

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:	01.08.2019
Project duration:	36 months
Project end date:	31.12.2019
Deliverable number:	D3.8
Deliverable title:	Report on reference zones IoT service deployment and operations
Document version:	Ver1.0
WP number:	WP3
Lead beneficiary:	4-FVH
Main author(s):	Veli Airikkala (FVH), Giuseppe Ciulla (ENG), Paolo Fosci (MIL), David Gomez (ATOS)
Internal reviewers:	Veli Airikkala, José Gato, Giuseppe Ciulla, Flavio Cirillo, David Gomez, Thomas Gilbert, Daniel Puschmann, Martin Brynskov
Type of deliverable:	Report
Dissemination level:	PU
Delivery date from Annex 1:	M35
Actual delivery date:	15.01.2020 (M37)



Co-funded by the
European Union

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

Co-funded by



Executive summary

One of the main methodologies of SynchroniCity has been to verify and validate the developed concepts through pilots. Each reference zone defined domains for their pilots. This deliverable is a follow up to the D3.7 “Pilot deployment plan” and aims to report on the deployment of the initial applications on each reference zone. The scope of the deliverable is also to share lessons learned and to outline future sustainability plans.

The pilot of the city of **Helsinki** aimed to extend the existing journey planning application with environmental parameters (air quality and noise).

The pilot from **Manchester** focused on understanding how IoT deployment could be utilized in the making of data driven policy decisions.

The municipality of **Milan** was involved in two pilot themes: HCTM and MMT. The first pilot aimed to use a range of data sources to improve cycling mobility themes and road safety. The second pilot was aimed to help disabled people to drive in the city.

The city of **Eindhoven** piloted a solution that connect different sensor data to improve both the bicycle experience and the municipality insights on cycling patterns.

The city of **Carouge** deployed two pilots with different themes: HCTM and CPS. The goal of the first was to deliver a smart parking service that uses real-time parking data from installed parking sensors and to integrate public transformation information. The goal of the second pilot was to measure noise levels on streets with restaurants and bars.

The city of **Porto** was also involved in two pilot areas: CPS and MMT. While the CPS pilot aimed to gather and analyse data for further refinement, the MMT pilot aimed to provide a personalised and customised multimodal assistant based on the user's preferences, requirements and choices.

The city of **Santander** deployed two pilots regarding the MMT theme. One pilot aimed to improve finding a parking place; the other pilot's goal was to combine urban commuting data for route suggestions etc.

The pilot in **Antwerp** aimed to update the existing Multi Modal navigator ‘Slim Naar Antwerpen’ with a more accurate guidance system for planning the route. The pilot was not deployed.

The Seoul satellite city **Seongnam** participated in Synchronicity by piloting a parking service and marketplace that used real time parking data.

Abbreviations

AUDP	ATOS Urban Data Platform
CDE	Customer Dedicated Environment
CPS	Community Policy Suite
D	Deliverable
EC	European Commission
FMI	Finnish Meteorological Institute
FVH	Forum Virium Helsinki
HCTM	Human Centric Traffic Management
HSL	Helsingin Seudun Liikenne (Helsinki Region Transport)
MMT	Multimodal Transportation
RZ	Reference Zone
UAT	User Acceptance Testing
WP	Work Package
WT	Work Task

Contents

1. INTRODUCTION	7
1.1 PURPOSE OF THE DOCUMENT	7
1.2 STRUCTURE OF THE DELIVERABLE	7
1.2.1 Pilot motivations and deployed use cases	7
1.2.2 Test results of pilot	7
1.2.3 Implemented infrastructure	7
1.2.4 Involved ecosystem partners and their roles	7
1.2.5 Internal and external dependencies	7
1.2.6 Deployment conclusions and lessons learned	8
1.2.7 Sustainability plan upon project completion and future improvements	8
2. HELSINKI	9
2.1 PILOT MOTIVATIONS AND DEPLOYED USE CASES	9
2.2 TEST RESULTS OF PILOT	11
2.3 IMPLEMENTATION ARCHITECTURE AND USED ATOMIC SERVICES	15
2.3.1 Implemented infrastructure	16
2.3.2 Changes and deviations of planned features of pilot	18
2.4 INVOLVED ECOSYSTEM PARTNERS AND THEIR ROLES	18
2.5 INTERNAL AND EXTERNAL DEPENDENCIES	18
2.6 DEPLOYMENT CONCLUSIONS AND LESSONS LEARNED	19
2.7 SUSTAINABILITY PLAN UPON PROJECT COMPLETION AND FUTURE IMPROVEMENTS	19
3. MANCHESTER	20
3.1 PILOT MOTIVATIONS AND DEPLOYED USE CASES	20
3.2 TEST RESULTS OF PILOT	21
3.3 IMPLEMENTATION ARCHITECTURE AND USED ATOMIC SERVICES	22
3.3.1 Implemented infrastructure	24
3.3.2 Changes and deviations of planned features of pilot	25
3.4 INVOLVED ECOSYSTEM PARTNERS AND THEIR ROLES	25
3.4.1 Required supportive actions and solved issues	25
3.5 INTERNAL AND EXTERNAL DEPENDENCIES	26
3.6 DEPLOYMENT CONCLUSIONS AND LESSONS LEARNED	26
3.7 SUSTAINABILITY PLAN UPON PROJECT COMPLETION AND FUTURE IMPROVEMENTS	27
4. MILAN	28
4.1 HUMAN-CENTRIC TRAFFIC MANAGEMENT APPLICATION "DECISION SUPPORT SYSTEM FOR CYCLE PATH PLANNING	28
4.1.1 PILOT MOTIVATIONS AND DEPLOYED USE CASES	28
4.1.2 TEST RESULTS OF PILOT	38
4.1.3 IMPLEMENTATION ARCHITECTURE AND USED ATOMIC SERVICES	38
4.1.3.1 Implemented infrastructure	39
4.1.3.2 Changes and deviations of planned features of pilot	41
4.1.4 INVOLVED ECOSYSTEM PARTNERS AND THEIR ROLES	41
4.1.4.1 Required supportive actions and solved issues	42

4.1.5 INTERNAL AND EXTERNAL DEPENDENCIES	42
4.1.6 DEPLOYMENT CONCLUSIONS AND LESSONS LEARNT	42
4.1.7 SUSTAINABILITY PLAN UPON PROJECT COMPLETION AND FUTURE IMPROVEMENTS	43
4.2 MULTIMODAL TRANSPORTATION APPLICATION "MULTIMODAL-NAVIGATOR FOR DISABLED PEOPLE"	43
4.2.1 PILOT MOTIVATIONS AND DEPLOYED USE CASES	43
4.2.3 IMPLEMENTATION ARCHITECTURE AND USED ATOMIC SERVICES	47
4.2.3.1 <i>Implemented infrastructure</i>	49
4.2.3.2 <i>Changes and deviations of planned features of pilot</i>	50
4.2.4 INVOLVED ECOSYSTEM PARTNERS AND THEIR ROLES	50
4.2.4.2 <i>Required supportive actions and solved issues</i>	50
4.2.5 INTERNAL AND EXTERNAL DEPENDENCIES	51
4.2.6 DEPLOYMENT CONCLUSIONS AND LESSONS LEARNED	51
4.2.7 SUSTAINABILITY PLAN UPON PROJECT COMPLETION AND FUTURE IMPROVEMENTS	51
5. EINDHOVEN	53
5.1 PILOT MOTIVATIONS AND DEPLOYED USE CASES	53
5.2 TEST RESULTS OF PILOT	54
5.3 IMPLEMENTATION ARCHITECTURE AND USED ATOMIC SERVICES	55
5.3.1 <i>Implemented infrastructure</i>	60
5.3.2 <i>Changes and deviations of planned features of pilot</i>	61
5.4 INVOLVED ECOSYSTEM PARTNERS AND THEIR ROLES	61
5.5 INTERNAL AND EXTERNAL DEPENDENCIES	62
5.6 INTEGRATING WITH OTHER EXTERNAL APPLICATIONS	62
5.7 DEPLOYMENT CONCLUSIONS AND LESSONS LEARNED	65
5.8 SUSTAINABILITY PLAN UPON PROJECT COMPLETION AND FUTURE IMPROVEMENTS	66
6. CAROUGE	67
6.1 PILOT MOTIVATIONS AND DEPLOYED USE CASES	67
6.2 TEST RESULTS OF PILOT	67
6.3 IMPLEMENTATION ARCHITECTURE AND USED ATOMIC SERVICES	71
6.3.1 <i>Implemented infrastructures</i>	72
6.3.2 <i>Changes and deviations of planned features of pilot</i>	73
6.4 INVOLVED ECOSYSTEM PARTNERS AND THEIR ROLES	74
6.4.1 <i>Required supportive actions and solved issues</i>	74
6.5 INTERNAL AND EXTERNAL DEPENDENCIES	74
6.6 DEPLOYMENT CONCLUSIONS AND LESSONS LEARNED	74
6.7 SUSTAINABILITY PLAN UPON PROJECT COMPLETION AND FUTURE IMPROVEMENTS	75
7. PORTO	76
7.1 PILOT MOTIVATIONS AND DEPLOYED USE CASES	76
7.2 TEST RESULTS OF PILOT	77
7.3 IMPLEMENTATION ARCHITECTURE AND USED ATOMIC SERVICES	79
7.3.1 <i>Implemented infrastructure</i>	80
7.3.2 <i>Changes and deviations of planned features of pilot</i>	81
7.4 INVOLVED ECOSYSTEM PARTNERS AND THEIR ROLES	82
7.4.1 <i>Required supportive actions and solved issues</i>	82
7.5 INTERNAL AND EXTERNAL DEPENDENCIES	82
7.6 DEPLOYMENT CONCLUSIONS AND LESSONS LEARNED	83
7.7 SUSTAINABILITY PLAN UPON PROJECT COMPLETION AND FUTURE IMPROVEMENTS	84

8. SANTANDER.....	85
8.1 PILOT MOTIVATIONS AND DEPLOYED USE CASES	85
8.2 TEST RESULTS OF PILOT.....	87
8.3 IMPLEMENTATION ARCHITECTURE AND USED ATOMIC SERVICES.....	88
8.3.1 <i>Implemented infrastructure</i>	89
8.3.2 <i>Changes and deviations of planned features of pilot</i>	89
8.4 INVOLVED ECOSYSTEM PARTNERS AND THEIR ROLES	90
8.4.1 <i>Required supportive actions and solved issues</i>	90
8.5 INTERNAL AND EXTERNAL DEPENDENCIES.....	91
8.6 DEPLOYMENT CONCLUSIONS AND LESSONS LEARNED	91
8.7 SUSTAINABILITY PLAN UPON PROJECT COMPLETION AND FUTURE IMPROVEMENTS	92
9. ANTWERP	93
9.1 PILOT MOTIVATIONS AND DEPLOYED USE CASES	93
9.2 TEST RESULTS OF PILOT.....	93
9.3.1 <i>Implemented infrastructure</i>	95
9.3.2 <i>Changes and deviations of planned features of pilot</i>	96
9.4 INVOLVED ECOSYSTEM PARTNERS AND THEIR ROLES	96
9.4.1 <i>Required supportive actions and solved issues</i>	97
9.5 INTERNAL AND EXTERNAL DEPENDENCIES.....	97
9.6 DEPLOYMENT CONCLUSIONS AND LESSONS LEARNED	97
9.7 SUSTAINABILITY PLAN UPON PROJECT COMPLETION AND FUTURE IMPROVEMENTS	98
10. SEONGNAM	99
10.1 PILOT MOTIVATIONS AND DEPLOYED USE CASES	99
10.2 IMPLEMENTATION ARCHITECTURE AND USED ATOMIC SERVICES	99
10.2.1 <i>Implemented Infrastructure</i>	100
10.2.2 <i>Changes and deviations of planned features of pilot</i>	101
10.3 INVOLVED ECOSYSTEM PARTNERS AND THEIR ROLES	101
10.3.1 <i>Required supportive actions and solved issues</i>	102
10.4 INTERNAL AND EXTERNAL DEPENDENCIES	102
10.5 DEVELOPMENT CONCLUSION AND LESSONS LEARNED.	102
10.6 SUSTAINABILITY PLAN UPON PROJECT COMPLETION AND FUTURE IMPROVEMENTS	102
11. CONCLUSION	104
REFERENCES	106

List of Figures & Tables

ROUTE SUGGESTION COMPARISON	10
ROUTE SUGGESTION COMPARISON.....	10
ROUTING REQUEST LOAD GRAPH.....	13
INITIAL APPLICATION ARCHITECTURE.....	15
IMPLEMENTATION ARCHITECTURE.....	23
MANCHESTER RZ FRAMEWORK DEPLOYMENT AND EPS TO ACCESS MANCHESTER’S RESOURCES.....	23
MANCHESTER DASHBOARD.....	24
MILAN - DECISION SUPPORT SYSTEM FOR CYCLE PATH PLANNING APPLICATION ARCHITECTURE	39
MILAN ICT INFRASTRUCTURE.....	40
MILAN - MULTIMODAL-NAVIGATOR FOR DISABLED PEOPLE APPLICATION ARCHITECTURE	48
SIGNCO DISPLAY ON THE STREETS OF EINDHOVEN	54
AN EXAMPLE OF USER INTERFACE AND TEXT VALIDATION BY USER WORKSHOP	54
CONCLUSIONS AND INSIGHTS FROM USER INTERACTION WITH EINDHOVEN’S “RING”.....	55
SCREEN CAPTION OF SOFTWARE	64
SENSOR DASHBOARD	65
DATA PROVIDED BY NOISE LEVEL SENSORS.....	68
TEMPERATURE SENSOR	69
HUMIDITY SENSOR.....	69
AIR QUALITY MEASURED IN CAROUGE.....	70
RESULTS PROVIDED BY THE PARKING ESTIMATOR.....	70
A NOISE LEVEL SENSOR IN PRIVACY APP	71
PORTO MULTIMODAL ASSISTANT APPLICATION (MOBILE VERSION).....	77
IMPLEMENTATION ARCHITECTURE.....	79
ROUTE SUGGESTION	86
SANTANDER “PARK & MOVE” APPLICATION - ROUTE PLANNING (MOBILE VERSION)	86
SANTANDER “PARK & MOVE” APPLICATION - PARKING ESTIMATION (MOBILE VERSION).....	87
SANTANDER “PARK & MOVE! APP” BASIC ARCHITECTURE.....	88
ARCHITECTURE OF THE INITIAL APPLICATION IN REFERENCE ZONE ANTWERP BY ROMBIT.	95
PLATFORM ARCHITECTURE	100
TABLE 1. ROUTING REQUEST LOAD EFFECTS ON THE SIZE OF GRAPHS.....	15
TABLE 2. DELTA	15
TABLE 3. HELSINKI CITY DATA ELEMENTS	17
TABLE 4. MANCHESTER THEMES AND SERVICES, INCLUDING SME PILOTS.....	25
TABLE 5. USE CASES OF MILAN DECISION SUPPORT SYSTEM FOR CYCLE PATH PLANNING APPLICATION	37
TABLE 6. USE CASE 8 - DATA ANALYSIS OF MILAN MULTIMODAL-NAVIGATOR FOR DISABLED PEOPLE APPLICATION	38
TABLE 7. DATA SOURCES USED IN MILAN DECISION SUPPORT SYSTEM FOR CYCLE PATH PLANNING APPLICATION.....	41
TABLE 8. USE CASES OF MILAN MULTIMODAL-NAVIGATOR FOR DISABLED PEOPLE APPLICATION.....	47
TABLE 9. DATA SOURCES USED IN MILAN MULTIMODAL-NAVIGATOR FOR DISABLED PEOPLE APPLICATION	50
TABLE 10. COMPONENT TYPES.....	56
TABLE 11. COMPONENT TYPE.....	57
TABLE 12. COMPONENT TYPE.....	59
TABLE 13. COMPONENT TYPE.....	59
TABLE 14. EINDHOVEN RZ.....	61
TABLE 15. CAROUGE RZ THEMES AND SERVICES	72
TABLE 16. TEST RESULTS PORTO RZ.....	78
TABLE 17. PORTO RZ.....	81

TABLE 18. TEST MATRIX FOR SANTANDER.....	88
TABLE 19. ANTWERP RZ THEMES AND SERVICES	96
TABLE 20. SEONGNAM AVAILABLE DATASETS	101

1. Introduction

1.1 Purpose of the document

The document contains the actions and processes that took place around the deployment of the IoT service pilots in each reference zone. It follows D3.7 [1] that outlined the plan for deployment and operational support. Services (or Initial Applications) were developed following the use cases and requirements, previously described, in D3.6 [2].

Each RZ performed validation tests that were reported in D4.2 [3]. Procedures and technical support processes are defined in WP4 and released in the same deliverable (D4.2)

1.2 Structure of the deliverable

The document is structured as follows. A short explanation of the described topics is provided.

1.2.1 Pilot motivations and deployed use cases

This subsection provides a brief description of the piloted use cases and outlines the motivations for the particular city to pilot that solution.

1.2.2 Test results of pilot

This section provides explanations on implemented tests, from concept validation to stress tests and outlines the results. Not all RZs made tests on their solution, their corresponding chapters will explain why.

1.2.3 Implemented infrastructure

This section lists the existing datasets used as well as resources installed particularly for the deployed pilots. In particular, RZs' applications leverage ICT infrastructure deployed in RZs themselves; these are compliant with SynchroniCity specifications. More detailed information about implemented ICT infrastructure in each RZ is reported in D2.9 [4].

1.2.4 Involved ecosystem partners and their roles

This section aims to give a picture on the player ecosystem and outlines the partners involved (either directly or indirectly).

1.2.5 Internal and external dependencies

This chapter explains the internal dependencies that affected the pilot operation and relations with other work packages and their deliverables (WP2, WP4, WP5) and dependencies within the deliverables, schedules and resourcing of other tasks of the same WP3 package.

The section also outlines the external dependencies such as links to outside data providers.

1.2.6 Deployment conclusions and lessons learned

An overall concluding chapter that wraps up the deployment and outlines the lessons learned.

1.2.7 Sustainability plan upon project completion and future improvements

Any plans for future implementation, adoption or future development of the solution is explained in this chapter. Additionally, arguments for not going further with the solutions are laid down for debate.

2. Helsinki

2.1 Pilot motivations and deployed use cases

Air quality has been a prominent theme in the last few years. Even though air pollution is not a major problem in the Helsinki city area, the pilot offered a good opportunity to test the concept and implementation to an already existing and widely used routing service (owned by the Helsinki Regional Transport Authority HSL¹). People with breathing challenges or other pollution related concerns could still find routing suggestions based on cleaner air useful and needed. The pilot application aimed to provide an extra feature to the existing journey planner by adding an air quality routing parameter.

Even though air pollution does not pose 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.

The deployed use cases follow the typical example set in D3.7:

Alice has a 6 months old baby so she usually moves along the city with her pram. She likes to walk with her baby in a big park on the other side of the city. If the weather is fine, she usually walks part of the way to the park. If there is a possibility of rain, she may combine a walk with a bus ride. The Journey Planner with air quality information app offers her all the possible combinations, including bus lines, metro and urban mobility infrastructures, so Alice can configure her path before leaving home. It also includes current information about noise and air quality, which allows her to avoid crowded, noisy and polluted streets.

During tentatively surveys cyclists and people with small children or respiratory challenges showed interest toward clean air optimisation services. The survey results have been confirmed with follow up surveys and interviews. The deployed uses cases mirror those findings.

¹ <https://www.hsl.fi/en>

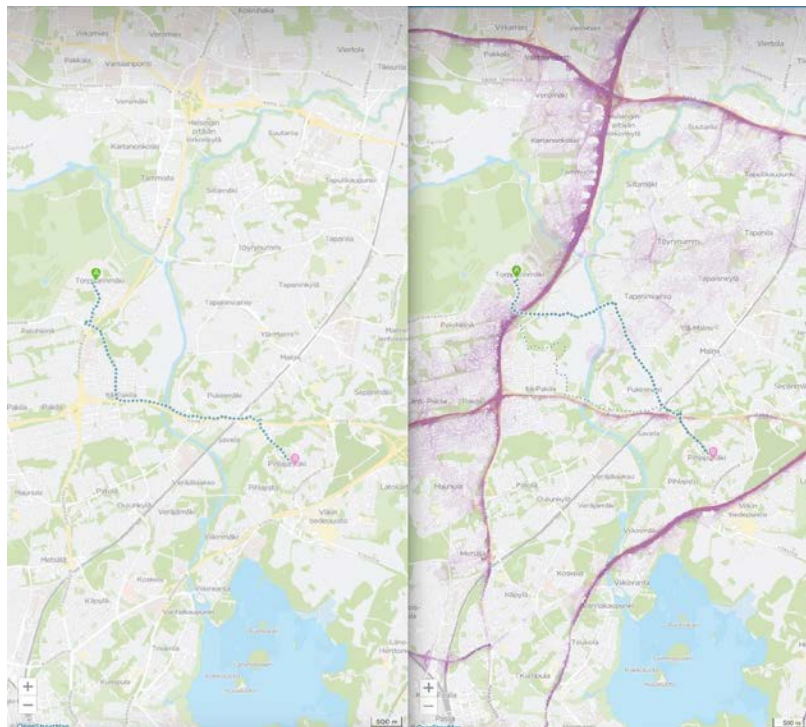


Figure 1. Route suggestion comparison

Figure 1 and Figure 2 show the changes in route suggestion when air quality is factored in. The Figure on the left is from the existing Routing app (Reittiopas²), the Figure on the right is from the RZ application. The purple layer is there to visualize pollution.

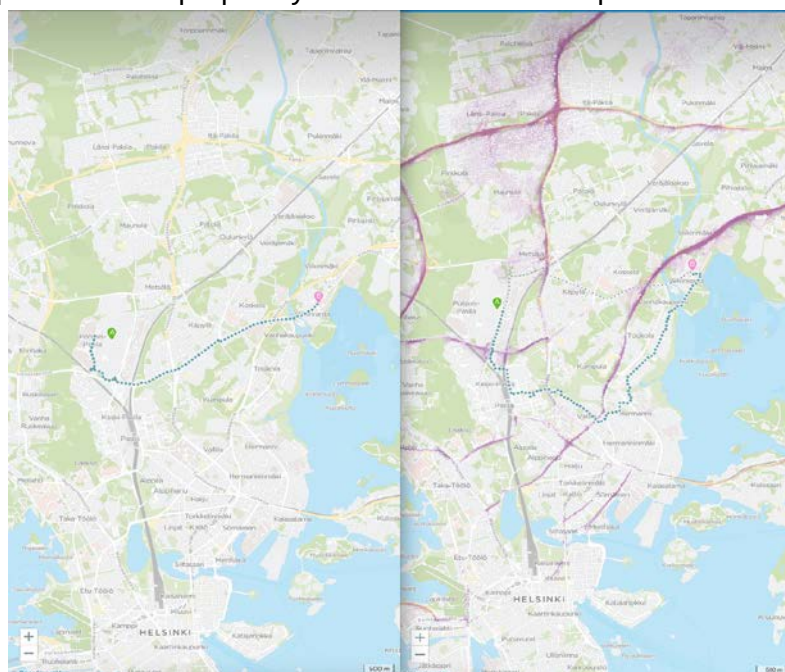


Figure 2. Route suggestion comparison.

² <https://reittiopas.hsl.fi/?locale=en>

2.2 Test results of pilot

Concept testing & validation

The pilot was concept tested by monitoring responses in social media regarding news on the service which was reported in several articles. Additionally, more than a dozen interviews were conducted in two (2) occasions: the clean air routing service was introduced and demoed after which a structured although free flowing conversation was conducted and documented.

According to the information gathered both from the social media responses and from the field interviews, the originally identified target groups (e.g. parents with small children) are the first choice among responders as well. Also, people with respiratory problems are identified.

All in all, the concept testing phase seems to confirm in part the hypothesis. Even in a city with relatively clean air, people with small children (for example parents and grandparents), people with respiratory problems and cyclists are open to the idea of using clean air routing on a daily basis.

Important feedback regarding usability and UI was also gathered through field tests. Listening to the feedback some visual elements that were introduced during the developing process were removed. Some suggested extra features were noted and duly documented.

A further end user study for 16 subjects was run to capture opinions in more quantitative manner. The project and journey planner background were presented shortly, then the subjects were asked to sign a consent form. Their demographics were then registered on a questionnaire. All journey plans in the tests were created for pedestrians.

Subjects were first shown the normal Helsinki Journey Planner (Reittiopas) and asked to create one route between two of their known locations and two predefined pedestrian routes, and describe them to a) establish their ability to understand the map interface and b) better immerse themselves with the routes. They were then shown the Clean Air Journey Planner, and asked to create the same three routes as for the normal Planner. This time, they would receive an alternative route that would provide cleaner air. They were again asked to describe the routes as if they were walking along the route. They were also encouraged to consider whether or not they would take the alternative path. A few more routes across the city, and free routes, were requested from the subjects.

After testing both the normal Journey Planner and the Clean Air version, subjects were asked to proceed with the questionnaire.

Of the subjects, almost half suffered from asthma or allergies (6/16). Almost all would avoid polluted areas if they could (15/16). Most people had not previously heard of clean air journey planners (11/16).

After testing, almost all (15/16) were happy with the map's air quality overlay in conveying the location of pollution in the city. Most also claimed that the overlay did not affect the readability of the map (11/16). Of note, **100% of respondents felt that the usability of the Clean Air version was the same as the normal version**. Of potential additional features, most users wanted to avoid road dust (12/16), over half would use routing to avoid allergenic particles

(9/16) and half would use routing for the asthmatic (8/16). Half also wanted to configure the importance of air quality themselves (8/16).

Qualitative feedback regarding the Clean Air Journey planner included observations on the actual routes. The system prefers indoors, as the air quality data always shows zero pollutants for building locations; thus, whenever possible, the routing engine will provide routes through buildings. Many subjects would not enter complex buildings as the routes were often quite complicated and artificial. Often these routes were also considerably longer, and sometimes it was unclear from the map whether the route would actually be in a tunnel or not. Also due to the low amount of pollutants during the tests, the map interface appeared quite clean, and some subjects needed further assurance that there actually is some pollutants in the air, as there was no clear visualization, while the suggested alternative route was clearly different from the shortest one.

Some subjects would have preferred to see some value regarding the exposure to be able to estimate the difference. Some suggested an enhanced pollutant visualization for the current view, and a legend that would establish numerical value for given colors, allowing a change in pollutant color ramps across different locations or zoom levels.

In general, the response from subjects was very positive, especially if they were given a chance to just quickly check the air quality. Those of the subjects who were especially sensitive to bad air quality, explained verbally that they are already avoiding certain areas where dust and exhaust cause irritation.

Stress testing

The test was executed using the Loader.io-service³. In the test, the server running the OpenTripPlanner was Scaleway's C2L (8 dedicated cores and 32 GB of memory).

The used server was Scaleway's "bare metal"-type server⁴ which means virtualisation doesn't affect the results.

During the test four (4) different versions of OpenTripPlanner⁵ (OTP from now on) were used to help make the comparison between the performance impact of different functions:

- No changes
- With air quality routing
- With noise level routing
- With air quality and noise level routing

Before running the tests, a graph was calculated for every version including only the information the version needed.

³<https://loader.io/>

⁴<https://www.scaleway.com/baremetal-cloud-servers/>

⁵ <https://www.opentripplanner.org/>

The same tests were run for every version:

1. A test where the load was increased with simultaneous routing requests; from 0 requests to 50 requests in one (1) minute.
2. A test where the load was increased with simultaneous routing requests; from 0 requests to 100 requests in one (1) minute.
3. A test where 200 simultaneous routing requests were run in one (1) minute.

Test results

Differences between systems were little and inside the measurement error margins. Variations in builds affected performance metrics so little, that they can be said to have no impact on the system performance.

The system configuration made for test purposes performed well in Test 1 and 2. Test 3 performance was on the limit: the system managed most of the requests but not nearly all of them. During the tests, a 15 seconds timeout was configured in the system so that routing requests that lasted over 15 seconds were interpreted as failures.

The system can momentarily handle 200 routing requests and without any significant slowing down 100 routing requests. The numbers don't directly scale with user amounts since individual users do asynchronous routing requests; the number of users the system can handle is in thousands in any case.

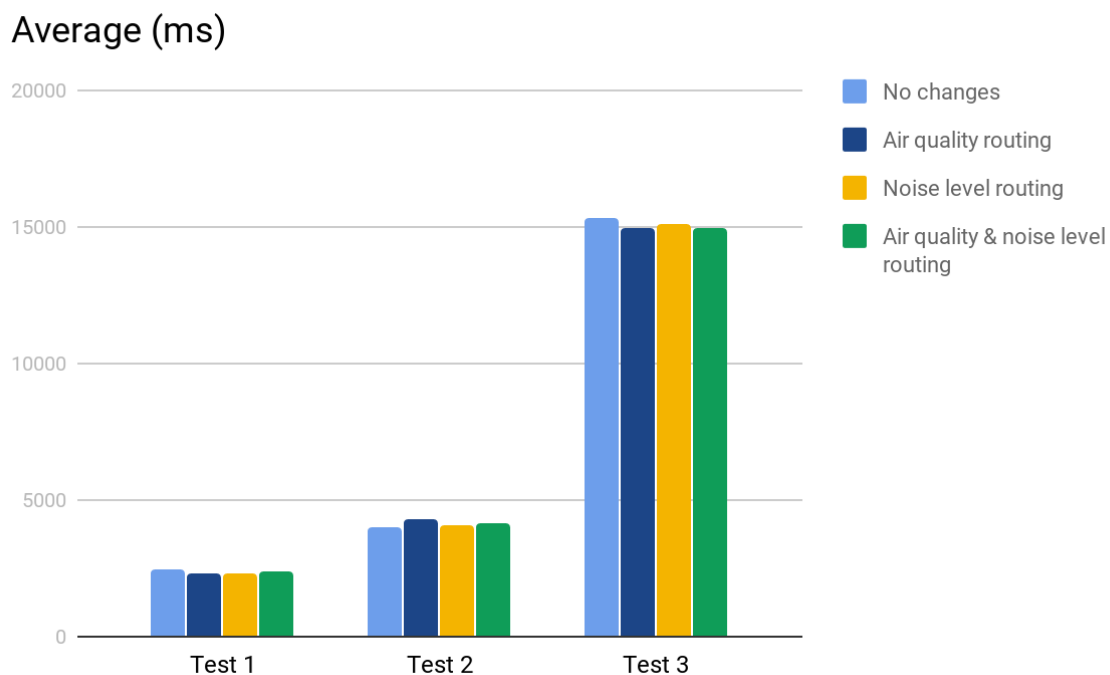


Figure 3. Routing request load graph

No changes

	Test 1	Test 2	Test 3
Average	2468	4039	15343
Minimum	571	545	2019
Maximum	4583	7776	20121
Successful	598	660	401
Errors	0.0 %	0.0 %	42.7 %

Noise level routing				Air quality routing			
	Test 1	Test 2	Test 3		Test 1	Test 2	Test 3
Average	2315	4074	15129	Average	2280	4320	14944
Minimum	450	470	1976	Minimum	580	516	2996
Maximum	4938	8348	21669	Maximum	4398	8116	21443
Successful	616	643	398	Successful	629	657	400
Errors	0.0 %	0.0 %	43.0 %	Errors	0.0 %	0.0 %	43 %

Air quality and noise level routing

	Test 1	Test 2	Test 3
Average	2374	4131	14994
Minimum	557	554	1486
Maximum	4672	8558	22007
Successful	600	636	397
Errors	0.00%	0.00%	43 %

Table 1. Routing request load effects on the size of graphs

Adding air quality and noise level data in the graphs had a minimal effect on the final size. The original graph was a little under 2 GB of size. Including air quality and noise level data affected less than 1%.

This table shows more detailed information:

Version	Size of graph	Change (delta)
No changes	2035681907	0.00%
Air quality routing	2045892950	0.50%
Noise level routing	2044771197	0.44%
Air quality and noise level routing	2054982325	0.94%

Table 2. Delta

2.3 Implementation architecture and used Atomic Services

The used architecture has been introduced in D3.5 (“Customized IoT service prototypes for lead reference zones”) and was implemented without changes.

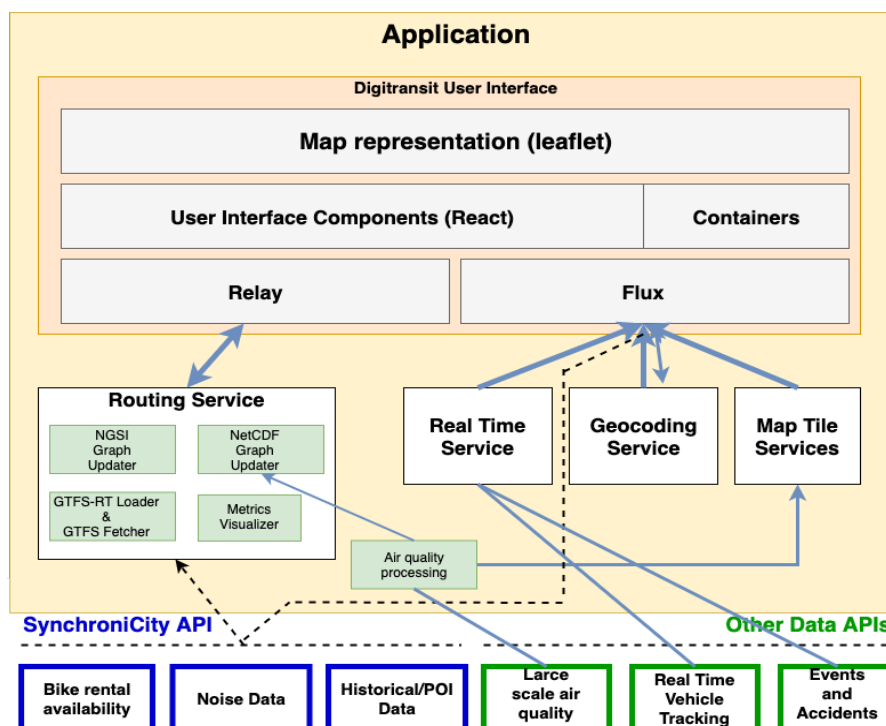


Figure 4. Initial application architecture.

Used Atomic Services

Routing

The Routing atomic service allows users to execute queries for finding routes, bus and taxi stops, cycling routes, city bike docking stations and disruption information. In the application the service is also used for generating multi-modal routing factoring static and dynamic information such as air quality model information.

This atomic service is based on the OpenTripPlanner⁶. Requests to this service can be done through the original OTP API or through GraphQL. City information is injected to the SynchroniCity framework using atomic services such as GTFS Fetcher⁷ and GTFS-RT Loader from NGSI.

Metrics Visualizer⁸ (Grafana)

This atomic service is a custom deployment of Grafana⁹ software. Through this service, a set of databases can be configured as data sources in order to create personalized charts and tables. The databases are filled using QuantumLeap and FIWARE Cygnus as device information subscribers.

GTFS-RT Loader from NGSI

This atomic service provides notifications of real-time information in a widely used format, GTFS-RT¹⁰. The atomic service consumes information following the SynchroniCity data model using an NGSIv2 API and generates GTFS-RT feeds from it. The real-time entities stored in the SynchroniCity instances are bound to a GTFS feed file that contains the static data, what allows the service to support real-time provisioning for multiple GTFS feeds. From a user perspective, the service is seen as a regular GTFS-RT provider.

This service can be used in combination with the Routing atomic service in order to feed it with transportation information stored within the SynchroniCity framework.

2.3.1 Implemented infrastructure

The infrastructure was deployed according to D2.9.

Sensors & data:

⁶ <http://www.opentripplanner.org/>

⁷ <https://hub.docker.com/r/synchronicityiot/gtfs-fetcher>

⁸ <https://hub.docker.com/r/synchronicityiot/grafanaatomicservice>

⁹ <https://grafana.com/>

¹⁰ <https://hub.docker.com/r/synchronicityiot/gtfsrt-loader-from-ngsi>

- **Air Quality**
The data from the Finnish Meteorological Institute¹¹ air quality sensors was directed to the NGSI Context Broker deployed for Synchronicity.
- **Noise**
A small number of Cesva TA120¹² -noise measuring sensors have been deployed around Helsinki along the years. The data from those sensors was redirected to the NGSI Context Broker deployed for Synchronicity and utilized in the routing application.
- **Bike Docking Station**
Availability of city bikes in the station with particular coordination.

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

Table 3. Helsinki City Data Elements

Platforms

- **Digitransit¹³**
An open source journey planning solution that combines several open source components into a route planning service.
- **OpenStreetMap**
An open map and community driven data provider
- **Orion context broker**
Synchronicity Context Broker for NGSI data.

¹¹ <https://en.ilmatieteenlaitos.fi/>

¹² <https://www.cesva.com/en/products/sensors-terminals/TA120/>

¹³ Digitransit homepage: <https://digitransit.fi/en/>

2.3.2 Changes and deviations of planned features of pilot

The integration of additional air quality data sources was not implemented as planned at some point. The Select4Cities¹⁴ (a EU-funded project in the Forum Virium Helsinki portfolio) platforms were not mature enough and could not provide the aggregating factor that was hoped for.

2.4 Involved ecosystem partners and their roles

Three partners worked in the Helsinki RZ initial application solution:

- Forum Virium Helsinki
A non-profit company owned by the City of Helsinki. Partner in the SynchroniCity consortium.
- Aalto University
A multidisciplinary University based in the Helsinki Metropolitan area.
- Metatavu Oy (subcontractor)
A Mikkeli-based software company specialized in open source solutions. Was subcontracted for 425 hours of developing work. Practically all hours have been used and a final product (working PoC of the clean air routing service) delivered.

Other organizations were involved indirectly by providing openly available technologies:

- HSL (Helsinki Region Transport)
HSL is a joint local authority that has developed an open source solution for journey planning called Digitransit
- Finnish Meteorological Institute
Enfuser¹⁵ is an air quality modelling tool developed by the Finnish Meteorological Institute and was released in 2018 under a Creative Commons license.

2.5 Internal and external dependencies

The pilot depends on WP2 in terms of architecture compliance, data marketplace availability. Pilot roll out and execution is strongly dependant on availability of atomic services and schedule for initial application development in each RZ. On validation protocols and procedures, the pilot is strongly dependant on WP4.

The external dependencies emerge mainly from data providers. In Helsinki case noise sensors were installed by the project. The air quality data is coming from the external provider (Finnish Meteorological Institute) and the city bike docking station status comes from HSL. The latter is not part of the core data sources since it doesn't affect routing.

¹⁴ <https://www.select4cities.eu/>

¹⁵ <https://en.ilmatieteenlaitos.fi/environmental-information-fusion-service>

2.6 Deployment conclusions and lessons learned

One thing that was learned during the process was that the air quality model used (FMI-Enfuser¹⁶) is not easily deployable in other cities. It requires for example a 3D model of the city for dispersion modelling. Nevertheless, more and more cities are developing a 3D model of the city so this issue, for one, will be solved in the future years. Furthermore, the used air quality model Enfuser was suitable for an existing open standard (NetCDF) that would have been scalable. For testing purposes and apparent motivations to use SynchroniCity developed technologies, the standard was not used and a wrapper (NetCDF2NGSI) was developed instead. Scalability, as for now with these specifications, is a challenge.

The best way to dissipate the air quality routing is to incorporate it into existing routing services (e.g. Reittiopas). As a standalone product it risks to become a niche service, especially in cities with relatively good air quality. The application was left as it is (a standalone service) to not “undo” the development done. It is ready to be chopped into relevant pieces for dissemination purposes though.

According to light and preliminary probing there is demand for this kind of particular niche service. The fact that the tested concept was visually very similar to the most popular routing service surely helped the testing audience welcome it more easily.

On a more technical note an important discovery was made regarding the usage of air quality models in the context of Synchronicity framework: virtual air quality measurement points do not scale in the data model and a lot of calculations are needed if a 10x10 meters measure point matrix is created over a City area. The computing power needed is, at least in the scope of this project, simply unrealistic.

In general, the Minimum Interoperable Mechanisms concept has been useful as a theoretical framework. The initial application pilot proved to be a useful process to validate and iterate this idea. Further improvements of the concept and the MIM technologies is welcomed.

2.7 Sustainability plan upon project completion and future improvements

Themes around air quality have been trending in the last year. FVH has been part of the efforts to disseminate information about the gathering of data, how to interpret it and how to develop services that utilize this data.

A project in the FVH portfolio, UIA HOPE¹⁷, is looking into developing a similar service. Contacts have been made to adjust goals and arrange competence and technology transfers.

¹⁶ <https://en.ilmatieteenlaitos.fi/environmental-information-fusion-service>

¹⁷ <https://www.uia-initiative.eu/en/uia-cities/helsinki>

The organisations involved are: University of Helsinki, City of Helsinki, Vaisala Oy, Finnish Meteorological Institute, Useless Company Oy¹⁸, FVH.

In the scope of UIA HOPE, the Enfuser model will see iterative improvements such as the incorporation of boat emissions into the air quality estimation in coastal areas. A new GitHub repository¹⁹ has been set up by the UIA HOPE partners and documentation procedures have been agreed upon. Regular communication exchange on the matter has already begun. As mentioned in the previous chapter, possible modulation of the service is needed to lower adoption barriers for HSL (or other similar organisation for that matter) into their service Reittiopas. Helping adoption of developed technologies will continue on a project level.

3. Manchester

3.1 Pilot motivations and deployed use cases

Additional detail can be found in D3.6 “Customized IoT service prototypes for lead ref. zones”.

The Manchester baseline application focused on the “community policy suite” theme which as the project developed became more clearly focused on a particular stakeholder approach in Manchester around “agile governance”.

Focus groups identified that city officers and partners were aware of a wide range of data, both real time, and static, but without expert help from data analysts in the council were unable to easily access it in a timely fashion in order to make planning and reporting decisions.

During 2016-18 Manchester had taken part in the UK funded IoT-UK programme and had become the city demonstrator. This meant that parallel to Synchronicity we were seeing

- new sensor data becoming available
- development of a city data platform (“The BT Hub²⁰”)

It was to take advantage of this new data, and to provide easy access to it for officers across the council that the pilot was developed.

Manchester City Council worked with Manchester Metropolitan University to develop the methodology around agile governance, and the aim was to deploy particular use cases to validate this, and at the same time integrate this with the Synchronicity framework so that there could be both a cross-service platform for different city data services and a cross-city platform implementing the Synchronicity marketplace.

The main points of validation were:

¹⁸ <https://useless.fi/>

¹⁹ <https://github.com/DigitalGeographyLab/hope-green-path-server>

²⁰ <https://datahub.rp.bt.com/>

To understand how “virtual teams” within the city council work and could have access to the same data despite often being in different departments with different data access permissions

Through enabling officers to create their own dashboards using a range of data sources whether it would provide them with the necessary tools in order to make better planning decisions.

Manchester Metropolitan University and Manchester City Council shadowed several teams the city’s Complaints Team, Freedom of Information Team, and Call Handling teams to validate the first of these, identifying the wide range of data sources they used, and the wide range of departments that they were in contact with in order to resolve particular issues and complaints. Manchester City Council identified that there were officers across the council who needed access to air quality and traffic management (people and vehicle counting, parking) on an ad hoc basis and often had no easy access to this.

In order to validate both the Synchronicity framework and the pilot application Manchester City Council agreed to focus on specifically IoT sensor data - with the aim of overlaying other data as and when the above services were in a position to adopt and access appropriate applications.

The pilot application allows city council officers to view data from sensors and analyze how changes in policy affect measurable results such as traffic flow, available parking and air quality. In addition to this, the software allows policy makers to document their findings through a feature which allows users to create and collaborate on documents.

3.2 Test results of pilot

Manchester City Council and Manchester Metropolitan University conducted a System Test in August 2019, which looked at the functional specifications of the pilot application developed by Bronzelabs and tested the functionality.

The test focused on a number of areas of the software:

1. User access permissions
2. Data availability
3. Dashboard creation
4. Virtual team allocation
5. Dynamic policy document

A brief summary is included:

User Access Permissions - different levels of access are provided by the service. These are intuitive and easy to use. We were able to set up different access levels. Certain functions do require to be set by the software company and in production version it might be good to make these available locally.

Data Availability - data availability via the Synchronicity framework and the Orion context broker. Some issues were present around the historical data, we are told by Bronze Labs that these issues were due to the context broker being updated to implement OAuth, Bronze Labs have re-integrated with the service and this issue has now been resolved. Manchester City Council does have access to additional data sources that are not currently visible (these have to be integrated by Digital Catapult) and the presentation of this data would need to be done in a more meaningful way before we could go live with the system.

Dashboard Creation - dashboards can be created and customized by users, so that they can view metrics and data that is relevant to them. The functionality is good and we were able to view data from the Synchronicity platform. Many kinds of counters and graphs can be added to a dashboard through a simple drag and drop interface. As there are so many options for customizing the graphs, it can take some time to become familiar with this function and get the full benefits of these features.

Virtual Team Allocation - this enables the creation of virtual teams and is successfully integrated with the user access permissions.

Dynamic Policy Document - this was a new development from the early iterations and addresses some of our previous concerns. We'd like to see it integrated more fully with the dashboard and able to access data in a dynamic way, but as an additional feature it provides some potential.

Overall from the user testing, Manchester City Council found the adapted framework⁴² software to work well and provide useful functionality. Dashboards allowed us to access data via the Synchronicity framework, making it Synchronicity compliant, however, it was not possible in the testing scenario to fully validate the service as the data was partial and there were issues with accessing historical data. We would expect these issues to be addressed prior to a full deployment and user acceptance testing.

3.3 Implementation architecture and used Atomic Services

Framework⁴²

Cloud-hosted smart city middleware which consumes data from the Synchronicity Framework (or the context management API, historical data API) through an NGSI interface. Users can browse and subscribe to available data-sets from the Synchronicity Framework, then set up dashboards, policies and workflows to automate alerts and actions based on real time data.

²¹ <https://www.bronzelabs.co.uk/framework42>

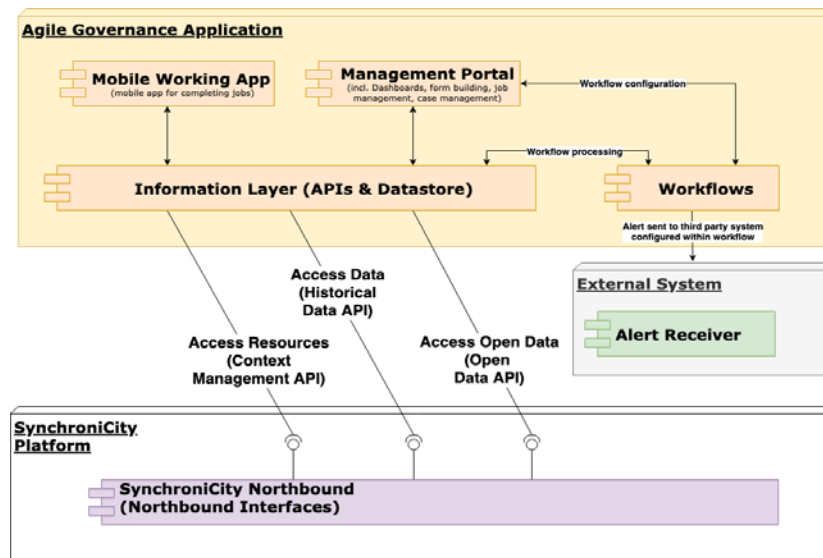


Figure 5. Implementation architecture.

Additional detail of Bronze Labs architecture present in D3.6.

Orion Context Broker

Manchester RZ's IoT infrastructure has been adapted to be SynchroniCity compliant by converting the original data models (based on HyperCat) in the SynchroniCity ones. To that aim, NGSI adaptors have been implemented in Node-RED²² by DigiCat.

Additional detail present in D3.5 [5].

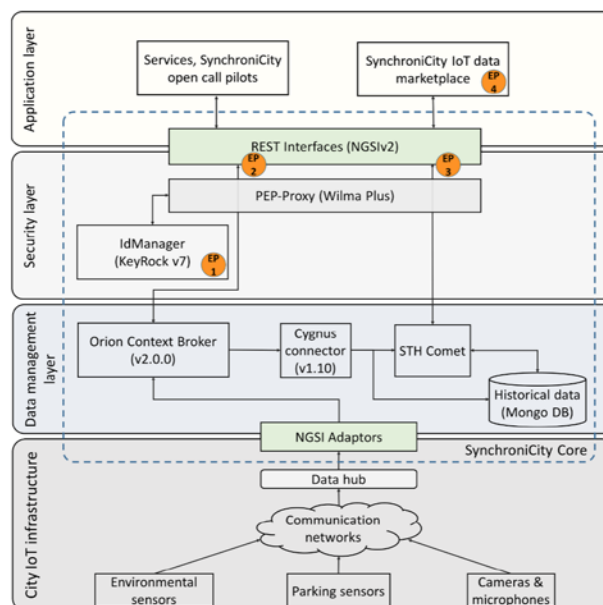


Figure 6. Manchester RZ framework deployment and EPs to access Manchester's resources.

Used atomic services

²² <https://nodered.org/>

Metrics Visualizer (Grafana)

Data visualization software allowing users to view NGSI data from the context broker.

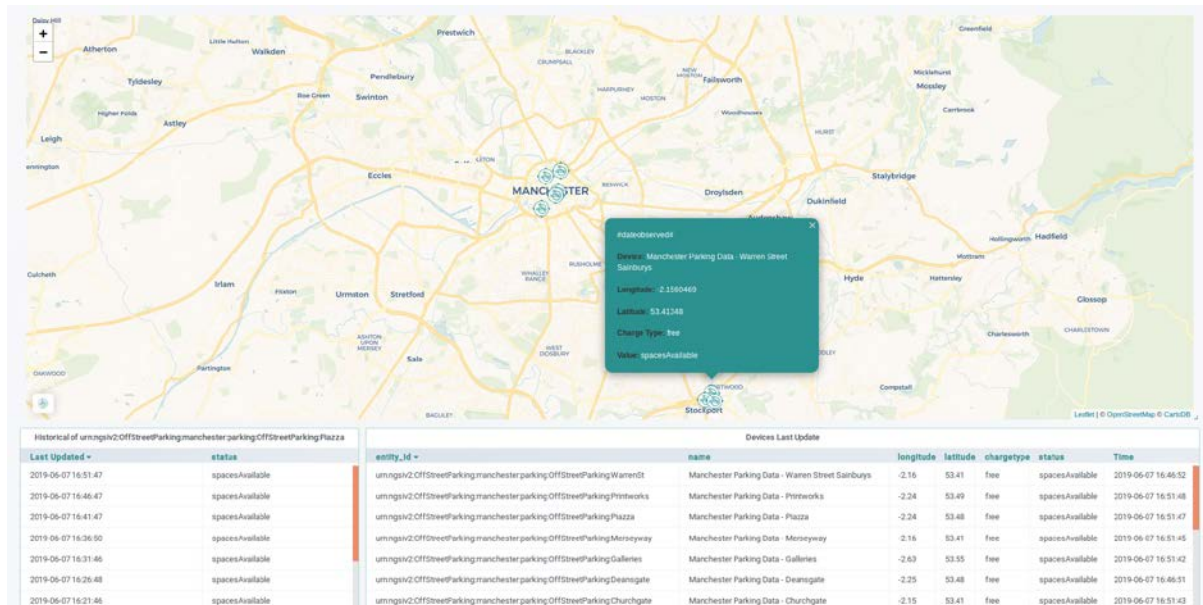


Figure 7. Manchester dashboard

3.3.1 Implemented infrastructure

Sensors & data

- Air Quality
- Traffic flow
- CrowdfLOW
- Parking availability

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

Smart Waste Management	Neighbourly	n/a
------------------------	-------------	-----

Table 4. Manchester themes and services, including SME pilots

Platforms

- Framework42 application
- Orion context broker

3.3.2 Changes and deviations of planned features of pilot

As detailed in D3.6, the citizen use case was removed to ensure the scope of the pilot was realistic and achievable.

3.4 Involved ecosystem partners and their roles

- Synchronicity Partners:
 - Bronze Software Labs: Application development, technical validation
 - Digital Catapult: Application development
 - Manchester City Council: Use cases, user validation
 - Manchester Metropolitan University: Use cases, user validation
 - Hopu: Grafana deployment
- Suppliers
 - BT: Sensors

3.4.1 Required supportive actions and solved issues

User testing was held between Bronze Labs, MMU and Manchester City Council, who are also representative of the target user group, several areas were highlighted where usability of the platform could be improved. One example, was on a webpage where the user selects which sensors they would like to view data for, initially the interface was a large dropdown list containing all sensor IDs. These computer-generated IDs made it hard to identify the sensors, some brainstorming highlighted a good option would be to instead show the user a map with a marker for each sensor. Allowing users to select sensors based on their physical location. Several other areas of the portal were also refined through the same process, this was completed in a face-to-face workshop in Manchester at the start of 2019. Another round of internal acceptance testing was completed by Manchester City Council and MMU in August 2019.

More recently we have been accommodating some changes with the Orion Context Broker hosted by Digital Catapult, this server hosts the API which framework42 consumes and saves sensor data from. The authentication method and several underlying API calls were recently improved, as a result our existing integration was no longer functioning which had broken some functionality in our online portal. Another round of development and technical communications was required to restore the missing functionality.

3.5 Internal and external dependencies

Internal dependencies

- Availability of context broker

External dependencies

- Data from existing sensor networks
- Availability of sensors

3.6 Deployment conclusions and lessons learned

The main lessons learnt for Bronze Labs were around the internal testing processes we follow, the technical testing was relatively successful, but internal usability testing was being completed by people already comfortable with the software. This led to oversights in areas where the user experience of the online portal was not ideal, this was highlighted when people unfamiliar with the adapted framework42 solution began testing. Internally we now have more stringent processes to resolve this, including the use of basic “user stories” which ensure basic functionality is easy to use.

Manchester City Council and MMU gave consistent feedback throughout the rollout of the framework42 platform, we were able to develop and release iteratively in order to make the platform more functional and easier to use. One issue that was highlighted was around the performance of the online-portal, pages initially took a long time to load which made the user experience frustrating. Bronze Labs then spent a large amount of time refining code which made a few specific pages much faster. The greatest benefits came from increasing the size of the deployment server, in hindsight we should have done this much sooner as the initial server was a very basic cloud offering. After moving to a production-level server the system performance was greatly improved, which made both using and testing the platform a much better experience for Manchester City Council.

From a city perspective we had some constraints given delays in the deployment of the software. Partly this was because of the above dependencies, but we think a minimum viable product (as agreed with the developers) was one way of ensuring we could undertake meaningful system testing with both live and test data, and could have then looked to some limited User Acceptance Testing (UAT) to trial the more advanced functionalities of the system.

In retrospect, the delays in an agreed specification for the Synchronicity framework, and difficulties in particular, in onboarding time series data, limited our ability to undertake a meaningful trial with end user departments. As stated above, final testing of the software was limited to system testing, and compliance with the Synchronicity framework.

As the above timeline indicates, the delivery user testing took place in August 2019. Our efforts on testing the agile governance methodology developed with MMU were by that time focused on the software provided through the open call. However, we will continue to look for suitable end user cases for the software suite.

3.7 Sustainability plan upon project completion and future improvements

Unlike several other application themes, we had a working basic version of our platform before the Synchronicity project which was used as a baseline. During the course of this project framework42 has evolved significantly through field testing and end-user feedback from the partner reference zones. The workshops hosted between ourselves and the Manchester partners have been especially useful in ensuring the product is easy-to-use and functional.

Internally at Bronze Labs our business strategy and direction have also changed greatly over the last few years, we are now developing software that operates within the Health & Social Care sector. Although this is some distance away from the Smart Cities industry, the software has provided a great foundation for our new product, which is able to integrate with existing council systems.

4. Milan

4.1 Human-centric Traffic Management Application "Decision support system for cycle path planning"

4.1.1 Pilot motivations and deployed use cases

In Milan there is a growing interest in improving the cycle network due to the increasing usage of bicycles, mainly thanks to the authorization in the last two years of new free-floating bike sharing services in the city. Moreover, the opportunity given by the Synchronicity project to gather data from sensors made possible the recognition of all the available data around the cycling issue and put them together.

Under these premises, the aim of the "Decision support system for cycle path planning" pilot application was to create a tool to gather all the information that could be interesting to the bicycle mobility to be used by the Mobility Agency (AMAT) of the Municipality of Milan.

Under this goal a set of use cases (or scenarios) reported in Table 4 have been developed; a detailed description of the use cases is available in deliverable "D3.5 Customized IoT service prototypes for lead ref. zones – basic". Furthermore, in addition to the use cases (scenarios) identified and described in D3.5, a new use case has been identified: UC. 8. This use case is based on data provided by the Open Call pilot application KissMyByke²³.

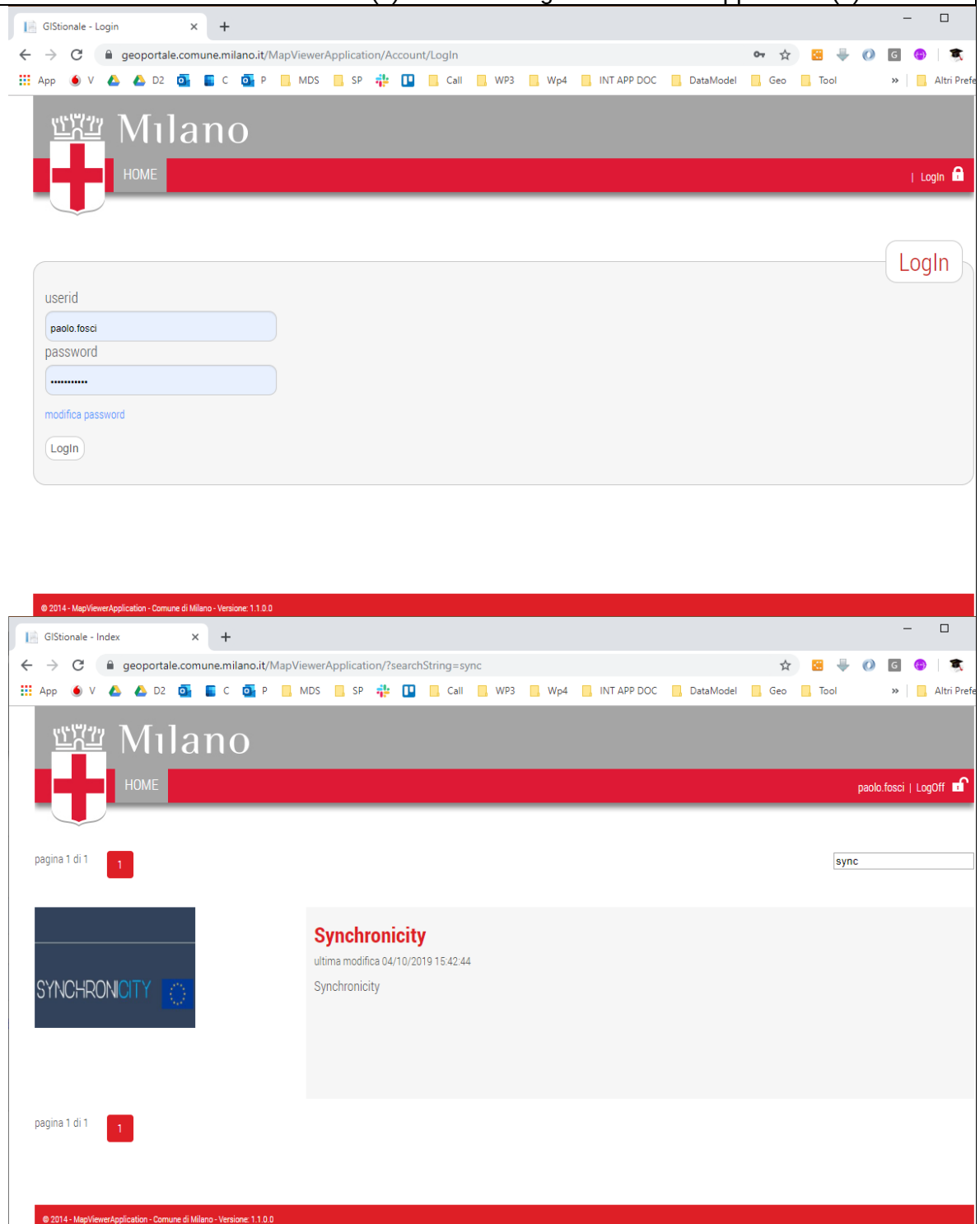
The KissMyBike pilot provided anti-theft devices to be installed on 15 bikes of the BikeMi²⁴ station-based bike-sharing service. The anti-theft device fed the Synchronicity platform of Milan with the GPS-position of the bikes for 5 weeks across May-June 2019. The KissMyBike data were unexpected when "*Decision support system for cycle path planning*" was designed and so when the KissMyBike experimentation has finished all the collected data could have been used to improve the application showing the actual position of the bikes and making statistical analysis on data.

²³ <http://www.kissmy.bike/en/>

²⁴ <https://www.bikemi.com/>

UC. 1: Application Access.

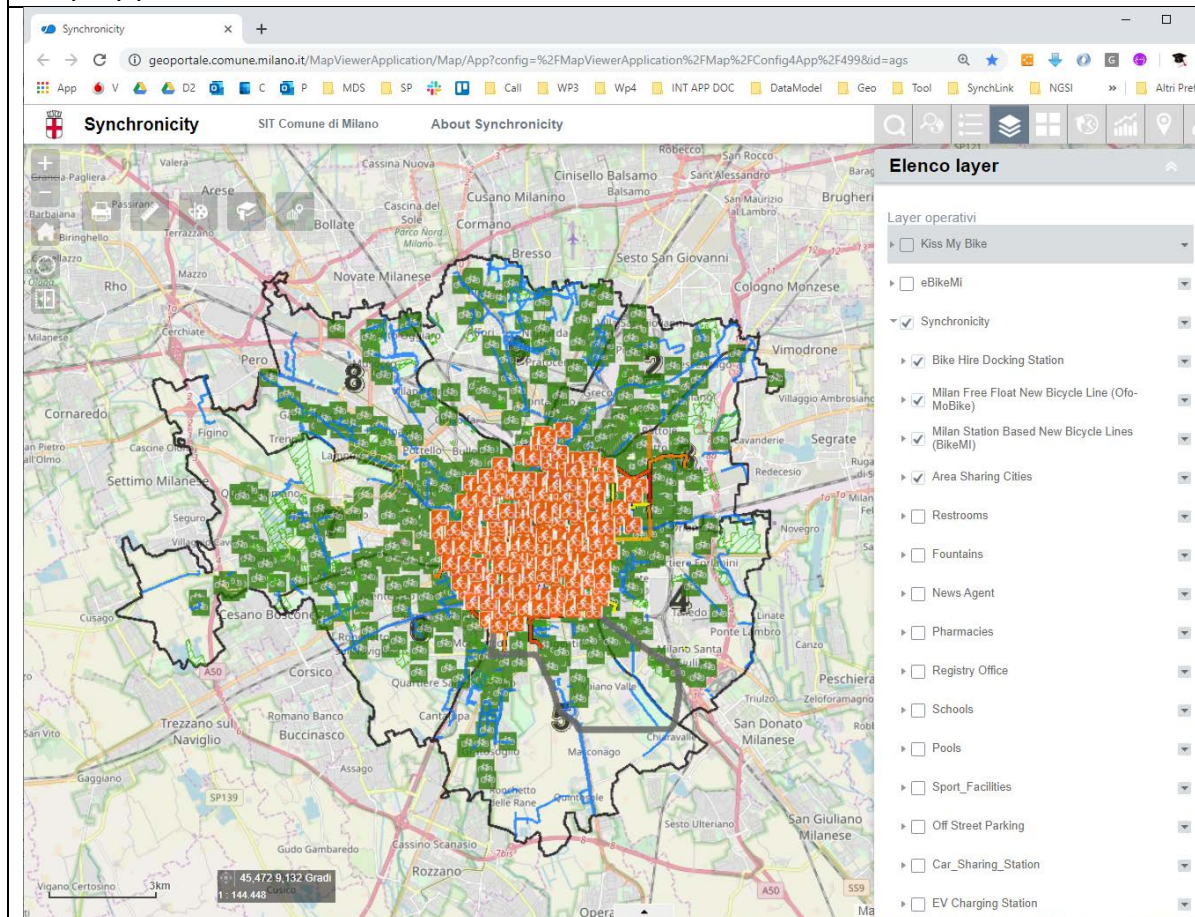
The application is available through Milan internal geoportal accessible via browser. The User must insert valid credentials (a) before having access to the application (b).

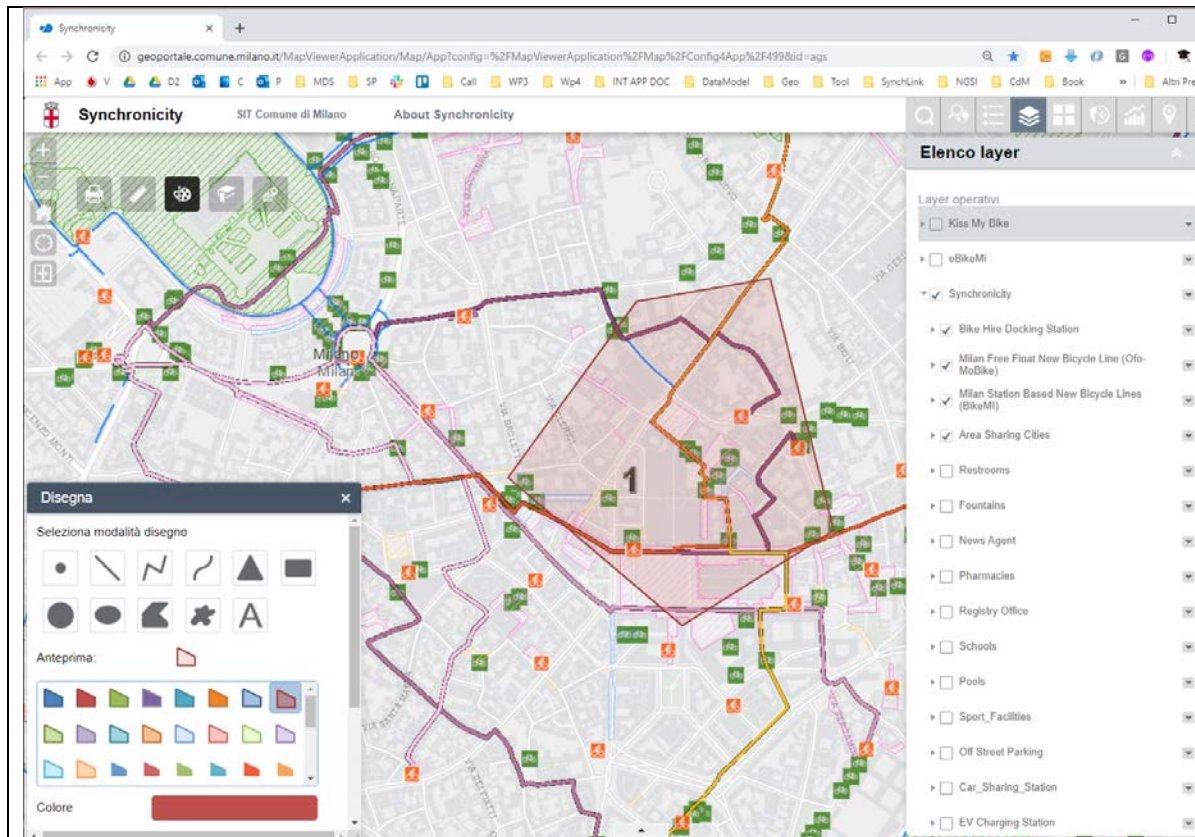
**UC. 2: Map interaction - Zooming, scrolling, and area selection.**

After the User has launched the application, the initial screen shows a map of the entire area of Milan with some information layers visible by default (a). The default information layers are bike docking station (green and orange icons), bicycle lines (blue lines),

Origin-Destination bike-sharing usage lines calculated by Synchronicity routing service (yellow-orange-red lines).

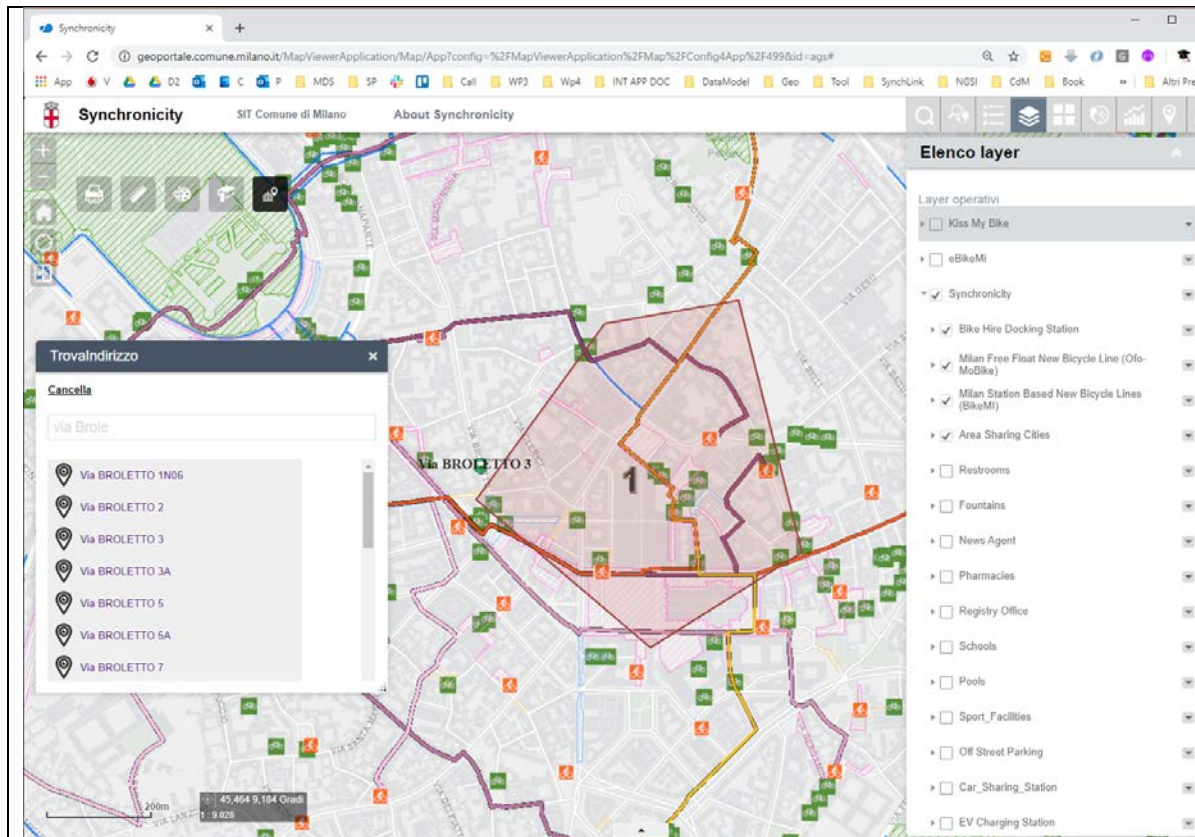
Then the user has the chance to zoom and scroll the map in order to show the area the User is interested in, and in case the User needs to highlight a specific area by drawing a shape (b).





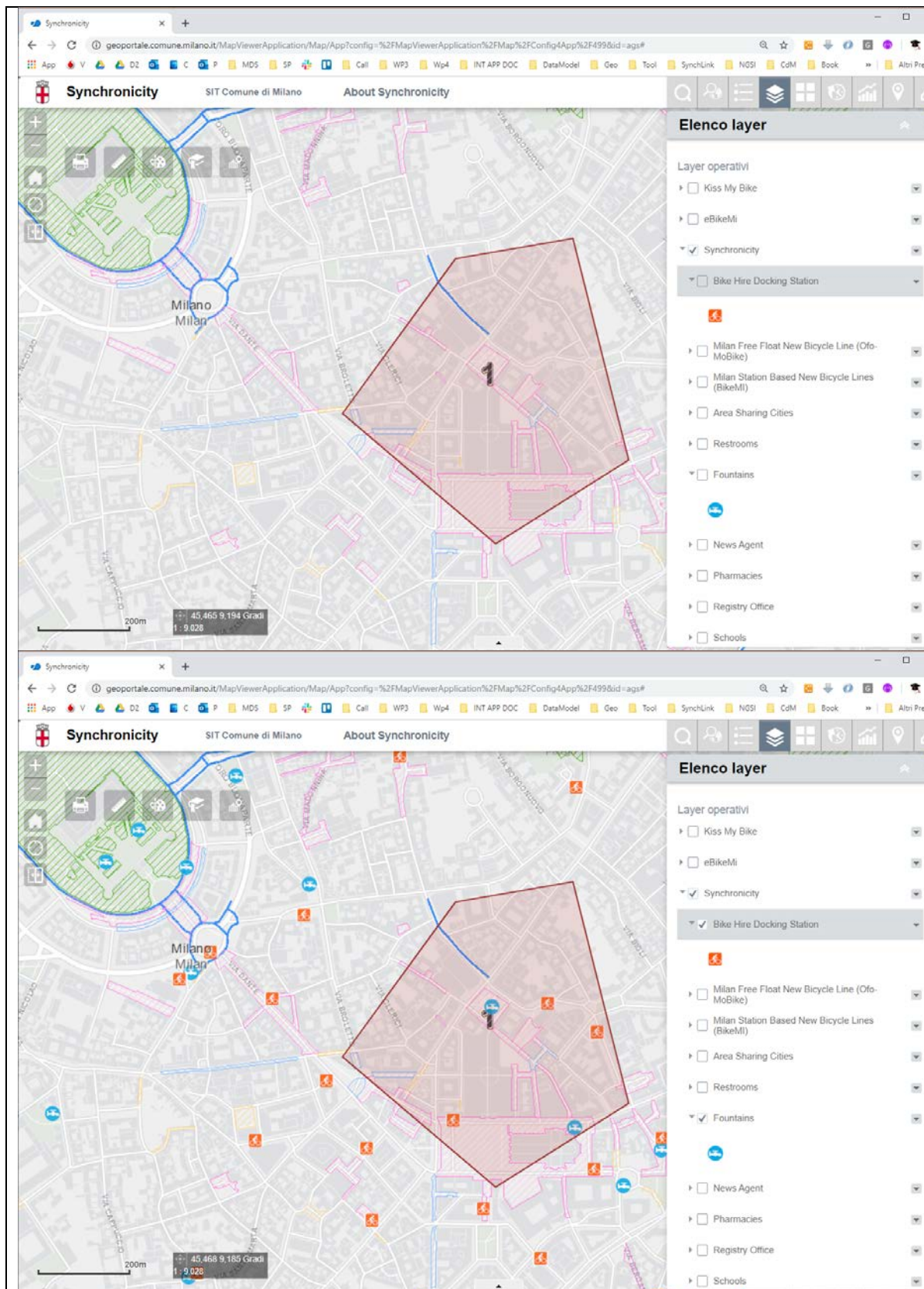
UC. 3: Searching address.

The User, by clicking an icon, can activate the searching address pop-up. While inserting the address the application shows to the User all possible matches. After the User has chosen an address the map is centered on that point and it is shown on the map (e.g. via Broletto, 3).



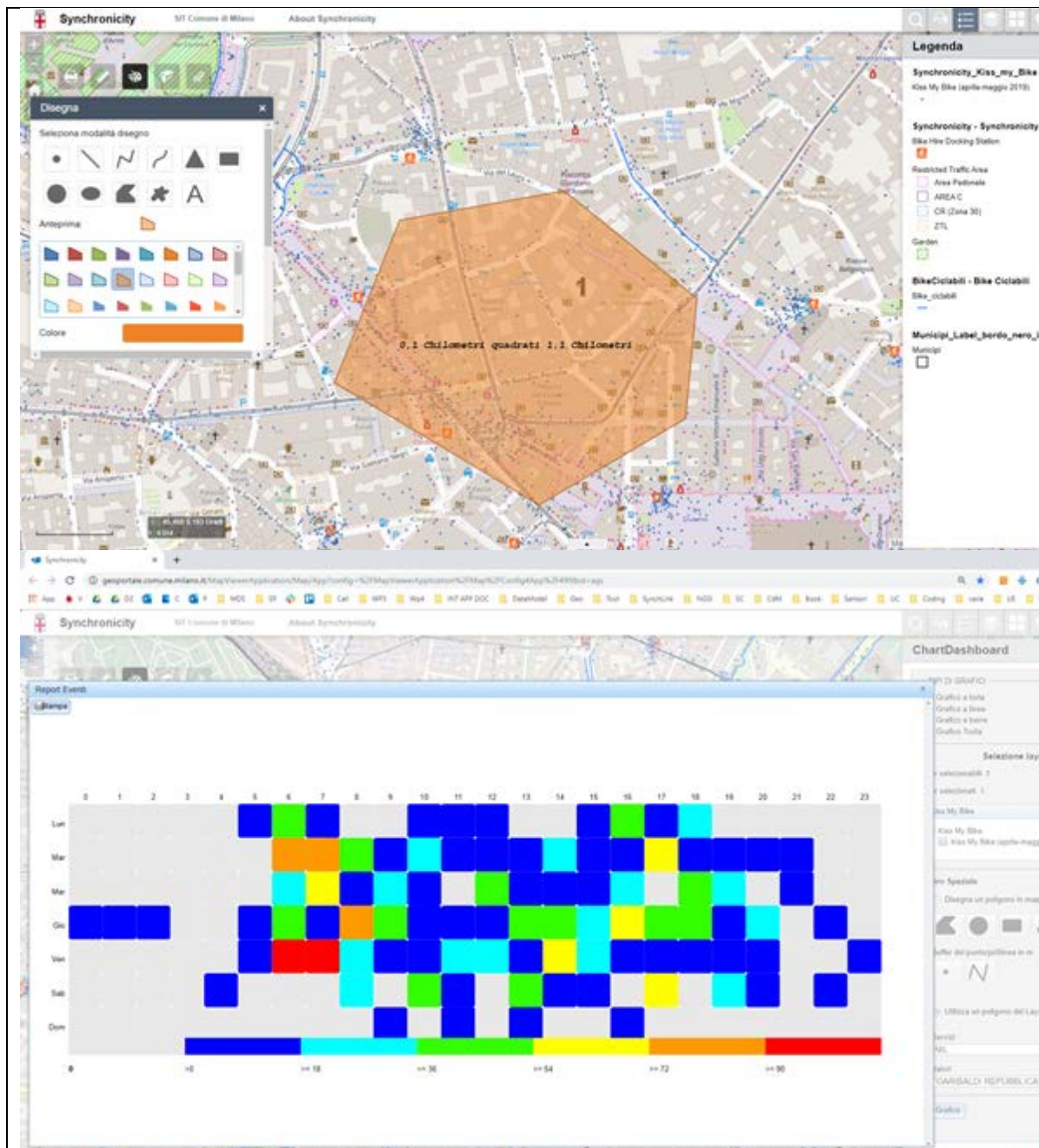
UC. 4: Toggling layers.

On the right side of the screen the User can have access to all information layers available, so he/she can decide which ones to toggle. In (a) the User toggles off all information layers except for bicycle lanes. In (b) the User toggles on the bike-sharing docking stations and the drinking fountains.



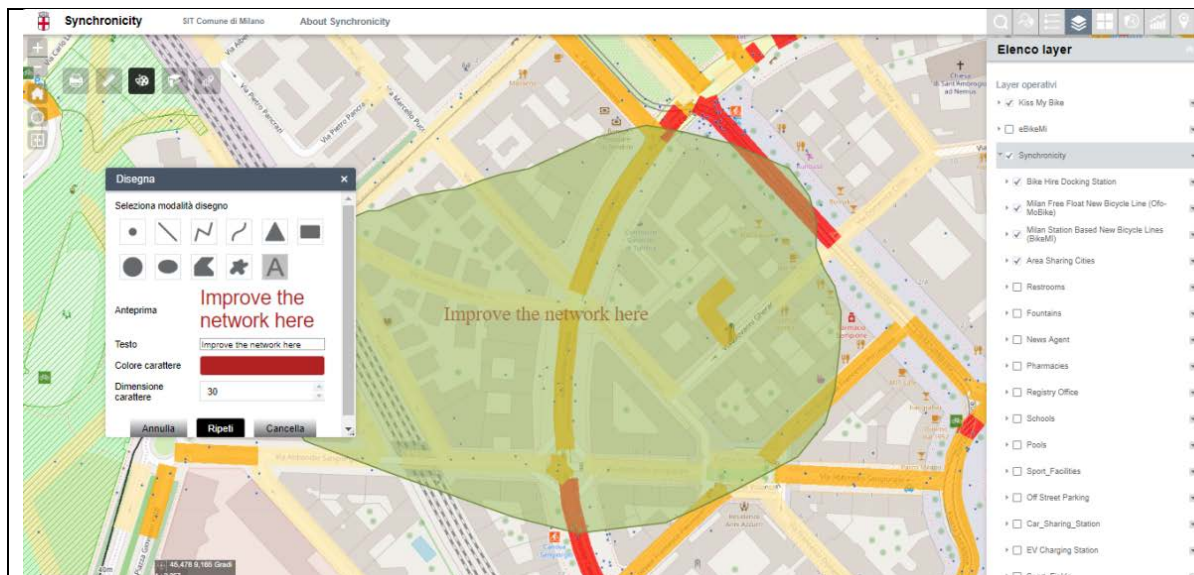
UC. 5: Layers interactions.

The User can select different information layers and select an area on the map (a). With the intersection of the data in the selected area the user can decide to perform statistical analysis such as showing bike usage data graph per day and day-time in the area.



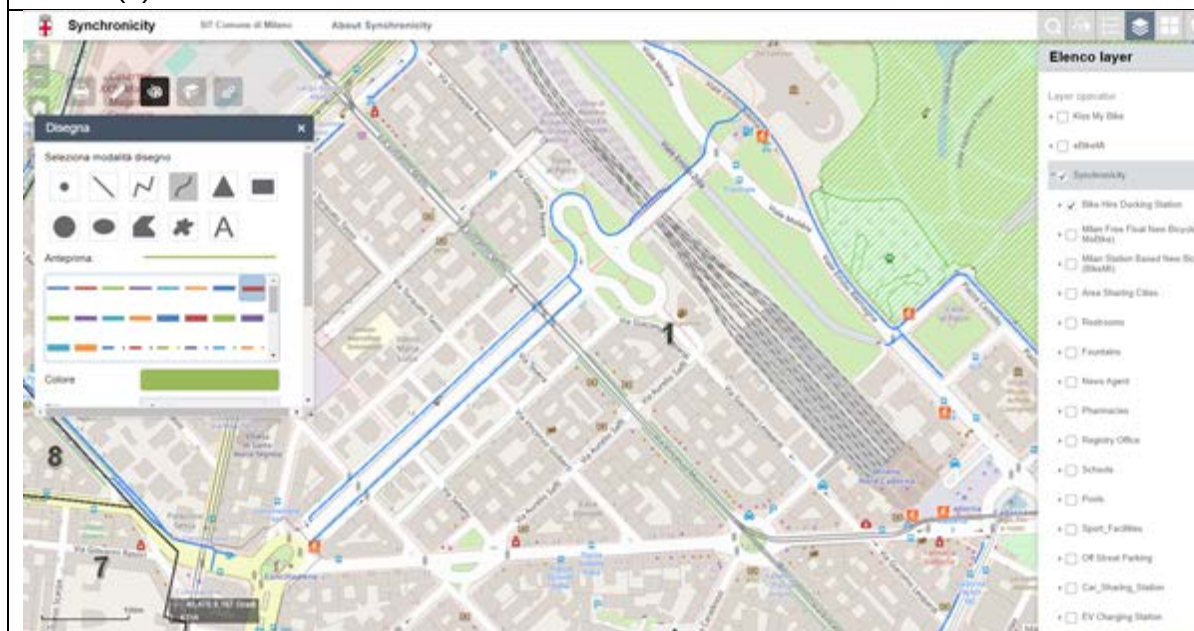
UC.6: Recording notes.

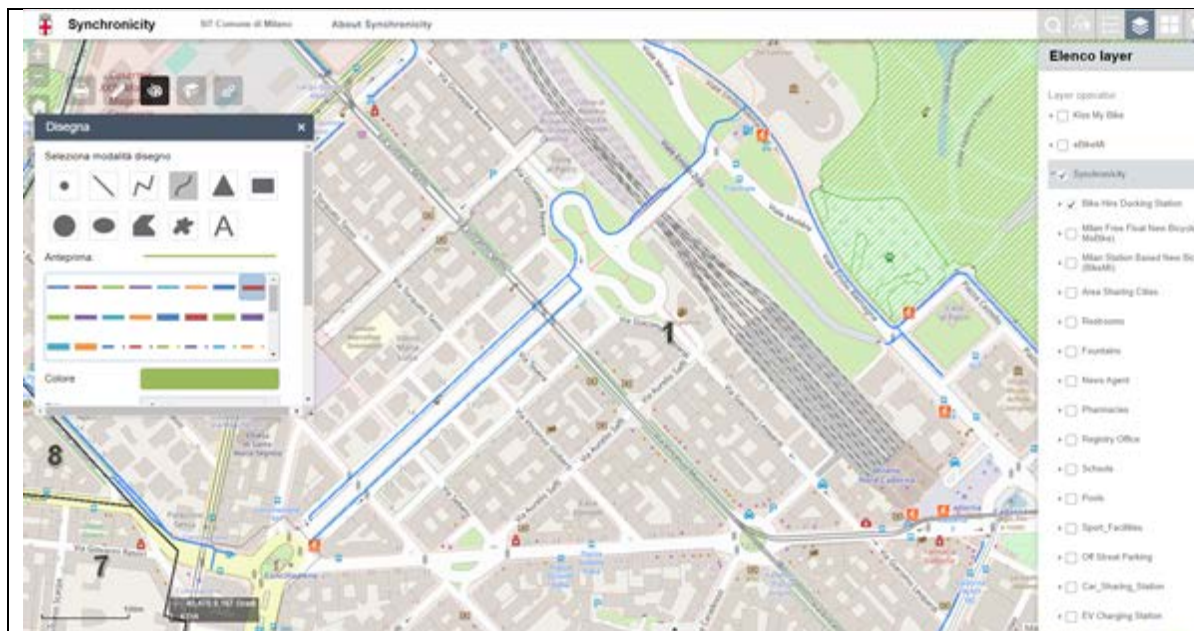
By clicking an icon, the User has the chance to activate the pop-up to insert notes that will be shown directly on the map.



UC. 7: Planning new bicycle routes.

The User can activate the drawing pop-up to draw and design a new bike lanes (a). Once the User has drawn a new bike lanes (dashed red), the total distance of the new lane has shown (b).

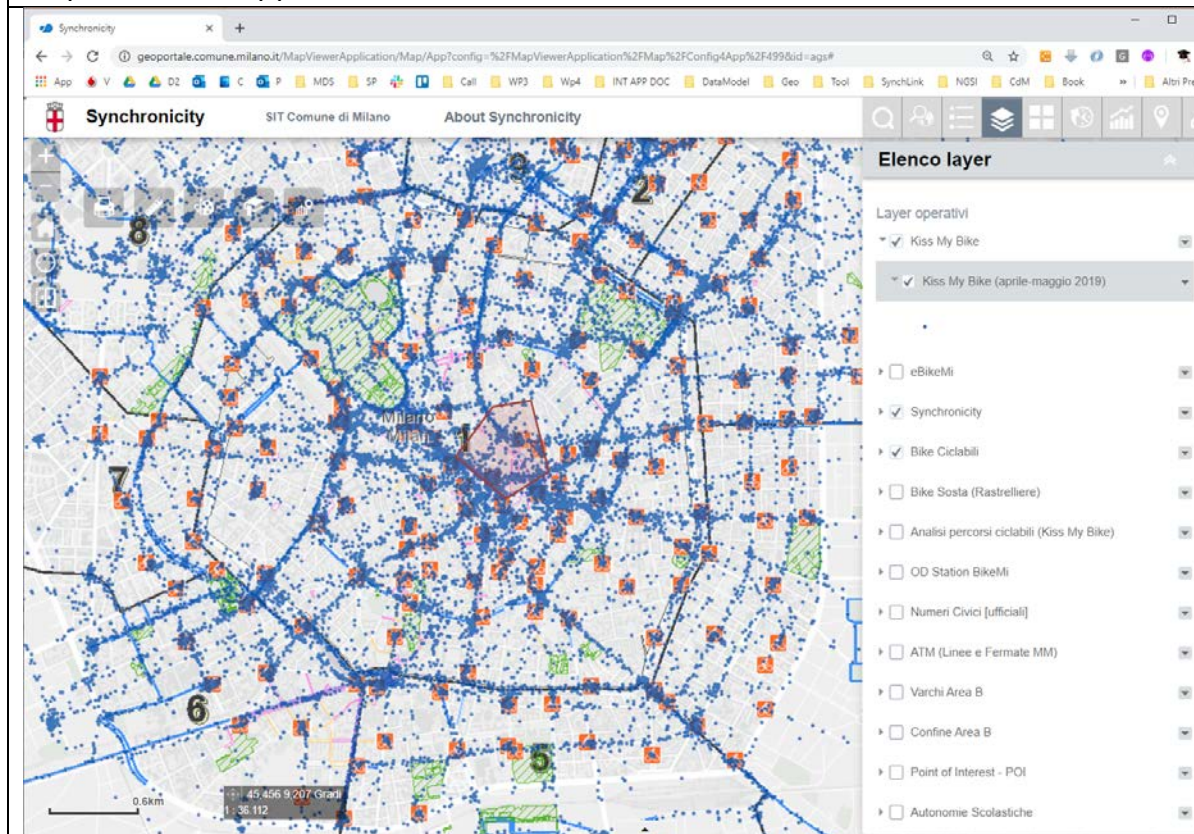




NEW UC. 8: Analysing bike usage data from Open call sensors (KissMyByke).

The user can activate the information layer containing the GPS position of bike-sharing bikes equipped with KissMyBike sensor (a – blue dots).

With this KissMyBike data it is possible to perform various analysis and visualize the results. One of the most significant analysis highlights with color a color-map the most used road-segments (b). Another significant analysis shows with a color-map the most frequented areas (c).



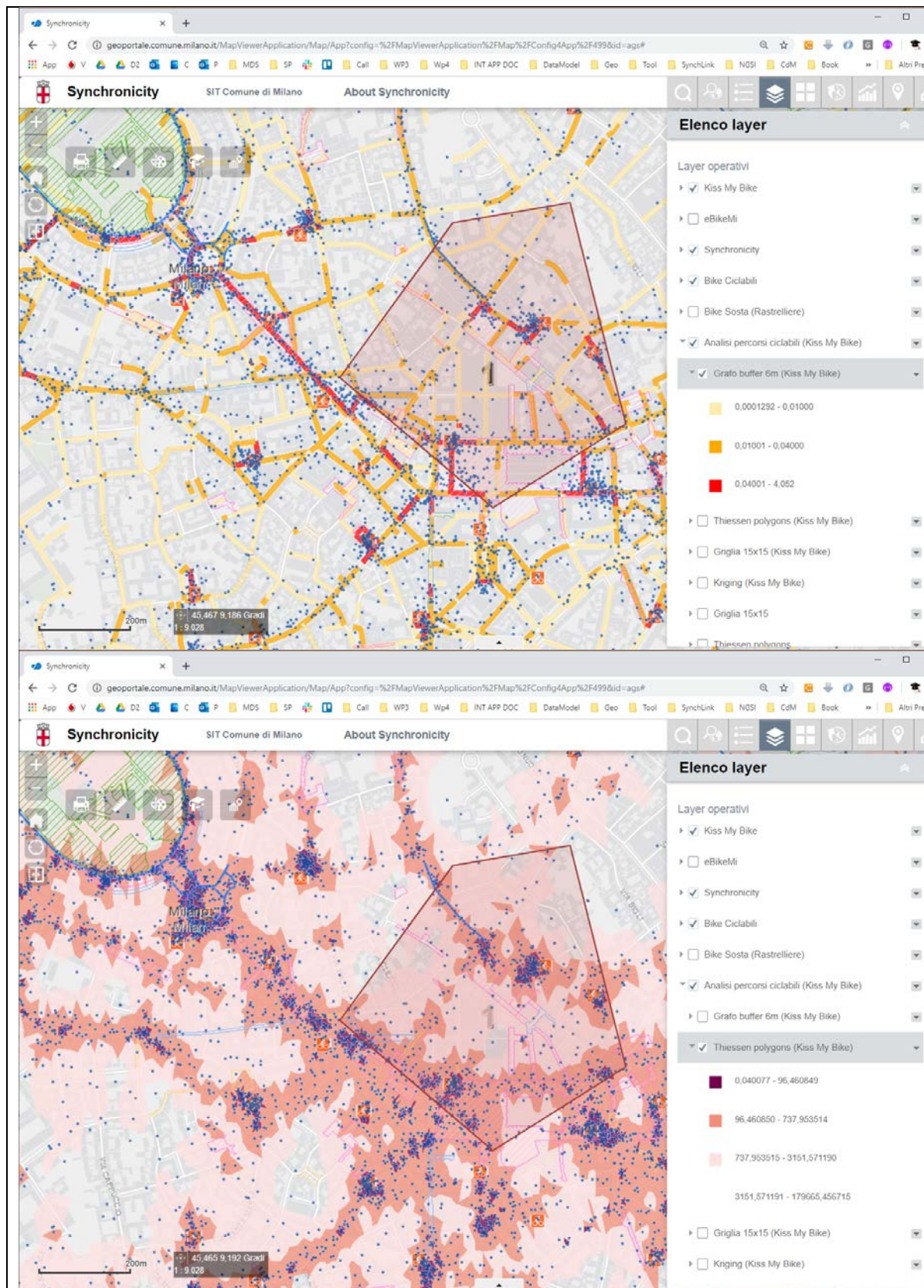


Table 5. Use cases of Milan Decision support system for cycle path planning application

As reported on the beginning of the chapter, by means of the new the KissMyBike GPS position data a new Use Case (n.8) has been developed. Its description is reported in the following Table 5.

Scenario title	Sc.8 – Data analysis
User profile	Planner
Background	David launches the application from his browser. The application shows a map of the area of the municipality in which David is interested. David wants to see the GPS position of the bikes equipped with sensors from KissMyBike pilot and he also wants to perform some analysis of the data.
Objective	David wants to see information related to the territory
Storyline	David presses a button on the screen with the mouse that shows the available information layers for the shown territory. David selects the information layer related to KissMyBike GPS data to show the hitmap of the recorded position of bikes. Then David wants to perform some analysis of the data and thus select the information layer that allows to select different kinds of analysis on data: heatmap, buffer-graph, Thyssen polygons, Kringing analysis. David can decide to show the different analysis one by one or overlap them.

Table 6. Use Case 8 - Data analysis of Milan Multimodal-navigator for disabled people application

4.1.2 Test results of pilot

During the software development of the pilot an issue has been reported with the routing service used to create path from Origin-Destination bike-sharing usage data, since it's not possible to communicate to the routing service that existing cycle network has priority with respect to the normal road network.

4.1.3 Implementation architecture and used Atomic Services

This section describes the final architecture of the "Decision support system for cycle path planning" application (Figure 8); the application leverages SynchroniCity APIs exposed by ICT infrastructure implemented in Milan (described in section 4.1.3.1). "Decision support system for cycle path planning" application makes use of the Routing (Atomic) Service to calculate new potential cycle paths within the city. The architecture does not differ from the one reported and described in D3.6; for more details about it and its components please refer to the mentioned deliverable.

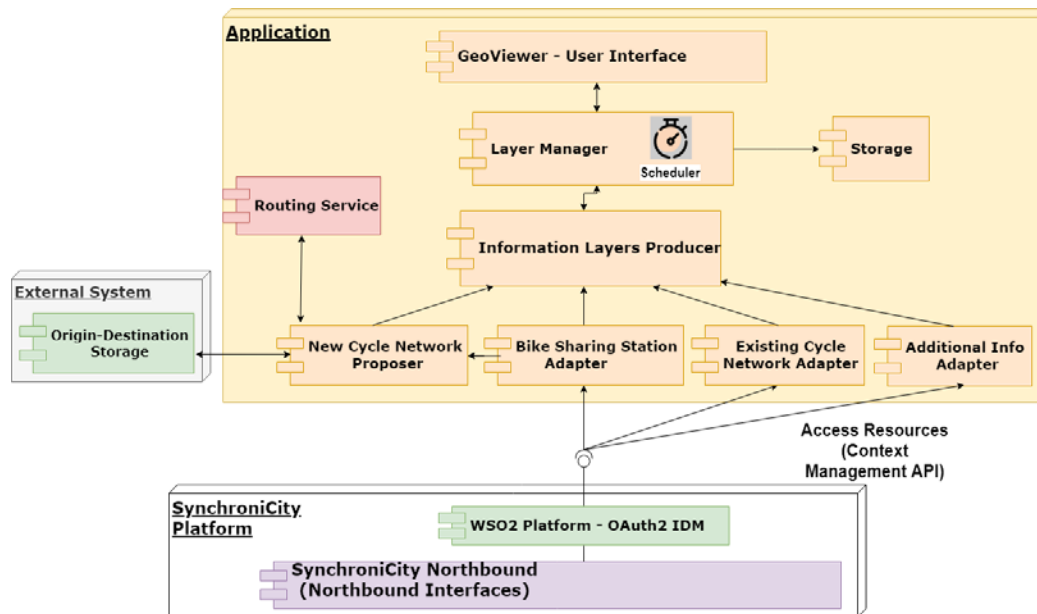


Figure 8. Milan - Decision support system for cycle path planning application architecture

4.1.3.1 Implemented infrastructure

"Decision support system for cycle path planning" and "Multimodal-navigator for disabled people" applications rely on an ICT infrastructure compliant with SynchroniCity specifications. The central component of the infrastructure is represented by a dedicated instance of the Context Broker. The ICT infrastructure is deployed as a fully dedicated VM instance running on a cloud environment provided by Engineering Ingegneria Informatica SpA (one of the partners of SynchroniCity project), and it has been integrated with the legacy WSO2 platform of the Municipality of Milan. An overview of the infrastructure is depicted in Figure 8.

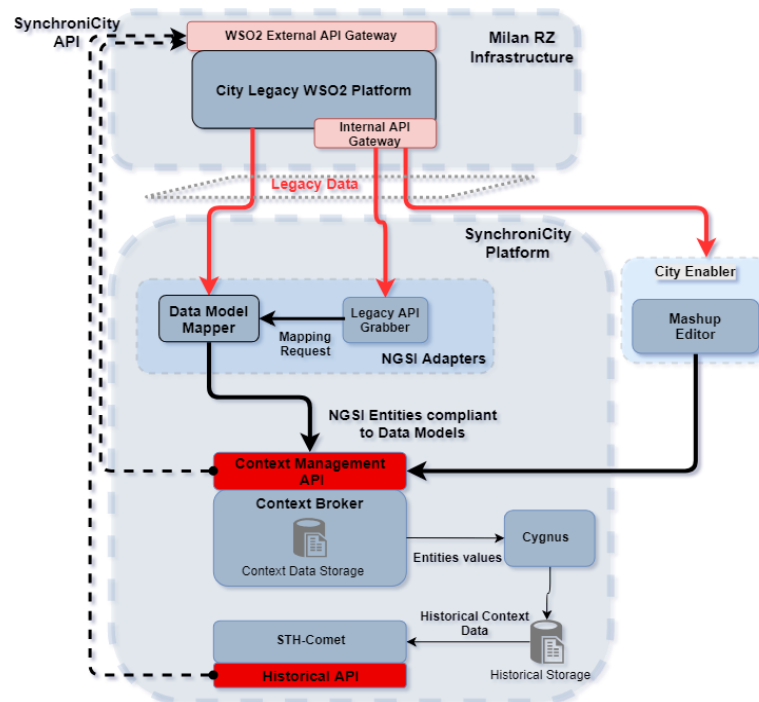


Figure 9. Milan ICT Infrastructure

The connection between Milan's legacy platform and SynchroniCity components is represented by the NGSI Adapters components: *Data Model Mapper*, *Legacy API Grabber* and *Data Mashup Editor* from City Digital Enabler. These translate Milan's legacy data into NGSI entities compliant to SynchroniCity Data Models.

Resulting NGSI entities are managed in the Context Broker instance, whereas data history (for a subset of NGSI entities) is performed through a subscription mechanism provided by the Context Broker and Cygnus Data Connector, in combination with STH-Comet.

Finally, Context Management APIs (provided by the Context Broker) and Historical APIs (provided by STH-Comet) are accessible by the Milan WSO2 API Manager (API Store), that is secured and publicly re-exposed with endpoints held by the Municipality of Milan. More detailed information about implemented ICT infrastructure in Milan are provided in Deliverable D2.9.

To feed the "Decision support system for cycle path planning" application, Milan's ICT infrastructure manages different data; a summary description is provided in Table 3.

Title	Description
Cycle network	GeoJson description of current cycle-network in the city
Bike sharing Stations	Position and description of the stations where to pick bikes up
Parks	GeoJson description of parks in the city

Restricted traffic and pedestrian areas	GeoJson description of traffic restricted areas and pedestrian only
Point of interests	Descriptions and location of valuables 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.
Tramway and Railway network	GeoJson description of tramways and railways network
Bike trajectories	Historical GPS position of free-floating bikes trajectories.
Bike Origin-destination	Historical data about the station-based bike sharing usage
KissMyBike Historical Data	Data about historical usage of KissMyBike pilot bicycle data. This data is coming from a Synchronicity Open Call winning solution.
ATM Tram Lines	Produced from UrbanMobility entities

Table 7. Data sources used in Milan Decision support system for cycle path planning application

4.1.3.2 Changes and deviations of planned features of pilot

During the implementation of the application new data were recognized as reported before from Kiss My Bike project. Differently from all other data used in the application that were mostly static since coming from OpenData, or definitive reports as Origin-Destination data, KissMyBike data were gps-real-time data coming from sensors. This gave the opportunity to monitor and analyse the real position of bikes and perform advanced analysis on them (UC 8).

4.1.4 Involved ecosystem partners and their roles

The partners involved in the execution of this pilot have been the ones in the consortium located in the city of Milan, assuming different roles:

● Municipality of Milan

- o **D-SIAD – Interoperability Department:** has been in charge to coordinate the Synchronicity project for Milan RZ, gathering data and defining the scenarios and use cases.
- o **SIT – Geographic Department:** a big support has given by SIT (Sistema Informativo Territoriale), which is managing the geoportal, in developing the user interface.
- o **AMAT –** Mobility agency of the Municipality of Milan, intended as final user of the application.

- **ENG** has been in charge to support Milan RZ developing the back-bone of the application and converting data from the Synchronicity context broker into informative layers to be ingested by the geoportal.

4.1.4.1 Required supportive actions and solved issues

Gathering data has been a real tough activity. The application scenarios had to be redefined several times in order to match with the actual available data. For instance, even though it was certain to have accident data available for the application, at the very last moment, it has been discovered that the data were quite old and not very precise because of various GDPR issues raised by the local police. It was good anyway that the application definition was robust to change and ready to be integrated with new extra data after its definition as in the case of KissMyBike.

4.1.5 Internal and external dependencies

- Internal dependencies for this pilot are the following
 - o WP2 for Synchronicity architecture
 - o WP3 for the defined scenarios and use cases
 - o WP4 for the data validation
 - o WP5 for detecting an Open Call solution according to the pilot's goals
- External dependencies
 - o KissMyBike for bike's GPS data
 - o Bike Sharing service for Origin-Destination data

4.1.6 Deployment conclusions and lessons learnt

Although gathering data has been quite hard, the application was developed on track since the good collaboration between the Engineering team, that provided all the back-end activities, and the SIT team that developed the user interface upon the features provided by the geoportal. The experience of the SIT department was precious in order to define the user functionalities and to define the standards by which Engineering succeeded in implementing the connectors between the Synchronicity platform and the geoportal.

The good implementing architecture demonstrate that the application can be integrated with unexpected data such as in the KissMyBike case reported before, even though a non trivial effort has been requested to define the better way to analyze and show the data to the final user on the application. Experts from the SIT on the field of data visualization has been fundamental.

The good success collected by the application between all the stakeholders to whom it was been presented to, generated requests of further data integration and improvements that go

beyond the original scope of the application, and this action will be planned in the next few months. In this case it is very important to achieve the collaboration of experts.

4.1.7 Sustainability plan upon project completion and future improvements

After showing the draft of the application to the various departments inside of the Municipality there is a growing interest around the application and its future development. In particular the application has been presented to the alderman of the Digital Transformation and Public Services and to the alderman of Mobility that both appreciated the tool beyond the specific purpose to improve cyclability and then asked to integrate the application with all the data coming from the other sharing services in the city.

Under this request the Municipality is trying to involve the car-sharing providers to share their data. Moreover, the Municipality changed the public tender to select new scooter-sharing service providers, that will start to operate in the next months, in order to request them GPS data of their scooters as part of the mandatory clauses of the authorization.

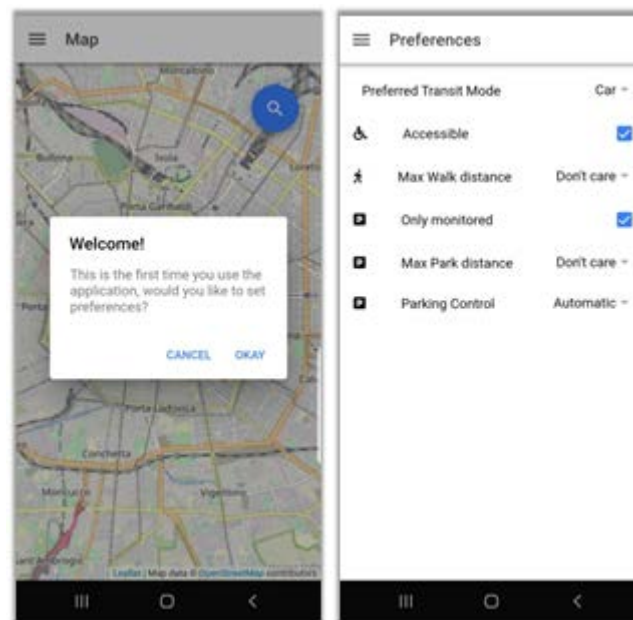
4.2 Multimodal Transportation Application "Multimodal-navigator for disabled people"

4.2.1 Pilot motivations and deployed use cases

The aim of the "*Multimodal-navigator for disabled people*" application is to facilitate the mobility of disabled people across the public services available in Milan. In details, Municipality of Milan allows disabled people to traverse the restricted traffic areas with their vehicles and to use specific road segments, such as bus lanes, that are prohibited to non-disabled people; indeed, currently available navigators usually do not provide such kind of features for disabled people. In addition, disabled people could be interested to use both their vehicle and the public transport system, and they need information about the closest available parking lot and accessible public transport.

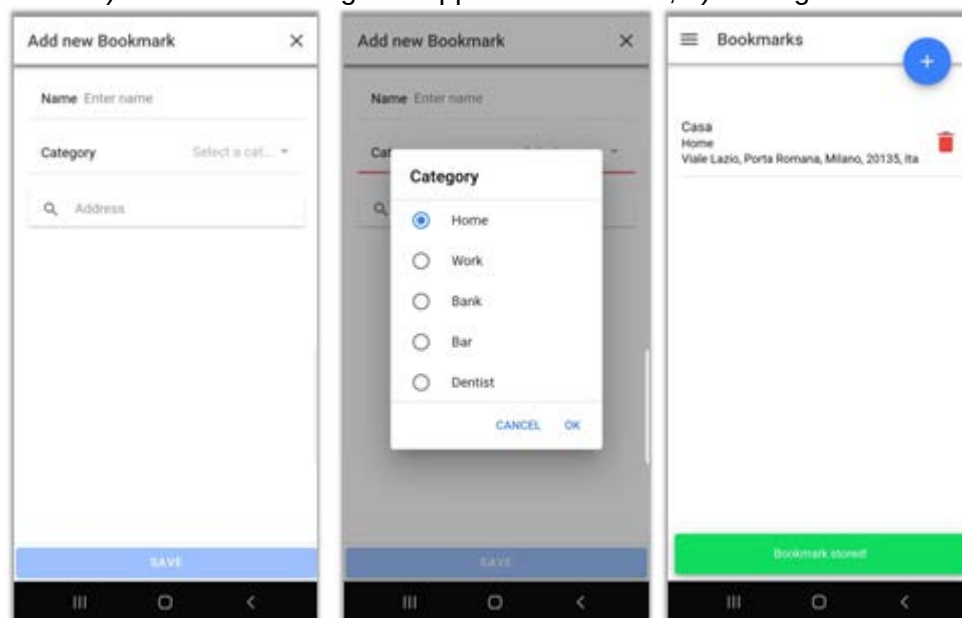
Under this goal the following use cases (or scenarios) have been developed; their complete description is reported in D3.6. A summary of developed use cases is reported in Table 4. For each use case screenshots of the application depicting the implemented functionalities are reported.

UC.1:	Saving	Preferences
-------	--------	-------------



Picture 3: Milan-MMT UC.1 - Saving Preferences

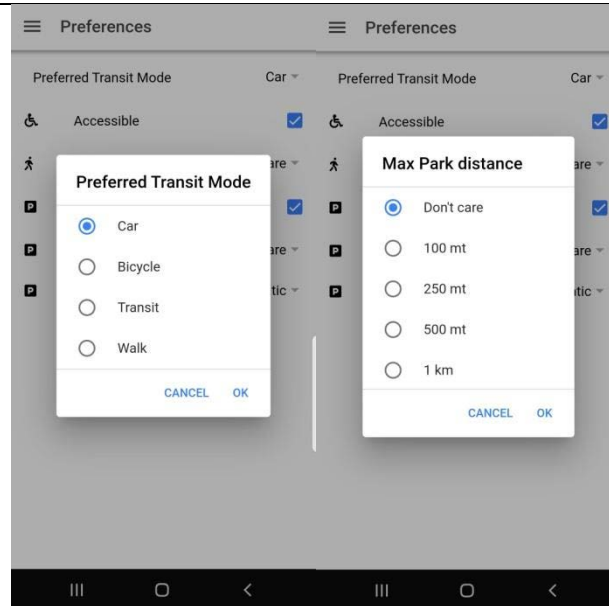
Picture 3 - a) Welcome message on application first run; b) Settings the Preferences



Picture 4: Milan-MMT UC.1 – Adding new Bookmark.

Picture 4 - a) New Bookmark Form; b) Selecting a category for the Bookmark; c) Visualization of the new stored Bookmark.

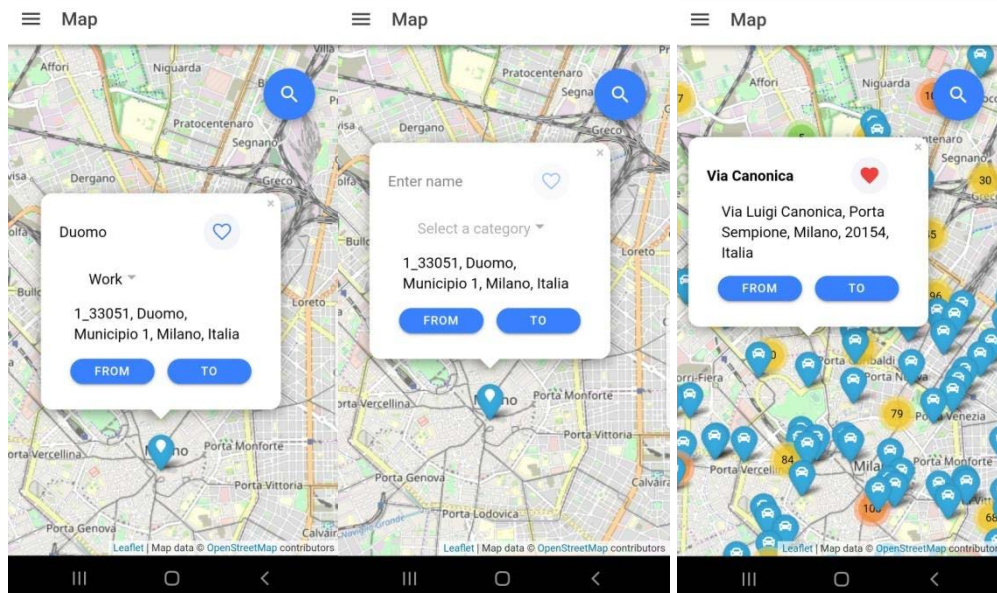
UC.2: Changing preferences



Picture 5: Milan-MMT UC.2 – Changing Preferences.

Picture 5 - a) Selecting the Preferred Transit Mode; b) Selecting the Maximum park distance.

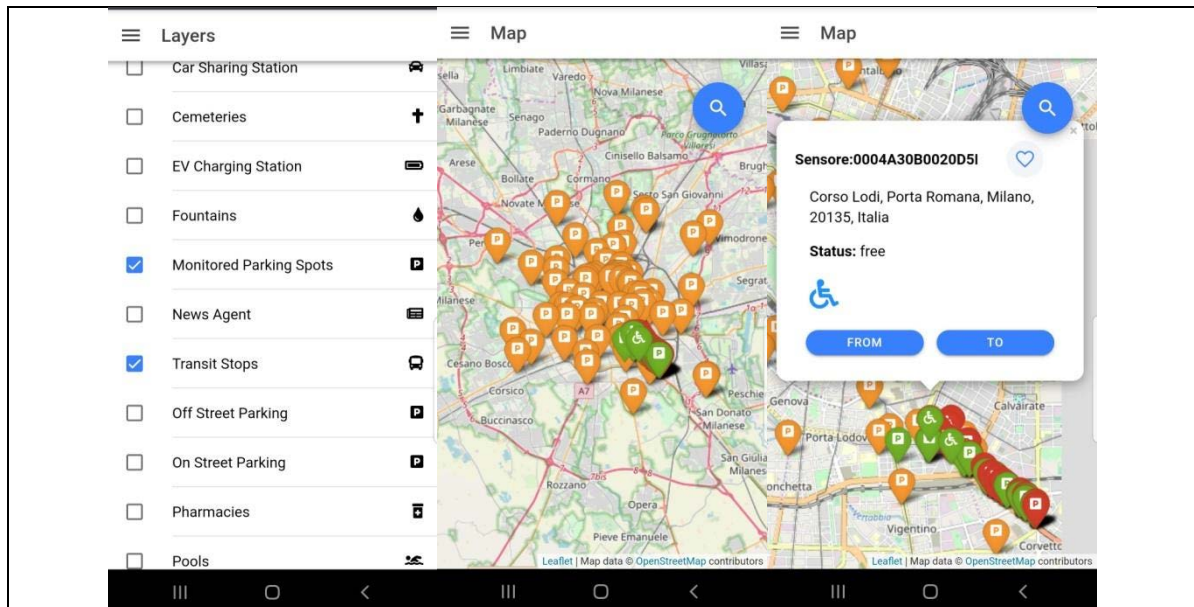
UC.3: Adding bookmark from the map



Picture 6: Milan-MMT UC.3 – Adding Bookmark from Map.

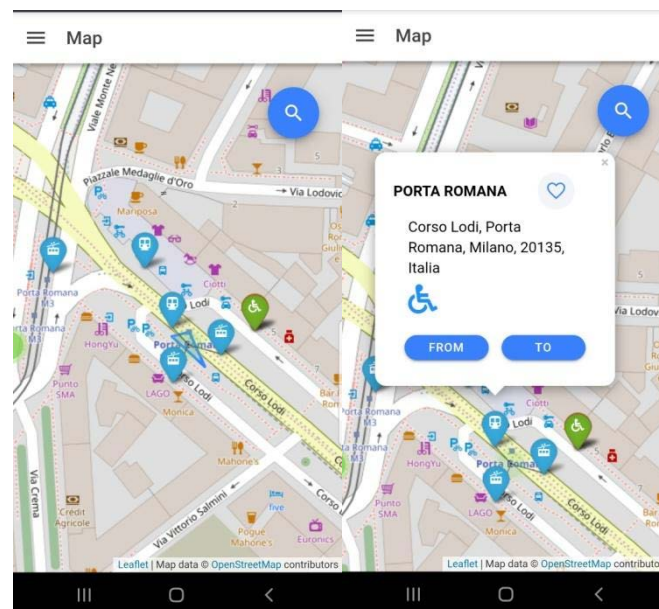
Picture 6 - a) By clicking on a point in the map a popup appears, and the user can fill the name and the category of the new Bookmark. b) The user fills the form of the new Bookmark, by clicking on the Heart icon, the bookmark will be stored; c) The user open check the information of an existing point and save it as Bookmark

UC.4: finding free parking



Picture 7: Milan-MMT UC.4 – Finding Free Parking (I)

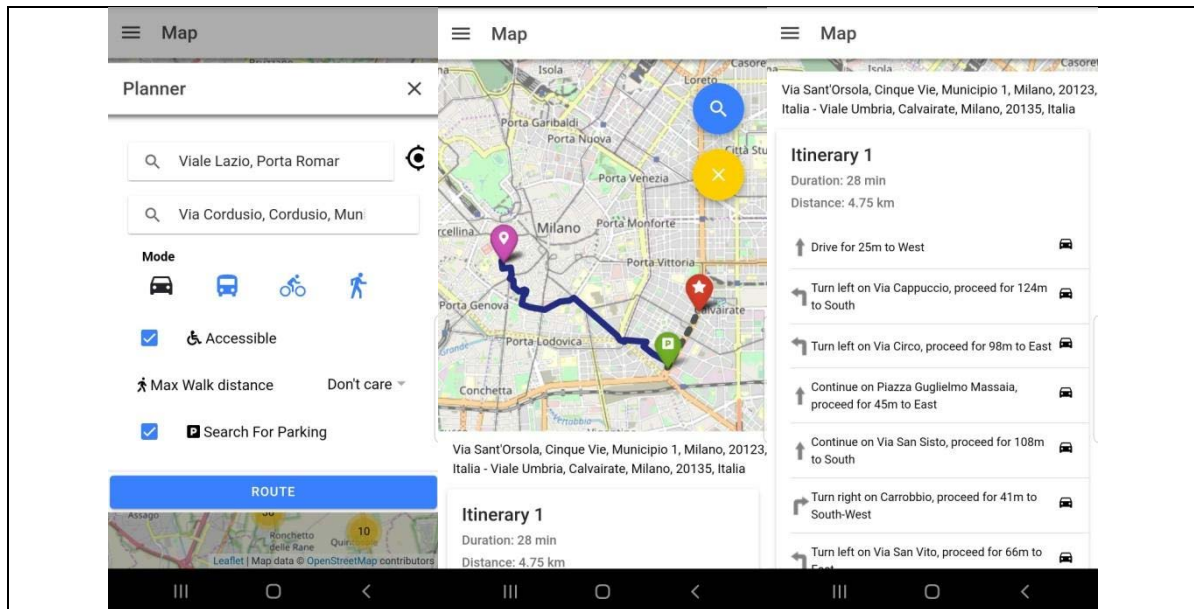
Picture 7 - a) The user selects the parking layers from the “Layer” section of the application; b) Parking layers are displayed on map; c) The user click on a parking marker and a popup appears with information about its status and its accessibility for disabled people, if any.



Picture 8: Milan-MMT UC.4 – Finding Free Parking (II)

Picture 8 - a) Information about transit stops are displayed on the map; b) The user selects a transit stop and its detail appears with information about the accessibility for disabled people, if any.

UC.5: Calculating routes plan between two stations



Picture 9: Milan-MMT UC.5 – Calculating routes plan between two stations

Picture 9 - a) The user fills the form selecting a starting point, the destination, the transit mode, the accessibility and if he/she needs to park; b) The application calculates the route suggesting a parking spot and show the result in the map; c) The application shows the details of the suggested path.

Table 8. Use cases of Milan Multimodal-navigator for disabled people application

The idea of the application has been initially presented the Mobility Agency (AMAT) and also to some local disabled people association in order to ask them contribution to better define requirements. Since there have been some delays in the development of the software due to the lack of collaboration of some data providers, the final application has been presented to AMAT at the very last moment. AMAT found it quite interesting and it is planning to integrate the application with some new data coming from further parking sensors that are about to be installed in Milan in 2020.

Unfortunately, the application is still at an early stage because there are only a few parking sensors concentrated in a relatively small area of the city and the actual map of reserved lanes that disable people can run across is not complete and updated, so the MMT application has been presented to the disabled people association as a proof of concept (POC) of the possibility. Thus, the application collected a good interest, but it is too early to be presented to people.

4.2.3 Implementation architecture and used Atomic Services

This section describes the final architecture of the "Multimodal-navigator for disabled people" application; the application leverages Synchronicity APIs exposed by ICT infrastructure implemented in Milan.

The final architecture of "Multimodal-navigator for disabled people" (Figure 9) includes some updates; in particular, the Parking Estimator Atomic Service has been removed since it has been decided to improve the main functionalities of the application. Its adoption will be investigated more in detail in a second phase.

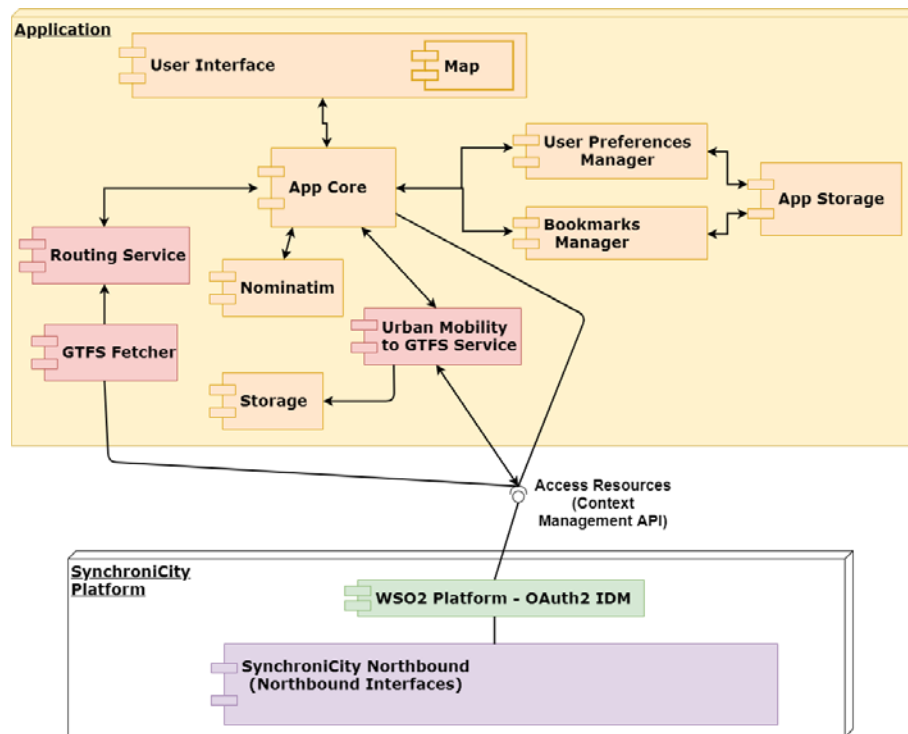


Figure 10. Milan - Multimodal-navigator for disabled people application architecture

The application retrieves data from SynchroniCity APIs; access to those APIs has been secured by the IdM module/component provided by the WSO2 platform already available in Milan Municipality which is compliant with OAuth2 standard according to the SynchroniCity Security layer specification explained in D2.10 [6].

Atomic Services used in "Multimodal-navigator for disabled people application" are:

- **Routing Service:** provides suitable routes, combining several transport information such as tram, train and bus lines.
- **Urban Mobility to GTFS Service:** 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. Generated GTFS Feed archive is stored in a dedicated Storage.
- **GTFS Fetcher:** imports GTFS files (static timetables, route shapes and related info) into the Routing Service, making this available for routing calculation.

Other components of the application are:

- **User Interface:** this component provides the users with functionalities to interact with the application. For instance, it allows the user to visualize and navigate a map, to look

for Point of Interests, parking, public transport stations and access Points, to request and visualize routes, etc.

- **App Core:** this component is the logical controller of the application; it manages all the requests coming from the users and coordinates the interactions with the other components of the application.
- **Nominatim:** a tool to support user searching OpenStreetMap data by name and address.
- **User Preferences Manager:** this component manages the user's preferences (e.g.: saving, retrieving, updating and deleting user preferences) as requested by the App Core on the base of user's actions.
- **Bookmarks Manager:** this component manages the bookmarks of the user (e.g.: saving, retrieving, updating and deleting bookmarks) as requested by the App Core on the base of user's actions.
- **App Internal Storage:** this component provides storage functionalities to manage both *user preferences* and *bookmarks*.

4.2.3.1 Implemented infrastructure

The ICT infrastructure implemented in Milan is the same of the "Decision support system for cycle path planning" application. This ICT infrastructure represents the foundation on top of which Milan's application are implemented. The ICT infrastructure is described in section [3.2.3](#).

To feed the "Multimodal-navigator for disabled people" application, Milan's ICT infrastructure manages different data; a summary description is provided in Table 8.

Title	Description
Point of interests and touristic itineraries	Descriptions of valuables places in an area and the suggested tourist itineraries.
Traffic Restricted areas	GeoJson description of traffic restricted areas.
Public transport metro network	GeoJson description of metro network and stations.
Public transport terrestrial network	GeoJson description of terrestrial bus/tram network and relative stops.
Car sharing parking areas	GeoJson description of car-sharing parking areas, with parking lots.
Parking areas	GeoJson description of parking areas.
Parks and gardens	GeoJson description of public parks and gardens.
City facilities	GeoJson description of facilities such as pharmacies, schools, congress centres.

Parking lots status	Information about the status of parking lots, such as availability.
Electric car charging stations	Information about the location of electric cars charging stations
Weather Stations and weather measurements	Information about weather stations and real-time measurements for each station.

Table 9. Data sources used in Milan Multimodal-navigator for disabled people application

4.2.3.2 Changes and deviations of planned features of pilot

During the developing phase some of the data providers in the E015 denied the usage of data from local transport service, such as time-schedule and status of bus and metro lines, and real time position of buses.

4.2.4 Involved ecosystem partners and their roles

The partners involved in the execution of this pilot have been the ones in the consortium located in the city of Milan, assuming different roles:

- **Municipality of Milan**
 - **D-SIAD – Interoperability Department:** has been in charge to coordinate the Synchronicity project for Milan RZ, gathering data and defining the scenarios and use cases
- **ENG** has been in charge to support Milan RZ developing the back-bone and the front end of the application and converting data from the Synchronicity context broker into informative layers to be used in the application;
- **Sharing Cities H2020 EU project**²⁵ – The municipality of Milan is involved in this H2020 project as well, and one of the results was to deliver parking sensors in reserved spot for disabled people in the city which can be used internally to develop this pilot application;
- **E015** – The E015 ecosystem²⁶ is a platform that gathers various local data providers. The role of the E015 eco-system is to deliver request to use data to the data provider under the presentation of a POC. Each data provider can decide autonomously if granting the access to the data or not.

4.2.4.2 Required supportive actions and solved issues

The application scope has been internally presented and shown also to some local disabled people association. As reported in the HCTM application the routing service cannot be informed that some roads have a higher priority. In the MMT application this case is referred to the dedicated lanes for public transport system that in Milan can be used also by disabled

²⁵ <http://www.sharingcities.eu/>

²⁶ <https://www.e015.regione.lombardia.it/>

people with a specific authorization grant. In this case the issue is about to be overcome by creating a parallel road network integrating public transportation lanes information to be used by the routing service if the final user has an authorization for disabled people. Anyway, as reported before, in testing phase, it has been realized that the map of the dedicated lanes for public transport needs to be completed and updated.

4.2.5 Internal and external dependencies

- Internal dependencies for this pilot are the following
 - o WP2 for Synchronicity architecture
 - o WP3 for the defined scenarios and use cases
 - o WP4 for the data validation
- External dependencies
 - o E015 eco-systems;
 - o Sharing cities H2020 EU Project;
 - o Disabled people association;

4.2.6 Deployment conclusions and lessons learned

The application still needs some refinements before being tested and deployed. Technical delays have to be addressed to the difficulty of gathering all sets of data needed that coming from different providers can be granted independently one from the other. Surely, anyway, the biggest issue in a project like this is to overcome the wide variety of resistances coming from different stakeholders. One of the reasons of the resistance is due to the large number of different proposals that implies two different behaviour from the stakeholders. The stakeholders directly involved in the alternative proposal sometimes consider the other proposals as concurrent and tend not to collaborate. On the other hand, there are the other stakeholders, such as the disabled people associations, that are asked to contribute to every initiative and since most of the times the initiatives are not sustainable, the disabled people associations tend not to dissipate their resources and so they are hard to be convinced in being involved.

4.2.7 Sustainability plan upon project completion and future improvements

There is a great interest in the municipality around the mobility for disabled people, so there are a lot of different proposals that sometimes (as in the E015 case reported before) are pushing against concurrent applications. One of the greatest strengths of the application is having the access to sensors of reserved parking spots for disabled people that are not available for all the other concurrent application. For this reason, the Municipality is planning together with the AMAT Mobility Agency to use the MMT application as a baseline to be integrated with all the new parking sensors that AMAT is about to install around the city.

Furthermore, there is a discussion inside of AMAT about a complete mapping of road and public transport system accessibility in the city and the MMT application could be a valid tool to be exploited to show the result of the mapping.

5. Eindhoven

5.1 Pilot motivations and deployed use cases

Although Eindhoven has a quite bicycle friendly approach by expanding and improving cycle paths and encourages people to cycle there are still more opportunities to encourage people to use a bicycle instead of going by car. For the Synchronicity pilot project on data-driven bicycle mobility, Eindhoven aims to do just that: use a connected platform to integrate smart thermal cameras, intelligent traffic light controllers and a large roadside display in order to improve the bicycle experience, and give the municipality deeper insight into cycling patterns. An in-depth overview of use cases can be found in D3.5 and D3.6. Below we provide an overview of the implemented use cases:

- | | |
|--|--------------|
| • Basic 1: Bicycle counter; | Deployed |
| • Basic 2: Dashboard for professionals; | Deployed |
| • Advanced 1: Average waiting time information for cyclists; | Deployed |
| • Advanced 2: Waiting time information for cyclists, | Deployed |
| • Advanced 3: Speed advice for cyclists; | Deployed |
| • Advanced 4: Green wave for cyclists; | Not Deployed |
| • Advanced 5: Extend green time for group of cyclists; | Deployed |
| • Advanced 6: Provide insight to achieve municipal policy objectives | Deployed |

As can be seen, we did not deploy the Advanced 4 scenario. The out-of-project funding that was needed to install extra roadside displays was sufficient to acquire one, but not three units. In use cases advanced 2, 3 and 5, a slight change in scope and name was adopted, due to feedback on the user and stakeholder validation processes we have executed for the WP4 tasks.

In the picture below the example is given for advanced 3; Speed advice for cyclists. The top three rows will give other information depending on the active use case (advanced 1, 2 or 3). The fourth row always displays the number of cyclists of that day and the bottom row always shows the total number of cyclists past here since the start (basic 1).



Figure 11. Signco display on the streets of Eindhoven

5.2 Test results of pilot

For the evaluation of the use case we have followed the WP4 pilot validation tasks as reported in D4.3 [7]/D4.5 [8]. A significant part of these tasks consisted of user oriented feedback and validation;

- 4 co-creation workshops with key stakeholders and citizens were held (1 to identify challenges, 3 to iterate the final product), to capture detailed insights and get a deep understanding of user experiences and expectations;
- Expert meeting to create persona and stakeholder mapping to understand who are the key stakeholders, what value is added by the use case;
- User tests to test the user interface of the application;



Figure 12. An example of user interface and text validation by user workshop

Main outcome of the processes were fine-tuned end user scenarios for the on-site use case; this consisted of more informative messages to cyclists and a change of message content to be more actionable (e.g. the message to speed up / slowdown was perceived as better than information on number of cyclists passed). Finally, the end users again emphasized and reconfirmed the cycling priority and speed advice as main user scenarios to focus on.



Figure 13. Conclusions and insights from user interaction with Eindhoven's "Ring"

From the validation of the on-site deployment, the impact on the bicycle mobility is still too early to quantify, however we do feel there can be significant benefits in optimizing the bicycle flow for situations where a high volume of cyclists intersect with a slightly less congested car road, and where policy indicates that situational priority can be given to the bicycle in the traffic light phases and timing. So, a detailed analysis on suitable sites for further deployment that satisfy the preconditions should be advisable.

5.3 Implementation architecture and used Atomic Services

The used components for the Synchronicity use cases are the following

- Data ingress: Containerised python scripts
- Data Queuing: Apache Kafka²⁷
- Data Preprocessing and rerouting for storage: logstash²⁸
- Data storage: Elasticsearch²⁹
- Event and context data: Orion Context broker
- Dashboarding and visualizations: Kibana³⁰
- Communication: Physical dashboard in the city
- APIs were provided for data access

Scalability and Performance are key design points for the system, so it can be expanded when we foresee more users of "things" and/or streams of data scaling both horizontally and vertically. In horizontal scaling other nodes could be added, where copies of the software will run on, ideally in a dynamic fashion so that nodes are added automatically when the need

²⁷ <https://kafka.apache.org/>

²⁸ <https://www.elastic.co/guide/en/logstash/current/transformation.html>

²⁹ <https://www.elastic.co/blog/found-dive-into-elasticsearch-storage>

³⁰ <https://www.elastic.co/guide/en/kibana/current/dashboard.html>

arises. In vertical scaling the system can store more data or have more memory to perform advanced computing.

Therefore, we have introduced Big Data Pipelines with Real time Analysis in the speed layer, Batch analysis with the Batch Layer and Combine it to Context management to elevate Orion context broker to scalability properties.

Data Ingress to the platform

The first step into this building block is data ingress. Currently it is running under Python script but migration to Apache nifi is planned.

Component type	Why	What
Data Ingress	1. Automated and an easy to use, reliable system to process and distribute data. 2. Highly configurable, it has web-based UI, it shows data provenance, possible consummation on connectors, data transformation from raw to NGSI is possible. 3. scalable and secure (HTTPS, SSL, encryption, etc.)	Python script

Table 10. Component types

Message Broker

After the data have been ingested in the platform it needs to go to the dedicated service and storage. As we are in a context of multi-client multi-use case proposition, we need to ensure that data never get lost on the platform and it is being handled in real-time. Therefore, we need Message Broker and Stream processing on it to do light weighted aggregations.

Component type	Why	What
Real-time data pre-processing	The system should have data collection and processing engine that is scalable and high performance providing near real time processing (<500 ms) It comes should come with a wide range of plugins that makes it possible to easily configure it to collect, process and forward data to different components.	Logstash

Component type	Why	What
Data ingestion	1. Message Broker enables asynchronous communication, which means that the endpoints that are producing and consuming messages interact with the queue, not each other. Producers can add requests to the queue without waiting for them to be processed. Consumers process messages only when they are available. No component in the system is ever stalled waiting for another, optimizing data flow. 2. Message Broker makes your data persistent and reduce the errors that happen when different parts of your system go offline. By separating different components with message queues, you create more fault tolerance. If one part of the system is ever unreachable, the other can still continue to interact with the queue. 3. It allows simple decoupling and granular scalability	Python script

Table 11. Component type

Data Storage and search engine

According to our functional requirements the architecture should address the storage of data from both platform and usage perspectives. From a platform point of view, data could be stored into an on-premise system, in a cloud service or in a hybrid system depending on restriction of data privacy. The system should also consider the latency of the network. The hybrid solution where some data will be saved in locally owned systems and some data in cloud services could be also possible. Currently our strategy to store all data on premise system (Customer Dedicated Environment). From a usage perspective, applications and services will need to process data in various formats. For instance, structured data carries specific information that may fulfil the immediate needs of an application or a service, whereas raw data can embed information that may be used in the near future. Thus, the architecture should consider different data formats, providing storage support for both unstructured (e.g., raw data) and structured data and API to access historical data in a uniform manner

In order to guarantee that data access is performed in accordance with customer's their license, policies of distribution and/or charging, the system should support different data categories based on restriction of their usage such as public or open data, private data and commercial data.

Component type	Why	What
Real-time and Batch data storage	Near real-time scalable and reliable data storage that must support different data formats, providing storage support for both unstructured (e.g., raw data) and structured data and API to access historical data in a uniform manner. Historical data must also be considered and stored for later consultation especially by 3rd parties.	Elastic Search

Component type	Why	What
Real-time data visualization and monitoring	1. This component will allow our clients to visualize the data coming into the platform, define KPIs on them through a wide number of possible KPIs 2. It needs to be (near)Realtime tool so that the client can have a high decision-making velocity. 3. The tool need to allow us to embed part of the reports in custom made application we will offer the client such as web apps. 4. The tool must be scalable and highly available as it will be part of the user interface. 5. Monitoring of the platform need to be handled in this tool	Kibana

Table 12. Component type

Context Data management with Context Broker

In the platform we will be hosting the context information coming from IoT devices and other public and private data sources, providing a context data access through a uniform interface. Context information contains status information about real world entities defined in a structured way. Context Data Management provides functionalities to enable access.

Component type	Why	What
Context Broker	Manage the full lifecycle of context information providing the most recent data on different sensors in standard data models. It provides APIs in pub-sub mechanism for authorized users to collect this event data as soon as they are published.	Orion Context broker and its database Mongo-db

Table 13. Component type

Used atomic services

No atomic services are used in RZ Eindhoven

5.3.1 Implemented infrastructure

IoT Data combined with context policies is a fast growing domain. Synchronicity project is intended to analyze streams of data from different data sources such as cameras and traffic sign systems on a large scale. Therefore, we have a considerable velocity of incoming data. The urban ecosystem might communicate inconsistent data or incomplete due to latency, deception or service failure, veracity of the data is an element to think of. The data collected in these projects is rich and will be used to bring more insight to the users to help them generate value in their domain. In consequence, the platform where Synchronicity Eindhoven will run on is a Big Data platform fulfilling the 5V characteristic of Big Data context: Volume, Velocity, Variety, Veracity and Value. Therefore, beside the SynchroniCity Framework Context broker and its MongoDB, there are highly available and highly scalable components such as Elasticsearch and Kibana. Allowing us to provide an implementation of the synchronicity project as a Big Data IoT Platform powered by the SynchroniCity Framework.

In Document D2.10 detailed requirements for Synchronicity project can be found. We can map these requirements into logical building blocks of the conceptual architecture. This Architecture includes the following standards:

- Decoupling of Services and Data
- IoT Network Management
- Automated data ingestion automation
- Big Data Architecture augmented with Context data management
- Orchestration of the components for automated scaling and load balancing
- Cluster based deployment
- Monitoring of the cluster and Reporting for error handling and usage monitoring

Sensors/data providers:

- Three thermal cameras³¹
- Signco³² display (see Figure 10)
- Existing Intelligent Traffic light³³

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

³¹ www.flir.com

³² www.signco.be

³³ <https://dynniq.com>

Table 14. Eindhoven RZ.

Platforms

- Synchronicity Marketplace

Software components

- Orion
- Kafka
- Nifi
- Elasticsearch
- Logstash
- Kibana

5.3.2 Changes and deviations of planned features of pilot

Use Case advanced 4 could not be completed due to insufficient funds for additional hardware. To complete this use case all three crossings should have a display sign. Fortunately, the city was able to provide funds for one crossing, to complete the other use cases.

During the development of the applications an additional feature was implemented. The thermal cameras are used as a detection to virtually push the button at the traffic light to have a green light.

As can be seen, we did not deploy the Advanced 4 scenario. The out-of-project funding that was needed to install extra roadside displays was sufficient to acquire one, but not three units.

In use cases advanced 2, 3 and 5, a slight change in scope and name was adopted, due to feedback on the user and stakeholder validation processes we have executed for the WP4 tasks.

5.4 Involved ecosystem partners and their roles

SynchroniCity Partners:

- The **Municipality of Eindhoven** is actively engaged in smart city research and innovation activities, where the city platform and pilot applications are applicable. In the scope of SynchroniCity project, the city has been participating in the Human Centric Traffic Management theme.
- **ATOS** is a European IT services corporation specialized in hitech transactional services, unified communications, cloud, big data and cybersecurity services. Atos has

been fully involved in developing the urban data platform for the city and visualizing the data collected on the street for the use cases developed.

- **Heijmans** has been mostly involved in on site installation of hardware and the user validation process.

Suppliers:

- **Dynniq**: Application provider of the Intelligent Traffic Light Controller software
- **Technolution**: Provider of the networked traffic management system of the whole city of Eindhoven
- **Signco** has provided the display communicating with the end users (non-motorized traffic participants) and advising them how to adapt their speed in order to get priority when passing by the intersection;
- **Flir**: supplier of three thermal cameras in order to count the number of cyclists passing by;

5.5 Internal and external dependencies

Internal dependencies

- Available data from second and third thermal cameras due to delay in delivery.
- Availability of platform

External dependencies

- Data from existing traffic lights and traffic management system
- Necessary software changes to display sign.

5.6 Integrating with other external applications

The previous chapters have covered the Initial Applications use cases defined from D3.5 involving the different systems and requirements from Eindhoven Municipality and ATOS. When finalizing the definition of the RZ User's Requirements, it was noticed that some extra Use Cases would be included, to demonstrate the interoperability with other external commercial solution for SmartCities (such as ATOS Urban Data Platform).

These extra Use Cases does not focus on making interoperable different systems, or gathering data into the Synchronicity Platform. These will focus on using an already existing Synchronicity Platform (like the one developed for the sections above) to connect it to another existing solution. Connecting and visualizing data into another solution and demonstrating the straightforward integration process through the Synchronicity Framework and the Minimum Interoperability Points, that ATOS would replicate through other cities and clients. With these ideas in mind, D3.6 defined the following implemented Use Cases, as an extra demonstration derived from the original objectives for Eindhoven. Which have been implemented and covered above as the official main objectives for this RZ.

The main functionalities of ATOS Urban Data Platform (AUDP) 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

Next sections describe D3.6 Extra Use Cases

Counter for Cyclist

Using the platform's data about Cyclist, the information feeds is fed into the Atos Urban Data Platform (AUDP) for visualization. The counter of bicyclist from a visual perspective is shown as a heatmap, keeping in mind the values measure of last the previous week (i.e. historical context)., this means that is showing current data. On the heat map appears the date of the week and a filter where the user could select the quantity of counts to show. Using the Minimum Interoperability Point was a straightforward implementation task, only having to develop new views in the dashboard to support the user's role.

Camera info for visualization

This dashboard is intended to be used by professionals.

Data received from the same components (i.e. bicycle counter) that counts bicycle's users from the sensor at the bicycle station. We collect the data being filtered and normalized by the Synchronicity's Platform, but showing the data in a weekly heatmap. The data is shown in a cartesian calendar heatmap, where the professional user (Traffic Management Professional Role from D3.6) can check the quantity of bicyclists in a general way. So, the user would make a general idea of which stations are working properly.



Overview:

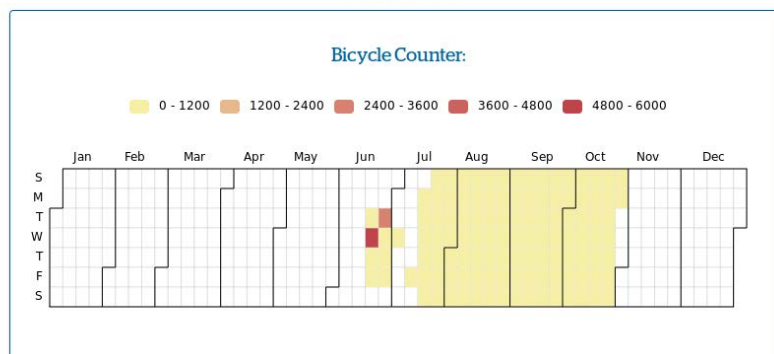


Figure 14. Screen caption of software

Clean bicycle routes

This dashboard is intended to be used by citizens

In this use case (see Figure 12), extra data sources (Eindhoven OpenPortal Portal³⁴) are used to be combined with the previous one provided by the Synchronicity Platform. In that way, data from different origins can feed into AUDP for more advanced use cases, (and interoperability demonstrations). Bicycle counters, routes and air quality are combined and unified into the platform. Air Quality Data is previously normalized, using the Context Management component, into the NGSI standard format.

Routes can be visualized together with Air Quality markers over a map. A user (with the Cyclist Role defined in D3.6) can approach to this page, select a single route and check the air quality of a specific area, for a better and cleaner route selection.

³⁴ <https://data.eindhoven.nl/pages/home/>

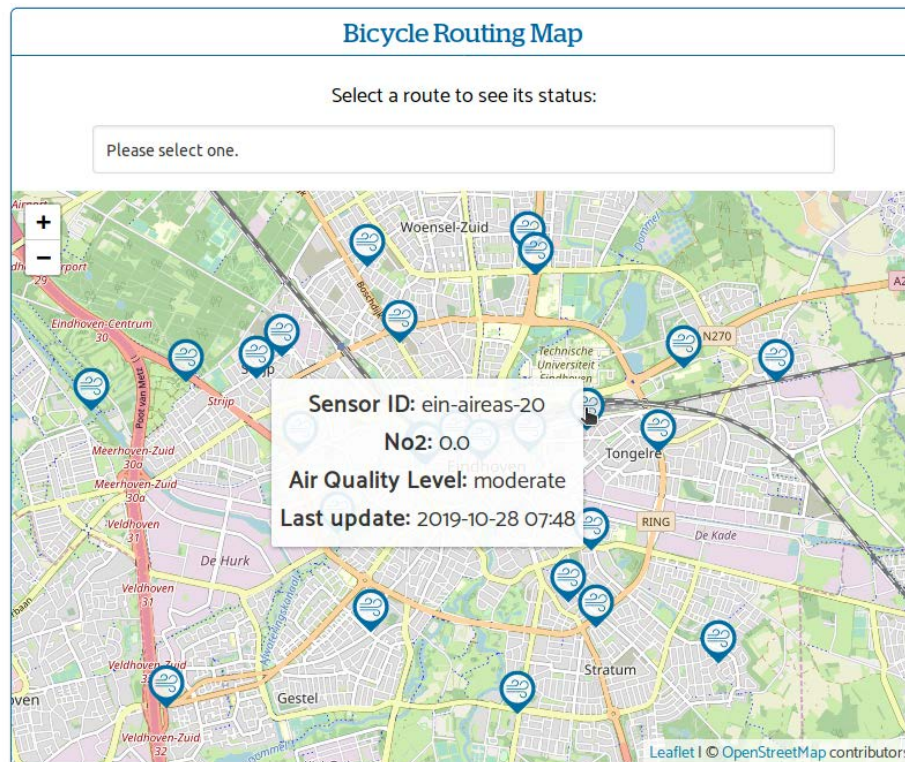


Figure 15. Sensor dashboard

Avoid parking in polluted areas

This dashboard is intended to be used by professionals

For this use case, data of parking areas in Eindhoven is added to the AUDP. This data source is also previously normalized using the Context Management component (offering one of the Minimum Interoperability Points). Using this component interface, data is again directed to AUDP to be shown over a map. Data from parking sensors and air quality sensors is showed in a combined way. A user (with the Mobility and Sustainability Policy Advisor role), would check the map to take different sustainability and mobility policies.

5.7 Deployment conclusions and lessons learned

Short delay on deployment by ATOS due to the transition to the new and stable environment CDE (Customer Dedicated Environment). RZ Eindhoven needed to connect third parties (traffic lights, display controller and traffic management system). The data specification and protocol design for these connections was out of scope of the initial workpage for the city of Eindhoven, so additional agreements and funding where necessary to get this data from the suppliers. This did take extra time.

Alignment of strategic goals for the Synchronicity project with the expected added value for our citizens proved more complex than anticipated. This had to do with shifts in the consortium for Eindhoven, and changes in the leadership and project team that put a strain on continuity.

We have found that a conceptual cliff exists between the worlds of Smart City Data platforms and the mobility solution providers we need to make it all a success. That is not easy to

overcome. More attention to the deliverables for a reference zone should be reflected in the project structure, instead of a purely technical setup. This may help shift focus to the added value of the practical solutions for citizens. Integrate better with technical partners during definition and development stages.

During the past years, ATOS did a great investment about FIWARE and the potential of this platform. This was an attempt to test new business models around platform concepts; to understand the use of open APIs and reusable components was part of the challenge of easing integration with different systems and accelerating the development of services based on cross-sectorial functionalities. The investment gave evidence of business potential specifically in the market of smart cities, where problems like vendor lock-in and limited innovation were clear. The work done about MIMS, data models and interoperability led Atos to increasingly invest in data analytics. One example of this, it is the ATOS Urban Data Platform (AUDP), where the data integration is now an easier task and the company can concentrate on data analytics and services implementation. This can be seen in the extra use cases that were added in D3.6 and covered in this deliverable in the [section 5.6](#). These extra use cases, designed to be integrated in the Eindhoven RZ, are a great playground to understand the deployment of MIMS as an extension of these concepts. With them Atos was looking to widen the spectrum of data sources and integration with a large variety of IoT infrastructures and ATOS own urban platform.

5.8 Sustainability plan upon project completion and future improvements

The city of Eindhoven plans to keep the focus on the improvement of bicycle mobility by having a better cycling experience and flow. The developed use cases will remain operational during after project close, at least for the time of the internal evaluation. This will include an analysis of suitable deployment sites and a sustainability proposal to continue operation until end-of-life of the synchronicity components. Also, alternative use case could be identified to add extra value, such as the identification of additional data sources to affect bike priority, such as traffic intensity of crossing traffic. Depending on the results of location analysis, the system can be moved to another location.

The main innovation for Eindhoven however is the deployment of the synchronicity data platform for multiple cross-silo purposes. We now have the intention to expand the developed the Synchronicity platform beyond the original use cases and use it for a new innovation project 'Inzicht Verlicht'. The data ingress is also being transferred from Python script to Apache nifi is planned. At a big intersection in Eindhoven new cameras and audio sensors are installed to collect better data. This will help the city to achieve higher level goals that connect mobility with sustainability and emission goals, and provide a detailed insight into the factors that contribute to the overall city goals. Also, we are converging legacy smart city solutions such as the city event sound dashboard to the platform to achieve even greater added value and datasets. Already, we have identified a dozen potential additional solutions to scale up after that. This is where Synchronicity will really shine and live on.

6. Carouge

6.1 Pilot motivations and deployed use cases

The complete description of the different use cases implemented in the Carouge reference zone is available in the SynchroniCity deliverables D3.5 and D3.6.

The first use case, in fact the first application deployed in the city of Carouge, is Smart Parking. The goal of this application developed in the context of the SynchroniCity project is to help reduce unnecessary traffic generated when the car drivers are searching for a parking slot. The application itself permits showing areas free parking spots are available currently. This is done through a parking estimator Atomic Service which provides the current status in the different parking areas in the Carouge reference zone. So, a driver coming in Carouge knows in advance where the free parking spots are available and can save his time to find a parking spot for his car.

In parallel, the users of the SynchroniCity platform installed for the Carouge reference zone can also see the different public transport stops (bus, trolleybus and tramways) and the related time tables on the map of the application. This allows to quickly determine the best way to go across the city within the public transportation. For example, a driver can leave his car on a parking spot outside the core of the city and come into the centre using a tramway.

The second application is the monitoring of the noise level near the bars and restaurants established in Carouge. The aim of this application is to determine which areas are the noisiest one in Carouge and so, the city council can take the right measures to reduce the noise pollution. The latest measurements for the noise level are displayed on a heatmap and historical data can be also found in the same manner.

On the other hand, Mandat International has also developed a new application named Privacy App³⁵. The goal of this application is to list all the IoT devices installed in the smart cities and to provide to the citizens all the information related to the privacy of these IoT devices. So thanks to this application, it is possible to identify all the IoT devices installed in the public space. Through Privacy App, the cities can communicate with their citizens all the IoT deployments and their purpose. Each citizen can use the application to report new seen IoT devices and to learn about the deployed IoT devices.

6.2 Test results of pilot

Different sources of data were integrated to the Carouge SynchroniCity platform instantiation and all the data are available in a central platform under the form of a Google map centred on the Carouge reference zone³⁶.

³⁵ <https://privacyapp.info/>

³⁶ <https://map.cityreport.org/?latitude=46.1829674&longitude=6.1378538999999982&zoom=14>

Each kind of data is pushed to the Carouge Orion Context Management API instance and then, used in the SynchroniCity application layer known as the services or applications. The most representative application is the map where different data sources are displayed (Figures 16-22). The complete data flow from an IoT sensor to the application was tested and the following screenshots are showing the final results.

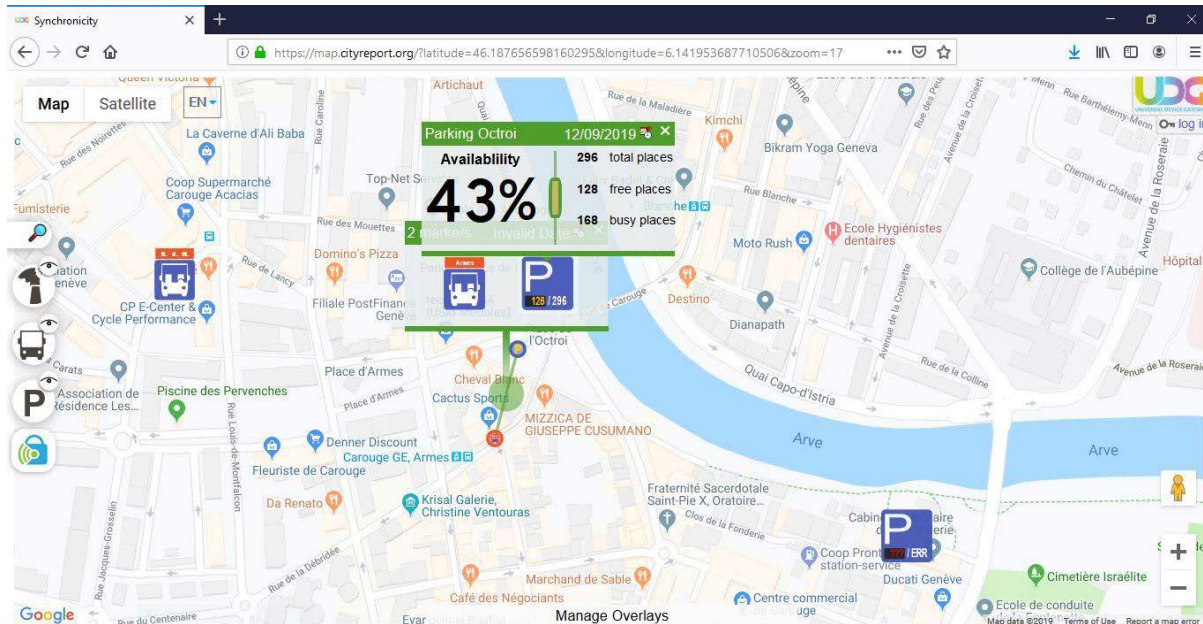


Figure 16

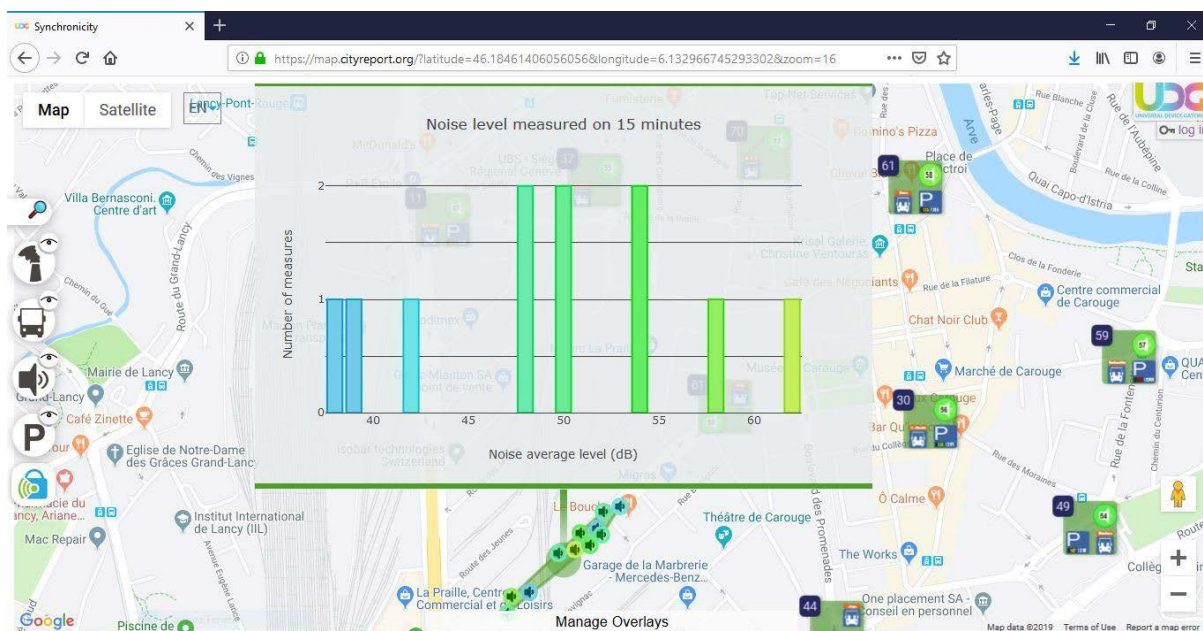


Figure 17. Data provided by noise level sensors

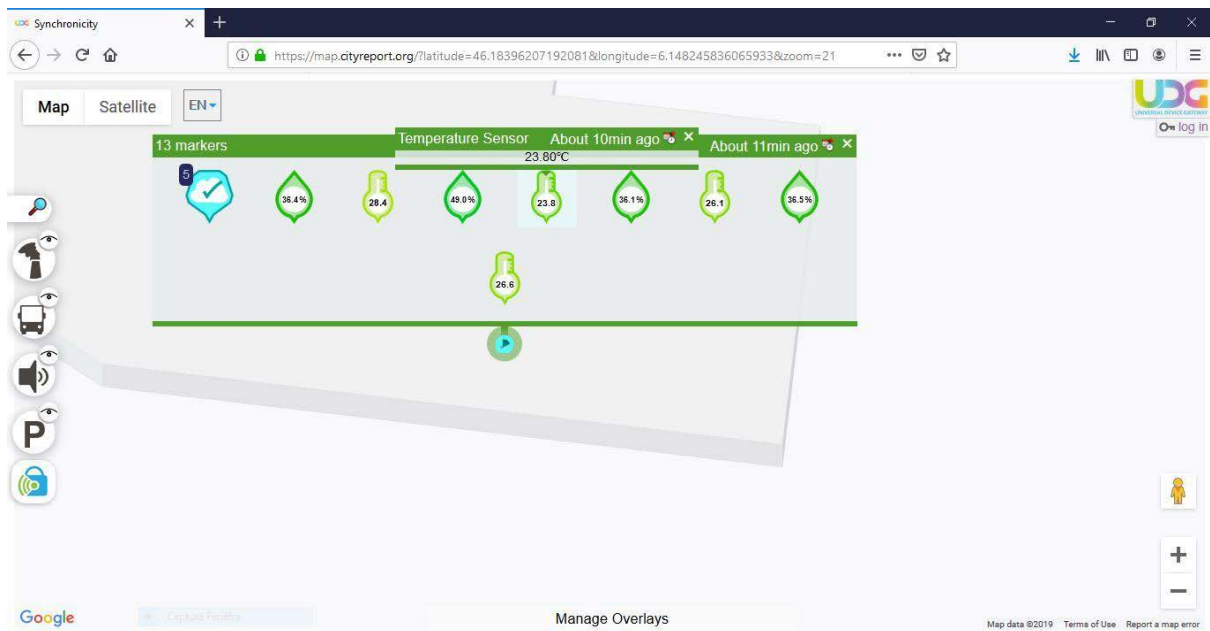


Figure 18. Temperature sensor

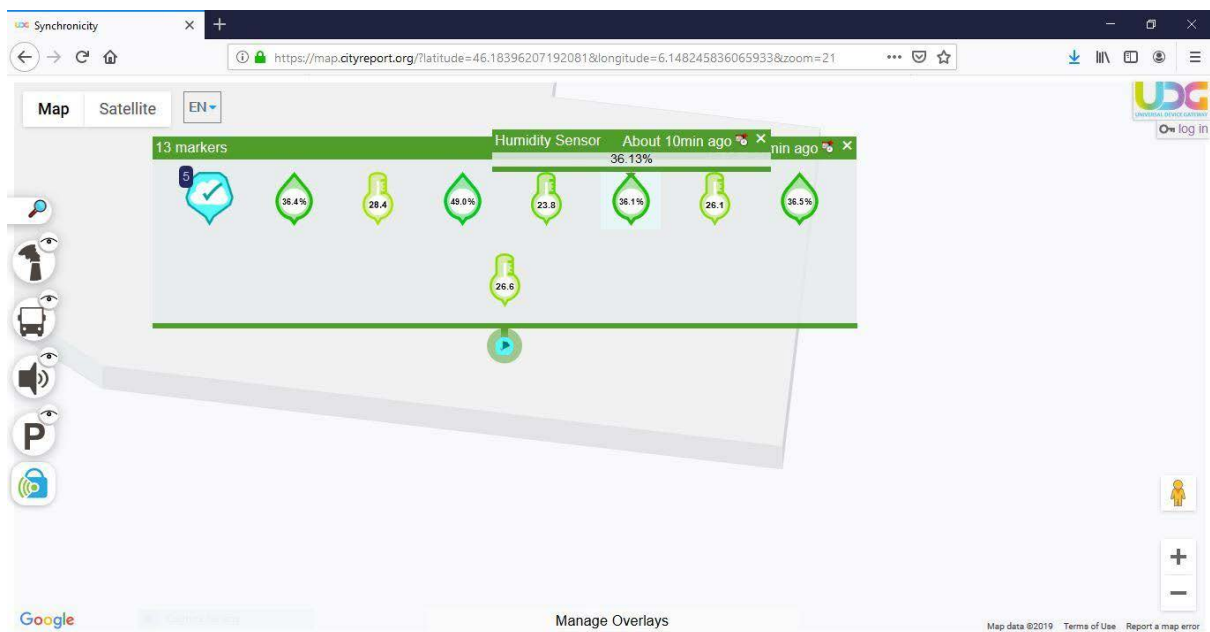


Figure 19. Humidity sensor

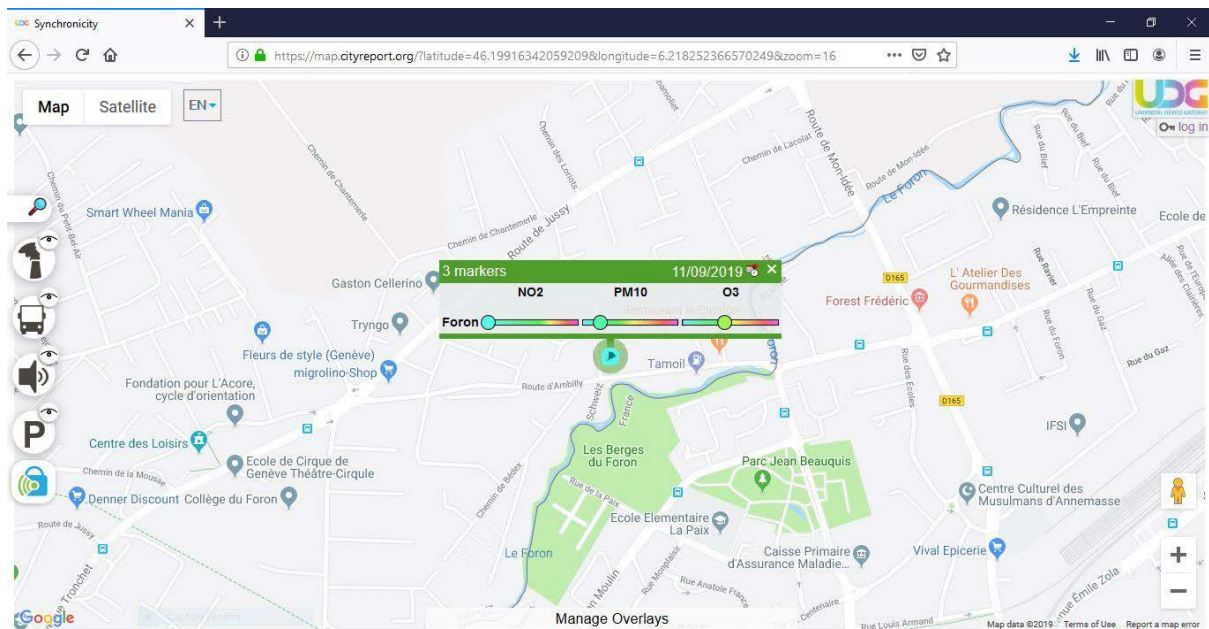


Figure 20. Air quality measured in Carouge

The map is also used to show the information provided by the parking estimator atomic service also illustrated below:

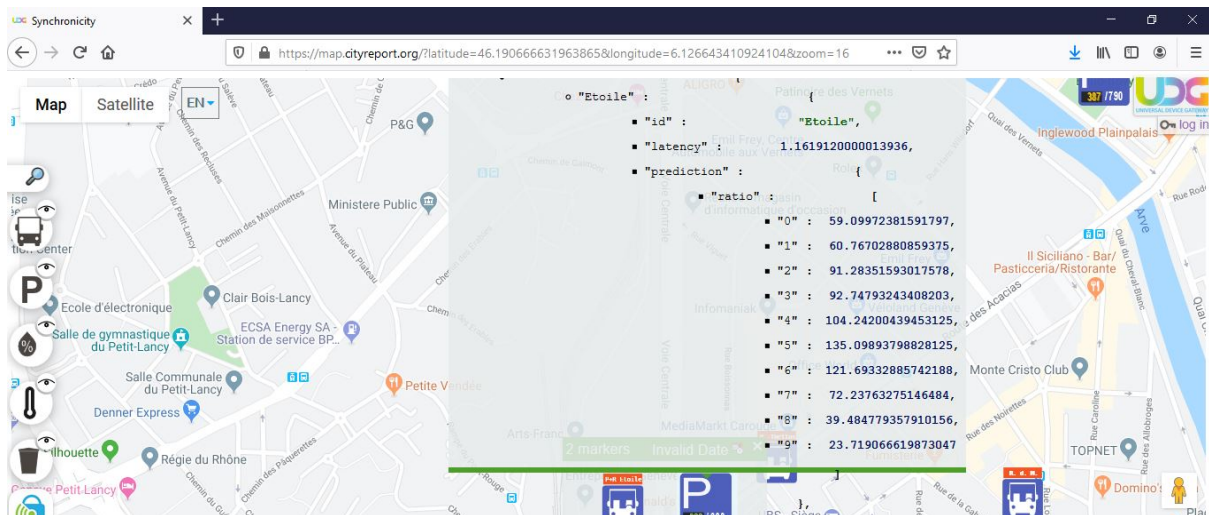


Figure 21. Results provided by the parking estimator

The second application is Privacy App which describes the sensors installed in a smart city, in our case in Carouge. The goal of Privacy App is to inform the citizens about the IoT devices deployed in their city.

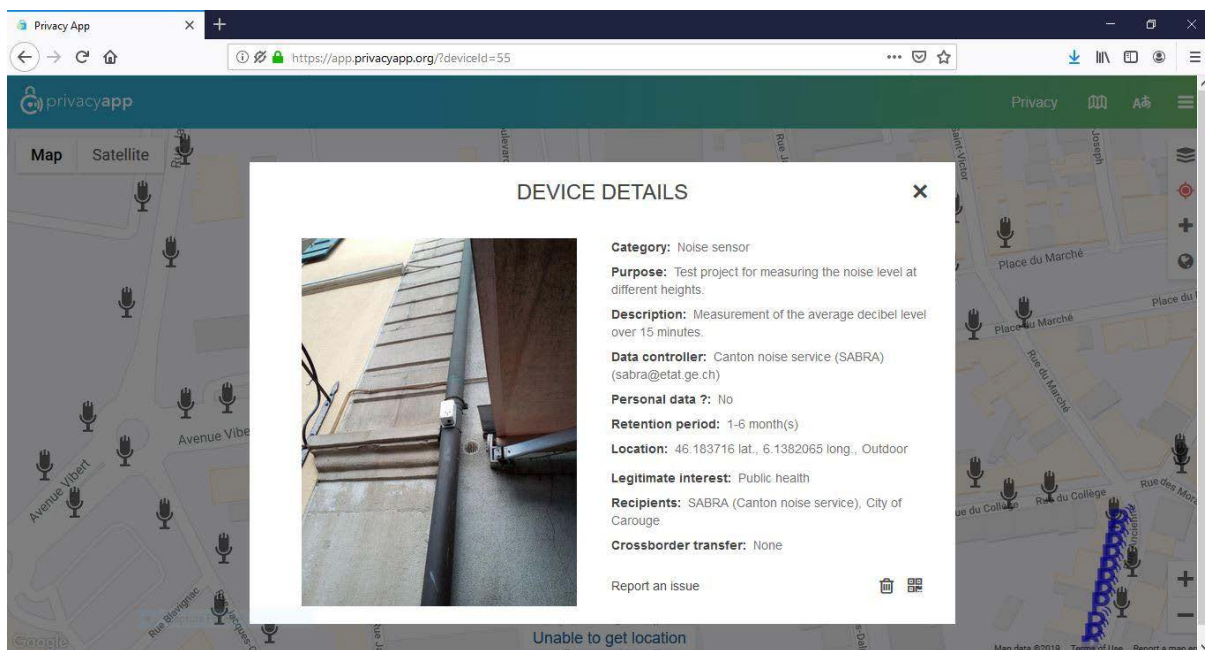


Figure 22. A noise level sensor in Privacy App

This application can be used by every city of the project consortium and worldwide. Indeed, the application does not need the datasets defined in the context of SynchroniCity, but all the information concerning the privacy of each IoT device liked shown above.

6.3 Implementation architecture and used Atomic Services

Used atomic services

Parking Estimation Service

The Carouge reference zone is using the parking estimator atomic service, developed initially for a different pilot and adopted at later stages by the SynchroniCity project pilots. The goal of the parking estimator is to provide an estimation of the future parking slots availability based on the data collected by the sensors installed in the Carouge reference zone. The parking estimator is also divided into two other versions: a noise estimator and a traffic flow estimator. These two versions have the same software basis as the parking estimator, but the data are coming from the noise level sensors and the traffic flow sensors installed in Carouge.

The architecture of the application described in this section is using of course the SynchroniCity framework instantiation installed in the Carouge reference zone. So, the data needed for this application are collected by the UDGAaS which is forwarding the data to Orion context Broker. Then Cygnus sends the data to the database used for historical storage (MongoDB by the way). Then a Web service implemented by a Python script automatically retrieves the data for the parking and gives them to the parking estimator atomic service which does the estimations. The results generated by the parking estimator are available under the form of a URL for each parking entities.

6.3.1 Implemented infrastructures

Some of the data sources existed before the start of the project:

- Public transportation schedules provided by the Transports Public Genevois (TPG) (<http://www.tpg.ch/>). The related data are provided as open data.
- Traffic flow and parking information are provided by the Système d'Information du Territoire à Genève (SITG) (<https://ge.ch/sitg/>), also available as open data at the level of the Canton of Geneva.

In the context of the SynchroniCity project, two companies based in Geneva have built sensors which were installed in the streets of Carouge:

- OrbiWise: Noise level sensors used in the frame of the noise monitoring near the bars and restaurants.
- IEM: Magnetic parking sensors installed on each parking slot in Rue Ancienne in Carouge.

Other companies involved in the context of the WP5 pilots provided sensors:

- Leapcraft: Air quality in the schools.
- Linc: Power consumption measurements.
- SmartImpact: Air quality in Carouge.
- Quamtra: Fill level of the rubbish bins.

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

Table 15. Carouge RZ themes and services

The data coming from the sensors made by Linc, Leapcraft and SmartImpact are collected, stored inside the Carouge Orion context broker, available through the MiM of the

SynchroniCity framework and the Context Management API and used in the map presented in the above section.

Some numbers can be given concerning the data sources:

- 100 sensors for air quality from different types.
- 18 public transportation stops.
- 453 noise level sensors.
- 45 parking sensors.
- 25 sensors for the humidity.
- 25 sensors for the temperature.
- 5 sensors for the energy consumption.
- 1 sensor for the trash bin level (work in progress when this document was written).

The Carouge SynchroniCity platform instantiation, based on the components issued from the WP2, includes:

- UDGaaS, which was certified through the FIWARE IoT ready programme. UDGaaS is responsible for collecting all kinds of data from different sources and to push them to the FIWARE generic enablers installed in the Carouge reference zone. It replaces the IoT agent defined by FIWARE.
- Orion context broker.
- Cygnus, installed on the same virtual machine than Orion context broker.
- Comet STH with the MongoDB database for the historical data.
- KeyRock, for the users management.
- Wilma, the PEP proxy.

Other components were added to the Carouge SynchroniCity framework instance to cover all the requirements asked by the smart city of Carouge:

- RESTHeart, used for the map to retrieve the historical data.
- CKAN, for Carouge open access data.

On the application level, the Carouge reference zone is using the following components:

- Google Maps with the Carouge data and sensors.
- Parking estimator and its variants.
- Privacy App

The server used for the Carouge SynchroniCity framework instantiation is also hosting the IoT Product marketplace, which is selling products (hardware and software) for IoT. This marketplace was put in place for all the project partners.

6.3.2 Changes and deviations of planned features of pilot

There are no major deviations in the planning made for the Carouge reference zone. A lot of time was spent solving the issues linked to the parking estimator. The development of Privacy App was scheduled at the very beginning of the project, but was completed before the end of the SynchroniCity project. Some changes were done in the data models, but this issue was

reported in WP2 and WP4. Indeed, the Carouge reference zone is now fully aligned with the data models defined in FIWARE and in the project.

6.4 Involved ecosystem partners and their roles

There are mainly three partners worked in the Carouge reference zone:

- The City of Carouge
- Mandat International
- UDG Alliance

The three partners are working closely together to achieve the objectives defined at the beginning of the SynchroniCity project in the different domains linked to the Internet of Things: IoT integration and interoperability, data protection, standardisation and exploitation.

The suppliers were described in the section named “Implemented Infrastructures”.

6.4.1 Required supportive actions and solved issues

The Carouge reference zone, through its various partners, has participated in numerous events presenting the work done in relation with the SynchroniCity project. The goal was also to involve the citizens in the project as they are the end-users of the applications and services developed in the frame of SynchroniCity. The different feedback collected in the events were used to improve the SynchroniCity platform instance deployed for the city of Carouge and the different applications. The complete description of the citizens' involvement and their feedback is mentioned in the deliverable D4.4.

6.5 Internal and external dependencies

There is no major internal dependency. For the external dependencies, the Carouge reference zone has worked with external suppliers for the sensors and the communication of the data generated by these sensors. There are project dependencies, specially for the WP3 where Carouge is associated with ATOS for the parking estimator.

6.6 Deployment conclusions and lessons learned

The first version of the parking estimator was installed in August 2018, but the tests were not good for different reasons: the usage of the resources on the dedicated virtual machine was too important, the time of processing was too high compared to the requirements expressed by the Carouge reference zone. The team of UDG Alliance has also modified the manner to feed the parking estimator atomic services with the data provided by the parking sensors. Finally, after a long work from ATOS and UDG Alliance, the parking estimator was improved. At the time of the writing of this document, a totally new version of the parking estimator was published by ATOS, but it was installed and is currently under tests in the Carouge reference zone.

The lessons learned are the following:

- The SynchroniCity and FIWARE data models must be strictly respected by everyone using the SynchroniCity components deployed in the Carouge reference zone.
- The tests made in the context of WP4 and WP5 are shown that not all the problems were detected immediately. Indeed, only when the final integration of the sensors and their data was realized in the Carouge reference zone, the technical team working for the city of Carouge have found errors. Typically, one of the most common errors appearing in Carouge was the inversion of the latitude and longitude in the data models used by the sensors provided by the pilots or third parties.
- The sensors must provide accurate and reliable data. Indeed, it is very important for the citizens to obtain correct information for the IoT devices installed in Carouge.
- The security and privacy should be considered more globally by all the project partners. The citizens expect to have a smart city without any failures in terms of data protection. Indeed, a breach in the data protection could have a very bad effect for the citizens who will not anymore believe in the concept of smart cities.
- The deployment of new technologies for smart cities should be clearly justified in terms of benefits, notably the good impacts for the environment and the health. In the context, the citizens are becoming more critical about the advantages brought by the Internet of Things of other related technologies.
- The large amount of different data sources, including several types of IoT sensors, permits to have a better understanding of the urban environment in the Carouge reference zone. The interactions between the different actors and stakeholders of the city can be better observed and new insights can be elaborated at the city level.

The deployment of the sensors, the collection of their data and the utilisation of the SynchroniCity platform in Carouge are considered as successful and a lot of objectives defined during the first phase of the project were accomplished during the project duration.

6.7 Sustainability plan upon project completion and future improvements

The SynchroniCity platform instantiated in the Carouge reference zone can be of course improved, notably to be more reliable and more manageable using a software like OpenStack. On the other hand, the work done during the three years of the SynchroniCity platform allows the different partners of the reference zone to get a lot of experience. The SynchroniCity platform will be re-used in new European projects in collaboration with the city of Carouge. Also, new business models for the SynchroniCity platform installed in the Carouge reference zone could be elaborated in the next future.

7.Porto

7.1 Pilot motivations and deployed use cases

Porto's metropolitan area is over two thousand square kilometres large and includes a wide number of transportation providers (public and private) operating in the area. Although these support a common multimodal ticketing system, it is difficult for someone to use public transportation in a comfortable and articulated way. Each of the transportation companies defines their own route and schedule, often the traffic jams cause significant delays and when there are service outages, people in bus stops are not informed or aware. There is a lack of information about the existing transports, its stops, routes, trips, schedules, etc. Not real-time or accurate information of the waiting time with the exception of the Metro system. Because none of the transportation providers covers all city areas, there's a real need of articulating mobility with 2 or more different operators. This is a problem that affects not only suburban areas as well as downtown areas.

This type of service level, in the past considered to be 'how things are', as ceased to be accepted. Society evolved and is today much more time consuming and stressful. Wealth increase has allowed for citizens acquire and use their own cars when going to work. From a financial point of view, the combined transportation costs are often similar to the private vehicle cost, and despite the added traffic pressure, especially during rush hour, it is faster to drive your own car than taking public transportation. The public transportation usage is increasing due to city sensitization campaigns but Porto still has a high rate of private vehicle usage.

Besides, current navigation applications do not properly combine the usage of private vehicle and public transportation in a seamless way, for example, by providing indoor and outdoor parking slot availability. They also don't provide real-time data, like real-time vehicle location or estimated time of arrival, or even take non-scheduled issues into consideration (such as roadblocks accidents or major events in the city). Accordingly, many private vehicle owners tend to decline the use of public transportation in at least some portions of their journeys.

Over the last few years, and in order to promote a sustainable mobility within the city, the Municipality of Porto has been engaging several initiatives that promote, incentivize and ease the usage of public transportation.

In this sense, Porto selected Multimodal Transportation as one of its use cases in SynchroniCity, proposing the implementation of a navigator application to ease the usage of different transportation means and discourage the use of private transportation when possible.

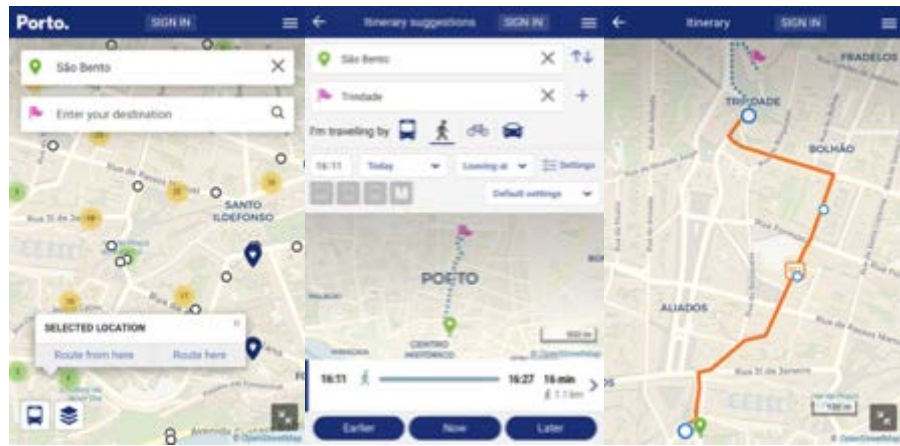


Figure 23. Porto Multimodal Assistant Application (mobile version)

The following sections intends to complement and update the details previously provided in deliverables D3.5, D3.6 and D3.7.

7.2 Test results of pilot

Concept testing & validation

Table 11 below summarizes the end-user's tests and opinions about the user experience of the app. Note that results were obtained before all functionalities were implemented. For instance, 24% of the test group mentioned that they would not use the app because real-time data was not available, which was due to the fact that real-time data was not being provided at that time the UX tests were performed.

What the participants liked most: ● Porto branding. ● Different alternatives of public transports from the same origin. ● Having calories displayed to incentive walking/cycling. ● 77% agree that the web app is easy to use and to get the right information. Top rated as true: ● The typography (lettering, headings, titles) is clear and has legibility. ● It is easy to find my way around the product. ● Information is written in a style that suits me.	Major issues of the pilot solution. (eg: technical problems, service, user experience, other) ● Problems in finding some addresses. ● The way search results are presented makes it difficult to find the right address. ● Too much information presented in mobile – clusters and POIs should be disabled. ● Map area is very small in mobile when the filters are open. ● Not a real time information.
--	--

● The product content interests me. ● The product is well-suited to repeat visitors.	
What the participants liked the least: ● 62% consider the platform is incomplete. ● 53% did not like the fact that this was not a native app (but a web app, instead). ● 24% would not use it because real time data is not available. ● 54% disagreed that the information is clearly arranged and structured. Top rated as false: ● It is fun to explore the product. ● Screens have the right amount of information. ● The product is designed with me in mind. ● The product reflects progressive, leading edge design. ● The product is exciting.	Recommendations for improvement: ● 53% would like to have the warnings and impediments available. ● 24% would prefer to use other apps (google maps, wase, moovit, etc.). Suggestions: ● Remove the clusters because most users did not understand them. ● The POIs' layer should only be visible after selecting it and not from the beginning because it represents visual noise. ● Provide bicycle paths visible when transportation by bike is selected. ● Provide an option to match the best route that contains certain POIs. ● Provide a greater range than only limited to the city of Porto.
Positive findings: ● 92% of the participants considered the platform useful but incomplete. ● UI is clean. ● Platform is useful and simple to use. ● 77% trust the data because it has the Porto branding.	

Table 16. Test results Porto RZ.

7.3 Implementation architecture and used Atomic Services

The following diagram updates the implementation architecture presented in D3.6, as significant changes were made (for clarity, the Context Management API is duplicated in the diagram, but there is only one endpoint for it).

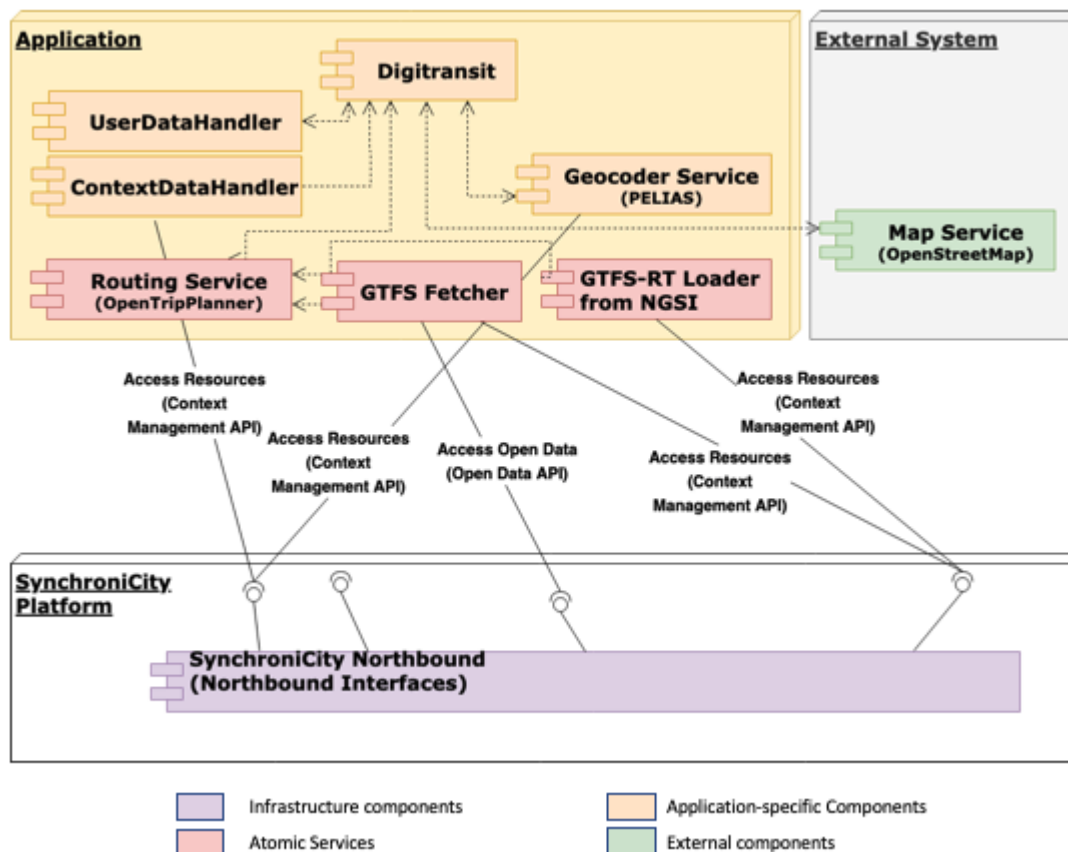


Figure 24. Implementation architecture

For the application itself, Porto opted to have its solution based on **Digitransit** which is an existing open source solution, that is also used in the city of Helsinki.

For Porto's implementation, **Digitransit** relies directly on five components, one of which is an Atomic Service and, indirectly, on another two (also SynchroniCity's Atomic Services):

The **Map Service** component is an external (public cloud based) service used for showing the map of the city.

For geolocation (searching origin/destination), an instance of the **Geocoder Service (PELIAS)** was installed. This component requires loading information like street names, but also Points of Interest (coming from the SynchroniCity platform) and its GPS coordinates.

Once the origin and destination are determined, the **Routing Service (OpenTripPlanner)** is used for route calculation. This service is fed by two additional Atomic Services:

- **GTFS Fetcher**, that provides an automated way to update GTFS information, by extracting data from GTFSTransitFeedFile entities and importing GTFS files (static timetables and related info) into the OpenTripPlanner.
- **GTFS-RT Loader from NGSI**, which consumes real-time urban transport entities (so far, ArrivalEstimation) and generates GTFS-RT feeds from it, so that, for instance, users can see real-time ETA for buses.

The **UserDataHandler** service is responsible, not only for storing user preferences but also for providing history and usage metrics (that allows the user to analyse indicators like calories spent and walking distance).

The **ContextDataHandler** is a new component. It is responsible for gathering city context information from the Context Broker and make it available as a map layer. For the moment, it only allows to retrieve and show information about POIs, but a refactor is planned, in order to make it useful for other data sources (e.g. noise, weather and traffic). Besides, it is not a reusable component, but it is expected to be as a result of the referred refactor operation.

7.3.1 Implemented infrastructure

The infrastructure for supporting the Multimodal Transportation Application comprises components across three areas: Data Providers, Platform and Software Components:

Sensors & data providers

- ROPI – Official Repository of Points of Interest (*Repositório Oficial de Pontos de Interesse*)
- GTFS files (directly provided by the transportation providers on open data portals)
- Mipweb – Official GIS database of the Municipality of Porto

Service	Atomic	Description	Status
Routing Service	Yes	Based on OpenTripPlanner ³⁷ it finds suitable routes combining taxi stops, buses info and bicycle routes (among other available possibilities) between two points and according to user's preferences.	Installed
(reverse) geo-coder Service	No	Based on Pelias ³⁸ , taking an input text, such as an address or the name of a place, it returns a latitude/longitude location on the Earth's surface for that place (and vice-versa).	Installed

³⁷ www.opentripplanner.org

³⁸ <https://github.com/pelias/pelias>

GTFS Fetcher	Yes	Extracts data from GTFSTransitFeedFile entities ³⁹ and imports GTFS uploaded files (static timetables and related info) into the OpenTripPlanner platform, making this available for routing calculation (within the Routing Service).	Installed
-----------------	-----	---	-----------

Table 17. Porto RZ

Platforms

- SynchroniCity platform: namely the context broker, installed in WP2 (please refer to D2.9 for implementation details). At the moment of writing this document, no other platform component is used, though it is expected that at least QuantumLeap⁴⁰ (the one responsible for storing history data) will be used. No component updates were required.

Software components (described earlier)

- Atomic Services (AS)
 - OTP – OpenTripPlanner (for route calculation)
 - GTFS Fetcher
 - GTFS-RT Loader from NGSI
- Other
 - PELIAS for geocoding and reverse geo-coding

7.3.2 Changes and deviations of planned features of pilot

The initial proposal for this application was very ambitious. For the reasons highlighted in D3.6, we had to rewrite the objectives for this application, within the scope of SynchroniCity. This was mainly due to the realization of the actual effort required to implement some features (for example, implement a system of green credits), the time required for breaking silos and get access to mobility data (in particular, from internal departments, the public transportation providers and the mobility regulation entities), and the effort to integrate data sources with the SynchroniCity platform.

Nevertheless, taking into consideration the role of this application in the planned strategy to improve the sustainable mobility within the city, Porto is highly committed to implement the initially proposed features, beyond SynchroniCity.

³⁹ <https://gitlab.com/synchronicity-iot/synchronicity-data-models>

⁴⁰ <https://github.com/smartsdk/ngsi-timeseries-api>

7.4 Involved ecosystem partners and their roles

Whilst implementing the Multimodal Application, the following partners were involved:

- Municipality of Porto: Being the entity that requires a Multimodal Application the municipality was responsible for requirements analysis, use cases description and user validation
- Synchronicity partners:
 - Porto Digital⁴¹: As partner for the innovation projects of the Municipality, Porto Digital was responsible for the deployment of the application and components, and user validation
 - Ubiwhere⁴²: Local company with expertise on Smart City Solutions provided Technical support
- Suppliers
 - Whitesmith⁴³: Local Software development company, responsible for UI development of some features

7.4.1 Required supportive actions and solved issues

Previous supportive actions (in particular, quantitative and qualitative research activities based on service design and design thinking methodologies, and tools) were described in D3.7.

After in depth analysis, the source code of Digitransit (the UI component), it was noticed that Porto Digital did not have the required qualified developers for implementing some of the proposed features. For that reason, and as indicated in the previous section, a local company (Whitesmith) was hired for that purpose.

7.5 Internal and external dependencies

Please refer to D3.7 for a detailed explanation of the previously identified internal and external dependencies. The main dependencies for this pilot are as follows:

- Internal dependencies:
 - WP2: SynchroniCity's reference architecture and data models.
 - WP3: Availability of Atomic Services.
 - WP4: Validation through defined mechanisms (technical, integration and replication, users and stakeholders, market).

⁴¹ <https://www.portodigital.pt/>

⁴² <https://www.ubiwhere.com/>

⁴³ <https://www.whitesmith.co/>

- External dependencies:

- Updated data from buses, metro and train lines routes, schedules, and real-time ETA, operating in the city.
- Points of Interest from ROPI (Official Repository of Points of Interest).

7.6 Deployment conclusions and lessons learned

Implementing such application revealed to be full of challenges and a lot more difficult than initially predicted. In addition to what was already mentioned in D3.6, here are the lessons learned:

- To some extent, the encountered difficulties are similar to other cities: breaking administrative silos and have data available (at the right time and with high quality).
- There's a big gap between the way transportation providers think and work when looking on how technology can help improve services to their users, and the objectives of SynchroniCity for this particular matter.
- Porto decided to use open source software as building blocks, and thus reducing the costs and development time of implementation. It is believed that these advantages will be more tangible thought time. Nevertheless, the integration of the transportation providers' systems and data with the city's platform took almost half the time required for getting this project up and running.
- Although the new GDPR law was written in 2016, it was introduced in may 2018. This required additional developments, which were not expected from the beginning, also with implications with the conclusion below. Changes in regulation can impact the development.
- Making data publicly available, requires a strong support and knowledge about the local policy for Open Data. In our particular case, a lot of the data required was not tagged as Open Data, and so before using it, we've gone tedious process, which added some delay.
- Using official data revealed to be a difficult task. As an example, although public/community's databases like OpentSreetMap were easier to use (essentially due to its well documented and open format), it was required to use the municipality's street geographical database (provided by a private vendor). Which, in turn, required efforts to extract data from.
- When having the same type of data provided by multiple sources, issues were detected even when they all use the same data model. In fact, having data sources reporting measurements in observation time intervals, requires some kind of harmonization, before using the data.
- It was difficult to find *de facto* data models, but a very important work was done in WP2.
- Since cities have common problems, the use of Atomic Services have a huge impact in the implementation of solutions, considering its replicability with minimum efforts.

7.7 Sustainability plan upon project completion and future improvements

Backing the description made in deliverable D3.7 for the sustainability plan (see “Description of sustainability mode upon project completion”), the development of this solution was followed by a validation process, which was able to provide a better understanding of the stakeholders and users involved, as well as the potential market.

A careful process was developed together with the potential users to define features, which could be technologically implemented (feasible) while solving their needs (needed). There was also a process to test the application in what comes to its user interface (UI) and usability (UX). In this sense, efforts were made to have a solution that addresses both the Municipality's and the citizen's needs, which improves problem/solution fit.

Given the feedback received upon tests and opinions gathered from the usability test group, most of the participants (92%) considered the application useful. The Municipality of Porto (CMP) itself strongly believes that the “Porto. Multimodal Assistant” can be an interesting tool to ease and encourage the use of public transportation and other new transportation means (such as, bike and scooter-sharing) in the city of Porto. Accordingly, Porto Digital and CMP will keep investing in new developments, features and releases of this application after the pilot phase and after the SynchroniCity project ends, including communication and dissemination activities.

Having validated the interest of the potential users and the ones of the potential owner of the solution, it is comfortable to say that this project is empowered enough for the municipality to maintain it after SynchroniCity's end.

The tests performed with users also showed a lot of potential for evolution. For future improvements, a lot can be done to increase the likeliness of adoption of the app by the users, which we split in two groups:

- Proposed features by the users involved in the tests (previously mentioned).
- Features determined by the municipality, to improve the sustainable mobility within the city (such as including information about weather, pollution, traffic, estimations, etc.)

With the results achieved, the city is willing to fund further technological, product and service development of the solution. Porto doesn't want to force users to use its own multimodal application. The goal is to deliver a set of base components (themselves being backed by the synchronicity platform) for other similar (and hopefully, better) third party applications.

It is expected that, in the near future, the “Porto. Multimodal Assistant” will not be a single and stand-alone application (as it will be within the context of the SynchroniCity project's pilot), but a set of features and functionalities of the Porto Urban Platform, and as such, Porto Digital and CMP will keep investing in its development, integration and deployment, either directly, or by subcontracting to a software company.

Furthermore, the platform (software and hardware) will need to be scaled out in order to cope with the demand. This demand may either be from the use of Digitransit based component directly or indirectly by other applications.

8. Santander

8.1 Pilot motivations and deployed use cases

Even though Santander, due to its size and number of inhabitants (around 172,000 inhabitants), cannot be considered a big city, the truth is that its particular conditions (such as its high number of hills) make transport a transcendental and fundamental issue for its citizens and the municipal corporation.

In the last few years, Santander City Council is investing to improve the sustainable mobility within the City. In this way, as hinted in D3.5 and D3.6, new mobility premises such as escalators and lifts have been deployed to ease the access different parts of this hilly city, as well as reinforce and redistribute city public transportation to better adapt to citizens requests. This is the reason why Santander RZ selected Multimodal Transportation use case, within Synchronicity scope, to exploit these new infrastructures and provide better mobility resources to its citizens and visitors.

Therefore, the use case deployed versed around this topic, developing an application called “Park & Move” which dealt with the multimodal transportation and route planning (see Figure 25) and looked for ways to complement it with a parking estimation (see Figure 26). This application will encourage some of Santander, and its immediate surrounding neighborhoods, citizens to drop the use of their own car downtown, thus making the city centre much more interesting for both citizens and visitors. In addition, those who mandatory require access the city centre with their own car will enjoy a comfortable way to find quickly the parking spots where to park it, since the results are offered in real time, updated every minute. All in all, this kind of pilot makes a lot of sense in a city with Santander’s characteristics.

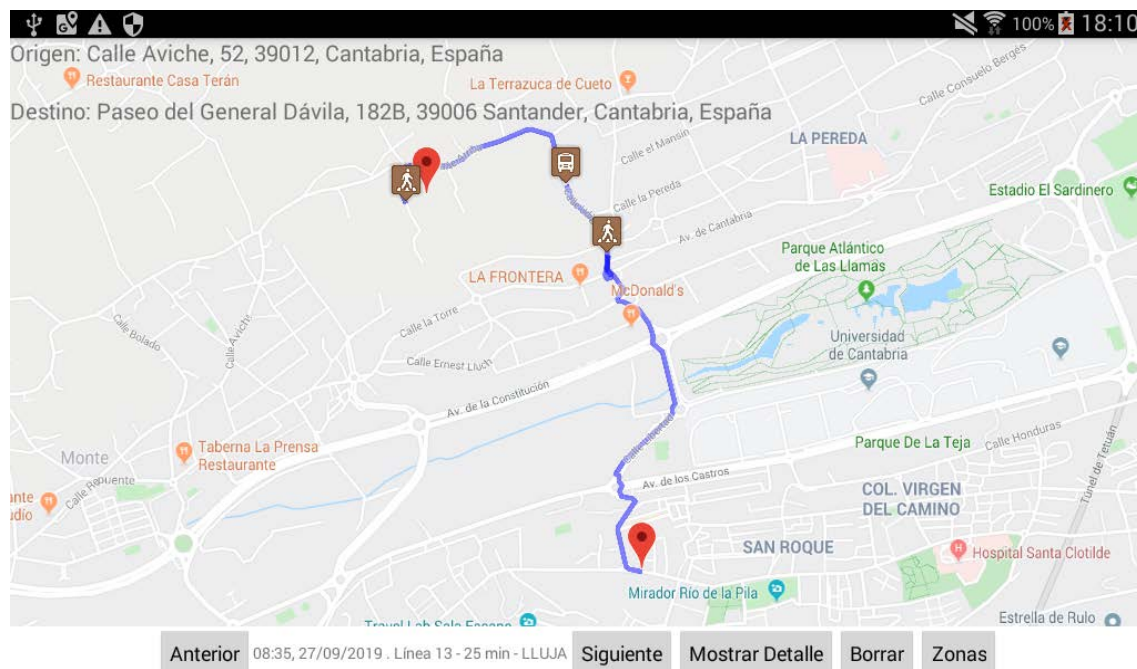
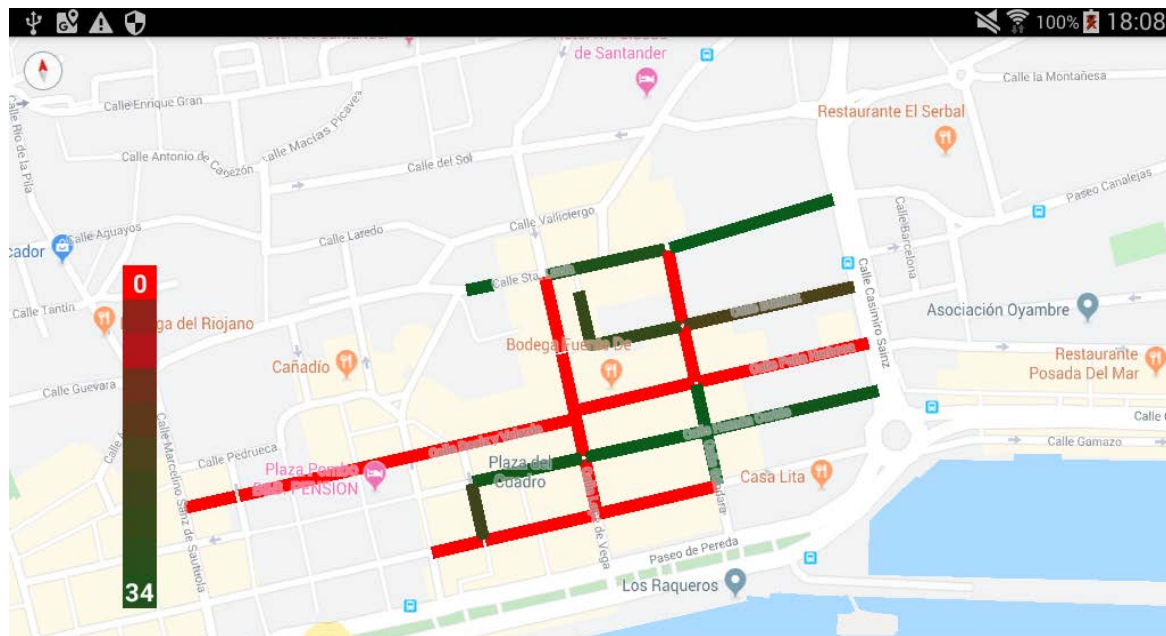


Figure 25. Route suggestion



Figure 26. Santander "Park & Move" Application - route planning (mobile version)



Mapa de Rutas

Figure 27. Santander “Park & Move” Application - parking estimation (mobile version)

8.2 Test results of pilot

Concept testing & validation

Table 18 below presents a recap to summarize the main tests and opinions related to the user experience of the app. Please note that these results were obtained before all functionalities were implemented. Nevertheless, they all help to sketch a quite accurate painting of the real situation.

Preferred features	• Different alternatives of public transports from the same origin. • Easy of use and to get the right information
Least attractive features	• Unfinished platform • Initial trials ran in a web app: users would prefer a native application
Major issues of the pilot solution	• Problems in finding some addresses and sketching the corresponding routes • Absence of real time information in some cases
Suggestions for improvement	• Seamless integration of other means of transportation when possible

	• Increase the range that currently is only limited to the city of Santander
--	--

Table 18. Test matrix for Santander

8.3 Implementation architecture and used Atomic Services

The basic architecture of the Santander application was already presented in D3.5 and partners stuck to that plan, shown in Figure 27.

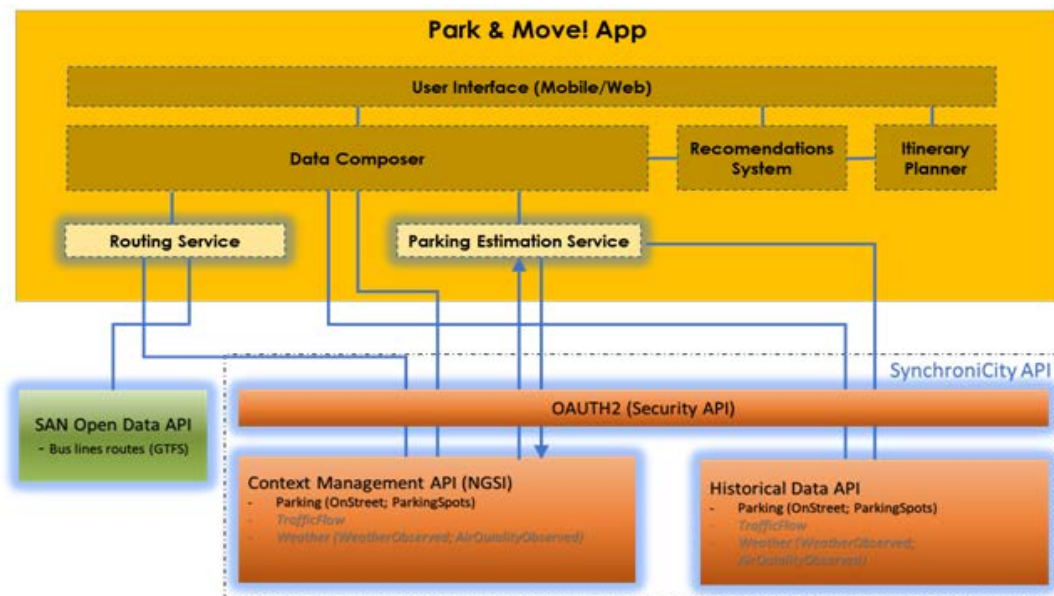


Figure 28. Santander "Park & Move! App" basic architecture

As for the used Atomic Services, the two initially considered were also dealt with when preparing the final application. On the first stage, the Routing Service was implemented while the Parking Estimation Service came in the second phase.

The **Routing Service** is based on OpenTripPlanner, it calculates multimodal routes within a city (and surroundings) involving public transportation (GTFS format), bicycle lanes and mobility premises. The provided routes consider also special requests (e.g., points to/not to go, sections to/not to cross, profiles, etc.) from the service consumer.

Furthermore, this Routing Service comprises two different atomic services:

GTFS Fetcher

In charge of providing an automated way to update GTFS information.

GTFS-RT Loader from NGSI

Dealing with the consumption of real-time urban transport entities and generates GTFS-RT feeds from it.

Parking Estimation Service

Calculates and provides the probability of finding a parking spot for a car within a given area in the city and in the requested time window (e.g.: the parking prob in the city centre in 20 minutes from now). Thought to be easily evolved to include more data sources and innovative algorithms, a first version is based on current parking status and historical data referred to city areas.

8.3.1 Implemented infrastructure

The implemented infrastructure to provide support for the Multimodal Transportation Application comprises components across mainly three areas: Sensors/Data Providers, Platform and Software Components. The last two have been deployed in servers dedicated to accommodate the required services and owned by the RZ participants.

Sensors/data providers:

- IoT equipment already deployed in Santander (e.g. OnStreetParking areas)

Platforms:

- Synchronicity Marketplace

Software components:

- OTP environment for the web version
- Parking estimator instance (see D3.4) [10]

8.3.2 Changes and deviations of planned features of pilot

During the preparation and development of the pilot some features initially considered were in the end dropped from the final version due to the appearance of unexpected or insurmountable hurdles. Moreover, the contact with interested stakeholders such as municipality representatives and Santander citizens with diverse backgrounds helped get an extra point of view and distinguish what features could be considered as really helpful and which ones did not make a real difference.

As an example of this situation, the integration of additional air quality or noise level data sources was not implemented, due to the fact that they did not complete the service in the way it was originally estimated. The kind of information they provided is apparently of not much use for the real-life users, so it's better to get rid of it and focus on what they are really fond of.

The same goes for the Weather Forecast data source, which was first considered as a potential ideal fit for the second version of the Santander multimodal application. So far, it has not been implemented.

8.4 Involved ecosystem partners and their roles

The partners involved in the execution of this pilot have been the ones in the consortium located in the city of Santander, assuming different roles. A brief explanation of such roles is offered in the list below.

- Synchronicity Partners:
 - City of Santander: Proposal of use cases apt for the city, conduct user validation process.
 - UC: Main role in the technical and user validation process
 - TST: In charge of the multimodal transportation application development and its technical validation.
 - Atos: Parking Estimator Service provider

8.4.1 Required supportive actions and solved issues

The Santander reference zone, through his various partners, has participated in various events presenting the work done in relation with SynchroniCity. The goal was also to engage and involve the citizens in the project as they are the end-users of the applications and services developed in the frame of the project. These events were also useful to retrieve valuable feedback used to improve the envisioned applications.

The main aim of these sessions was to gather the views of Santander's citizens about their life in the city and also to collect a set of improvement proposals obtained from brainstorming and further discussion.

In order to broaden the perspective of ideas and interests and to enrich the discussions, a diverse selection of citizens (age, occupation, location ...) was invited to participate in the sessions.

After a brief introduction to SynchroniCity project, summarizing its main goals and objectives, it was explained how the involvement and engagement of Santander citizenship is a core requirement of the SynchroniCity project goals. Afterwards, an open discussion was carried out with participants in order to know their daily habits and ask for their personal experiences within the city, extracting interesting conclusions from the different neighborhoods represented.

At last, some Smart City concepts were introduced to the attendees and their inputs were collected regarding what technical possibilities they would be interested in using to somehow improve their lives in Santander.

The sessions produced a significant amount of information. In general, citizens perceive Santander as a nice city to live in, mainly due to the fact that it's a medium size city with a good range of cultural activities on offer and also thanks to its natural resources, like green areas, beaches and the bay.

On the other hand, when talking about the main issues detected, there was a recurring mention to transportation, from the bus frequencies knowledge and their timely updates to the available parking spaces, especially downtown, dealing as well with the proper definition of accessible areas. All in all, topics that were addressed by the developed pilot, which tried to provide citizens with a useful tool to find a solution to these woes.

8.5 Internal and external dependencies

Internal dependencies for this pilot are the following:

- The pilot depends on WP2's defined architecture
- Availability of atomic services (WP3)
- Validation from WP4 through their defined mechanisms

As for its external dependencies:

- Updated data from bus lines routes in the city

8.6 Deployment conclusions and lessons learned

A delay on the implementation of the application happened due to the fact that getting to fully grasp OTP was a task tougher than expected. Furthermore, it was also difficult getting to know the precise location and characteristics of every mean of transportation in the city, especially the escalators and slopes. In addition, the creation of particular GTFS files for those means of transportation that do not have it can be tricky sometimes.

With regard to the lessons learnt throughout the execution of this activity, the following can be highlighted:

- Pay an even greater attention to citizen groups opinions, since they are the ones who would use the proposed solutions in the end and it will help technical partners focus on the added value of the practical solutions for citizens.
- Create a climate where different partners, technical and not, involved in the solution development are capable or getting on the same page easily during both definition and development stages.
- Find a balance between the options offered by the whole set of data available and the one piece really useful for each service to provide: it's usually better to focus on certain information and not overwhelm end users with not so relevant details.
- Understand the different politics behind the diverse means of transportation in the city and supply them all with an overall perspective which in the end will lead to the citizens benefit.

8.7 Sustainability plan upon project completion and future improvements

Start with the consolidation of the solution and verify the approval of the majority of citizens, to confirm there is a real interest in a service like this. Once proved, check the kind of improvements and features those users would like to have and evaluate the possibility of implementing them or not in further iterations.

In the end, another goal should imply exporting the solution to different cities, considering the diverse means of transportation that can be found in them, thus requiring adaptation in the generic application. On the other hand, the creation of new business models related to the employment of the SynchroniCity platform will be analyzed by the members of the consortium in the Santander RZ.

Furthermore, the SynchroniCity platform and its features could be used again in new collaborative projects, both in an European and a cross-border level, in collaboration with the city of Santander.

9. Antwerp

9.1 Pilot motivations and deployed use cases

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 TrafficFlowObserved format.

The objectives of the city 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. This is the use case that was decided upon for the RZ Antwerp: Bicycle patterns in the city of Antwerp.

To realise this use case, an application will be developed. The background conditions this application needs, are useful, consistent bicycle data. The 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. 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.

The bicycle data for Antwerp is delivered by tracking Ring-Ring⁴⁴ bicycles. Ring-Ring is a company for which users can ride their bike and transfer their kilometers into discounts at local stores. Ring-Ring is currently not active in Antwerp anymore, but delivered a historical dataset of 22 million data lines from activities in Antwerp in the recent past (2017).

9.2 Test results of pilot

Meaningful data has proven to be the true pain point in this work package for Antwerp. To start with, it was a true challenge to find a use case that mapped with the city needs as well as with the availability of data. A lot of time was spent on finding the right use case(s). Several use cases were formulated and analyzed. Sometimes the use case lacked usefulness for the city. Sometimes the use case was a good idea, but there was no data to back it up. Or there was not enough data, or the data was not reliable enough. Often it was a combination of several of these impediments.

Finally, the above mentioned use case, Bicycle patterns in the city of Antwerp, was chosen because it was believed to be useful for the city and to have data available for its development. However, some projects that related to cycling safety were discontinued, only delivered real-time data from 1 intersection or could not guarantee sufficient accuracy. Finally, after a manual control of the raw data, the Ring-Ring data was chosen as the main data set that would tick

⁴⁴ <https://ring-ring.nu/>

all checkboxes. It was a very large amount of data (22 million records) that seemed relevant (e.g. good spread) and relatively reliable (e.g. clean and unambiguous data). And although it should always be kept in mind that it was not a statistically correct sample, it could give some interesting insights in bicycle usage.

Already during the development phase, data issues arose again. Historical data was a part of the Synchronicity framework, but the primary focus lay on real-time data. The framework was proven less fit for large amounts of historical data to be uploaded at once. Up until now the city of Antwerp is still trying out alternative solutions for the historical data component, compliant with the Synchronicity framework.

When the development was demonstrated new concerns were raised. The data experts of the City of Antwerp expressed their concerns on the accuracy of the data and the relevance of the visualisation of the data. W.r.t. to the accuracy, it was found that about 18% of the data was lost after conversion/ingestion to NGSI.

Several discussions between the stakeholders led to the conclusion that the only sensible decision was to discontinue the work on the application and on this work package.

9.3 Implementation architecture and used Atomic Services

Atomic services used: none. 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.

Atomic services developed: Map Visualisation (+ datetime-filter). 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 data model categories Vehicle, TrafficFlowObserved and/or BikeHireDockingStation data.

The Vehicle data (i.e. the bicycle 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.

A web application has been developed. (As shown in Figure 23). The main Application Components are:

- **ETL (Extract, Transform, Load):** Extraction transform and load of the different data sources, which need to be transformed to NGSI models. This is done by using Apache NiFi ETL⁴⁵ 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 to the market place.

⁴⁵ <https://nifi.apache.org/>

- **Data validation:** Data is validated on correctness and validated and filtered on what data can be shown public. After the validation the GPS data is transformed to the NGSiv2 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 the 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.

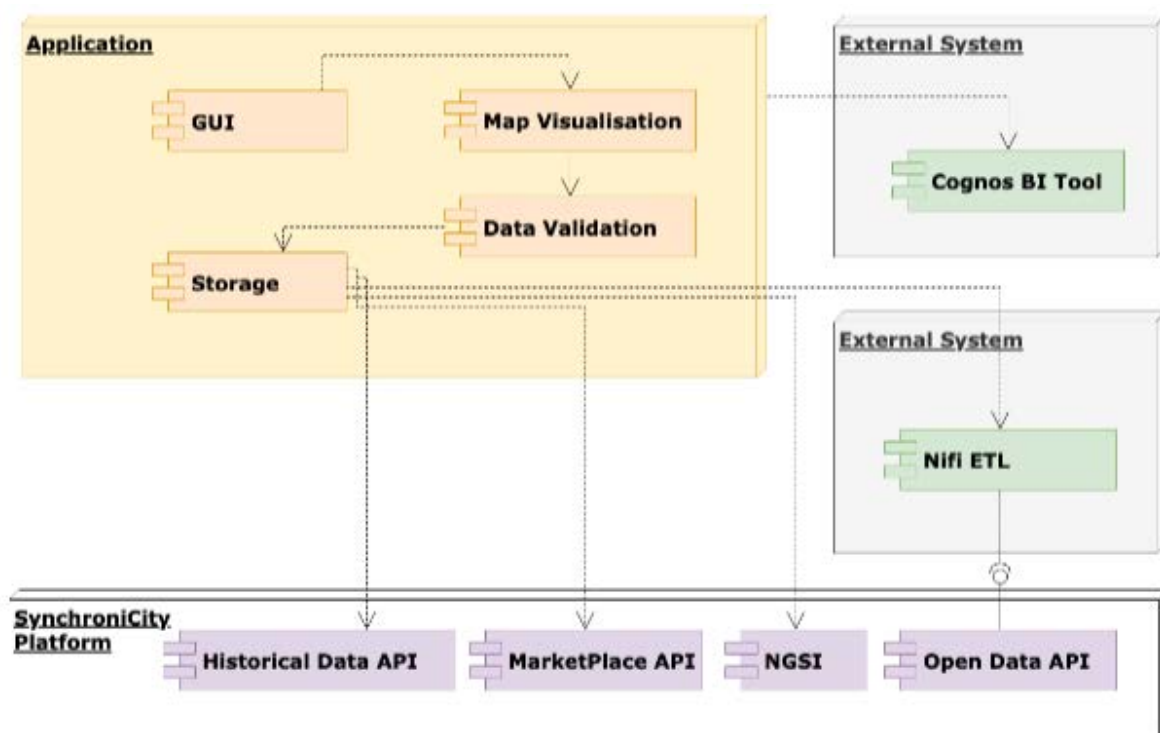


Figure 29. Architecture of the initial application in Reference Zone Antwerp by Rombit.

9.3.1 Implemented infrastructure

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 the above diagram.

Sensors & data providers:

- Ring-Ring data set. The background conditions this application needs, are useful, consistent bicycle data which is delivered by tracking Ring-Ring bicycles. 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.
- Data from bicycle hiring stations. The BikeHireDockingStation data is gathered and visualized through icons dispersed throughout Antwerp and were obtained from an open api which holds a lot of different data about bike hire stations.

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

Table 19. Antwerp RZ themes and services

Platforms:

- Kubernetes: This container management platform was used to host the frontend visualization and was also used to aggregate all the bicycle data to form TrafficFlowObserved data.

Software components:

- Map Visualization: As described above in the previous section, is a map component is used to visualise the different data sources.
- Apache NiFi: As described above in the previous section, Apache Nifi was used to transform all the incoming data to NGSI format.
- 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 basis of the gained data and other datasets of the city of Antwerp.
- Orion Context Broker: This server side concept was used to store all NGSI data and enables to search on time and data of TrafficFlowObserved data.

9.3.2 Changes and deviations of planned features of pilot

Final deployment of the developed Initial Application on the Antwerp city stack was not executed. The lack of usefulness of the development lead to the decision to discontinue the work on the application and this work package (see [8.2 Test results of pilot](#)).

9.4 Involved ecosystem partners and their roles

- Synchronicity Partners:

- Stad Antwerpen / Digipolis Antwerpen: coordination, use cases, user/technical validation
- Rombit: application development, technical validation
- Imec: coordination, use cases (+ co-creation)
- Suppliers:
 - Ring-Ring: supplied a historic (static) data set of bicycle data

9.4.1 Required supportive actions and solved issues

Via internal co-creation workshops between imec and the city of Antwerp / Digipolis Antwerpen, the selection of a use case as well as the definition of the functional requirements of the application were facilitated. Second, many technical meetings have been set up between all internal stakeholders in order to refine the requirements of the application and to search for relevant data.

9.5 Internal and external dependencies

Internal dependencies

- Handling of historical data, and large historical datasets more specifically, in the Synchronicity framework requires additional investigation in/for the City of Antwerp. Historical data was a part of the Synchronicity framework, but the primary focus lay on real-time data. The framework was proven less fit for large amounts of historical data to be uploaded at once. At the time of writing this deliverable the city of Antwerp is still trying out alternative solutions for the historical data component, compliant with the Synchronicity framework.
- Additionally, a recent hypothesis arose, that still requires additional investigation: during the execution of the project, the development of a local NGSI compliant IoT stack was ongoing. NGSI data conversion/ historical replay was in particular a part of that data stack where at the time of the development of the application still many uncertainties existed (e.g. at what data rate could data be replayed over Orion) which may have an impact on data availability/quality.
- Availability of reliable and relevant data to support relevant use cases for the city

External dependencies:

- Usability and quality of the Ring-Ring data

9.6 Deployment conclusions and lessons learned

The result of the realization of the Initial application raised a lot of questions on the usefulness of the app. Therefore, Antwerp decided not to continue with this track. As a result: an Initial Application was developed for the Antwerp use case, but it was not deployed in production.

The only data available for the use cases was a large dataset of historical data. This was not the main focus of the Synchronicity framework and a lesser fit. Alternative data options have been thoroughly analysed during the initial application meetings between the Antwerp partners but it was concluded that all options would finally not lead to a more useful application. Some examples of the paths that were investigated: real-time traffic data from 1 intersection was made available but it was decided that this was too scarce for a city-level application; a track to extend bicycles with proper sensors to collect data was too complex to realise (solely) within the Synchronicity timeframe/context, and several others.

The lack of useful data and the loss in data conversion (18%) to the NGSI-standard was an impediment. A lot of effort/time was lost in trying to search/generate useful data. A second critical path during the evaluation of the developed application was the aspect of data accuracy. The process of translating non-compliant data to a NGSI compliant format appeared to lead to data loss. E.g., it was found that around 18% of the data was not available in Orion after conversion. The process of identifying possible issues in this process appeared to be resource intensive. In addition to the previous point: transformation from all static data to NGSI (TrafficFlowObserved) was more computationally expensive than initially anticipated. It needed special care to be run in a decent amount of time.

9.7 Sustainability plan upon project completion and future improvements

There are no concrete plans to continue the development of the app nor to deploy the app to facilitate city services. However, as both Stad Antwerpen / Digipolis Antwerpen and imec are (jointly) accelerating in exposing data through Fiware NGSI compliant data models and APIs, there could be a future opportunity where reuse/adoption of the application could be re-assessed.

10. Seongnam

10.1 Pilot motivations and deployed use cases

Seongnam is one of the satellite cities near the capital city, Seoul, in South Korea. Therefore, cities around Seoul have problems on parking as well as heavy traffic. For citizens in Seongnam, the SynchroniCity Korean consortium (KETI, SNIP and ULike) worked on the parking service and data marketplace with collected real-time parking data.

The parking recommendation service is deployed as the extension to the previous parking service, commuter parking. It provides the parking spot recommendation to users while they're trying to find free parking spaces. Based on real-time availability, congestion level of cars in parking lots and user preference on parking area, the parking application recommends parking spots to park easily in a short time. In the KETI headquarter testbed, parking event data is collected from the LoRa parking sensors and the image recognition system which consists of a number of IP cameras and deep-learning machines. The video stream from an IP camera is analyzed by deep-learning algorithms to get the congestion level of parking blocks. The number of roaming cars for parking is used to get the congestion index and the index is used to provide the parking recommendation.

Smart parking service is focusing on commuters who are using their cars to transfer to the public transportation. Through the smartphone application, drivers can check not only the available parking spots but also information of public transportation including schedules of intercity bus, metro and intercity bus terminal. The in-depth description of this use case can be found in D3.6.

Data marketplace is available on the oneM2M⁴⁶ based platform. Users can sell or buy data through the marketplace web portal. The features implemented on the extended oneM2M platform contains managing the data sales, limiting maximum data usage (retrieval and notification) count, access checking for purchased data.

10.2 Implementation architecture and used Atomic Services

Platform architecture provided in Seongnam is illustrated below (Figure 24). Details of the application components can be found in D3.6.

The three adapters used in this architecture:

- **oneM2M Adaptor** collects data from city infrastructures (e.g. IoT devices, data portals).
- **Mca-Ckan Adaptor** transfers data from the oneM2M APIs to CKAN.

⁴⁶ <http://www.onem2m.org/>

- **Mca-NGSI Adaptor** works as an interworking proxy entity between oneM2M platform and Context Broker. The proxy supports both set of APIs from oneM2M and NGSI.
- **Data marketplace** provides data distribution mechanism for oneM2M platform. Users can sell or purchase data on the marketplace portal. Purchased data item can be retrieved from the oneM2M platform by smart city applications.

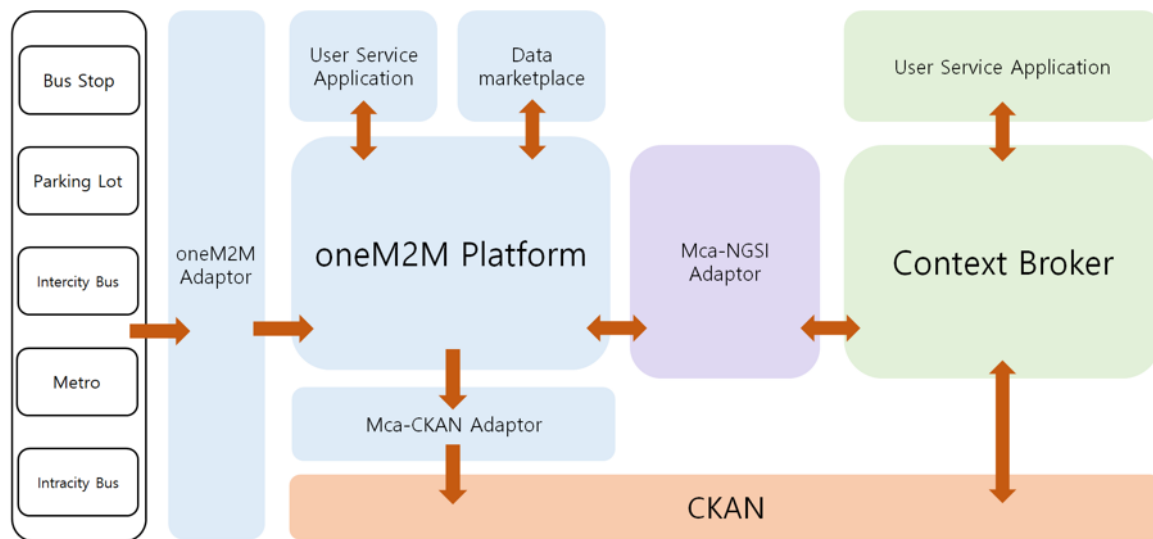


Figure 30. Platform Architecture

10.2.1 Implemented Infrastructure

Sensors & data providers

- About 400 parking sensors deployed parking lots near the Yatap station.
- About 130 parking sensors deployed in the KETI parking lot.
- 10 IP cameras deployed in the KETI parking lot.
- Public transportation data are collected from open data portals.

Entity Type	Reference	Data Source	Accessibility	Ent ries
OffStreetParking	OffStreetParking	Proprietary parking service server	public	2
ParkingSpot	ParkingSpot	Proprietary parking service server	public	231
BusStop	BusStop	GyeongGi Bus Information:	public	4

		http://www.gbis.go.kr/		
Intra_bus	BusLine	http://www.gbis.go.kr/	public	96
Inter_bus	BusLine	Open Data Portal: https://www.data.go.kr/dataset/15000590/openapi.do	public	105
Metro	-	Open Data Portal: https://www.data.go.kr/dataset/15000754/openapi.do	public	2

Table 20. Seongnam available datasets

Platforms

- oneM2M platform
- Context Broker

Software Components

- Mca-NGSI Adaptor
- Mca-CKAN Adaptor
- oneM2M Adaptor
- Data marketplace

10.2.2 Changes and deviations of planned features of pilot

- LoRa parking sensors had connectivity issues with gateways, so to get more reliable parking event data, IP cameras were installed and applied image recognition with deep learning

10.3 Involved ecosystem partners and their roles

The SynchroniCity Korean consortium composed of three organizations:

KETI (Korea Electronics Technology Institute)

- Developing oneM2M based smart city extension platform
- Developing an adaptor to interwork between Context Broker and oneM2M platform
- Developing application for users which gets data from the Context Broker
- Coordinating the SynchroniCity Korean consortium

ULike

- Developing data marketplace that uses APIs of the extended oneM2M platform
- Developing smartphone application for users which gets data from the oneM2M platform

SNIP (Seongnam Industry Promotion Agency)

- Linking consortium with civic groups, local businesses
- Working with public organizations to prepare testbeds in downtown
- Collecting user experiences of the smart city services

10.3.1 Required supportive actions and solved issues

With the prior experience on oneM2M-based smart city project in Korea, the main goal of the Korean consortium is to show the integrated smart city architecture of oneM2M platform and FIWARE open source components. It has been developed the interworking proxy between oneM2M and FIWARE platforms. The proxy enable datasets in one platform can be used in the other platform. This realizes interoperable smart cities with digital single market.

10.4 Internal and external dependencies

Internal dependencies comprise of all the atomic services defined for in Seongnam Architecture.

10.5 Development conclusion and lessons learned.

One key factor gained from the project was the bridge built between the oneM2M-platform and the Synchronicity Context Broker. This will be beneficial in future smart city development since it broadens the solution scope with a single digital market.

The European marketplace idea along with the digital single market concept have been influential. Similar features have been built on top of the oneM2M platform.

10.6 Sustainability plan upon project completion and future improvements

Some plans and actions have been identified for the future. Future plans include engaging SMEs to promote the platform and services as well as help their business with open source software offerings from the project.

The marketplace billing service will be implemented with the possibility of adding a private blockchain for the exchange of tokens between sellers and buyers.

The parking infrastructure near Yatap will be managed in the future by a public organisation while the oneM2M server will be run by KETI. A machine learning approach will be introduced and data such as weather observations and forecasts will be used in estimating parking occupancy.

11. Conclusion

This document means the end of a long journey that started a long time ago (almost three years). At that (almost) clean-slate beginning, a number of cities (or, in the context of Synchronicity, Reference Zones), followed their own IoT infrastructures, potentially in a non-interoperable manner among each other. Namely, Helsinki, Milano, Porto, Santander, Manchester, Antwerp, Eindhoven, Carouge and Seongnam were part of the original line-up of reference zones. At the time of closing this deliverable, and hence this Work Package number 3, cities have streamlined their legacy solutions in order to be compliant with the framework proposed in the scope of Synchronicity. Thanks to this paradigm shift, a global IoT market of data and services loom.

As stated above, this deliverable is the last stop on a path of milestones, represented in the following list of deliverables:

- D3.1 introduced the three Application Themes (i.e. Human-Centric Traffic Management, Multi-modal Transportation and Community Policy Suite) that cities had to address when it came to define their own fully-fledged applications (or so-called “Customized IoT service prototypes”).
- D3.2 and D3.3 presented the concept of “atomic services” (formerly baseline services), small pieces of software where replicability and interoperability are their main targets.
- D3.4 settles down the guidelines that embrace the steps to be done on the definition and subsequent implementation of the various “atomic services”.
- D3.5 and D3.6 tackled the breakthrough of the different applications, carried out by the aforementioned reference zones, spanning the technical aspects that shape the underlying development of the prototypes.
- D3.7 covered a first report on the deployment of the different applications, but covering a logistic/managerial plane. This first loop sought internal and external dependencies among cities and partners, and glimpsed a first view on what RZs expect after the project comes to an end.

On its behalf, D3.8 concludes the documentation process initiated by D3.7, focusing on the final outcomes and remarks of each RZ application. In this case, we pay attention to things like testing/feedback, lessons learnt during the process, etc. Finally, we delve into actual cities’ sustainability plans.

One of the things that is worth bringing up is the difficulty of achieving all the plans and goals that were initially pledged. Every RZ cluster is composed of a handful of teams, usually from a wide variety of skills and backgrounds; apart from that, multiple agents (e.g. dependencies coming from external projects or municipality duties, etc.) may jump in at any point and jeopardize the regular roadmap each city nailed down at the beginning of the project lifetime. That being said, the general trend says that in the vast majority of the cases cities have reached the minimum expected milestones of their respective pilots.

Reference zone pilots were designed to behave as real testers and validators of the Synchronicity framework. This part can be considered as done across a constant and iterative assessment, whose outcomes and feedback were conveyed to their corresponding addressees (mainly, WP2 activities). This way, the concept of Minimal Interoperability

Mechanisms (MIMs) has been settled down, thus leading to a holistic and conceptual framework extrapolable to every single pilot. Especially the ideas of Context Information Management (MIM #1) and Shared Data Models (MIM #2) made ground throughout the development of the various applications.

Moreover, these pilots have also harnessed the availability and flexibility of Atomic Services, which were gradually integrated as part of the whole RZ ecosystem in a fair portion of the cases. Thanks to this adoption process, RZs relied on external (although internal in the scope of the project) sources to implement a core part of their own applications (e.g. Carouge by using traffic and noise estimators); hence they could mainly focus on the critical points that had to do with their own cities (e.g. getting feedback from citizens, public administrations, etc.).

In parallel, during the development of RZs respective development phases, they have harvested a number of worthy lessons and experiences to be considered in the future. For instance, one of the common perceptions is that many public municipal administrations behave in an isolated manner, as often the case in modern organisations. In all the cases, it is a belief that an important effort to digitize and make data available is of utmost importance. Hence, a good point would be to start from this critical issue.

Moving to the information plane, the amount of available data produced and provided by the cities was also overestimated. While the hypothesis was that open city data was going to be underused, in many cases one challenge was to identify meaningful use cases based on the data available. The real value of initial applications for users is quite much based on the available data on RZs. Real-time traffic data, for instance, was lacking in many instances and is one particular domain that needs to be addressed in the future. This challenge has been identified in many cities and actions have been taken to address it. The SynchroniCity concept and framework learnings and experiences were archived regardless of data limitations.

Shifting to a less technical standpoint, some reference zones concluded that a more citizen-centered (human-centric) approach is often needed when it comes to design a smart city solution or service. This leads to a call for a better alignment of strategic goals between the project mission and the expected added value for the citizens in question.

Last but not least, we can state that most of the pilots succeeded in providing either valuable knowhow and learnings to city stakeholders, even leading to a solution sought by the municipalities. For example, in Porto further development of the Multimodal Assistant is being resourced between Porto Digital and the Municipality of Porto which sees the application as a tool to encourage the use of public transportation and other forms of shared infrastructure. In some cases, a handover of the technologies is being planned like for example in the instance of Helsinki, where the main journey planner application owner HSL will be offered the “add on” -service of route optimisation by air quality. Other projects will also take over the developing work started by the SynchroniCity initial applications (RZ pilots).

References

- [1] SynchroniCity, "D3.7 - Pilot Deployment Plan"
- [2] SynchroniCity, "D3.6 - Customized IoT service prototypes for lead ref. zones – advanced"
- [3] SynchroniCity, "D4.2 - Technical Validation"
- [4] SynchroniCity, "D2.9 - Report on Advanced Reference Zones platform deployment and operational plan"
- [5] SynchroniCity, "D3.5 - Customized IoT service prototypes for lead ref. zones – basic"
- [6] SynchroniCity, "D2.10 - Reference Architecture for IoT Enabled Smart Cities"
- [7] SynchroniCity, "D4.3 - Technical Validation (Phase2)"
- [8] SynchroniCity, "D4.5 - Technical Validation of the SME projects"
- [9] SynchroniCity, "D3.1 - Specification and Design of Initial IoT Applications"
- [10] SynchroniCity, "D3.4 - Common methodology and toolset for city service customization"