mobility.
User02	Car drivers: they park their vehicles and attempt to transfer to different transportation such as subways, city buses, intercity buses to either enter Seoul or move to another area

3.6.1.2 Scenarios

Table 43: Scenario "Find parking place Solutions" of Smart Parking application

Scenario title	Find parking place Solutions
User profile	User02
Background	There are three large transfer parking areas at Yatap, and they show high congestion at commute time. Since users do not know the real-time information of each parking lot, if there is no space in the visited parking lot, they have to move to another one without convince.
Objective	Parking sensors have been built on three transfer parking lots located in Yatap, and real-time parking information is provided by linking sensors and platform server.

⁵¹ <https://github.com/loTKETI>

Storyline	The user moves to Yatap by car for transfer and receives real-time parking information of the three transfer parking lots through the app. It guides to move to a parking lot with low congestion.
------------------	--

Table 44: Scenario "My Route Solutions" of Smart Parking application

Scenario title	My Route Solutions
User profile	User02
Background	There are many transfer systems in the vicinity of Yatap. Many citizens park their cars in Yatap and then move to other cities or Seoul through other modes of transportation
Objective	To help users who want to use parking lots near the Yatap station. To provide real-time information so that the user can make an effective transfer to another mode of travel.
Storyline	Provides real-time transfer information according to the "My Route" set by the user. Subway arrival times, city bus schedules, and intercity bus schedules are linked in real time. Users can check the data of public transport easily, by setting the My Route once.

Table 45: Scenario "Transfer parking lot information " of Smart Parking application

Scenario title	Transfer parking lot information
User profile	User02
Background	Current system does not effectively provide detailed information (rate information, number of parking spaces, route guidance) of the transfer parking lots.
Objective	To provide static and dynamic data of the Yatap Transfer parking lots to the users effectively. To provide information and directions to the three parking lots (navigation app linkage).
Storyline	The user confirms information (including charges, etc.) of the transfer parking lots in real time and decides which parking lots they are going to use. The app provides information about the parking lot user chose and links to the navigation app user is familiar with.

3.6.2 Requirements analysis

3.6.2.1 Functional requirements

Table 46: Functional requirements of Smart Parking application

ID	Description
FR-1	The application MUST enable the creation of a user account
FR-2	The application MUST enable saving user's transportation preferences
FR-3	The application MUST provide available parking space in real time
FR-4	The application MUST provide transportation schedule which include Inter-City bus, Intra-City bus and metro information
FR-5	The application MUST record the time spent in the parking lot
FR-6	The application MUST save user meta data

3.6.2.2 Non-functional requirements

Table 47: Non-functional requirements of Smart Parking application

ID	Description
NFR-1	Every user MUST be authorized
NFR-2	Data MUST be saved on both oneM2M platform and Fiware Context Broker
NFR-3	The application MUST support multiple languages
NFR-4	The application MUST provide terms and conditions before providing services

3.6.3 Available data sources

Table 48: Data sources used in Smart Parking application

Title	Description	Open Data API /	Adopted SynchroniCity Data Model	Notes
Parking Spot Number	Number of available parking spaces, exact location and real time status of parking spot	API	OffStreetParking, ParkingSpot	
Public transportation	Public transportation data (inter-city bus schedules, intra-city bus schedules, metro schedules, etc)	API	BusStop, BusLine	

3.6.4 Application architecture

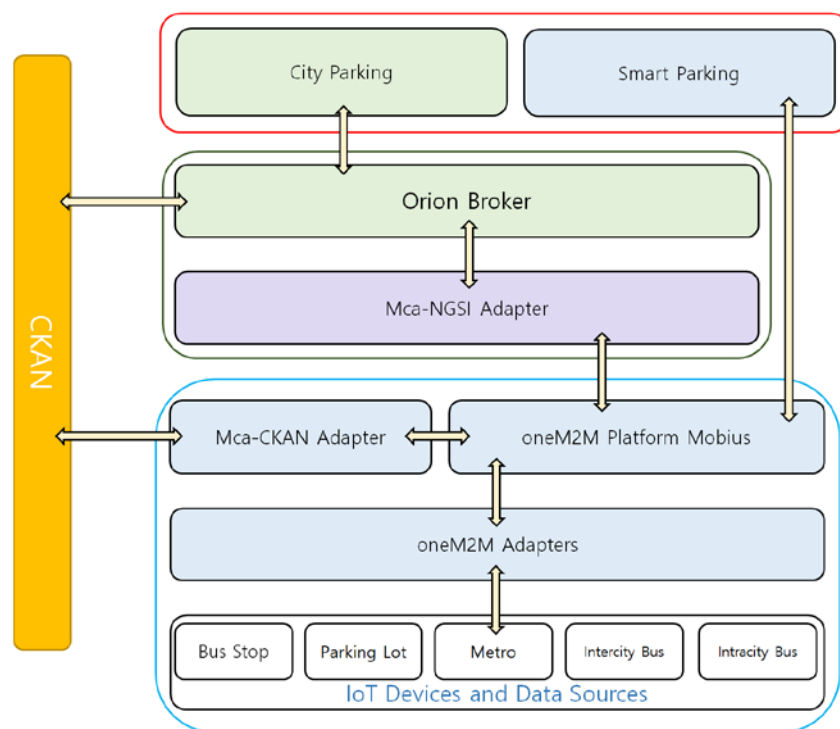


Figure 45: Seongnam SynchroniCity Platform Architecture

Overview

Seongnam SynchroniCity Platform Architecture has two User Application Services and a list of Adapters that translate the raw data from the IoT devices into either oneM2M[16] or NGSI model. The data generated by the IoT Device layer is stored in a CKAN open data market place. The Orion Context Broker is directly connecting with CKAN using CKAN's plug in.

Application Components

- **City Parking** is a user application service designed using Ionic Cordova[17]. It supports Web, Android and iOS devices. The application provides users with Map Service, Parking-Lot Service and Transportation Schedule; for more details refer to Section 3.6.6
- **Smart Parking** is a user application service designed for Android and iOS devices which communicates with Mobius[18]. It saves user data such as transportation, user preference information on the Mobius. The application also gets the real-time data of parking lot, parking spot, schedule of Inter-City bus, Intra-City bus and metro from Mobius.
- **Mca-NGSI Adapter** is an interworking proxy for the Mobius and Orion Context Broker. Mca is, in oneM2M standard, communication flow between an application entity and a server platform entity. Using subscription/notification feature in Mobius, whenever the data is created/updated in the Mobius, it sends notifications to the Mca-NGSI Adapter. The Mca-NGSI Adapter then modifies the data format to fit Orion Context Broker and requests to store it to the Orion Broker over NGSI.
- **Mobius oneM2M Platform** is an open source IoT server platform based on the oneM2M standard. As stated in oneM2M specification, Mobius provides common service functions such as registration, data management, subscription & notification and security as middleware to IoT applications of different service domains.
- **Mca-CKAN Adapter** is a plugin tool to transfer oneM2M compliant data to CKAN in accordance with administrator's setting.
- **oneM2M Adapters** is a translating tool that provides communication service between heterogeneous IoT devices and oneM2M server platform by following interface which defined in oneM2M specification.
- **IoT Devices and data sources** are a group of sensor services or devices that generate data at the lowest level which then remap by oneM2M Adapter to store in oneM2M compliant Mobius platform. In this scenario, we have parking sensor devices to generate parking spot status such as occupied or available, Inter-city Bus, Intra-city bus, bus stop, metro are services that generate the data regarding transport information, all this data is tunneled to oneM2M Adapter for translation.

Component Diagram of Applications

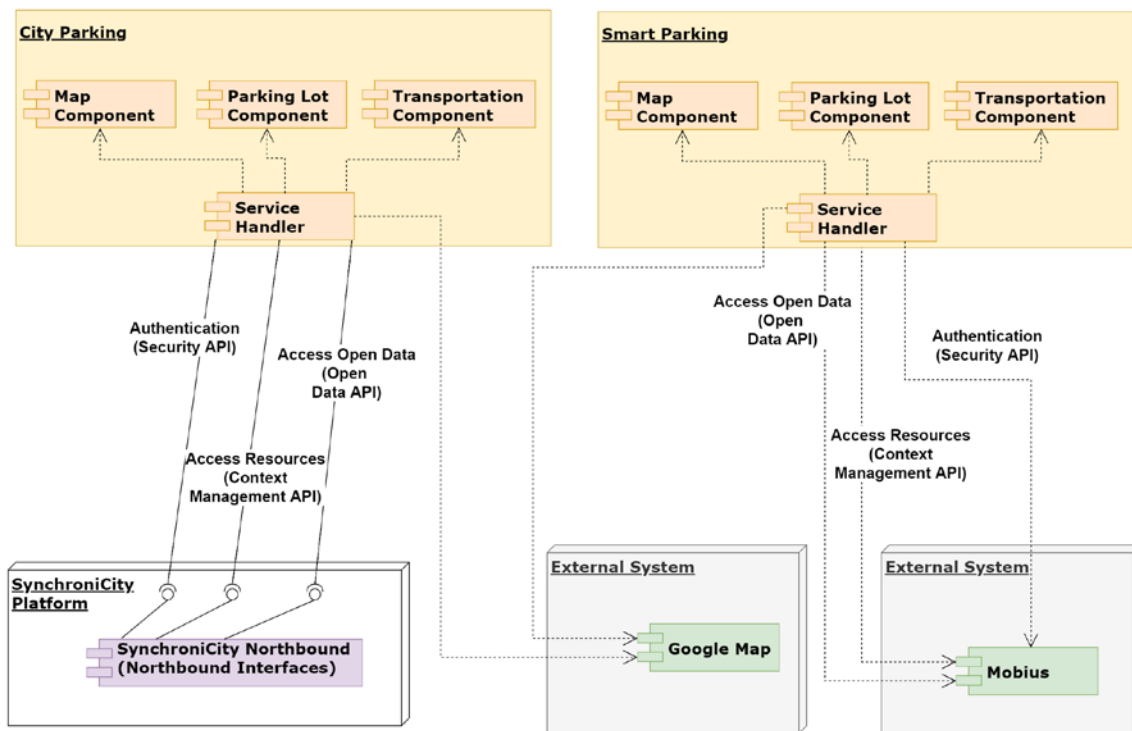


Figure 46: Application Architecture

Table 49: Application Components in Seongnam SynchroniCity platform

Application Components	Rationale
Map Component	Map Component uses Google Map-based API and provide services such as map view, map direction, parking hot spots markers and Transport hot spot markers.
Parking Lot Component	Parking Lot Component provides service selection of parking lot, notification of update and suggestion.
Transportation Component	Transportation Component provides services such as selection of transportation medium, notification of update and suggestion.
Service Handler	Service handler is a gateway in the application which is dealing with all incoming and outgoing requests.

3.6.5 Sequence diagrams

- **Sequence Diagram of City Parking:** This sequence diagram depicts interactions among the components of the application in order to provide City Parking functionalities such as Map service, Parking lot service and Transportation schedule service.
 - **Step 1:** oneM2M Adapter requests data from IoT Device Services.
 - **Step 2:** Mca-NGSI Adapter creates the subscription resource on the Mobius. By using subscription/notification feature, whenever the data is created/updated in the Mobius, it sends notifications to the Mca-NGSI Adapter.

- **Step 3:** IoT Device Services respond with raw data that they generated, once the data is available at the oneM2M Adapter the adapter will then send the data to the Mobius by using interface which defined in oneM2M specification.
- **Step 4:** The oneM2M Adapter will send save request to the Mobius.
- **Step 5:** Mobius would then notify the Mca-NGSI Adapter for all updates.
- **Step 6:** Mca-NGSI Adapter would then extract the data from the notification and translate it into NGSI format. The formatted data is then transferred to Orion Broker.
- **Step 6:** City Parking application would send either event and schedule-based data requests to Orion Broker.
- **Step 7:** Orion Broker would then respond with data appropriate for the request.

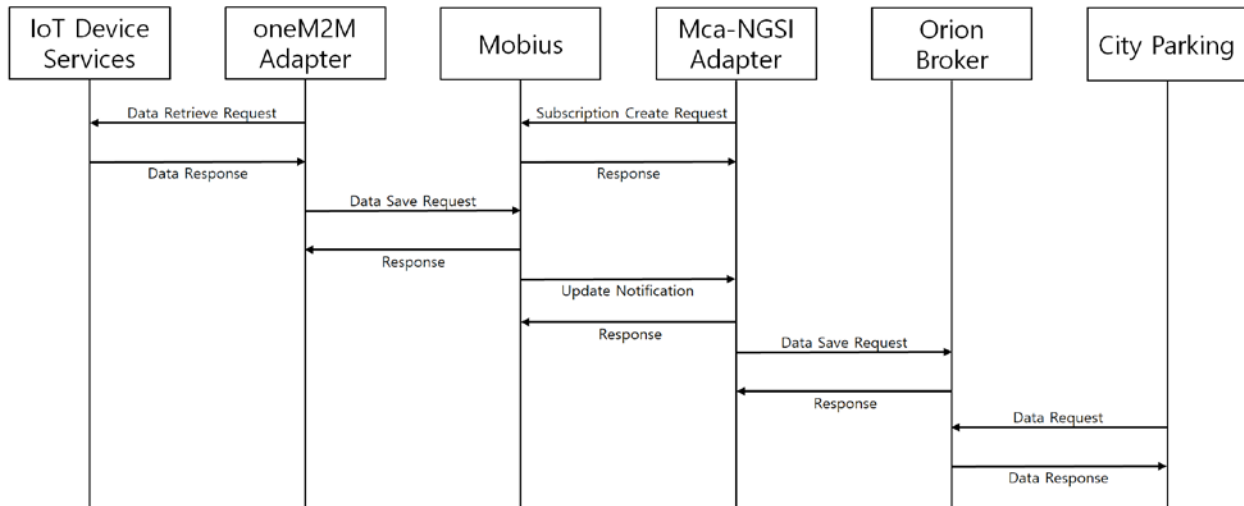


Figure 47: Sequence Diagram with City Parking

Sequence Diagram with Smart Parking: This sequence diagram depicts interactions among the components of the application in order to provide Smart Parking functionalities such as Map service, Parking lot service and Transportation schedule service.

- **Step 1:** oneM2M Adapter requests data from IoT Device Services.
- **Step 2:** IoT Device Services respond with raw data that they generated, once the data is available at the oneM2M Adapter the adapter will then send the data to the Mobius by using interface which defined in oneM2M specification.
- **Step 3:** The oneM2M Adapter will send save request to the Mobius.
- **Step 4:** Smart Parking application would send either event or schedule-based data requests to Mobius.
- **Step 5:** Mobius would then respond with data appropriate for the request.

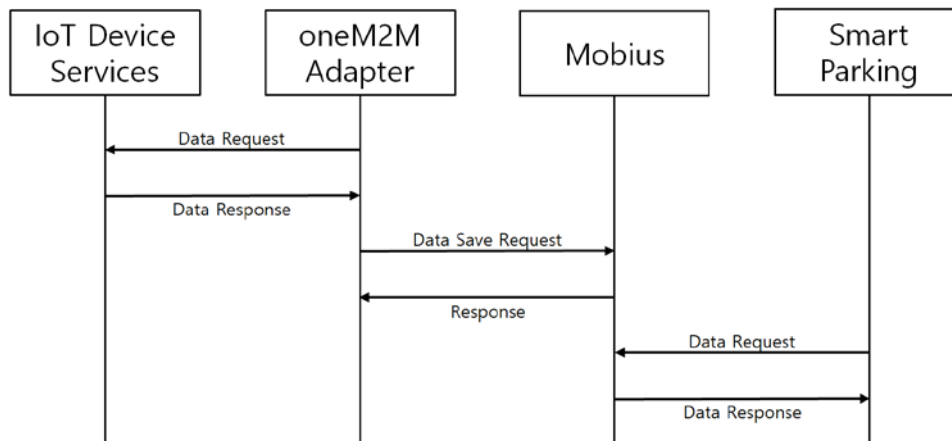


Figure 48: Sequence Diagram with Smart Parking

● **Sequence Diagram with CKAN:** This sequence diagram depicts interactions among the components of the application in order to store data in CKAN.

- **Step 1:** oneM2M Adapter requests data from IoT Device Services.
- **Step 2:** Mca-NGSI Adapter creates the subscription resource on the Mobius. By using subscription/notification feature, whenever the data is created/updated in the Mobius, it sends notifications to the Mca-NGSI Adapter.
- **Step 3:** IoT Device Services respond with raw data that they generated, once the data is available at the oneM2M Adapter the adapter will then send the data to the Mobius by using interface which defined in oneM2M specification.
- **Step 4:** The oneM2M Adapter will send save request to the Mobius.
- **Step 5:** Mobius would then notify the Mca-NGSI Adapter and Mca-CKAN Adapter with updates.
- **Step 6:** Mca-NGSI Adapter would then extract the data from the notification and translate it into NGSI format. The formatted data is then transferred to Orion Broker, while Mca-CKAN Adapter would extract the same data as Mca-NGSI and store the data in CKAN, following CKAN system admins settings.
- **Step 7:** Orion Broker would store the data updates in CKAN following the CKAN system admin settings.

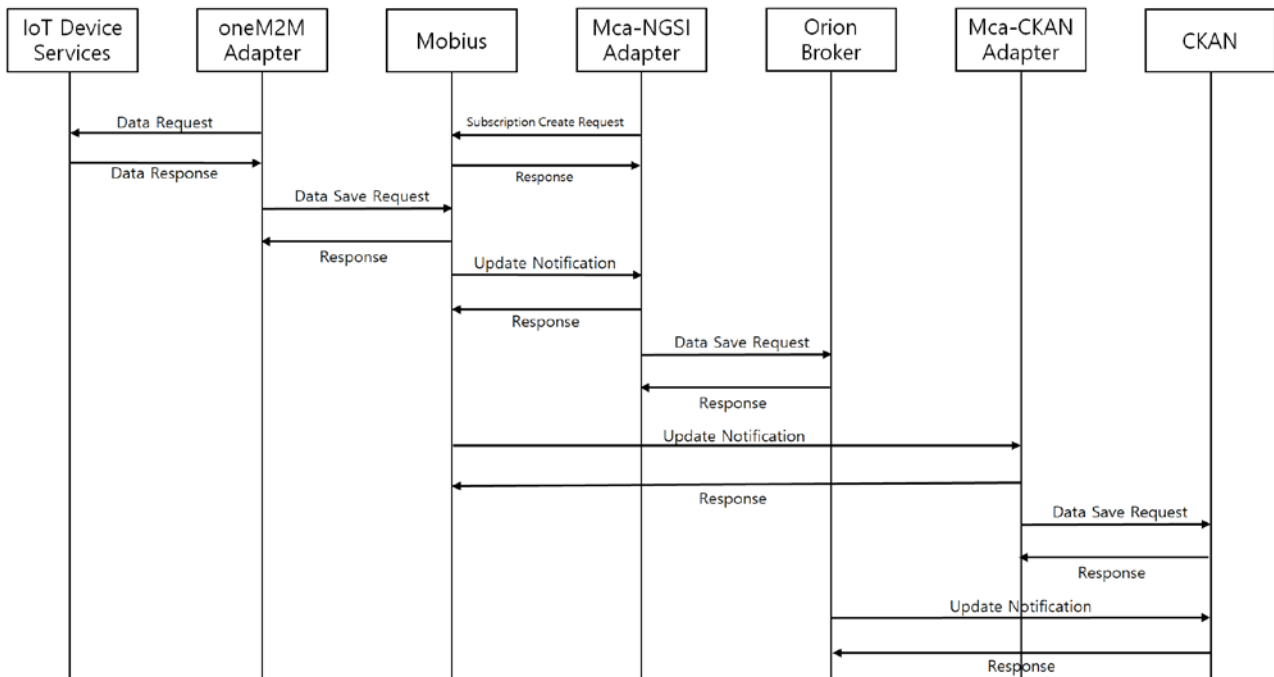


Figure 49: Sequence Diagram with CKAN

3.6.6 User Interface

Seongnam provides two applications: Smart Parking and City Parking. As described in section 3.6.4, Smart Parking and City Parking get information from oneM2M based platform and FIWARE Orion Context Broker respectively.



Figure 50: First Screen of Smart Parking

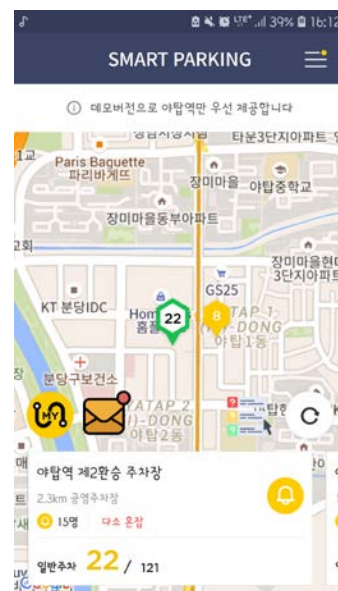


Figure 51: Main Screen of Smart Parking

Figure 50 and Figure 51 are from Smart Parking application. By using the application, the user can check not only real-time parking availability but also the schedule of transportation. The Smart

Parking application also provides service called My Route which allows the user to save and check the preferred transportation schedule.



Figure 52: Parking Lot Check in City Parking

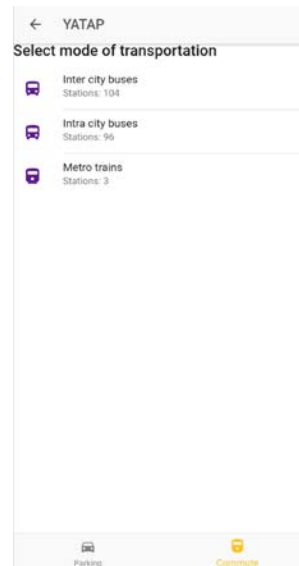


Figure 53: Transportation information in City Parking



Figure 54: City Parking Tablet Screen Capture

Figure 52, Figure 53 and Figure 54 are from City Parking. The second application Seongnam provides is called City Parking. The City Parking application provides similar services as Smart Parking application but through the different endpoint. The City Parking application supports Android, iOS and Web application.

3.6.7 Data Management

Authorization

Basically, user data is saved on Mobius. As oneM2M specification has authorization control, Mobius gives limited access to the user data.

Authentication

The application manages login by using Google OAUTH⁵².

Application saves the following user data on the server

- Email
- Licence plate number
- Preference of transportation

3.6.8 Conclusion

Seongnam, a Non-Reference Zone of SynchroniCity, is deploying SynchroniCity platform from 2018. This city is focusing on extending their legacy platform by using both SynchroniCity platform and oneM2M, global standard of IoT, platform. In this point, Seongnam is developing interworking proxy to contribute to the SynchroniCity project. oneM2M is widely used and well known IoT standard in Korea. Busan, Goyang and Daegu city in Korea built their smart city infrastructures based on oneM2M. As another example, United Kingdom made One Transport based on oneM2M⁵³. The interworking proxy used in Seongnam's application is designed to translates protocols from oneM2M to SynchroniCity platform. Using subscription/notification feature of oneM2M specification, whenever the data is created/updated in the oneM2M platform, it sends notifications to the Mca-NGSI Adaptor. The Mca-NGSI Adaptor then modifies the data format to fit SynchroniCity platform and requests to store it to the SynchroniCity platform over NGSI. The City Parking application which has been mentioned above is getting data from SynchroniCity Platform.

In 2018, Seongnam SynchroniCity platform is verified and certified by TTA (Telecommunications Technology Association). Some examined point are as follows: Connection of parking sensors, Mca-NGSI Adapter, real-time updates. TTA has also verified the dynamic nature of both the applications like the reflection of the information in the application with the change in the parking spot status, notification of the transport information.

⁵² <https://developers.google.com/identity/protocols/OAuth2>

⁵³ <https://onetransport.io/>

4 Application Theme: Community policy suite

Main aim of this application theme is the design and implementation of an approach to develop innovative IoT-based solutions that enable local authorities abilities to manage policies, improve decision making and help with the administration of the city. Leveraging on real-time information provided by IoT sensor and on historical collected data, cities can 1) identify hidden problems to be solved or new and more efficient solution for well known problems and 2) improve the efficiency of day-to-day operational functions.

Three RZs participate in this application theme are Carouge, Manchester and Porto. In particular:

- Manchester aims to break information silos by the application of an agile approach to city governance, in order to obtain a more efficient system to coordinate cross-departmental teams; this objective will be reached by leveraging on existing infrastructures and data and on the application of a new methodological approach. (Section 4.1)
- Porto aims to implement a solution to enable a proactive approach for problem solving and for a more agile management and policy processes, but also to increase transparency and to enable a bidirectional interaction and communication between the Municipality of Porto and the local communities, citizens, companies and institutions; the objective is to support the digitization process and to break information silos of the traditional approaches, in order to realise a integrated and more effective management system. (Section 4.2)
- Carouge aims to realise a noise monitoring application in order to monitor noise pollution and to detect which commercial activities (such as bars and restaurants) have a significant impact on noise level; this application will help Carouge to identify noise sources and to adopt adequate decisions in order to reduce the generated noise. (Section 4.3)

It is important to underline that Manchester and Porto are approaching this application theme through the implementation of new methodologies, whereas Carouge is designing a more classic application to facilitate day by day activities.

4.1 Manchester's Application "Agile Governance"

OVERVIEW

The Agile Governance methodology, approach and co-creation is covered in detail in D3.5, alongside various scenarios that were explored during the earlier period of the project.

This reiteration deliverable provides an update on the Manchester application that has taken place since D3.5. was issued.

CO-CREATION

Further co-creation with Manchester City Council stakeholders was required, as we identified different service needs that could be served by the software provided as baseline application within the Community Policy Suite theme.

The aim was to test the BronzeLabs software in September 2018, with users in the appropriate departments to show multiple team working and on-boarding of different data sources to the platform.

Technical delays meant that in Autumn 2018 we were only able to provide the software functionality - through the on-boarding of air quality sensor data in real time onto the platform - and to create virtual teams, and a series of "jobs" or other workflows for a cross-disciplinary team within the city. This test scenario was presented at the Antwerp project review.

Following on from this we looked to work with an extended range of teams within the city council to identify new scenarios for using the platform.

These additional co-creation activities included a visualisation and ideas workshop which we had undertaken prior to the review, as we wanted to include a number of new scenarios in the agile governance theme, following delays in the proposed LCO (Local Care Organisation) reorganisation proposed in D3.5.

At this initial scoping workshop we invited the following:

- city centre management
- waste management
- Greater Manchester Police
- events management
- Transport for Greater Manchester
- customer relationship management
- air quality officers

We were aware that it would not be feasible to work with all of the new scenarios identified within this co-creation activity, but we wanted to broaden the awareness of the project within the city - and it was successful in identifying a range of potential use cases for data-driven agile governance.

MANCHESTER INTEGRATION

Following the completion of the CityVerve⁵⁴ programme in June 2018, it was vital that we had a roadmap for continuing availability of city real time data that could be used by the Manchester pilot.

Our IoT data platform provider BT, agreed to retain the platform and access to it for the duration of SynchroniCity. This enabled us to work with Digital Catapult on the conversion layer and a more durable endpoint, which is detailed in Work Package 2.

PROBLEM IDENTIFICATION TO TEST THE METHODOLOGY

As detailed above, we have continued problem identification following the methodology detailed in D3.5.

The following paragraph is included again as an aide memoire - as it was important that the criteria for problem identification follows the same format as previous work.

Over the last 18 months MCC and MMU have been identifying with stakeholders a range of potential “problem statements.”

Throughout this, it has been necessary to work with stakeholders to ensure that the “problem statement” to be tested within the baseline application has the following characteristics:

- *“Buy in” from internal stakeholders to ensure that there is a commitment to use the community policy suite tool in a real world situation*
- *Availability of “real time” data for populating the application*
- *Need to develop “virtual team” structures to improve decision making*
- *Timescales of the “problem” aligned with SynchroniCity deployment timetable (D3.4)*
- *The problem is specific and the impact is measurable*

⁵⁴ <https://cityverve.org.uk/>

As identified above the LCO use case detailed in D3.5 is no longer the one that we are pursuing as the reorganisation into the local care organisations took far longer than was initially planned and therefore we were informed that they were unable to introduce any new technologies in line with the timeline of delivery required by WP3.

Our initial concern with this, was alleviated through the co-creation workshop above, where we identified some smaller, more manageable use cases.

In some ways, this allows us to move on from the initial concept of D3.5 which was determined upon the deployment of the particular software tools provided by Bronzelabs, to D3.6 which is a more advanced version of the scenario, where a range of city problems can be addressed through different datasets, and in line with the description of work for the theme, look at a range of software solutions to provide different applications and visualisations, arising from the service needs. This included, looking at the atomic service provided by Grafana, who have worked with Digital Catapult to look at consuming the Manchester air quality data and mapping it accordingly. We have also been able to identify 3rd party providers who the city has been working with in other contexts, and identifying how they can consume MCC data.

For the scenario that we have prioritised since the delays in the software deployment, we have worked closely with our Freedom of Information team, Events Management Team and Customer Relationship Management teams. These teams had a lot in common in that they are all dealing with a wide range of external clients, customers and the general public, and with requests coming in from a range of different channels.

We shadowed the FOI (Freedom of Information), Events and CRM staff and as part of the broadening out of the service to a wider problem base, MMU will process map where agile governance can provide added value across these service areas. We continue to identify the needs to develop the data platform with Bronzelabs and the new scenarios are in line with this deployment.

The real value here has been in our stakeholder engagement and looking at how we can provide data-driven solutions to the wide range of challenges that these teams have identified.

Once the Bronzelabs software is delivered we would look to test the team building and workflow within a controlled environment with these teams.

In order to ensure that we have a model for supporting the other scenarios coming from these teams, where the real interest is in management information in order to make business decisions on resource deployment and business cases for investment, we are working with MMU to provide some advanced visualisations, using people and vehicle counting data, aligned with the historical events data that we have.

Licence

Closed source. Licence provided free of charge for the duration of the SynchroniCity project.

4.1.1 Scenarios

4.1.1.1 Users profiles:

The fourth user type from D3.5, Citizen, has been removed. It was decided for Manchester to have the MCC officers as the end-user to ensure the scope of the application is focused and achievable.

Table 50: User profiles of Agile Governance application

User profile ID	User description
User Manchester 1 (City Council Officer)	The user is a City Council Officer who works in the Growth and Neighbourhoods directorate. Operational officers work within the processes that have been agreed for their service, but the same problems occurs time and again. They need access to data and be able to analyse citizen needs Officers know what the problems are and are unable to alter their patterns of working. It can be difficult for the team to get an overview of the problems and ensure that they are making the best decisions with the limited data available and the reduced resources to address the problem. Cross-department communication can be difficult and knowledge of who has the best skills and experience to address a specific challenge in the city is usually based upon their tacit knowledge and personal connections. They also do not have good feedback loops to show effectiveness of a policies applied in their day to day work.
User Manchester 2 (City Council Team Leader)	Team leader is working with a diffuse team that are set in their ways and they meet them only occasionally. They have limited resources and staff and have to use them in the most effective way. They know that certain staff have “easier” workloads and want to encourage better understanding amongst the team. They want to ensure that the resource is in the right place at the right team and the data can help them do this. They need access to one transversal platform in order to give access to the necessary data and reduce piecemeal research and guesswork .
User Manchester 3 (Policy Maker)	Senior manager is receiving regular complaints from citizens and from elected politicians and each one needs investigating taking up a lot of time. There is poor record keeping , and sometimes the monthly data from subcontractors does not give a picture that is the same as people are feeling in the city, they need to understand the present situation in the city. On the other hand, the service has recently improved with more resources and better training - yet people's complaints are based on isolated incidents. The policy maker wants to use good practice in this part of the city to influence strategy elsewhere in the city. There is a need for one transversal platform for city data.

4.1.1.2 Scenarios:

Table 51: Scenario "Monitoring the City / Real-Time Response" of Agile Governance application

Scenario title	Monitoring the City / Real-Time Response
User profile	User Manchester 1 – City Council Officer
Background	John is meeting his team for their weekly meeting reviewing the status of projects and new issues that need addressing
Objective	John wants to get a better handle of what they should prioritise and to gain better situational awareness of current activities.
Storyline	John is new in the team. He is dealing with a situation that he is sure must have

	been encountered before. He is able to bring up the necessary data on dashboard and filter it according to location. The software generates real-time insights from the data-feed. He can compare with the data from his own situation and identify similarities. He is able to contact the appropriate team to find out how the situation was solved.
--	--

Table 52: Scenario "Management Information" of Agile Governance application

Scenario title	Management Information
User profile	User Manchester 2 – Team Leader
Background	Ann has recently joined a team that has been looking after events in the city for a number of years. They have an established approach to event management and will have a debrief after each event.
Objective	Ann wants to easily assign tasks, assess the effectiveness of their response and enable them to better plan for future events where there are unexpected numbers attending.
Storyline	Ann has asked for data to be provided to the Agile Governance software for the weekend of the event. She is then able to sit down with her team to identify how the data confirms and enhances their understanding. They had previously underestimated how quickly people arrive and depart the event, and this gives them valuable information for planning future events. She can also recommend to the agile governance software what new data-feeds would help to gain a better understanding of the issue.

Table 53: Scenario "Dynamic Strategy" of Agile Governance application

Scenario title	Dynamic Strategy
User profile	User Manchester 3 – Policy Maker
Background	Atif has been working for the council for twenty years and holds the position of Strategic Director. The Council has grown in size and has become increasingly siloed with this expansion. He feels he does not have enough awareness of how his department is addressing the systemic challenges and how effectively they are implementing policy.
Objective	Atif wants to assess how successfully policy has been implemented across the various departments that he oversees.
Storyline	Atif accesses the Agile Governance dashboard that enables him to view the variety of projects that are in progress and the associated policy with each. He reviews insights from officers that are tackling the problems on the ground and reviews the suggestions for policy amendments in the Live Policy Document. He can also view what issues citizens are most concerned with and the areas in the city that are affected. He is able to compare these with historical figures and see that certain things have improved, but other areas have bigger problems because of changes in population density since the strategy was written. He is able to present a refresh of the strategy based upon these changes and able to show where improvements have been successful, and where the new priorities are meant to be.

4.1.2 Requirements analysis

The requirement analysis, made after the updates in the application reported in this document, led to no updates in functional and non-functional requirements.

4.1.3 Available data sources

There are no updates, since data sources for the application were consolidated in D3.5 and are still valid.

4.1.4 Application architecture

Bronze Software Labs is deploying a version of its smart city middleware, framework.⁴², with features specifically developed for SynchroniCity for use by Manchester. Architecture depicted in Figure 55 in common between Manchester and Porto applications. framework.⁴² is a cloud based, IoT platform that will accept NGSI formatted data from the SynchroniCity API to answer the various scenarios put forth by the cities.

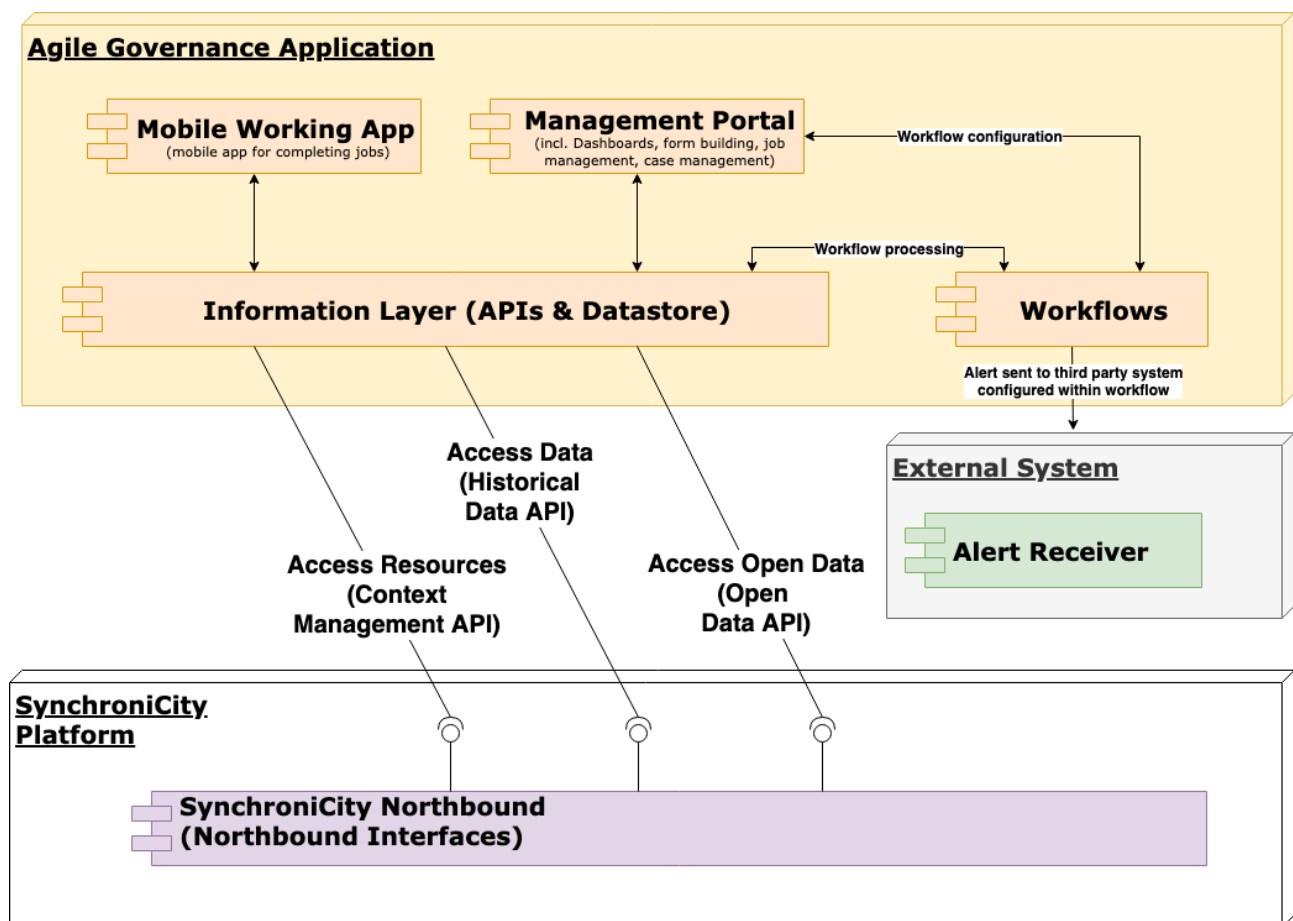


Figure 55: Manchester - component diagram of Agile Governance application

Overview

The architecture of framework.⁴² revolves around a centralised information layer which receives and stores data from user input and third-party APIs. This information is then distributed internally to both the management portal and the mobile working application. Workflows run in realtime on the server, data is processed by workflows as the information is received via the SynchroniCity northbound interfaces.

Application Components

- **Information Layer:** This consists of two main parts, the Framework.⁴² system has a centralised datastore which is used to distribute information to the management portal, mobile application and third-party systems via workflows. The second component is the public and private APIs that allow data to flow between applications securely.
- **Mobile Working App:** This is an android application that can be used for completing remote work digitally. Workers can view their schedule of work and complete forms attached to a job which are submitted back up to the portal on completion.
- **Management Portal:** The portal contains the bulk of Framework.⁴²'s functionality. Users can set up customisable dashboards, which supports both high-level and more detailed analytics. The forms to be completed by mobile workers are created in the portal, with a simple drag and drop interface, the form builder also includes a versioning mechanism to allow management of form iterations and rollouts.
- **Workflows:** Workflows simply allow a user to set up a trigger, and an action. There are many kinds of triggers, including: a job has been completed, or a sensor threshold has been exceeded. If this trigger occurs, the chosen actions would then be automatically sent. Actions can include a notification to a third-party system, an email or SMS alert, or even creation of a job. Workflows are processed in real-time as data comes into the system, as the live data is fed from the SynchroniCity platform it is compared against active workflow triggers.

External Systems

- Workflows within the system can be used to issue alerts to a user-configured external system. This is useful where setting threshold triggers, or job management alerts. In addition to this, an SMS action exists which sends a text message to notify the appropriate users, this action is achieved through the Twilio web-service.

4.1.5 Sequence diagrams

The following sequence diagrams illustrate the technical processes that are carried out to fulfil a user interaction within the system.

- **Subscription Creation:** Creation of a new data subscription in the management portal, the chosen entity data is then queried and processed by the framework.
- **Workflows:** This diagram depicts both the user configuration of a workflow within the framework, as well as the ongoing automated processing that occurs to check whether a workflow should be triggered by incoming data.

Subscription Creation

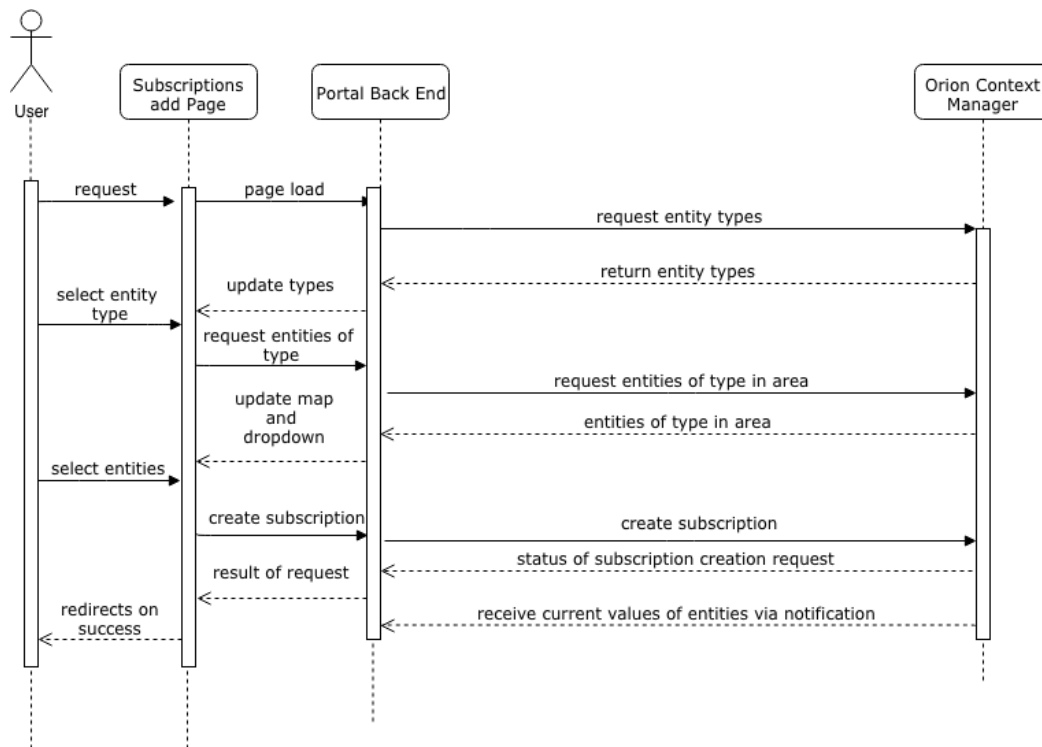


Figure 56: Manchester - Subscription Creation sequence diagram of Agile Governance application

A user has the ability to create a new data subscription in the management portal, upon opening the creation page the framework loads the available entity types. After a user selects their desired entity type (data series) entity data is downloaded and the user is presented with a map showing all available entities (sensors) for their working area. The user can select a subset or all of the available sensors to generate their new data subscription. This subscription then means the sensor data is continually monitored going forwards, the data can be fed into a workflow to trigger actions or set up as part of a policy.

Workflows

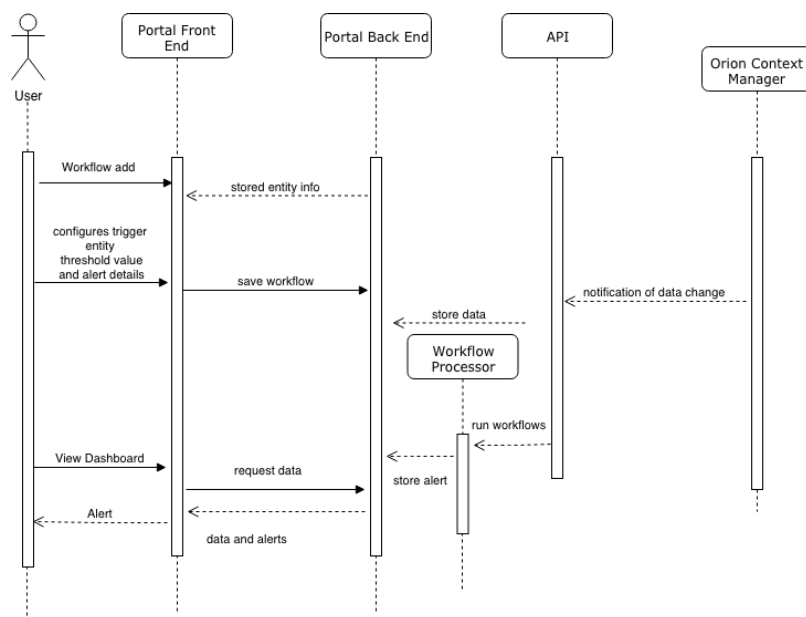


Figure 57: Manchester - Workflows sequence diagram of Agile Governance application

Workflows begin with user configuration; an administrative user can create a workflow which consists of two main parts. A trigger and an action, the trigger is a customizable element that defines when automated action is required. This could be based on incoming sensor data, job status or form input as a few examples. Then a user can add several actions which could be an alert via email, SMS or a trigger to a third party system.

Once the workflow has been configured and made active, it is processed every time the relevant data changes within the system, after a job is completed for example.

4.1.6 User Interface

Dashboard

The dashboard section of the portal offers a customizable drag and drop interface for creating both high and low level analytics. Users have a drop down towards the top right which they can use to switch between their custom dashboard pages, as well as create a new dashboard page. After creating a page, the plus icon in the bottom right can be clicked to bring up a menu of available “widgets”. These widgets are drag and drop elements which can be added to the page, examples include: maps, charts and numeric counters. The size of the widgets can also be altered, users could for example have four high-level counters across the top reporting on overall activity, with a map below providing a method to drill-down into lower level data. The map seen in Figure 58 is showing a policy that contains “on street parking” data within Manchester, a policy could contain several data series so it can be assessed whether there is a correlation. In this case the dropdown beneath the map can be used to toggle the visible data series. Underneath the map to the right, is a time scrubber which allows users to browse the data across the period defined when creating the policy.

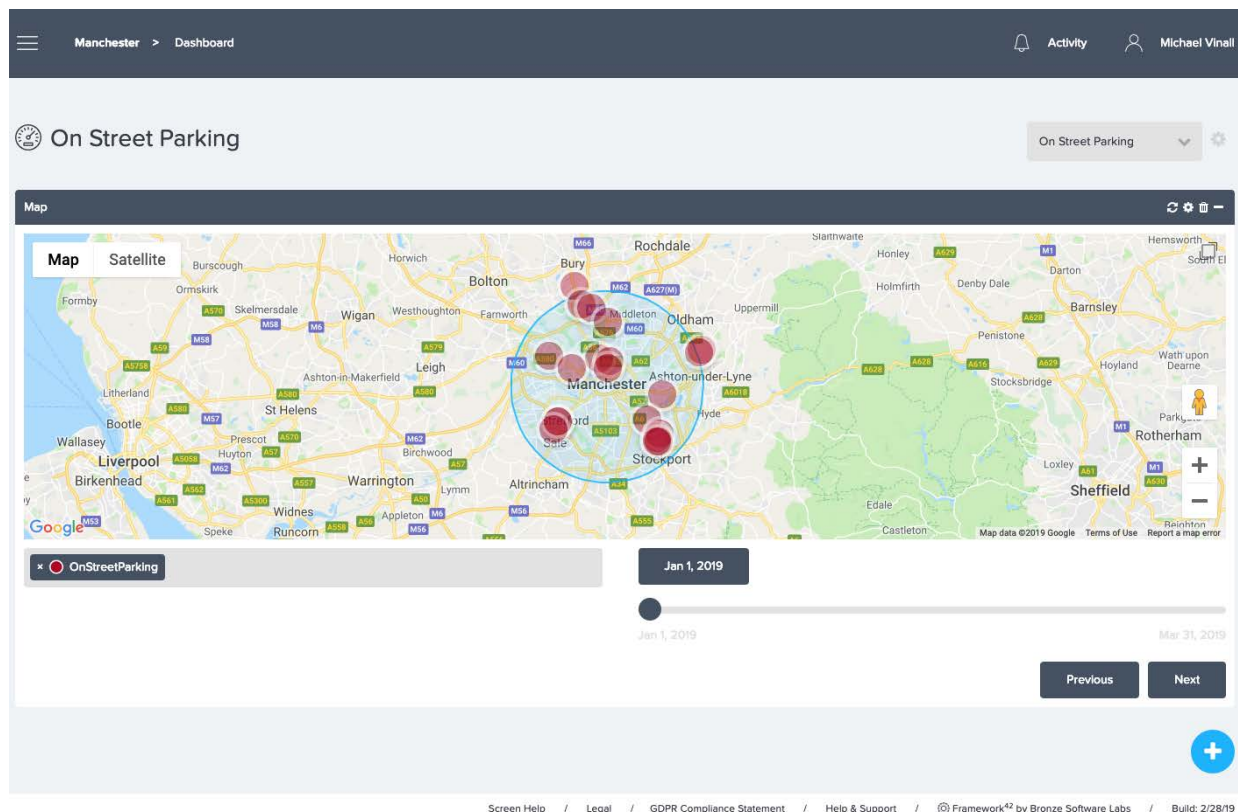


Figure 58: Manchester - High-level view of “On Street Parking” data within the dashboard

Policy Configuration

The policy configuration section is relatively straight forward, it consists of a fairly basic form that allows users to specify what data the policy should consist of, or take into account. Over what period the policy will be monitored, and which users should have visibility of the policy that is being created. Below this is where users set up the live policy document, a document made up of sections which will allow evidence-based decision making and documentation. Sections within this document can be accessed by the configured users as depicted in Figure 59.

The screenshot displays the 'Manchester - New policy configuration' interface. At the top, there's a navigation bar with 'Manchester > Policies > Add Policy' and user information 'Michael Vinnall'. The main heading is 'Michaels Parking Policy' with 'Download PDF' and 'Back' buttons. The form is divided into two main sections: 'Policy Details' and 'Live Policy Document Sections'.

Policy Details:

- Name:** Michaels Parking Policy
- Users:** Amir Khorasani, Adrian Slatcher, Michael Vinnall
- Subscriptions:** Michaels Parking Subscription
- Duration:** This Year

Live Policy Document Sections:

- Main:** Michael Vinnall
- Conclusion:** Amir Khorasani

At the bottom, there's an 'Add Policy' button and a footer with links: Screen Help / Legal / GDPR Compliance Statement / Help & Support / Framework 42 by Bronze Software Labs / Build: 2/28/19.

Figure 59: Manchester - New policy configuration

Subscription Configuration

A subscription is created to indicate data should be collected and processed from a group of sensors or entities. To begin, users provide their new subscription with a name in the portal. Then, they select their first entity type which in this case is the data series they are interested in. For Manchester some examples would be traffic flow or on street parking. After selecting this, a map is populated which displays of the available sensors returning data of the chosen type. A user can then choose to have all sensors of that type within their policy, or select them individually based on their location on the map, or the name attached to the sensor. Figure 60 shows a user who has selected an entity type of "On Street Parking" and now has visibility of those sensors on the map. Once a subscription has been created, you can view the subscription to check when data was last received.

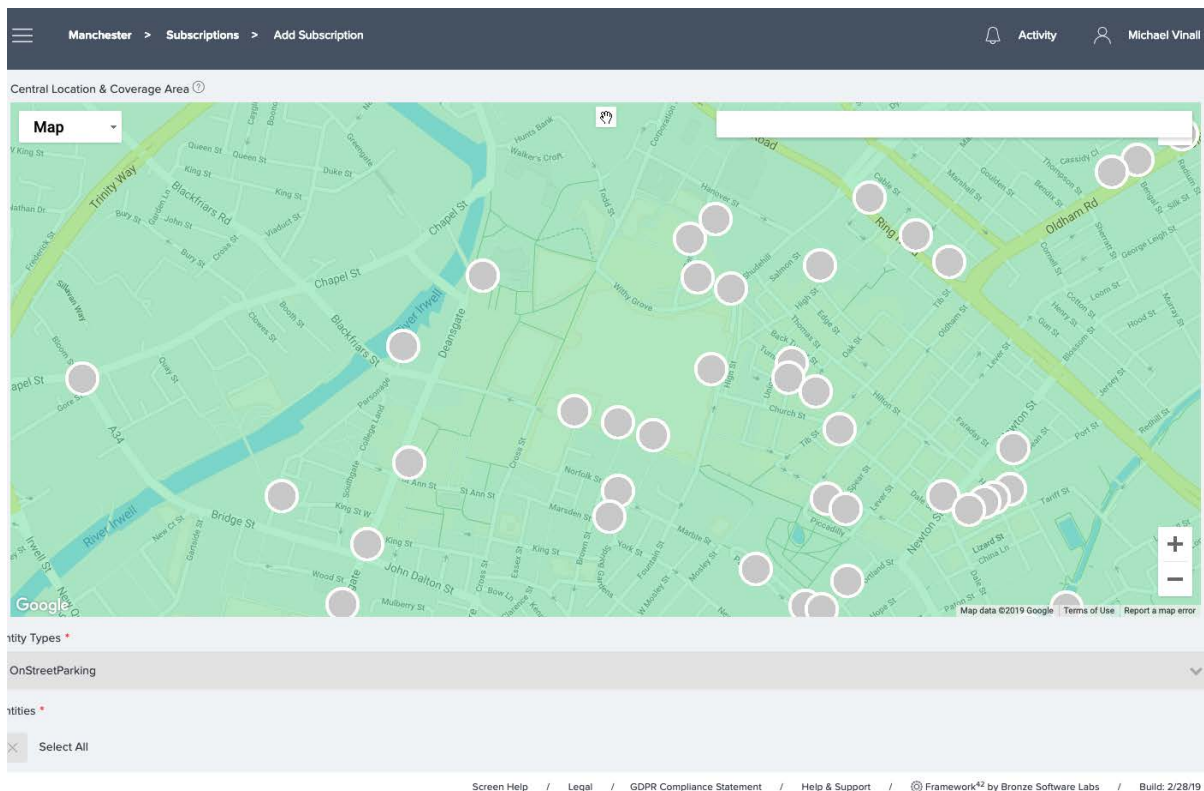


Figure 60: Manchester - New subscription configuration (choose an entity type, and select from available sensors)

Workflow Configuration

As detailed above within the Workflow sequence diagram section, users can define triggers based on incoming data and automated actions the system should take. A trigger could be based on sensory data, coming in from the SynchroniCity API, exceeding a threshold. If this happens, several actions can be triggered, an email could be sent, a job could be raised in the system. Therefore, as an example use case, we could create a workflow to monitor the battery health of a sensor. If battery health is identified as poor, we could automatically create and assign a job to a mobile worker user within the area. This job would appear on the users tablet for completion and contain all of the relevant paperwork required for digital sign-off.

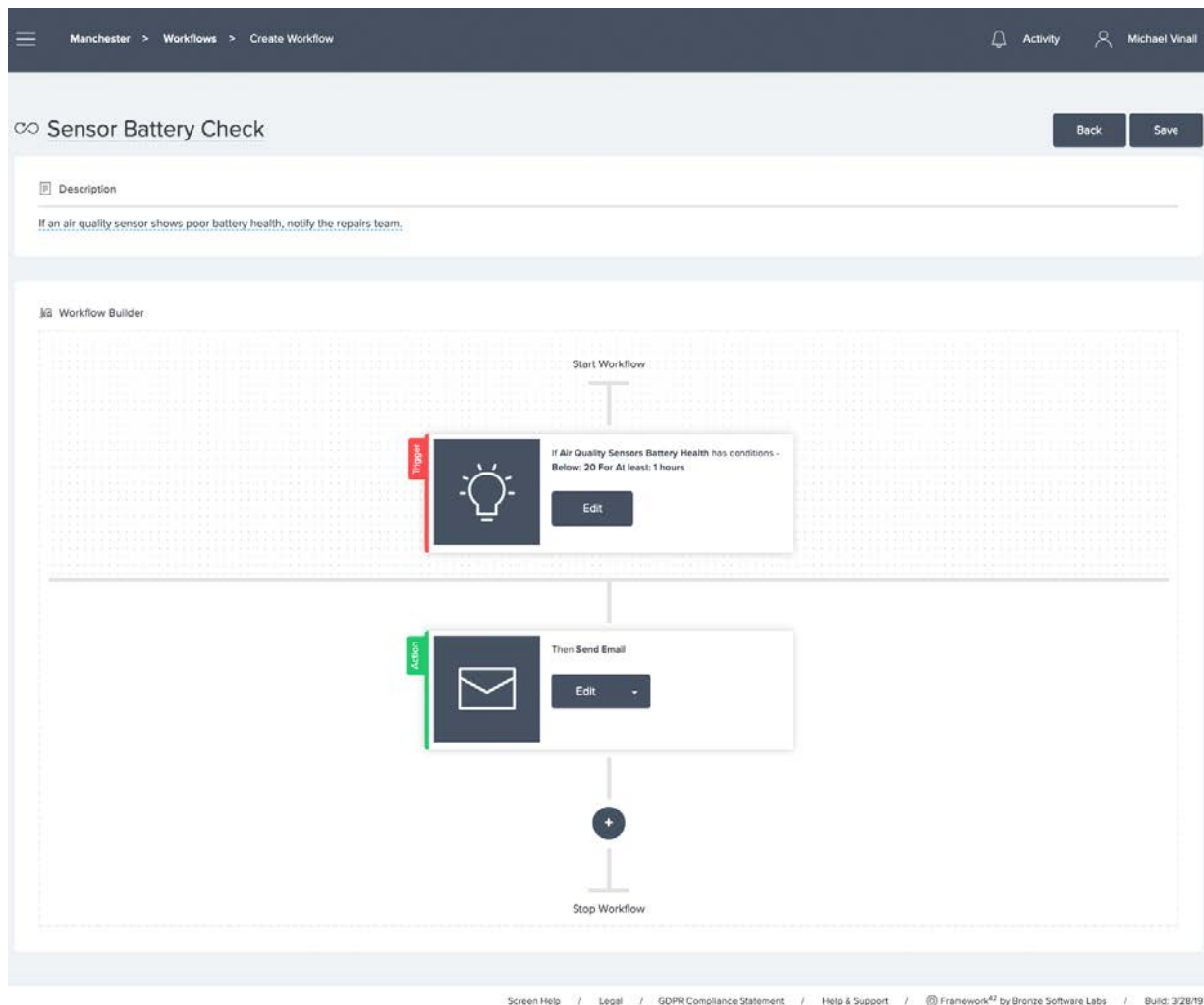


Figure 61: Manchester - Workflow configuration

4.1.7 Data Management

Our system processes and stores little user data, though users are able to control and access their personal information in line with GDPR. Our policies ensure the security of personal information that is processed within the system, they also ensure we provide a compliant and consistent approach to data protection. Our existing policies and processes which were based on modern best practices have been revised to comply with GDPR. As well as ensuring the security of data, we have the processes to ensure we can comply with requests from users to erase or present the data currently held about a user.

The framework.⁴² platform is hosted on a Google architecture which is ISO 27001, ISO 27017, ISO 27018 and GDPR compliant, illustrating that enterprise level security is in place on both a software and hardware level. The data is held on servers within the UK, though in the unlikely event Google's UK network went down, data would be distributed to servers within the EU if possible, but potentially outside of the EU if unavoidable to prevent a loss of service. Our GDPR compliance statement, and privacy statement are both accessible at the bottom of every page within the application.

The predominant user types are administrators using the management portal, and mobile workers using the mobile application. Both kinds of users provide their data with consent and approval of both the linked privacy policy and GDPR compliance statement. The data entered and stored does not include any sensitive data, it consists of basic profile information which is not shared with any third parties. To expand on this, as the software is primarily used in a working environment by

employees of a company, there is no incentive to collect masses of personal data and the data input in all areas is basic.

4.1.8 Conclusion

With the application based on an existing software platform the application design at a high-level has largely been inherited with tweaks and additional functionality introduced for Manchester and Porto applications. The generic framework used as a base already included features that enabled mobile working and analytics. Essentially, we had a strong starting point for introducing collaborative features to allow Manchester to create and shape policies in an evidence-based way with agile working processes.

Users are given a large amount of power, they have an interface into the SynchroniCity framework through creation of data subscriptions, a user selects a group of sensors then data is then pulled from the SynchroniCity platform. They can then setup a policy and include multiple users to track what effect is produced from policy changes, the outcomes can then be entered into a policy document within the system.

Leveraging the use of existing software provided some strong positives and negatives for this application. From the start the application contained a lot of functionality and customizability that could be leveraged with SynchroniCity datasets. However, in some areas the adopted application design did not fit perfectly to the new requirements, which in some cases slowed delivery of additional features for the Agile Governance application.

4.2 Porto's Application "Porto Open Interactive Map"

The **Porto Open Interactive Map** application will gather, process and analyse city data, providing valuable information to the public servants of Porto's City Council, such as strategic managers & administration and local policymakers.

The main city data sources that are expected to be handled by this application are related with environment (noise, air quality and meteorological parameters), traffic flow and constraints, POIs, events and hardware (such as APs and electric charging stations).

This application will be a useful analysis tool of updated, real-time and trusted city data to the City Council (in particular, managers, administration and policy makers) and to the Municipal companies, in order to better inform and implement decision and policy making. This solution will enable a proactive approach to problem solving, contributing to a more agile management of operational and policy processes.

The flow of available information will not only support the digitization process but also improve the "de-siloing" trend of the traditional structures, thus promoting an integrated and more effective management system.

The innovative character of this application is based on the fact that it brings a new 'intelligence' layer to city governance, and thus augmenting the impact of the IoT network deployed in the city. This solution enables a reliable and more consistent usage of interconnected data, which enables all the actors (City Council and Municipal Companies) to interact with each other, by identifying and monitoring cities events and transformations.

Trends and patterns can be integrated in the decision-making process and result in an improvement of the quality of life of the citizens.

Licence

Closed source. Licence provided free of charge for the duration of the SynchroniCity project.

4.2.1 Scenarios

Compared to D3.5 user profiles and scenarios of **Porto Open Interactive Map** application have not been modified; they are reported in this section for reasons of clarity and completeness.

4.2.1.1 Users profiles:

Table 54: User profiles of Porto Open Interactive Map application

User profile ID	User description
User Porto 1 City council (<i>Strategic managers & administration</i>)	This user works at the Municipality of Porto and knows exactly what he needs to do to achieve relevant results but can be frustrated by not having the right information at the right time. Part of his/her job motivation is to have access to city data, analyse citizens' behaviours and needs, and provide a quick and effective answer to city occurrences and activities. He/she wants to feel confident by doing his/her job, and for that to happen, he/she believes that related data must be provided in just one transversal platform to reduce the need for piecemeal research and guessing work. This user is mainly interested in finding particular information to analyse, without being distracted by other data information; keeping up to date with the latest information; looking for time series and data comparison (e.g. noise levels, air quality, pollution levels) in order to be able to predict future changes; having a powerful research tool to quickly find data information and apply a data analysis process (in a simple and straightforward way), to support and answer different situations.
User Porto 2 City council (<i>Local policymakers</i>)	This user works at the Municipality of Porto and works hard to identify and develop key policy areas, and also to interpret and analyse information. This user job involves understanding the present situation of the city and the factors that contribute to it, so that he/she can be able to develop and implement policies and strategies. This user believes that the city data must be provided in just one transversal platform to reduce the need for piecemeal research and guessing work. This user is mainly interested in using data for benchmarking or comparison amongst local communities; having access to clear and trusted information; seeing high level summaries, narratives and key charts that provide local context; keeping up to date with latest data information; and finding long time-series of data to create personalised over-time series of data analysis, in order to be able to predict future changes.

4.2.1.2 Scenarios:

Table 55: Scenario "Noise monitoring" of Porto Open Interactive Map application

Scenario title	Noise monitoring
User profile	Porto 1 (City council / Strategic managers & administration)
Background	The Integrated Management Center (CGI) of the Municipality of Porto provides real time information and promotes an integrated action amongst different public stakeholders and services, such as security (national and municipal Police), emergency (civil protection, medical emergency and fire department), public transportation, and services of the Municipality of Porto (e.g. environment and waste, mobility and traffic, public space management and fleet management).
Objective	Monitor noise in residential areas close to nightlife zones and take the necessary measures when the noise levels reach or exceed a defined threshold for a given amount of time.

Storyline	Noise levels are measured by noise sensors, which are part of the city's Environmental Sensor Network (ESN). A workflow, available in the platform provides alerts and displays on the interactive map, mapping the location and noise levels every time they reach or exceed defined thresholds for a given amount of time, during the night. This workflow also emails a person located within the council that is able to deal with it.
------------------	--

Table 56: Scenario "Policy assessment" of Porto Open Interactive Map application

Scenario title	Policy assessment
User profile	Porto 2 (City Council / Local policymakers)
Background	The Municipality of Porto measures the traffic levels with vehicle counters and speed meters, and the pollution levels with noise and air pollutant sensors. The latter two are part of the city's Environmental Sensor Network (ESN).
Objective	Detect correlations and causalities between traffic and pollution levels, and assess (validate) targeted mobility policies after they have been put in place (ex-post assessment).
Storyline	The platform gathers and processes data from the mobility sensors (vehicle counters and speed meters) and from the ESN (noise and air pollutants, the latter including gas and particle pollutants). This data is reviewed by the Mobility Department of the Municipality of Porto via the interactive map. Accordingly, the Mobility Department decides to take appropriate measures in order to restrict or close the traffic in those two streets during the day. After that, the Municipality of Porto continues to monitor thoroughly the traffic and pollution levels in those two streets, in order to assess these mobility policies. The platform provides real time feedback, via the sensors on the state of these streets allowing the Municipality to take informed conclusions and measures: 1) there was a real causality between traffic and pollution in street A, and thus decides to keep the street A with traffic restrictions; 2) although there was a correlation between traffic and pollution in street B, the results show that there is not a causality between the two (the pollution levels remain more or less the same), and thus decides to reopen the street B; 3) the high levels of pollution measured in street B are actually caused by an industrial facility located nearby (which only works during the day), and the issue is now handled to the Environmental Department of the Municipality, so that it can take the appropriate measures.

Table 57: Scenario "Preventive action" of Porto Open Interactive Map application

Scenario title	Preventive action
User profile	Porto 1 (City council / Strategic managers & administration)
Background	Several major events take place in the city of Porto during the year, such as, music festivals, football matches, student festivities, Saint John's Eve and New Year's Eve. These events involve significant amounts of people, traffic and noise in the event area, but also have significant impacts in the overall city in terms of traffic.
Objective	Take preventive actions in order to: 1) control the traffic and manage bottlenecks; 2) avoid high traffic close to the hospitals, and assure the emergency lanes of the hospitals closer to the event are accessible; 3) avoid and punish unauthorized parking; 4) assure the safety of the high amounts of citizens by restricting or closing some of the streets; 5) dispatch preventive emergency teams; 6) assure the public order and citizens' safety, by dispatching national Police teams.

Storyline	An event organiser informed the Municipality of Porto that a music festival will take place on a given date, at a given time and location. The platform gathers live traffic data from sensors around the city and sends alerts to the Municipality when a certain threshold is reached for a certain period of time. Based on this data, the Integrated Management Centre (CGI) of the Municipality of Porto takes the following actions during the event day: 1) the Municipal Police locally monitors and controls the traffic and parking infractions, in particular, close to the venue; 2) the national Police locally assures the safety of the citizens; 3) the mobility and traffic services take the necessary measures in order to increase the public transportation offer and restrict or close some of the nearby streets; 4) the emergency services (civil protection, medical emergency and fire department) are informed and put on-hold (including making available some vehicles and teams nearby) for any eventual and emergency need. Besides, during the event, the CGI monitors the area with traffic sensors and traffic monitoring video cameras, and displays the main events on a map.
------------------	--

4.2.2 Requirements analysis

This section report the final requirements of **Porto Open Interactive Map**. Both functional and non-functional have not been modified; they are reported in this section for reasons of clarity and completeness.

4.2.2.1 Functional requirements

Table 58: Functional requirements of Porto Open Interactive Map application

ID	Description
FR-1	Real time data MUST be provided
FR-2	Alerts based on data thresholds MUST be enabled
FR-3	Access to datasets MUST be able to be set by admin level users
FR-4	A public dashboard MUST be available without a login wall
FR-5	All data sets MUST be made available to the map widgets
FR-6	Workflows MUST be customized and generated
FR-7	An info bubble MUST open when clicking on an object
FR-8	Time-lapse animation ('data scrubber') of historical data MUST be enabled

4.2.2.2 Non-functional requirements

Table 59: Non-functional requirements of Porto Open Interactive Map application

ID	Description
NFR-1	The service SHOULD be accessible via the latest Safari and Chrome browsers
NFR-2	The service SHOULD be accessible via Android and iOS tablet devices
NFR-3	The service SHOULD be available in English and Portuguese
NFR-4	The service MUST be compliant with EU data protection laws
NFR-5	The service SHOULD have intuitive and easy to use UI
NFR-6	The service MUST be secure
NFR-7	The service MUST have a search tool

4.2.3 Available data sources

The main city data sources that are expected to be handled by this application (Table 60) are related with environment (noise, air quality and meteorological parameters), traffic flow and constraints, POIs, events and hardware (such as APs and electric charging stations). In particular, compared to D3.5 the data sources "Vehicle location and speed" and "Public transportation" have been removed, because the application will no longer use them, whereas the data sources "Devices" and "Electric charging stations" have been added.

Table 60: Data sources used in Porto Open Interactive Map application

Title	Description	Open Data/API	Adopted SynchroniCity Data Model
Noise	Noise levels (LAeq, 5 min)	Open data	NoiseLevelObserved
Air pollutants	Air pollutants measurements such as: PM2.5 particles, PM10 particles, Ozone (O ₃), Nitrogen dioxide (NO ₂), Carbon monoxide (CO)	Open data	AirQualityObserved
Meteorological parameters	Meteorological measurements: Solar radiation, ultraviolet radiation, wind direction and speed, rainfall, atmospheric pressure, air temperature and humidity measurement	Open data	WeatherObserved
Vehicle count	Number of vehicle (count) in a road segment.	Open data	TrafficFlowObserved
Traffic constraints	Temporary traffic constraints (scheduled events, accidents, etc.)	Open data	Open511
POIs	Points of interest (POIs)	Open data	PointOfInterest
Events	Events (concerts, football games, cultural events, etc.)	Open data	Event
Devices	Wi-Fi access points (location, model, ID, category, etc.)	Open data	Device
Electric charging stations	Electric vehicles charging stations (location, capacity, socket type, status, etc.)	Open data	EVChargingStation

4.2.4 Application architecture

Bronze Software Labs is deploying a version of its smart city middleware, framework⁴², with features specifically developed for SynchroniCity⁵⁵ for use by Porto. The application architecture is depicted in Figure 62 and is common between Manchester and Porto's applications. framework.⁴² is a cloud based IoT platform that accepts NGSI formatted data from the SynchroniCity API to answer the various scenarios put forth by the cities.

⁵⁵ <https://synchronicity.framework42.com>

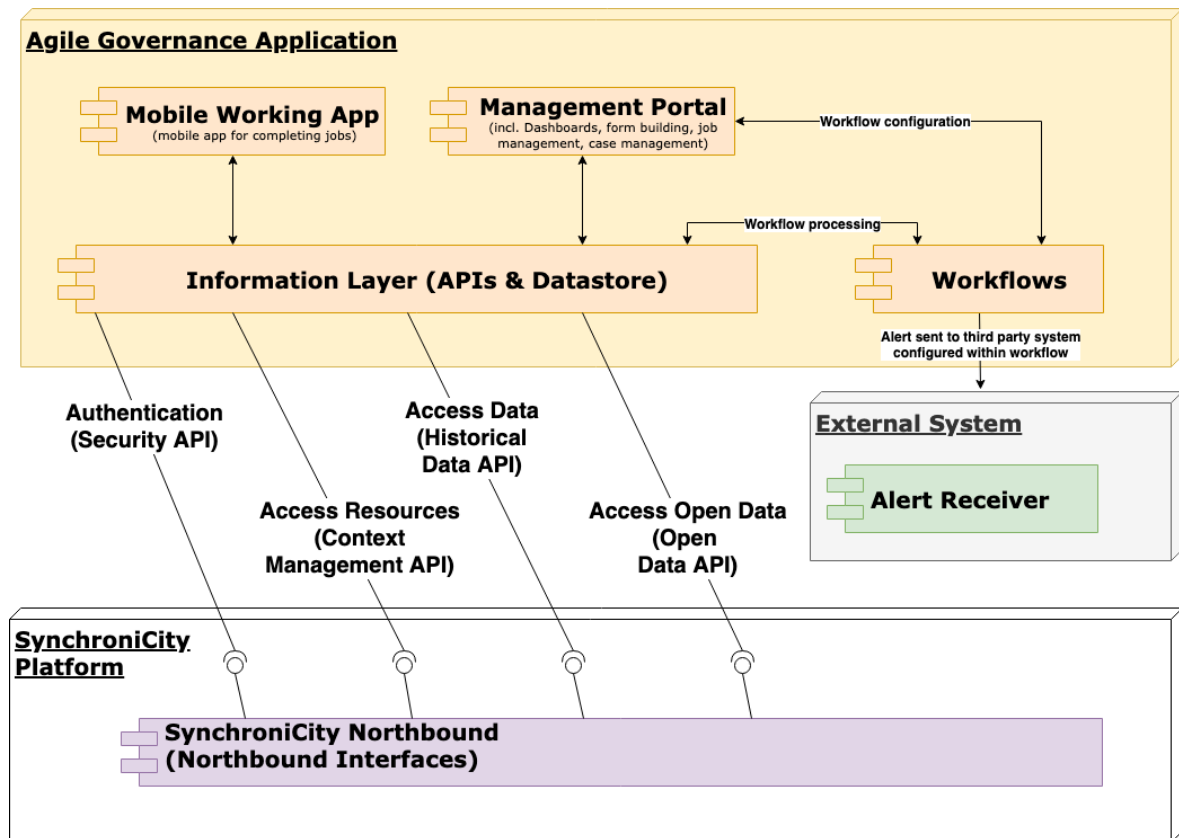


Figure 62: Porto - component diagram of Porto Open Interactive Map application

Overview

The architecture of framework.⁴² is common between Porto's and Manchester's Community Policy Suite applications, and it is documented under section 4.1.4.

4.2.5 Sequence diagrams

The following sequence diagrams illustrate the technical processes that are carried out to fulfill a user interaction within the system.

- **Public dashboard:** The public dashboard displays events to the user directly from SynchroniCity, as well as a map containing sensory data and POIs data across Porto.
- **Subscription creation:** Creation of a new data subscription in the management portal, the chosen entity data is then queried and processed by the framework.
- **Workflows:** This diagram depicts both the user configuration of a workflow within the framework, as well as the ongoing automated processing that occurs to check whether a workflow should be triggered by incoming data.

Public dashboard

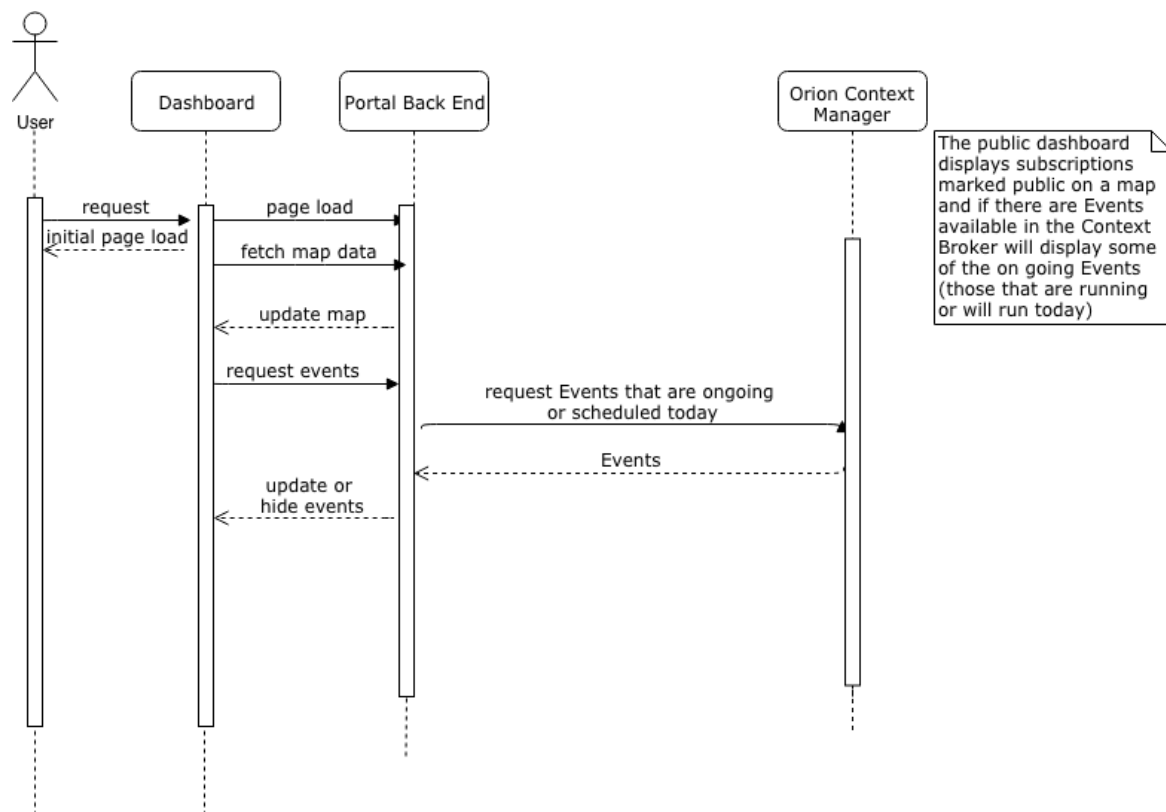


Figure 63: Porto - Public dashboard sequence diagram of Open Interactive Map application

Members of the public are able to view a map consisting of multiple reference zone datasets. This currently includes a map showing POIs in the area, as well as an event calendar that highlights what is going on in Porto for that day. As it can be seen on the diagram, the application backend requests events and stores them to reduce load times for users.

Subscription creation

This sequence diagram is a generic function of the software, which is documented under section 4.1.5 of Manchester's Agile Governance application.

Workflows

This sequence diagram is a generic function of the software, which is documented under section 4.1.5 of Manchester's Agile Governance application.

4.2.6 User Interface

The user interface Porto Open Interactive Map application is in common with Manchester's application, since both are based on Framework⁴². Despite of this, Porto Open Interactive Map application provides a dedicated functionality based on the use of a map, the dedicated user interface is depicted in Figure 64.

Public citizen map

The public page has two main sections: 1) across the top is a calendar that is showing events taking place within Porto, and 2) below is the map itself, which shows data as configured by the region. After a data subscription is created within the portal, the user has the option to make it visible within the public citizen map. The functionality of the map itself is similar to that of a dashboard map detailed below, where the data series visible on the map can be toggled, and a time scrubber is present that allows the user to view data across a period of several months by moving the slider across days.

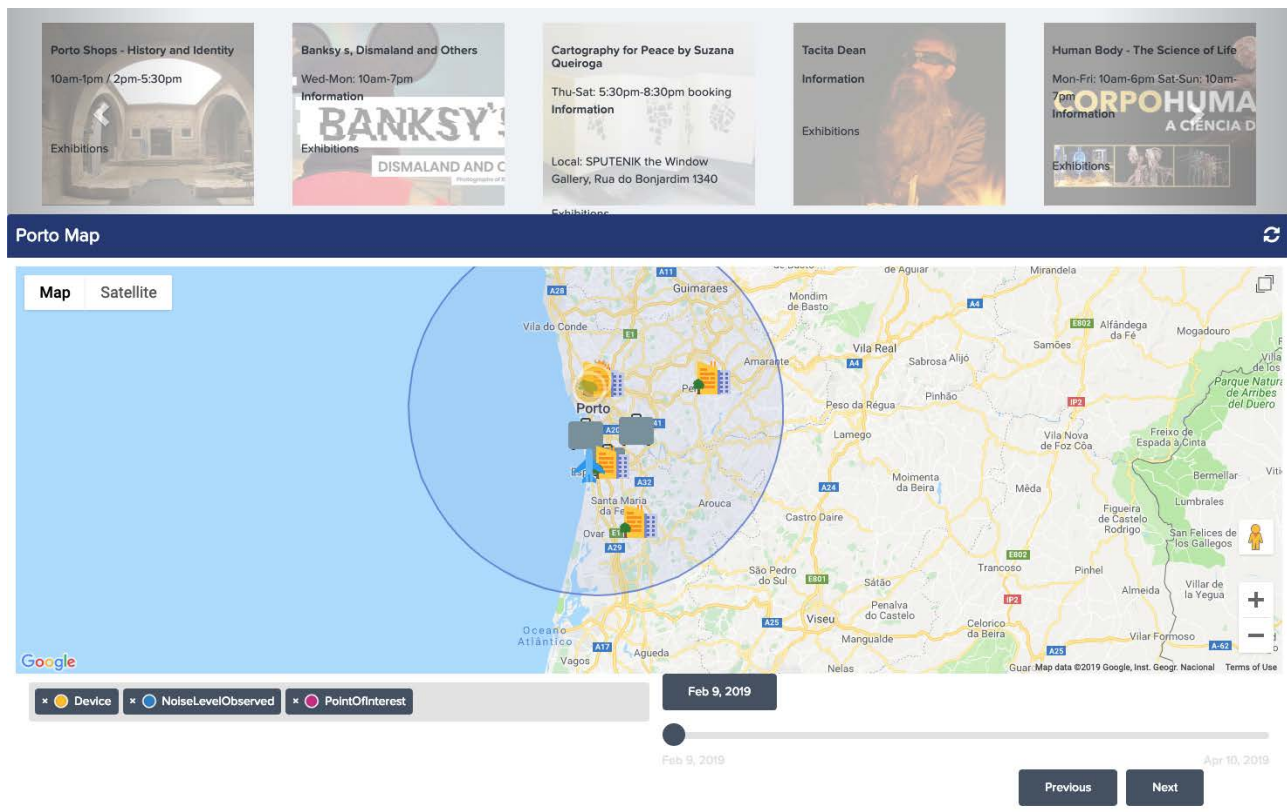


Figure 64: Porto - Public citizen map with event calendar and sensory data

For parts of the user interface dedicated to Dashboard, Policy configuration and Subscription configuration, it is possible to refer to description of Manchester's user interface (Section 4.1.6)

4.2.7 Data Management

From a technical and process perspective data management is common with the Manchester's application as reported in 4.1.7.

4.2.8 Conclusion

Bronze Software Labs (developer)

The Porto application allows policy officers to choose datasets that should be available through a citizen-facing dashboard. This dashboard also shows events happening within Porto, this event data is loaded directly from the SynchroniCity APIs to reduce overheads. Below the event calendar is the public map, this overlays Porto datasets as configured by the policy officers within the system. The underlying solution design is based on Bronze labs existing software platform (framework.⁴²), the application inherits a large amount of features around mobile working, form capture and data analytics. The public map functionality, data subscriptions and data policies have all been developed as new features for the Porto application, our existing software platform essentially gave us a robust starting point to introduce the required features to deliver against the citizen engagement use case for Porto.

Policy officers have the ability to interface with the SynchroniCity framework through the portal interface. Data subscriptions can be created, this interface loads a list of possible data types from the SynchroniCity platform, after a user selects a data type they are presented with a map displaying all available sensors. Once a subscription is created, data is collected by our application close to realtime. This data can be consumed via dashboards or automated workflows which can trigger user

alerts & notifications. A policy can also be created in the system, which enables policy officers to track how data changes over a period of time and control access to that data within their departments and teams.

The base of this application and the application design, the Framework.⁴² platform, normally acts as a central datastore. For the use cases presented by both Manchester and Porto all sensory data is coming from the SynchroniCity platform, instead of directly from the sensors. This has been one of the main adaptations required from our existing design. In some areas, the existing design has shown a lack of flexibility which has slowed delivery of features. Additionally, the user interface of the administration portal offers a high level of customisation. Having a wide feature set from the beginning has also meant initial scope of the application and use cases was very ambitious, and has not been fully realised as of yet.

City of Porto (user)

So far, the city of Porto has been cooperating with the Bronze Software Labs company in the specification of the application, co-definition of the functional and non-functional requirements, and co-creation of the scenarios. As an end-user, the city of Porto has also been performing high-level tests of the application, in particular, in terms of data upload, workflows and policies creation, and other general functionalities. Several problems and malfunctions have been identified and reported to Bronze Software Labs to be solved over the past months; the majority of them have been solved and the remaining ones are being resolved at the time of writing.

Now that the application is fully developed, we will then start the testing and deployment phase in real scenarios.

Apart from the scenarios reported in the introduction, we identified a particular scenario in which the application is expected to be very useful, and which is straightforward to implement (for logistical and operational reasons). Accordingly, the city of Porto will kick-off the testing and deployment of the application with a particular use case: maintenance managing of the wireless access points (APs), which are part of the public Wi-Fi network deployed in the city.

The Wi-Fi network is a service that will benefit from the integration of the application, as minor changes (e.g. device failures and malfunctioning, low/high data transfer) can be instantly detected and reported via workflow alerts and traced through the interactive map. APs device replacements and maintenance is a recurring task that would greatly benefit from the adoption of the application, as it would enable an easier management through the use of the work orders assignments and the forms that go along with it. Besides, the proper monitoring of the network APs would also enable the city to create policies related with their operational performance, e.g. by increasing/decreasing the number of APs and/or their location in the city.

Some additional problems and malfunctions are expected to be identified during this stage, which will be reported and solved by the developer in an agile way. Besides, we also expect to make some additional fine-tuning in terms of functionalities and features, once we start working with real-life scenarios.

After fully testing and validating the application with this initial scenario, we expect to be able to extend the use of this application to other internal services (both of Porto Digital and the Municipality of Porto), in particular, to the scenarios described in the application scenarios.

4.3 Carouge's Application "Noise monitoring near bars"

The noise monitoring application will provide functionalities to measure noise level in each area of Carouge and to detect which bars and restaurants are noisy more than the allowed level, producing noise pollution. The application will inform the city council about the noisiest areas in Carouge; this will help the city council to take decisions in order to reduce the noise generated by the bars and the restaurants. For instance, the schedule for opening and closing time of the noisiest bars and restaurants could be adapted based on the measurements results and the information provided by the noise monitoring mobile application.

Licence

No licence has yet been chosen in the context of this application deployed in Carouge.

4.3.1 Scenarios

4.3.1.1 Users profiles:

There are no updates, since the user profiles were consolidated in D3.5 and are still valid.

4.3.1.2 Scenarios

There are no updates, since the scenarios for the application were consolidated in D3.5 and are still valid.

4.3.2 Requirements analysis

The requirements analysis has been made following the goals defined by the city of Carouge and the scenarios described in this document.

4.3.2.1 Functional requirements

The requirement analysis, made after the updates in the application reported in this document, led to no updates in functional requirements.

4.3.2.2 Non-functional requirements

Only the NFR-1 has been updated

Table 61: Non-functional requirements of Noise monitoring near bars application

ID	Description
NFR-1	The software used for this application SHOULD not consume too many resources on the server where it is installed.
NFR-2	The service MUST be accessible in diverse mobile devices and web browsers.
NFR-3	The service MUST be compliant with privacy data protection policy in EU and Switzerland.
NFR-4	The service SHOULD be intuitive and easy to use.
NFR-5	The service SHOULD be easy to extend itself and easy to integrate into other city services.
NFR-6	The service SHOULD provide multiple languages used in Switzerland

4.3.3 Available data sources

Table 62: Data sources used in Noise monitoring near bars application

Title	Description	Open Data API	Adopted SynchroniCity Data Model	Notes
Sound Level Average	Noise pressure level sensors which take a measurement each 15 minutes in different places of the city.	API	NoiseLevelObserved	

4.3.4 Application architecture

There are no updates, since the architecture of the application was consolidated in D3.5 and is still valid. For sake of completeness Figure 65 depicts the component diagram of Smart Parking application, refer to D3.5 for any further detail.

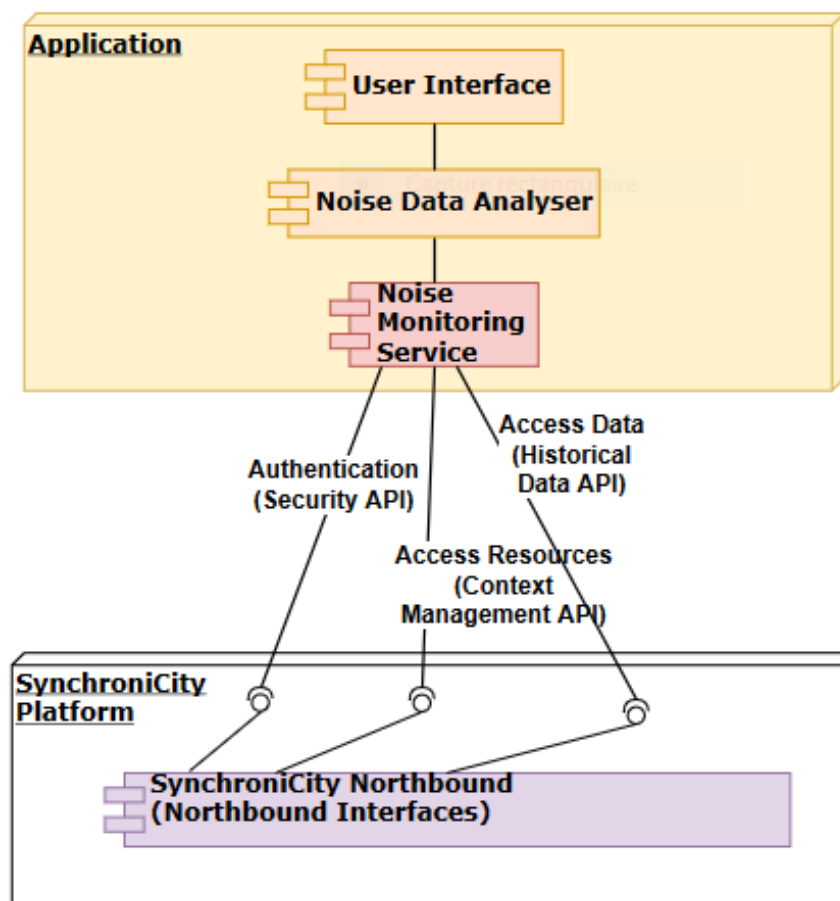


Figure 65: Carouge - component diagram of Noise monitoring near bars application

Reusable Atomic Services

The Carouge reference zone is using the Noise Estimation Atomic Service. Basically, a noise estimation or prediction is realised using previous historical data provided by the Synchronicity Framework. This is a new Atomic Service developed after releasing D3.4, and therefore was not included in that deliverable. In any case, the list of Atomic Services is maintained updated and live in the webpage and software repository[19]. This new service is derived from the Parking Estimation Atomic Service and leverage its capabilities. In fact, the Noise Estimation Atomic Service is using

the same principles; instead of parking sensors data, it takes in input data from the noise level sensors installed in Carouge. Currently, the Carouge reference zone is testing the Noise Estimation Atomic Service with the real data generated by the noise level sensors.

Table 63: Atomic Services adopted in Carouge Noise monitoring near bars application

Atomic Service	Rationale
Noise Estimation Atomic Service	The Carouge reference zone is using the new Noise Estimation Atomic Service to estimate and predict noise levels; this Atomic Service is derived from the Parking Estimation Atomic Service.

4.3.5 Sequence diagrams

Among the functionalities offered by this application, the more relevant one is represented by the possibility to monitor and analyse noise levels in the city areas. The sequence diagram of this functionality is reported and described in this section. The following sequence diagram shows the relations between the different components used for the Noise monitoring near bars application deployed in the Carouge reference zone.

Noise monitoring near bars sequence diagram

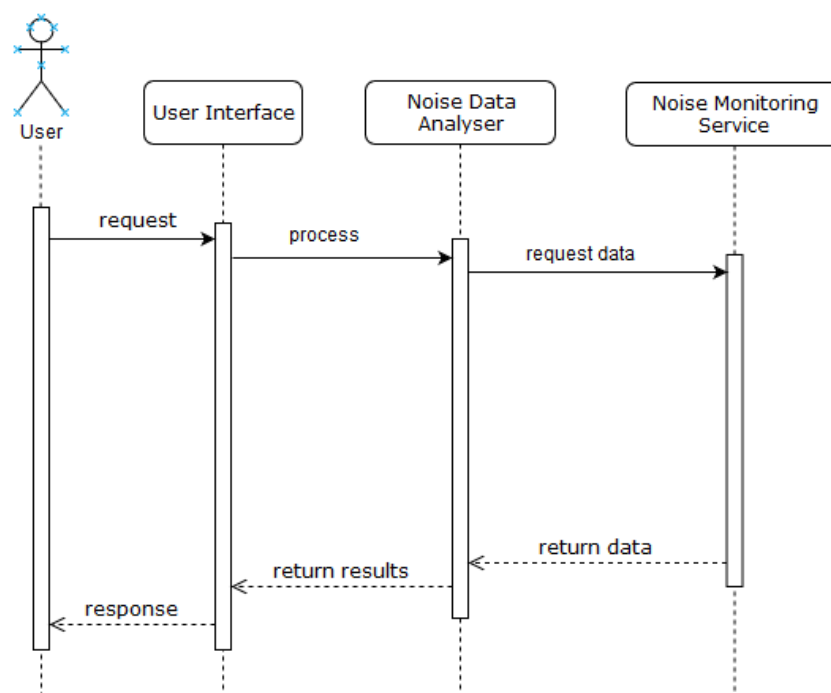


Figure 66: Carouge - Noise monitoring near bars sequence diagram of Noise monitoring near bars application

The user can basically select on the map a noise sensor or a group of noise sensors located in the same area of the Carouge city. From the map, the user can see the live data provided by each of the noise sensors or the historical data saved in the Carouge instantiation of the SynchroniCity framework. The user interface on the user's Web browser triggers each time some requests to the components installed for the handling of the noise level in the Carouge virtual machines. After the processing of the noise level data, the results are sent and displayed accordingly to the user's request. This implies the indirect use of the components like the Orion context broker and Comet STH MongoDB database on the Carouge backend.

4.3.6 User Interface

The user can see the data related to this application on a map hosted in the Carouge reference zone⁵⁶.

The following screenshot displays, as an example, the data related to a sensor measuring the noise pressure level in Carouge:

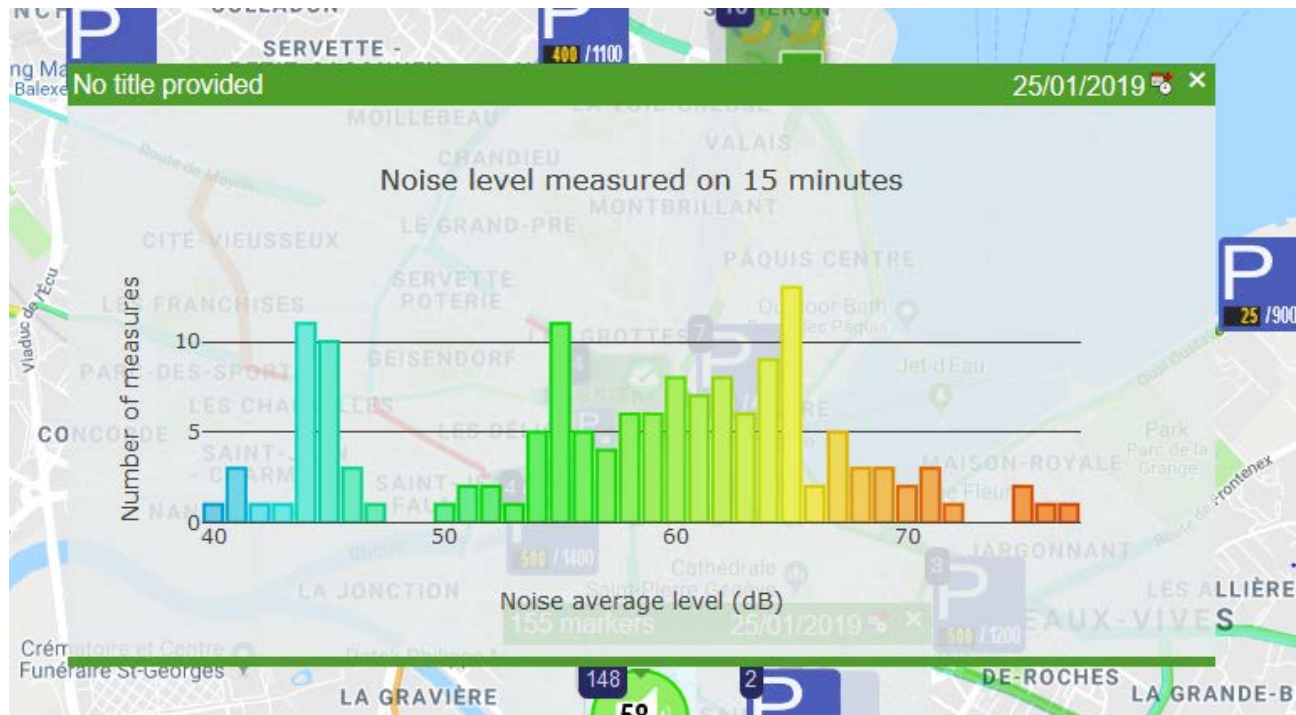


Figure 67: Carouge - a screenshot of Noise monitoring near bars application

4.3.7 Data Management

The access to the data is secured and limited to the required people. This is achieved through KeyRock and Wilma, but also with the use of passwords to access the map. Of course, there is no personal data stored in the Carouge reference zone server, so this reference zone fulfils the requirements given by the GDPR.

4.3.8 Conclusion

The application described here will try to reduce the problems linked to the noise emitted near the bars and restaurants in the city of Carouge. The application will make the citizens aware of the too high sound level encountered in the streets and to educate the citizens to be less noisy when talking near the bars. At the end, the quality of life will be improved thanks to the Noise monitoring near the bars application. So, the SynchroniCity platform brings to the citizens living in Carouge some improvements and the experience with the SynchroniCity platform is very good. In particular, the platform and its atomic service used in the context of this application ease the development of the use case about the noise monitoring. The use of the noise prediction based on the data collected near the bars and restaurants helps the city of Carouge to take decisions to try to reduce the noise in these places and so at the end, to improve the comfort for the citizens living in Carouge.

⁵⁶ <https://map.cityreport.org:9443/>

5 Conclusions

This deliverable describes the final design of the applications developed by the various RZs exploiting SynchroniCity platform APIs and Atomic Services. Each application belongs to one of the three main SynchroniCity application themes:

- Human Centric Traffic Management
- Multimodal Transportation
- Community Policy Suite

Twelve applications are reported in this document, specifically:

- three applications for the Human Centric Traffic Management theme and designed by Antwerp, Eindhoven and Milan;
- six applications for the Multimodal Transportation theme and designed by Carouge, Helsinki, Milan, Porto, Santander and Seongnam;
- three applications for the Community Policy Suite theme and designed by Carouge, Manchester and Porto.

This document focuses on the main differences between the first and the final version of the applications. Moreover, it aims to provide insights about technical aspects, such as the interaction among the application, the users and the SynchroniCity platform, reporting specific applications' sequence diagrams and the user interface of the applications.

Summarizing the conclusion of each RZ's application, it is important to underline that, despite some challenging experience in loading the data into the framework, the exploitation of SynchroniCity's APIs and of Atomic Services by developers eases the provision of applications and services to the citizens with reduced effort. Moreover, the reusability of the different Atomic Services between the smart cities involved in the SynchroniCity permits to exchange experience and knowledge. Table 64 summarizes the adoption of the Atomic Services by RZs. The table underlines the need for the Routing Service which finds suitable routes between two points according to users' preferences. Moreover, new Atomic Services might be created starting from the existing ones reported in Table 64. An example is the Noise Monitoring Estimator, built by Carouge RZ (4.3) starting from the existing Parking Probability Estimator Atomic Service.

An important example of deployment of the platform is the Seongnam scenario. In this context the platform was verified and certified by TTA (Telecommunications Technology Association) and it is used, in conjunction with OneM2M, to help people finding a parking space and the commute route such as minimizing the transfer time.

It is important to underline that some of the application reported in this document are still in development phase or pre-release phase, so their user interface might be susceptible to changes.

Table 64: Adoption map of Atomic Services in applications designed by RZs

		Routing service	Parking Probability Estimator service	Noise Monitoring Service	Traffic Flow Estimator service	Smartcities Dashboard service	Metrics Visualizer (Grafana) service	GTFS Fetcher service	NGSI Urban Mobility to GTFS	GTFS-RT loader from NGSI
Human- centric traffic mgmt	Antwerp									
	Eindhoven									
	Milan									
Multi-modal transportation	Helsinki									
	Milan									
	Porto									
	Santander									
	Carouge									
	Seongnam									
Community Policy Suite	Carouge									
	Porto									
	Manchester									

6 References

- [1] SynchroniCity, “D3.5 Customized IoT service prototypes for lead ref. zones – basic”.
- [2] SynchroniCity, “D3.3 Suite of baseline implementations - advanced”.
- [3] SynchroniCity, “D3.1 - Documentation of service designs”.
- [4] ATOS, “Atos Urban Data Platform,” [Online]. Available: <http://booklet.atosresearch.eu/content/atos-urban-data-platform-audp>.
- [5] “D2.8 Report on Basic Reference Zones Platforms Deployment and Operational Plan,” [Online]. Available: https://synchronicity-iot.eu/wp-content/uploads/2018/09/SynchroniCity_D2.8.pdf.
- [6] “The WebSocket Protocol - Request for Comments: 6455,” Internet Engineering Task Force (IETF) , December 2011. [Online]. Available: <https://tools.ietf.org/html/rfc6455>.
- [7] “Milan GeoViewer,” [Online]. Available: <https://geoportale.comune.milano.it/geoviewer/>. [Accessed 05 04 2019].
- [8] “Esri,” [Online]. Available: <https://www.esri.com/en-us/home>.
- [9] “Milan Open Data Portal,” [Online]. Available: <http://dati.comune.milano.it/>. [Accessed 05 04 2019].
- [10] “Agenzia Mobilità Ambiente e Territorio - AMAT,” [Online]. Available: <https://www.amat-mi.it/it/>. [Accessed 05 04 2019].
- [11] “Node.js,” [Online]. Available: <https://nodejs.org>.
- [12] “Express.js,” [Online]. Available: <https://expressjs.com/it/>.
- [13] “OpenTripPlanner - Multimodal Trip Planning,” 27 7 2018. [Online]. Available: <http://www.opentripplanner.org/>.
- [14] “OpenStreetMap,” 27 7 2018. [Online]. Available: <https://www.openstreetmap.org>.
- [15] “E015 API Catalogue,” [Online]. Available: <http://www.e015.regione.lombardia.it/site/api-catalog>. [Accessed 05 04 2019].
- [16] “oneM2M,” [Online]. Available: <http://www.onem2m.org/>.
- [17] “Ionic Framework,” [Online]. Available: <https://ionicframework.com/>.
- [18] “Mobius,” [Online]. Available: <http://developers.iotocean.org/archives/module/mobius>.
- [19] “Atomic Service Noise Estimator,” [Online]. Available: <https://gitlab.com/synchronicity-iot/noise-estimator>.
- [20] SynchroniCity, “D2.1 - Reference Architecture for IoT Enabled Smart Cities”.

[21] SynchroniCity, "D3.2 Suite of baseline implementations - basic".

[22] "GDPR, Recital 26," [Online]. Available: <https://gdpr-info.eu/recitals/no-26/>. [Accessed 05 04 2019].