URBANITE

Supporting the decision-making in urban transformation with the use of disruptive technologies

Deliverable D5.1

Detailed requirements specification

Editor(s):	Juncal Alonso, Iñaki Etxaniz
Responsible Partner:	Tecnalia
Status-Version:	Final – v2.0
Date:	26.12.2020
Distribution level (CO, PU):	PU

Project Number:	GA 870338
Project Title:	URBANITE

Title of Deliverable:	Detailed requirements specification
Due Date of Delivery to the EC:	31/12/2020

Workpackage responsible for the Deliverable:	WP5 - URBANITE ecosystem integration and DevOps
Editor(s):	Tecnalia
Contributor(s):	Juncal Alonso, Iñaki Etxaniz, Sonia Bilbao, Jorge Garcia (Tecnalia), Fritz Meiners (FhG), Alina Machidon (JSI), Giuseppe Ciulla (ENG)
Reviewer(s):	Heli Ponto (FVH)
Approved by:	All Partners
Recommended/mandatory readers:	WP3, WP4, WP6

Abstract:	This document will contain all the functional, non-functional and technical requirements of the URBANITE ecosystem and all the components to be developed in the context of the social and technical WPs (WP2, WP3, and WP4). This deliverable is the result of task 5.1.
Keyword List:	Functional requirements, Non-Functional Requirements, Technology, Baseline tools.
Licensing information:	This work is licensed under Creative Commons Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0) http://creativecommons.org/licenses/by-sa/3.0/ The document itself is delivered as a description for the European Commission about the released software, so it is not public.
Disclaimer	This document reflects only the author's views and neither Agency nor the Commission are responsible for any use that may be made of the information contained therein

Document Description

Document Revision History

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
v0.1	04/05/2020	First draft version	Tecnalia
v0.2	30/06/2020	Contributions added in section 3, requirements gathering for the different components	Fraunhofer, JSI, ENG, Tecnalia
v0.3	16/07/2020	Content in section 2.2 added, including the actors and the generic processes	Tecnalia
v0.4	28/07/2020	Agreed content added from the different partners' contribution	Tecnalia
v0.5	01/11/2020	Content added by partners	Fraunhofer, JSI, ENG, Tecnalia
v0.6	03/12/2020	Content added through all the sections by the partners	Fraunhofer, JSI, ENG, Tecnalia
v0.7	09/12/2020	Compilation of information gathered through the shared tables for section 5 and the Appendix	Tecnalia
v1.0	09/12/2020	Sent for review	Tecnalia
v1.1	10/12/2020	Updates on the tables	Tecnalia
v1.2	15/12/2020	Updates on the tables	FhG, Engineering, JSI
v1.3	21/12/2020	Address the comments of the internal review	Tecnalia
V2.0	26/12/2020	Version ready for submission	Tecnalia

Table of Contents

Table of Contents 4
List of Figures 5
List of Tables 6
Terms and abbreviations 7
Executive Summary 8
1 Introduction 9
1.1 About this deliverable 9
1.2 Document structure 9
2 Requirements management in URBANITE 11
2.1 Methodology for requirements elicitation 11
2.1.1 Requirements gathering and prioritization 11
2.1.2 Requirements documentation 14
2.2 URBANITE Main processes 15
2.2.1 URBANITE actors 15
2.2.2 URBANITE process 1: Select and combine available dataflows 16
2.2.3 URBANITE process 2: Visualize information in the map 17
2.2.4 URBANITE process 3: Data Analytics & events prediction 17
2.2.5 URBANITE process 4: Simulate (traffic) situations based on selected data and analyze obtained KPIs 19
2.2.6 URBANITE process 5: Collaborate in the Social Policy Lab 19
3 URBANITE ecosystem requirements 21
3.1 Functional requirements 21
3.1.1 Virtual SoPoLAb 21
3.1.2 Data Harvesting Component 24
3.1.3 Data Curation Component 25
3.1.4 Data Fusion/Aggregation Component 27
3.1.5 Data Storage Component 28
3.1.6 Data Retrieval Component 28
3.1.7 Data Projection component 29
3.1.8 Data Clustering component 29
3.1.9 Self-Organizing Map component 30
3.1.10 Regression component 30
3.1.11 Traffic Simulation component 30
3.1.12 Advanced Visualization component 31
3.1.13 Recommendation Engine component 31
3.1.14 Prediction component 32

3.1.15 Policy Simulation and Validation component 33
3.1.16 URBANITE UI Component 34
3.2 Non-functional requirements 35
3.3 Use Cases requirements 38
3.3.1 Helsinki Use Case 38
3.3.2 Bilbao Use Case 39
3.3.3 Messina Use Case 41
3.3.4 Amsterdam Use Case 42
4 Benchmarking of existing tools 45
5 Requirements alignment and prioritization 48
5.1 Functional requirements and KRs alignment matrix 48
5.2 Functional and use cases alignment matrix 49
5.3 Requirements prioritization matrix 52
6 Conclusions 55
7 References 56
APPENDIX: Tools description 57
Tool 1: ARX 57
Tool 2: Piveau *consus* 58
Tool 3: Piveau *hub* 59
Tool 4: Piveau *ui* 60
Tool 5: Piveau *incognito* 61
Tool 6: SUMO 62
Tool 7: MATSim 63
Tool 8: WEKA 64
Tool 9: Orange 3 65
Tool 10: Idra 66
Tool 11: DataModel Mapper 67
Tool 12: CaPe 68
Tool 13: Digital Enabler Platform 69
Tool 14: Micro Proxies 70
Tool 15: Decidim 71

List of Figures

FIGURE 1. URBANITE PROCESS FOR REQUIREMENTS GATHERING AND PRIORITIZATION 12
FIGURE 2. REQUIREMENTS PRIORITIZATION MATRIX (EXCERPT) 13

List of Tables

TABLE 1. TRACEABILITY MATRIX KEY RESULT – REQUIREMENT (EXCERPT) 13
TABLE 2. TRACEABILITY MATRIX FUNCTIONAL-USE CASE REQUIREMENTS (EXCERPT) 13
TABLE 3. RELATIONSHIP BETWEEN UC USERS' PROFILES AND URBANITE PLATFORM ACTORS. 15
TABLE 4 OVERVIEW OF THE NON-FUNCTIONAL REQUIREMENTS PER COMPONENT IN URBANITE 36
TABLE 5.OVERVIEW OF THE FUNCTIONAL REQUIREMENTS COVERAGE BY THE PROPOSED BASELINE TOOLS 46
TABLE 6. FUNCTIONAL REQUIREMENTS AND KRS ALIGNMENT MATRIX 48
TABLE 7. FUNCTIONAL AND UCS ALIGNMENT MATRIX 50
TABLE 8. FUNCTIONAL REQUIREMENTS PRIORITIZATION MATRIX 52

Terms and abbreviations

AGPL	Affero General Public License
API	Application Programming Interface
CKAN	Comprehensive Knowledge Archive Network
CLI	Command Line Interface
CSV	Comma-Separated Values
DKAN	Open Data CMS
DoA	Description of Action
DSS	Decision-Support System
EC	European Commission
FR	Functional Requirement
GIS	Geographic Information System
GNU	GNU's Not Unix
JS	JavaScript
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
KR	Key Result
M	Month
MoSCoW	Must, Should, Could and Would
NA	Not Applicable
NFR	Non-Functional Requirement
NGSI	Next Generation Service Interfaces
NGSI-LD	NGSI for Linked Data
ODMS	Open Data Management Systems
PA	Public Administration
PM	Post-Meridiem
UC	Use Case
UI	User Interface
VSPL	Virtual SoPoLab
WP	Work Package

Executive Summary

The objective of the present document is to provide the first version of the functional and non-functional requirements of the URBANITE Ecosystem and its components as well as the methodology used for its gathering and prioritization. Functional requirements define the basic system functionalities while non-functional requirements define the performance attribute of a software system such as scalability, availability, reliability, etc.

URBANITE Ecosystem, to be developed in the context of WP5, will comprise the URBANITE components implemented (KR1-Virtual Social Policy Lab, KR3-Data Management Platform, KR4-Algorithms) and their integration into the URBANITE ecosystem (KR5). URBANITE ecosystem and the components and other results from the other work packages will be validated in the case studies defined in WP6.

This document includes the description of the iterative process followed to acquire the functional and non-functional requirements and the first version of the functional, use cases and non-functional elicited requirements. It also includes a set of traceability matrices, where the relationship between the different requirements is shown. In the annexes, the analysis of the known tools by technology providers that will serve as a baseline for the URBANITE components is included.

The main results of this document consist of the first version of the functional and non-functional requirements for the URBANITE ecosystem and the related components.

In this document, the requirements elicitation process is described as an iterative process where both the technology providers and use case providers' have participated.

All through the deliverable, the different requirements and the process to obtain them are explained. For the functional requirements a combined approach has been followed: 1) a top-down approach led by the technology provider partners, who have defined the first set of functional requirements and 2) a bottom-up approach where the needs of the Use Cases have been monitored (in close collaboration with WP6) and UC initial requirements have been extracted. For the Non-Functional Requirements, these have been detailed per component, including relevant aspects, such as performance, usability or resources needs for deployment.

All these requirements will serve for the continuous development and improvement of the URBANITE ecosystem, through the different releases, validation processes and reviews of the requirements.

1 Introduction

1.1 About this deliverable

This document presents the first version of the detailed requirements specification of the URBANITE ecosystem. By detailed specification, we refer to the definition of the requirements and explanation of the process to be followed to manage the lifecycle of the requirements, from their elicitation to their prioritization and validation. It covers both the description of the process followed to gather the requirements and the description of the requirements themselves.

The main objectives of WP5 are to:

1. Design the overall architecture of the URBANITE ecosystem
2. To identify URBANITE's technical, functional and non-functional requirements, based on the input from the use cases (WP6)
3. To set up the DevOps continuous integration environment and strategy
4. To integrate and validate the components of the URBANITE ecosystem
5. To design and develop the user interface (UI) of the URBANITE ecosystem

This document includes the first version of the initial set of functional and non-functional requirements elicitation (WP5 objective 2), the benchmarking of the identified tools that will serve as the basis for the URBANITE ecosystem components and an initial analysis of the prioritization of the requirements to be implemented in the different releases. The content of this deliverable will be updated in subsequent deliverables in the context of the WP5.

- D5.2-Detailed Requirements specification v2 (M20)

1.2 Document structure

The document is structured as follows. Section 2 presents the methodology followed in URBANITE for the elicitation, alignment and prioritization of the different requirements gathered for the design and implementation of the URBANITE ecosystem. This section also presents the generic processes that the different users of URBANITE ecosystem (URBANITE actors) can perform when using the different components. In the context of URBANITE, generic processes describe different high-level activities that a URBANITE user can execute when using the (some) URBANITE ecosystem components. A generic process is understood as a complete workflow to achieve a goal. It also includes who is responsible of that workflow and if there is a precondition to be fulfilled before the process can be triggered

In section 3, the current version of the URBANITE ecosystem requirements is presented, including the functional requirements, the use cases requirements, and the non-functional requirements.

Section 4 analyses how the different tools proposed by the technology partners for the basis of the URBANITE components fulfil the elicited Functional Requirements. The overview of each tool is presented in the Appendix.

Section 5 presents a set of alignment matrixes including: the alignment performed between the functional requirements of the components and the KRs described in the DoA [1], the alignment between the use cases requirements and the functional requirements and the prioritization of the different requirements based on both, the knowledge/planning of the technology providers in URBANITE and the requirements coming from the use cases.

The document concludes with the description of the next activities to be performed in the context of WP5.

2 Requirements management in URBANITE

2.1 Methodology for requirements elicitation

URBANITE follows a combined top-down and bottom-up approach with respect to the requirements gathering process. That is, on the one hand, generic functionalities of the URBANITE ecosystem are described to present the features that URBANITE aims to offer as its value proposition. On the other hand, the use cases expressed their requirements, namely, what they expect from URBANITE as end-users of the URBANITE ecosystem. Eventually, these two strands have to merge.

2.1.1 Requirements gathering and prioritization

The URBANITE ecosystem will include all the components for data management, analysis and support to the decision making that are going to be created/developed/implemented in the context of the URBANITE project. The first version of this deliverable will be updated in another deliverable of WP5 (D5.1-Detailed requirements specification v2).

Several sources will be used to elicit the requirements for the URBANITE ecosystem:

- **Requirements coming from the URBANITE action specification**: These requirements cover the functionalities described in the URBANITE action specification. The first version of these requirements has been described by the Technology providers partners (Fraunhofer, Tecnalia, JSI and Engineering) based on the URBANITE approach and high-level architecture description included in the URBANITE action.
- **Requirements coming from the Use Cases**: The Use Case will propose functionalities for the URBANITE ecosystem, so that the features offered can cover their needs.
- **Requirements coming from the co-creation sessions (SoPoLabs)**: It is expected that some requirements may be derived from the Sopolabs that will be conducted in the context of WP2. If relevant, these requirements will be considered for the URBANITE ecosystem and prioritize for their implementation.

The requirements covered in this D5.1 gathers the requirements coming from the URBANITE action specification. This document also includes the first draft version of the Use Cases requirements. The official set of Use Cases requirements will be reported in M12 in D6.1-URBANITE use cases requirements and evaluation methodology. Nevertheless, an initial version of these UCs requirements, derived from the ongoing activities in WP6 has been included in section 3.3 of the current document.

The next figure depicts the process followed for the gathering of the requirements in URBANITE according to the following legend: activities performed in this work package are marked in green, while activities marked in red are performed in work package 6 (use case validation).

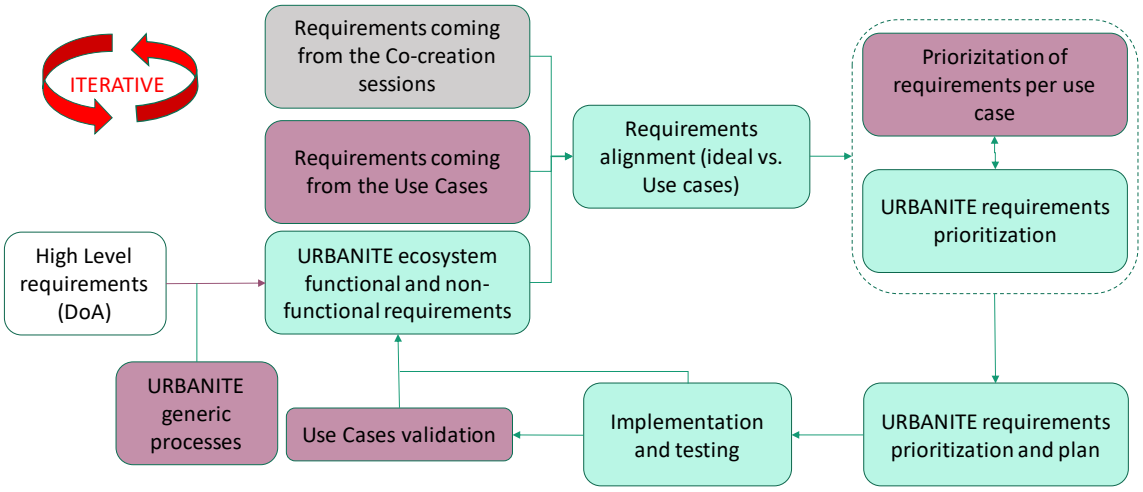

Figure 1. URBANITE process for requirements gathering and prioritization

The process followed in URBANITE for the elicitation of requirements is described as follows:

1. High-level requirements, that is, what URBANITE aims to offer, are elicited from what it is written in the Description of Action [2]. These requirements involve both functional and non-functional aspects that the URBANITE ecosystem should feature.
2. Definition of the generic URBANITE processes. “Generic processes” is understood as a complete workflow to achieve a goal. It also includes who is responsible of that workflow and if there is a precondition to be fulfilled before the process can be triggered.
3. The result of the activities described beforehand have been decomposed into technical (software) requirements, expressing both functionality needs and non-functionality aspects.
4. The technical partners then align the generic requirements elicited in previous steps with the requirements gathered by the use cases, and reformulated them, when needed, to end up with a consensus version.
5. Based on the final set of requirements (agreed in the previous step), use cases will prioritize them, taking into consideration their major interests. This prioritization will be reported and analyzed D5.2.
6. In parallel, technical partners will prioritize the requirements. This prioritization is needed because even if a requirement does not look like it adds anything to the system, it could be a baseline requirement that affects other requirements, and in the case, it was not implemented, the related functionality would not be delivered in a successful manner. Examples of this kind of requirements are, for instance, security requirements.
7. The outcome of both activities 5 and 6 results in a prioritization matrix (section 5.3 of the current document) that indicates for each release which requirements will be implemented and will therefore be able to be validated by the use cases.
8. Requirements will then be implemented into functionalities. In the context of this diagram, the implementation has been simplified and involves the architectural design, the coding, testing (unit and integration) and deployment. During this phase, and especially during the testing activities, new requirements or improved requirements may arise. These new requirements are carefully analyzed by the technical partners in order to avoid scope creep before deciding if they can or cannot be accepted.
9. Use cases validate the functionalities following the evaluation plan defined in D6.1 [3]. The evaluation will result in new requirements, as well as in updated versions of the

current requirements. Updated versions of requirements can include changes of the UI, while new requirements may involve new functionalities. As in step 8, these new requirements are carefully analyzed by the technical partners to avoid scope creep, before deciding if they will or will not be accepted.

Furthermore, several traceability matrixes are maintained.

On the one hand, in order to check if all URBANITE Key Results have associated requirements and not to forget any core functionality, the project maintains a traceability matrix between KRs and Requirements. An example of this traceability matrix is presented next. The final traceability matrix is reported in section 5.2.

Table 1. Traceability Matrix Key Result – Requirement (excerpt)

	KR1	KR3	KR4	KR5
VSPL.01	X			
VSPL.02	X			
VSPL.03	X			

Another traceability matrix maintained relates the requirements elicited and the ones prioritized by the use cases. An example of this traceability matrix is presented next. The final traceability matrix is reported in section 5.2.

Table 2. Traceability Matrix Functional-Use Case requirements (excerpt)

	Helsinki UC UC.HE.	**Bilbao UC** UC.BI.	**Messina UC** UC.ME.	**Amsterdam UC** UC.AM.
Data Harvesting				
DH.01	01,03,04,05-	01,02,03,05,06, 07,09	01,03,04,09,10,11	01,08,
DH.02	-	-	-	-
DH.03	-	-	-	-
DH.04	-	-	-	-

The last matrix maintained is the one that relates the requirements with the delivery date, allowing the developers to focus on the core aspects for each of the iterations.

Next, an excerpt image of the initial prioritization of the URBANITE functional requirements is shown. The final prioritization of the requirements is included in section 5.3.

Virtual SoPoLab	M15	M27	M33
VSPL.01	X		
VSPL.02	X		
VSPL.03	X		
VSPL.04	X		

Figure 2. Requirements prioritization matrix (excerpt)

2.1.2 Requirements documentation

Documenting requirements is a key issue in every software project. In the case of URBANITE, the requirements are defined to provide an understanding of what will URBANITE do, i.e., the functionalities.

For the prioritization of the requirements, from the point of view of the technology providers, the MoSCoW method [4] has been followed. The MoSCoW method allows to define clear priority levels while also determining which functionalities will be developed in each of the project iterations. The prioritization levels of MoSCOW can be defined as follows:

- M (Must): mandatory requirements. These requirements will be included definitely in the release.
- S (Should): requirements that should be included in the release or in the final version. The inclusion of this type of requirements must not affect the 'must' requirements and they will only be included in the case there is enough time and effort.
- C (Could): requirements that could be included, because they provide nice-to-have functionalities. These shall only be implemented when the M and S have been successfully implemented.
- W (Won't have): requirements that will not be included but could be delivered some time as additional or extended functionalities.

In the context of URBANITE Must and Should requirements are prioritized for the first versions of the components and considered all as Must requirements. Could requirements will be added in the final versions of the components.

This prioritization technique has allowed to prioritize the URBANITE requirements also taking into consideration the following constraints:

- Baseline functionalities (requirements prioritized as Must) will be implemented in the initial versions, while more complex functionalities depending on the basic ones will be implemented in subsequent releases.
- Functionalities which are core for the use cases have been prioritized for the initial releases (requirements prioritized as Must and Should), based on the use cases description and brainstorming sessions with the use cases. This prioritization of the functionalities will be accordingly updated as the description and needs of the use cases evolve.
- Functionalities with strong dependencies with other tools (i.e. interfaces related functionalities) (requirements prioritized as Must and Should) will be implemented in early stages of the project so that the integrated framework can be built up and the components can test the dependencies with other components.

In URBANITE, requirements are reported in a requirements document (this one) and are described as follows:

- **Requirement id**: unique identifier of the requirement
- **Short title**: short description of the requirement
- **Description**: more detailed description of the requirement. This attribute is especially relevant for the creation of the test cases.
- **Status**: Proposed / Accepted / Rejected / Implemented / Work in Progress /Finished
- **Priority**: Must have / Could have / Won't have
- **Related KR**: which URBANITE result is affected by this requirement

2.2 URBANITE Main processes

This section includes the generic processes that a user (defined as actors) of the URBANITE ecosystem can perform when using any of the components of the platform. These processes are based on the specific use cases, and cover the different scenarios defined in WP6; they cover the basic/generic functionalities that the URBANITE ecosystem will support.

- Process 1 –Select and combine available dataflows
- Process 2 – Visualize information in the map
- Process 3 - Data Analytics & events prediction
- Process 4 - Simulate (traffic) situations based on selected data and analyze obtained KPIs
- Process 5 - Collaborate in the Social Policy Lab

Sections 2.2.2, 2.2.3, 2.2.4, 2.2.5 and 2.2.6 provide the initial detailed description of these processes and the relationship with the use cases scenarios. These tables will be updated in the subsequent version of this deliverable (D6.2).

2.2.1 URBANITE actors

In this section, the URBANITE platform actors are described. These cover the definition of the different users of the URBANITE platform to perform any of the previously described:

- PA (Public Administration): This actor is the user from the public administration, usually the civil servants.
- Citizens: This actor is the citizen that is using any of the components in the platform.
- Platform administrator: This is the administrator of the platform who can install components, check the status of the included components, etc.

This roles, of course, are simplifications and generalizations of the real ones in the Use Cases.

It is envisioned that in the future, more roles are going to be identified as users of the URBANITE ecosystem. These will be included in the next version of the deliverable if necessary.

In the following table, the relationships between the Use Cases user's profiles defined in the context of WP6, the scenarios and the URBANITE platform generic actors is presented. A detailed description of the scenarios and the UCs users' profile will be presented in D6.1.

Table 3. Relationship between UC users' profiles and URBANITE platform actors.

UC users' profiles	UC	Scenario title	URBANITE platform actors
Urban mobility planner	Amsterdam	Analyzing and visualizing data for good policies	PA
Urban planner/ policy maker for various planning departments	Amsterdam	Combining dataflows for experts & data participation	PA
Project developer/ policy maker working for various departments	Amsterdam	Data for good	PA
Citizen (Expert/non-expert)	Amsterdam	Accountable and participatory tooling/ Data for good -2	Citizen
Traffic engineer (TE)	Helsinki	Analysing data for good planning	PA/Platform administrator

Traffic researcher	Helsinki	Toggling map layers, deep analysis	PA
Head of Traffic Planning	Helsinki	Toggling map layer, generating a 'big picture	PA
Urban planner planning traffic	Helsinki	Analysing data for good planning	PA
Urban planner	Helsinki	Analysing data for good planning	PA
Mobility Manager (MM)	Messina	Toggling map layers Toggling map layers 2 ("hybrid scenario")	PA/ Platform administrator
Noise Manager (NM)	Messina	Toggling map layers 1 ("public transportation flows") Toggling map layers 2 ("noise sensing") Toggling map layers 2 ("hybrid scenario")	PA/ Platform administrator
Sustainable Urban Mobility Director	Bilbao	Toggling Mobility Actions	PA
Transport and Mobility Assistant Managers	Bilbao	Implementing Mobility Actions	PA
Sustainable Urban Mobility Forum Management	Bilbao	SUMP Forum Management – 1	PA/ Platform administrator
Bilbao Mobility Forum Business and Citizens	Bilbao	SUMP Forum Management – 2	PA

2.2.2 URBANITE process 1: Select and combine available dataflows

Section	Description
Process id	P01
Process name	Data sources selection and combination
Objective	To select different data sources (and all the related information) and combine them.
Description	The data management platform will be designed in a data-format agnostic way. If data from new domains and sources are processable by the recommendation engine developed in WP4, specific adapters need to be integrated into the data management platform. Harvesting the data is only one aspect. Custom scripts may be required to homogenize and anonymize incoming data. This step takes place before storing data into the management platform. No single data structure can serve as a model for all kinds of data. Instead, domain-specific solutions are used, for example, DCAT-AP for metadata. Cross-linking multiple data sources requires case-by-case evaluation and manual intervention. As such, no automated process will analyse data and establish relationships.

Actors	PA
Pre-Conditions	Available source data to be represented in the maps. (e.g. in a UI, maps, etc.). Maps and GIS information available (for required data sources).
User scenarios	Combining dataflows for experts & data participation (Amsterdam) Data for good/ Data for good 2 (Amsterdam) Accountable and participatory tooling (Amsterdam) Toggling map layers (Helsinki) Toggling map layers (Messina) Toggling map layers 2 (“hybrid scenario”) (Messina) Toggling Mobility Actions (Bilbao)

2.2.3 URBANITE process 2: Visualize information in the map

Section	Description
Process id	P02
Process name	Data (traffic, public transportation, urban data (traffic lights)) representation in a map
Objective	To obtain a map of a specific area with different data represented on it.
Description	The application shows available data typology and data sources The user presses a button to access available information layers of the map, selects the desired ones (e.g. related to the traffic, noise, pollution, planned and development of works, etc.) and confirms the selection. The application updates the map, showing the information layers selected by the user and related analytics (if any) alongside it. Through proper commands, the user zooms the map to a specific district of the city to access detailed information about it or filter information by neighbourhood or district.
Actors	PA
Pre-Conditions	Available source data to be represented in the maps. Maps and GIS information available
User Scenarios	Analyzing and visualising data for good policies (Amsterdam) Toggling map layers (Helsinki) Analysing data for good planning (Helsinki) Toggling map layers (Messina) Toggling map layers 2 (“hybrid scenario”) (Messina) Toggling Mobility Actions (Bilbao) Implementing Mobility Actions (Bilbao)

2.2.4 URBANITE process 3: Data Analytics & events prediction

Section	Description
Process id	P03

Process name	Data (traffic, public transportation, urban data (traffic lights, etc.)) analysis and events prediction.
Objective	To obtain analytics of data and predict events based on this analysis. The following sub-objectives are considered: • Automatic data clustering: This process groups similar data according to user-defined criteria. This process can produce sets of data with similar characteristics. • Automatic dimensionality Reduction and relevant features identification: The goal is to remove/include from the data the features which are not relevant for an analysis or a purpose retaining the important ones. • Automatic short-term (based on historical data) or long-term (based on clustering and classification) predictions.
Description	The user will select the data sources. Based on the data selected, the possible data analysis will be shown: • Automatic data clustering: Once the data is selected, the user will press the Clustering button and will get groupings of such data with similar characteristics (clustering). An example of this process could be: *given a mobility data set, four clusters are identified, a) working day, b) Friday, c) Saturdays, d) Sundays From these relevant conclusions can be taken i.e. the traffic on bank holidays and August belongs to the same cluster, have similar profiles* • Relevant features identification: The user will select the goal (i.e. number of cars in a concrete point of the city map)to be used for the identification of the relevant features (i.e. the weather, bank holidays, school holidays, events in the city, etc.) impacting this goal. A list of the relevant features and their impact percentage will be shown to the user. An example of this process could be: *Which are the relevant features affecting the traffic in a concrete street?* • Events prediction: The user will select from the selected dataset the variable to be predicted (i.e. the number of cars). The prediction of this variable in certain (short or long term) time steps will be provided. Past predictions compared to the actual value can also be generated. An example of this process could be: *Given a data set of the historical recordings of the number of vehicles in a particular road at certain times, predict the number of vehicles in a future time.*
Actors	PA: traffic managers, urban mobility planners, traffic researchers, etc.
Pre-Conditions	Available source data in the required format to be analyzed.
User Scenarios	Analysing and visualising data for good policies (Amsterdam) Toggling map layers 1 (“public transportation flows”) (Messina) Toggling map layers 2 (“noise sensing”) (Messina) Toggling Mobility Actions (Bilbao) SUMP Forum Management – 1 (Bilbao) SUMP Forum Management – 2 (Bilbao)

2.2.5 URBANITE process 4: Simulate (traffic) situations based on selected data and analyze obtained KPIs

Section	Description
Process id	P04
Process name	Simulate (traffic) situations based on selected data and analyze the obtained KPIs
Objective	To simulate traffic (in general, pedestrian, vehicles, public transport…) flows and analyse the consequences on some KPIs (to be defined by the user).
Description	The application shows available simulation options (e.g. the situations to be simulated such as: opening a pedestrian street at certain times, location of electric charging stations, or bike-sharing points; the time, e.g. for certain days of the week or hours intervals, etc.) and parameters (KPIs) to be assessed (e.g. related to traffic volume, air quality, noise level, etc.). The user selects the options desired and confirms the selection. Based on the options selected by the user, the simulation runs, and various parameters are assessed (e.g. related to the traffic, noise, pollution, etc.). The application updates the map showing the information provided by the simulation algorithms (on different map layers) and the related parameters (KPIs) values evaluated alongside the map. An example of this process could be: *given a mobility data set for a city, and a specific time interval e.g. Friday from 3PM to 6PM, the simulation shows the traffic volume and air quality index after placing new bike-sharing points in the city,* thus helping the user to foresee different mobility situations in the city and their impact.
Actors	Citizens Pas
Pre-Conditions	Available source data to be represented in the maps. Maps and GIS information available
User Scenarios	Data for good -2 (Amsterdam) Implementing Mobility Actions (Bilbao)

2.2.6 URBANITE process 5: Collaborate in the Social Policy Lab

Section	Description
Process id	P05
Process name	Collaborate in the Policy Lab sessions

Objective	Participate in the Social Policy Lab sessions through the URBANITE online tools.
Description	The system will show the participants in the sessions, both the materials (documents, reports, graphics) necessary to document themselves for the session, as well as the threads and open discussions in each participating city as well as at the European level. Users will log into the system through a form and will be able to view the different functionalities: from participating in a forum (thread/debate) to solving a challenge that may have arisen through the debates. Users can participate and network in the discussions forums as well as upload solutions to challenges. They will also be able to consult the documentation that will be uploaded by the Virtual SoPoLab facilitators. Some of the sub-processes or activities envisioned in this process are: • Uploading content (reports, documents, graphics) onto the platform (PAs and Managers) • Create and Manage Challenges (PAs and Managers) • Upload solutions (Stakeholders) • Create and manages threads in the discussion forums (PAs and Managers) • Participate in each of the city's discussion forum (all users from that pilot city) • Participate in European Level discussion forum (all users)
Actors	All identified stakeholders PAs (Pilot City representatives) Virtual SoPoLab Managers (Tecnalia/WAAG)
Pre-Conditions	The facilitating users in each pilot city must be previously identified as they will have a different role capable of creating discussion threads and managing challenges. Each of the participating users must have access to the platform through a unique user and password.
User Scenarios	All

3 URBANITE ecosystem requirements

3.1 Functional requirements

The following sections collect the first set of functional requirements of the different modules of the URBANITE ecosystem. This first iteration is based on the analysis of the DoA and initial ideas based on the knowledge and expertise of the technology providers (components developers from the URBANITE consortium). This initial list will serve as the basis for the specification of the URBANITE processes and will be complemented by the Use Case providers requirements to be defined in WP6 as shown in section 2.1 from the current document.

ID's explanation:

- Virtual SoPoLab (VSPL)
- Data Harvesting (DH)
- Data Curation (DC)
- Data fusion/aggregation (DF)
- Data storage (DS)
- Data retrieval (DR)
- Data projection (DP)
- Data clustering (DCL)
- Self-Organizing Map (SOM)
- Regression (REG)
- Traffic simulation (TS)
- Advanced Visualisation (AV)
- Recommendation Engine (RE)
- Prediction (PRED)
- Policy Simulation and validation (PSV)
- Urbanite UI (UUI)

3.1.1 Virtual SoPoLAb

The virtual SoPoLAb component will provide the required functionalities to implement and support the different activities to be carried on for the co-creation sessions in the context of URBANITE.

Requirement id	**VSPL.01**
Short title	Enable co-creation
Description	VSPL should allow collaboration among its users, enabling co-creation approach. In the case of URBANITE, the co-creation sessions will be oriented to address and analyse the issues/barriers/ lack of trust in the usage of disruptive technologies in the public sector.
Status	Proposed
Priority	Must have DoA (part A – page 16 – T2.3)
Related KR	KR1

Requirement id	**VSPL.02**
Short title	Report needs

Description	The users of the VSPL should be able to report needs in the context of the analysis of the attitudes, trust and barriers in the use of disruptive technologies
Status	Proposed
Priority	Must have DoA (part A – page 16 – T2.3)
Related KR	KR1

Requirement id	**VSPL.03**
Short title	Launch Challenges
Description	The VSPL must allow to create challenges to solve the needs expressed related to the usage of disruptive technologies in the Public Sector.
Status	Proposed
Priority	Must have DoA (part A – page 16 – T2.3)
Related KR	KR1

Requirement id	**VSPL.04**
Short title	Report ideas (possible solutions)
Description	The users of the VSPL should be able to report ideas (possible solutions) to address the lack of trust, usage reticence, problems, needs of the usage of disruptive technologies in the Urban Mobility context.
Status	Proposed
Priority	Must have DoA (part A – page 16 – T2.3)
Related KR	KR1

Requirement id	**VSPL.05**
Short title	Evaluation of ideas (possible solutions)
Description	The VSPL must allow to evaluate proposed ideas to address the problems /needs related to the usage of disruptive technologies by the PAs for urban mobility.
Status	Proposed
Priority	Must have DoA (part A – page 16 – T2.3)
Related KR	KR1

Requirement id	**VSPL.06**
Short title	Selection of the ideas (possible solution)
Description	The VSPL must allow to select the best ideas to be refined and implemented in the context of the usage of disruptive technologies by PAs for urban mobility.

Status	Proposed
Priority	Must have DoA (part A – page 16 – T2.3)
Related KR	KR1

Requirement id	**VSPL.07**
Short title	Refinement of the ideas (possible solutions)
Description	The VSPL must allow to suggest refinements for selected ideas.
Status	Proposed
Priority	Must have DoA (part A – page 16 – T2.3)
Related KR	KR1

Requirement id	**VSPL.08**
Short title	Ask for Implementation
Description	The VSPL must allow to select ideas to be implemented in the context of the usage of disruptive technologies by the PAs for urban mobility.
Status	Proposed
Priority	Should have DoA (part A – page 16 – T2.3)
Related KR	KR1

Requirement id	**VSPL.09**
Short title	Resource hosting
Description	The VSPL must allow to host different kinds of resources created by the project, i.e. guidelines, methodologies and best practices.
Status	Proposed
Priority	Must have DoA (part A – page 16 – T2.3)
Related KR	KR1

Requirement id	**VSPL.10**
Short title	VSPL Forum
Description	The VSPL must allow the exchange of information between different participants of different nodes and cities.
Status	Proposed
Priority	Must have DoA (part B – page 6 - 25)
Related KR	KR1

3.1.2 Data Harvesting Component

The data harvesting component is the first part of the chain of processing data. Its task is to download data from data sources. This data can be heterogeneous with regards to the APIs it is fetched from, its domain, and format/structure. Also, data may change over time. Therefore, data must be fetched from the sources at regular intervals. Since the nature of the required data, as well as their availability, may change over time, the harvesting component should be designed in a modular fashion. This approach allows the harvesting of new data sources by extending the component at a later point in time.

Requirement id	DH.01
Short title	Data Harvesting from heterogeneous data sources
Description	The harvesting component will retrieve data from various sources (municipal services, open data portals, GIS, city private service providers) of varying formats (e.g. JSON, XML) from different data sources (e.g. open/private data portals, GIS data source, raw data from APIs or data coming from sensors).
Status	Proposed
Priority	Must have DoA (part A – page 20 – T3.1 ; part B – page 13)
Related KR	KR3

Requirement id	DH.02
Short title	Pagination
Description	Data Harvester should allow pagination of large amounts of data. This means that in case some data source APIs cannot provide data in bulk, the harvesting component should be able to fetch only chunks of limited size until all data has been harvested.
Status	Proposed
Priority	Must have DoA (part B – page 13)
Related KR	KR3

Requirement id	DH.03
Short title	Data Harvesting extensibility
Description	Data Harvester should be extensible with new connectors if new, unsupported, data sources are discovered.
Status	Proposed
Priority	Must have DoA (part A – page 20 – T3.1)
Related KR	KR3

Requirement id	DH.04
Short title	Data Harvesting supported protocols
Description	Data harvester must support at least HTTP(S) and MQTT protocol to fetch the data.
Status	Proposed
Priority	Should have DoA (part B – page 13)
Related KR	KR3

Requirement id	DH.05
Short title	Scheduled data fetching
Description	For client/server APIs, the harvester will download data from the configured APIs at recurring intervals of varying length (e.g. daily, weekly). The schedule will depend on the volatility of data. For example, weather data will change more frequently than map data highlighting current road construction work.
Status	Proposed
Priority	Must have DoA (page 11)
Related KR	KR3

3.1.3 Data Curation Component

Once the data is harvested, it needs to be processed in a variety of ways. This is required as the raw data may not be fit for immediate analysis, the reasons ranging from privacy issues to incompatibilities regarding the format. First, regulations and laws must be met by means of anonymization. Only then can data-wrangling commence. This is the process of cleaning raw data and subsequent transformation into formats most suitable for further analysis. Additionally, enriching data with metadata and performing quality checks can aid analysis even further.

Requirement id	DC.01
Short title	Data transformation after harvest
Description	The harvested data may not be in a format and/or structure suitable for data storage. In this case, the data will need to be transformed in an automated way.
Status	Proposed
Priority	Must have GA DoA part B (page 13)
Related KR	KR3

Requirement id	DC.02
Short title	Data Cleaning
Description	Data curation module should be able to clean the data coming from the harvester eliminating duplicates or error.
Status	Proposed
Priority	Must have DoA (part B - page 13)
Related KR	KR3

Requirement id	DC.03
Short title	Data Annotation

Description	Data curation module should add an annotation in the form of metadata to data to help the analysis. This metadata will be included in the data itself.
Status	Proposed
Priority	Must have DoA (part B - page 13)
Related KR	KR3

Requirement id	**DC.04**
Short title	Data anonymization
Description	This module shall anonymize or pseudonymize data. Data anonymization could be done at the source or before storing it, depending on the use case. In any case, URBANITE platform will provide the anonymization functionality for users (UCs) to use it before the data is uploaded/used by the URBANITE platform
Status	Proposed
Priority	Must have GA DoA part B (page 13, 27)
Related KR	KR3

Requirement id	**DC.05**
Short title	Data validation and quality check
Description	The data curation module must be able to validate the data provided by the data harvesting module and its quality based on a defined format
Status	Proposed
Priority	Must have DoA (page 27) Part A (page 20)
Related KR	KR3

Requirement id	**DC.06**
Short title	Data Interoperability
Description	Data curation module should provide data transformation functionalities to transform cleaned and annotated data to common semantics and data models to guarantee interoperability. It is important to note that there will not be one single common format that all data will be transformed into. Instead, established formats within the various domains will be targeted for transformation.
Status	Proposed
Priority	Must have DoA (page 13 – page 26 – part B page 32 (Interoperability by default))
Related KR	KR3

Requirement id	**DC.07**

Short title	Data license support
Description	The module must check the data licenses and provide understandable information to the owners and the data user. For combined data sets with different licenses, it detects possible compatibility issues and informs users how to use and share the data.
Status	Proposed
Priority	Must have
Related KR	DoA (part B – page 27)

Requirement id	**DC.08**
Short title	Pipeline between data harvesting and curation modules
Description	The data curation module must provide an API (REST service or MQTT endpoint) so that the data harvesting module can forward the data that has been retrieved.
Status	Proposed
Priority	Nice to have
Related KR	KR3

3.1.4 Data Fusion/Aggregation Component

Data aggregation is the process of gathering data and presenting it in a summarized format. The data may be gathered from multiple data sources to combine these data sources into a summary for data analysis. Examples of aggregate data from a website are statistics on customer demographic and behaviour metrics, such as average age or number of transactions.
Data fusion is the process of integrating multiple data sources to produce more consistent, accurate, useful information and sophisticated models than that provided by any individual data source. This means that the result of the data fusion process, once the N different datasets are integrated, should be worth more than the sum of each single dataset's result. Examples of data fusion are combining physical tracking data with environmental data, combining client identifier data with purchase history, etc.

Requirement id	**DF.01**
Short title	Aggregation
Description	The component should allow to aggregate curated data coming from different data sources if needed.
Status	Proposed
Priority	Must have DoA (part A – page 20 – T3.3)
Related KR	KR3

Requirement id	**DF.02**
Short title	Deduplication
Description	The component should allow the deduplication of the data.
Status	Proposed
Priority	Must have DoA (part A – page 20 – T3.3)
Related KR	KR3

Requirement id	DF.03
Short title	Data mapping
Description	The data should be mapped into EU vocabularies
Status	Proposed
Priority	Must have DoA (part A – page 20 – T3.3)
Related KR	KR3

Requirement id	DF.04
Short title	MetaData mapping
Description	The metadata should be mapped into DCAT-AP metadata
Status	Proposed
Priority	Must have DoA (part A – page 20 – T3.3)
Related KR	KR3

3.1.5 Data Storage Component

Once the data that is fetched by the harvesting component is curated and transformed, it needs to be persisted for further analysis. This task is handled by the data storage component. It acts as a database wrapper that handles CRUD operations, providing APIs for both, manipulating and reading data.

Requirement id	DS.01
Short title	Big data storage
Description	The harvested data should be persisted to a big data capable storage solution.
Status	Proposed
Priority	Must have DoA (page 9)
Related KR	KR3

Requirement id	DS.02
Short title	DCAT-AP compliance
Description	The data storage component should be able to process and store DCAT-AP compliant metadata.
Status	Proposed
Priority	Must have GA DoA part B (page 20,27)
Related KR	KR3

3.1.6 Data Retrieval Component

The data retrieval component will allow obtaining and querying data from the different URBANITE's repositories, i.e. semantic, relational and NoSQL repositories. This component will offer APIs to query data from a single repository or combined from multiple ones.

Requirement id	DR.01
Short title	Data Retrieval
Description	The data retrieval component must expose API to retrieve and query the data stored in the different repositories.
Status	Proposed
Priority	Must have DoA (part A – page 20 – T3.3)
Related KR	KR3

Requirement id	DR.02
Short title	Data Hub
Description	The metadata stored in the repositories should be accessible through a data hub in a uniform way taking advantage of DCAT-AP standard and related profile.
Status	Proposed
Priority	Must have DoA (part B – page 27)
Related KR	KR3

3.1.7 Data Projection component

Data projection component will provide dimensionality reduction methods for a better understanding and interpretation of the data. The inputs for this component will be tabular data and GIS data.

Requirement id	DP.01
Short title	Projection
Description	Data projection component will provide dimensionality reduction methods for a better understanding and interpretation of the data.
Status	Proposed
Priority	Must have
Related KR	KR4

3.1.8 Data Clustering component

Data clustering component will provide clustering methods that will enable clustering the data based on user-defined set of attributes. The inputs for this component will be tabular data and GIS data.

Requirement id	DCL.01
Short title	Clustering
Description	Data Clustering component will provide methods that will identify groups of similar objects in the data (based on user-defined attributes) and interactively present them to the user.
Status	Proposed
Priority	Must have
Related KR	KR4

3.1.9 Self-Organizing Map component

The Self-Organizing Map component will allow inspecting multidimensional data in a highly intuitive way. Highly dimensional data points will be nonlinearly projected into a two- or three-dimensional space that will reflect the topological shape of the data. This enables the user to visualize the data and identify clusters in a lower dimension.

Requirement id	SOM.01
Short title	Self-Organizing Map
Description	The Self-Organizing Map will provide the user with a visual topological representation of the data, highlighting potential clusters.
Status	Proposed
Priority	Must have
Related KR	KR4

3.1.10 Regression component

The regression component will provide the ability to see relationships among different variables.

Requirement id	REG.01
Short title	Regression
Description	The regression component will enable the user to investigate the relationship between different variables, and to actively search for causal relations in the data.
Status	Proposed
Priority	Must have
Related KR	KR4

3.1.11 Traffic Simulation component

The traffic simulation component will provide city traffic simulations based on the collected data and the city layout of both current situations and hypothetical ones.

Requirement id	TS.01
Short title	Current traffic simulation
Description	Traffic Simulation component will provide urban traffic simulation based on the collected data, describing the traffic flow locally, for specific parts of interest in the city and combining it in a hierarchical manner.
Status	Proposed
Priority	Must have
Related KR	KR4

Requirement id	TS.02
Short title	Hypothetical traffic simulation

Description	Traffic Simulation component will provide the ability to simulate hypothetical situations and the effects of different measures.
Status	Proposed
Priority	Must have
Related KR	KR4

3.1.12 Advanced Visualization component

Visualization component will provide visualization for mobility patterns, highlighting important events and results of policies.

Requirement id	**AV.01**
Short title	Data Visualization
Description	The harvested data should be visualized.
Status	Proposed
Priority	Must have GA DoA part B (page 11)
Related KR	KR3

Requirement id	**AV.02**
Short title	Mobility pattern visualization
Description	The component must allow to visualize the analysis results on a combination of map layers, heat maps, traffic flow graphics and other kinds of visualization.
Status	Proposed
Priority	Must have DoA (part A – page 25 – T4.4)
Related KR	KR4
Requirement id	**AV.03**
Short title	Visualization interactions
Description	The component must allow users to interact with the visualized data by, for instance, zooming, highlighting and displaying additional information.
Status	Proposed
Priority	Must have DoA (part A – page 25 – T4.4)
Related KR	KR4

3.1.13 Recommendation Engine component

The Recommendation Engine component will provide the user with a decision support system. The main objective of the Recommendation Engine is to identify and predict important or problematic events related to mobility and provide suggestions to tackle the identified issue.

Requirement id	**RE.01**
Short title	Recommendation Engine component

Description	The Recommendation Engine will provide suggestions to tackle the potential problems in the city traffic. This component will also provide support for identifying possible policies that tackle events based on specific criteria.
Status	Proposed
Priority	Must have DoA Part A (page 25)
Related KR	KR4

Requirement id	**RE.02**
Short title	Event prediction
Description	The recommendation engine must identify and predict events related to mobility (samples could be congestion situations, high-emission scenarios, unbalanced modal share, etc.), based on the analysis of existing models and/or simulated data. Such analysis will be supported by the previously mentioned component as those related to regression, clustering, simulation or additional ones.
Status	Proposed
Priority	Must have DoA (part A – page 24 – T4.2)
Related KR	KR4

Requirement id	**RE.03**
Short title	Suggestions
Description	The recommendation engine should provide support and suggestions to the policymakers for identifying possible policies that tackle identified problems and undesired events related to mobility based on specific criteria. Effective hierarchical multi-criteria decision models based on aggregated data and a rule-based reasoning will be adopted.
Status	Proposed
Priority	Must have DoA (part A – page 24 – T4.2)
Related KR	KR4

3.1.14 Prediction component

The prediction component will provide the ability to predict problematic events and the effects of new policies in the traffic flow.

Requirement id	**PRED.01**
Short title	Prediction
Description	The prediction component will provide an engine to produce a prediction for a traffic/mobility variable defined as a time series considering a series of time defined features.
Status	Proposed
Priority	Must have GA page 7, 9, 14,29,32 of the DoA part B

Related KR	KR4

3.1.15 Policy Simulation and Validation component

The Policy Simulation and Validation component will enable validation of policy by scoring the simulation and prediction results based on the user-defined KPIs.

Requirement id	**PSV.01**
Short title	Policy support system
Description	The component should support policy-makers for identifying possible policies that tackle events based on specific criteria.
Status	Proposed
Priority	Must have DoA (part A – page 24 – T4.2)
Related KR	KR4

Requirement id	**PSV.02**
Short title	Policy Changes evaluation
Description	The component should predict and classify traffic flow changes based on the changes in the new policies.
Status	Proposed
Priority	Must have DoA (part A – page 25 – T4.3)
Related KR	KR4

Requirement id	**PSV0.3**
Short title	KPI (criteria) selection
Description	User must be able to select the defined KPIs to evaluate policies.
Status	Proposed
Priority	Must have DoA (part A – page 25 – T4.3)
Related KR	KR4

Requirement id	**PSV0.4**
Short title	Validation engine – score assign
Description	The component must assign a score to each policy to help the decision-making process
Status	Proposed
Priority	Must have DoA (part A – page 25 – T4.3)
Related KR	KR4

Requirement id	**PSV0.5**
Short title	Policy Selection
Description	Policy-makers will be able to make an informed decision about which policies should be deployed in the city.
Status	Proposed
Priority	Must have

	DoA (part A – page 25 – T4.3)
Related KR	KR4

3.1.16 URBANITE UI Component

This component will guarantee the uniform access to every URBANITE tool thanks to a responsive User Interface. All the components developed in the URBANITE ecosystem that will provide specific graphical interfaces will be made accessible through this component.

Requirement id	**UUI.01**
Short title	Uniform Access
Description	The UI must provide uniform access to URBANITE tools and components.
Status	Proposed
Priority	Must have DoA (part A – page 28 – T5.4)
Related KR	KR5

Requirement id	**UUI.02**
Short title	DSS Integration
Description	The UI must be integrated with the DSS visualization capabilities.
Status	Proposed
Priority	Must have DoA (part A – page 28 – T5.4)
Related KR	KR5

Requirement id	**UUI.03**
Short title	User Profiling
Description	The UI must support different user profiles, offering different functionalities for administrators and the final user.
Status	Proposed
Priority	Must have DoA (part A – page 28 – T5.4)
Related KR	KR5

Requirement id	**UUI.04**
Short title	Responsive UI
Description	The UI must be responsive to support different types of devices.
Status	Proposed
Priority	Must have DoA (part A – page 28 – T5.4)
Related KR	KR5

Requirement id	**UUI.05**
Short title	UI Personalisation
Description	The UI must allow personalization through dynamic dashboards.
Status	Proposed
Priority	Must have DoA (part A – page 28 – T5.4)

Related KR	KR5

3.2 Non-functional requirements

This section covers the initial definition of the non-functional requirements (NFRs) of the URBANITE ecosystem and their components. With this respect, relevant non-functional aspects have been analyzed from each functional component perspective. The main outcomes are summarized below and will establish the basis for the definition of the URBANITE non-functional requirements as a whole.

To this end, we have selected the following non-functional requirements and related metrics, as these are the ones more relevant in the context of URBANITE ecosystem and the specific components:

- Performance: Response time in terms of Processing time and Data access time. (Average values)
- Security: Issues related to authentication, authorization and anonymization have been analyzed.
- Deployment needs: In terms of resources size, software needs, etc
- Availability: Percentage of time available.

These non-functional requirements have been analyzed per component (when relevant) and will be considered when studying and proposing the deployment configuration and requirements for the URBANITE ecosystem, both as the reference platform and for the instances to be deployed in the URBANITE UCs. These requirements are expressed from the perspective of the related component; nevertheless, they will also be impacted by the NFR of the underlying technologies. That is why these NF requirements will be further analyse when studying the deployment requirements for the URBANITE ecosystem and their components. An overview of this analysis is presented in the following table and will be updated in the D6.2.

Table 4 Overview of the Non-Functional Requirements per component in URBANITE

URBANITE component	Performance	Security	Deployment	Availability
Virtual SoPoLab	< 1,5 sec	Only authenticated and authorized users can access the VSPL	Ubuntu 18.04LTS - Ruby v2.5.1 - PostgreSQL v10 8GB Ram - HD 40GB - 2CPU	100
Data Harvesting	> 5 datasets/sec	Must be able to query secured APIs	Docker	95
Data Curation	NA	Must be capable of anonymization	Docker	95
Data Fusion/Aggregation			Docker	95
Data Storage		Only authenticated and authorized users can read/write data	Docker, Apache Hadoop, Keycloak	100
Data projection	< 1,5 sec	Only authenticated and authorized users can trigger processing	Docker, WEKA	95
Data clustering	< 1,5 sec	Only authenticated and authorized users can trigger processing	Docker, WEKA	95
Self-Organizing Map	< 1,5 sec		Docker, WEKA	95
Regression	< 1,5 sec	Only authenticated and authorized users can trigger processing	Docker, WEKA	95
Traffic simulation	off-line (up to a day)	Only authenticated and authorized users can trigger processing	Docker, Java	95
Advanced Visualisation	< 1,5 sec		Angular - Spring Boot - Docker	95

Recommendation Engine	1 minute	Only authenticated and authorized users can trigger processing	Docker, DEXi	95
Prediction	< 1,5 sec	Only authenticated and authorized users can trigger processing	Docker, WEKA	95
Policy Simulation and validation	1 minute	Only authenticated and authorized users can trigger processing	Docker, WEKA	95
Urbanite UI	< 1,5 sec	Only authenticated and authorized users can access the UI	Angular - Spring Boot - MongoDB - Docker	100

3.3 Use Cases requirements

This section covers the definition of the Use Cases requirements of the URBANITE ecosystem and its components. For each city, an initial analysis has been performed regarding the needs and features required in the Use Case from a high-level perspective in the context of WP6. For that, tools like the value proposition canvas have been utilized. The results are summarized here and will be detailed in D6.2.

These requirements are foreseen to be extended and consolidated as part of the WP6 job and will be reported on the future deliverables of the WP6 as the User Scenarios are being revised at the moment of editing this deliverable.

3.3.1 Helsinki Use Case

Requirement id	UC.HE.01
Description	The application MUST provide a map
Status	Proposed
Priority	Must have
Source	Helsinki

Requirement id	UC.HE.02
Description	The application MUST allow the user to zoom on areas and districts of the map
Status	Proposed
Priority	Must have
Source	Helsinki

Requirement id	UC.HE.03
Description	The map MUST allow the user to visualise thematic layers (e.g. parking, transport, bike-sharing, planned development vs actual, urban planning, traffic planning, etc.)
Status	Proposed
Priority	Must have
Source	Helsinki

Requirement id	UC.HE.04
Description	Thematic layers and base maps SHOULD be enriched with a description
Status	Proposed
Priority	Should have
Source	Helsinki

Requirement id	UC.HE.05
Description	The application SHOULD provide pre-packaged simulations (e.g. impact of urban planning on the traffic)
Status	Proposed
Priority	Should have
Source	Helsinki

Requirement id	UC.HE.06
Description	The application SHOULD provide a wizard to create charts

Status	Proposed
Priority	Should have
Source	Helsinki

Requirement id	**UC.HE.07**
Description	The application MUST provide a unique point of access to the data offered by the different departments
Status	Proposed
Priority	Must have
Source	Helsinki

Requirement id	**UC.HE.08**
Description	The application SHOULD automatize part of the analysis performed on the collected data (e.g. extract relevant information and provide it in a more usable manner)
Status	Proposed
Priority	Should have
Source	Helsinki

Requirement id	**UC.HE.09**
Description	The application SHOULD allow the user to download collected raw data
Status	Proposed
Priority	Should have
Source	Helsinki

Requirement id	**UC.HE.10**
Description	The application SHOULD highlight relevant information reducing the time to search it
Status	Proposed
Priority	Should have
Source	Helsinki

3.3.2 Bilbao Use Case

Requirement id	**UC.BI.01**
Description	The application MUST be able to show different charts and graphs in the same view
Status	Proposed
Priority	Must have
Source	Bilbao

Requirement id	**UC.BI.02**
Description	The application MUST provide a data catalogue to allow the user to discover the data (e.g. by transport mode).
Status	Proposed
Priority	Must have
Source	Bilbao

Requirement id	UC.BI.03
Description	The application COULD check if data is updated (e.g. check time constraints)
Status	Proposed
Priority	Could have
Source	Bilbao

Requirement id	UC.BI.04
Description	The application SHOULD allow the user to define custom KPIs
Status	Proposed
Priority	Should have
Source	Bilbao

Requirement id	UC.BI.05
Description	The application MUST provide a unique point of access to the data collected from scattered data sources
Status	Proposed
Priority	Must have
Source	Bilbao

Requirement id	UC.BI.06
Description	The application SHOULD allow to perform pre-processing of collected data
Status	Proposed
Priority	Should have
Source	Bilbao

Requirement id	UC.BI.07
Description	The application SHOULD allow to setup and execute simulations
Status	Proposed
Priority	Should have
Source	Bilbao

Requirement id	UC.BI.08
Description	The application SHOULD allow the users to exchange information
Status	Proposed
Priority	Should have
Source	Bilbao

Requirement id	UC.BI.09
Description	The application MUST harmonise the collected data according to common and well-defined data models
Status	Proposed
Priority	Must have
Source	Bilbao

3.3.3 Messina Use Case

Requirement id	UC.ME.01
Description	The application MUST provide the user with map layers to "visualise" information related to frequent transfer (e.g. well-trodden roads), pollution (noise, air, electromagnetic), traffic levels, road network quality
Status	Proposed
Priority	Must have
Source	Messina

Requirement id	UC.ME.02
Description	The application SHOULD allow the user to create customized dashboards
Status	Proposed
Priority	Should have
Source	Messina

Requirement id	UC.ME.03
Description	The application SHOULD offer simulations/analysis to support the identification of multimodal transportation paths
Status	Proposed
Priority	Should have
Source	Messina

Requirement id	UC.ME.04
Description	The application SHOULD offer simulations/analysis to estimate the number of transportation (vehicles) and driver available in a specific moment
Status	Proposed
Priority	Should have
Source	Messina

Requirement id	UC.ME.05
Description	The application COULD allow the user to make publicly accessible selected charts, graphs, map layers
Status	Proposed
Priority	Could have
Source	Messina

Requirement id	UC.ME.06
Description	The application COULD allow the user to enrich publicly accessible selected charts, graphs, map layers with a description
Status	Proposed
Priority	Could have
Source	Messina

Requirement id	UC.ME.07
Description	The application SHOULD allow the users to share map layers, charts, graphs, etc
Status	Proposed
Priority	Should have
Source	Messina

Requirement id	UC.ME.08
Description	The application MUST provide a unique point to different departments to access collected data
Status	Proposed
Priority	Must have
Source	Messina

Requirement id	UC.ME.09
Description	The application SHOULD support the user to identify problems in public transportation (e.g. delays, broken vehicles, traffic jams)
Status	Proposed
Priority	Should have
Source	Messina

Requirement id	UC.ME.10
Description	The application SHOULD support the user to identify roads and streets that need maintenance interventions
Status	Proposed
Priority	Should have
Source	Messina

Requirement id	UC.ME.11
Description	The application SHOULD support the user to identify intervention for public safety
Status	Proposed
Priority	Should have
Source	Messina

3.3.4 Amsterdam Use Case

Requirement id	UC.AM.01
Description	The application MUST allow to setup and execute analysis/simulations
Status	Proposed
Priority	Must have
Source	Amsterdam

Requirement id	UC.AM.02

Description	The application SHOULD allow to access without restrictions to specific datasets in a consistent and not fragmented way
Status	Proposed
Priority	Should have
Source	Amsterdam

Requirement id	**UC.AM.03**
Description	The application SHOULD allow to publish results of analysis/simulations
Status	Proposed
Priority	Should have
Source	Amsterdam

Requirement id	**UC.AM.04**
Description	The application SHOULD allow to publish datasets
Status	Proposed
Priority	Should have
Source	Amsterdam

Requirement id	**UC.AM.05**
Description	The application SHOULD allow to enrich published datasets and results of analysis/simulations with a description (about the importance of mobility data and data sovereignty)
Status	Proposed
Priority	Should have
Source	Amsterdam

Requirement id	**UC.AM.06**
Description	The application SHOULD allow the user to enrich metadata of collected data to facilitate the identification of relevant data (e.g. data related to bike usage)
Status	Proposed
Priority	Should have
Source	Amsterdam

Requirement id	**UC.AM.07**
Description	The application SHOULD allow the users to exchange information
Status	Proposed
Priority	Should have
Source	Amsterdam

Requirement id	**UC.AM.08**
Description	The application COULD allow the users to combine different types of data (e.g. energy, air quality, waste, health, wellbeing, etc..)
Status	Proposed
Priority	Could have
Source	Amsterdam

Requirement id	**UC.AM.09**

Description	The application SHOULD provide a unique point of access to the data collected from scattered data sources
Status	Proposed
Priority	Should have
Source	Amsterdam

4 Benchmarking of existing tools

This matrix represents the coverage of the different functional requirements by the tools that have been analyzed in the context of WP5. The green colour represents that the functional requirement is fully covered by the tool, the yellow colour means that the functional requirement is partially covered, and the empty cell represents that the functional requirement is not covered by the tool. Even if those requirements marked as green are supporting the functionality defined, adaptations and changes on the tools will still be needed so these baselines tools to can be integrated into the URBANITE ecosystem and interact with all the required components. The information on the integration needs or requirements is included in the overview of each tool in the Appendix and it will also be used to support the URBANITE partners on selecting the baseline tools and technologies.

The analysis of this matrix and other relevant information (i.e. licenses, integration needs) has served for the initial selection of the baseline technologies to be used and extended for some of the URBANITE ecosystem components. The table also provides an overview of the functionalities that will be built from scratch.

The overview of the tools is presented in the Appendix.

Table 5.Overview of the Functional Requirements coverage by the proposed baseline tools

RQ ID	ARX	Piveau Consus	Piveau hub	Piveau UI	Piveau incognito	SUMO	MATsim	WEKA	Orange 3	Idra	Data model Mapper	CaPe[1]	Digital Enabler Platform	Micro Proxies[2]	Decidim
VSPL.01															
VSPL.02															
VSPL.03															
VSPL.04															
VSPL.05															
VSPL.06															
VSPL.07															
VSPL.08															
VSPL.09															
VSPL.10															
DH.01															
DH.02															
DH.03															
DH.04															
DH.05															
DC.01															
DC.02															
DC.03															
DC.04															
DC.05															
DC.06															
DC.07															
DC.08															

[1] Still under review how this tool can support different requirements.
[2] Still under review how this tool can support different requirements.

DF.01															
DF.02															
DF.03															
DF.04															
DS.01															
DS.02															
DR.01															
DR.02															
DP.01															
DCL.01															
SOM.01															
REG.01															
TS.01															
TS.02															
AV.01															
AV.02															
AV.03															
RE.01															
RE.02															
RE.03															
PRED.01															
PSV.01															
PSV.02															
PSV.03															
PSV.04															
PSV.05															
UUI.01															
UUI.02															
UUI.03															
UUI.04															
UUI.05															

5 Requirements alignment and prioritization

This section presents the alignment between the different types of requirements elicited in previous sections. This alignment is being updated continuously.

5.1 Functional requirements and KRs alignment matrix

The functional requirements are defined in section 3.1. Following the list of Key Results (from the DoA) is presented:

- KR1: Social Policy Lab
- KR3: URBANITE Data Management Platform
- KR4: URBANITE Algorithms and simulations
- KR5: URBANITE ecosystem

As a result, it can be affirmed that all the elicited Functional Requirements are assigned to one/or more specific KR.

Table 6. Functional requirements and KRs alignment matrix

	KR1	KR3	KR4	KR5
VSPL.01	X			
VSPL.02	X			
VSPL.03	X			
VSPL.04	X			
VSPL.05	X			
VSPL.06	X			
VSPL.07	X			
VSPL.08	X			
VSPL.09	X			
VSPL.10	X			
DH.01		X		
DH.02		X		
DH.03		X		
DH.04		X		
DH.05		X		
DC.01		X		
DC.02		X		
DC.03		X		
DC.04		X		
DC.05		X		
DC.06		X		
DC.07		X		
DC.08		X		
DF.01		X		
DF.02		X		
DF.03		X		
DF.04		X		
DS.01		X		
DS.02		X		
DR.01		X		

DR.02		X		
DP.01			X	
DCL.01			X	
SOM.01			X	
REG.01			X	
TS.01			X	
TS.02			X	
AV.01			X	
AV.02			X	
AV.03			X	
RE.01			X	
RE.02			X	
RE.03			X	
PRED.01			X	
PSV.01			X	
PSV.02			X	
PSV.03			X	
PSV.04			X	
PSV.05			X	
UUI.01				X
UUI.02				X
UUI.03				X
UUI.04				X
UUI.05				X

5.2 Functional and use cases alignment matrix

This section shows the alignment between the functional requirements and the Use Cases.

In theory, it could be thought that for each functional requirement defined -let's call it FRx-, should be at least one-Use Case that requires FRx for its correct implementation (otherwise, the requirement FRx can be considered "useless"). This straightforward conclusion needs to be analyzed in detail, because URBANITE ecosystem needs to support not only current UCs but also future users of the platform and for this reason some requirements are already envisioned (i.e. support for standardized protocols or models, extension capabilities etc.). In addition, some

At the same time, if some Use Case requirement is not covered by any functional requirement, it means that we are missing some extra functional requirements not yet reflected in the table. This case needs to be further analyzed too, as some UCs needs may be out of the scope of the URBANITE project.

This table will serve for the components developers to identify potential missing functionalities form the UCs needs perspective and to align both the bottom-up perspective and the top-down approach for the elicitation of the requirements. The results of the detailed analysis of the table will be provided in the next version of the deliverable D5.2.

In the case of the Virtual SoPoLab requirements, as the purpose of this component is different (it is supporting the activities in WP2), it has not been included in the table, but, all the Use Cases will use this component during the implementation of the use cases.

Table 7. Functional and UCs alignment matrix

	Helsinki UC UC.HE.	**Bilbao UC** UC.BI.	**Messina UC** UC.ME.	**Amsterdam UC** UC.AM.
Data Harvesting				
DH.01	01,03,04,05	01,02,03,05,06,07,09	01,03,04,09,10,11	01,08,
DH.02	-	-	-	-
DH.03	-	-	-	-
DH.04	-	-	-	-
DH.05	-	-	-	-
Data Curation				
DC.01	01,03,04,05	01,02,03,05,06,07,09	01,03,04,09,10,11	01,08
DC.02	01,03,04,05,	01,02,03,05,06,07,09	01,03,04, 09,10,11	01,08
DC.03	01,03,04,05	01,02,03,05,06,07,09	01,03,04,09,10,11	01,08
DC.04	2	-	05	03, 04
DC.05	01,03,04,05	01,02,03,05,06,07,09	01,03,04,09,10,11	01,08
DC.06		-	-	-
DC.07	-	-	05	03,04
DC.08	-	-	-	-
Data fusion/aggregation				
DF.01	01,03,04,05	01,02,03,05,06, 07	01,03,04, 09,10,11	01,08
DF.02	01,03,04,05	01,02,03,05,06, 07	01,03,04,09,10,11	01,08
DF.03	(4)	09	-	-
DF.04	-	09	-	-
Data storage				
DS.01	01	-	-	-
DS.02	-	09	-	-
Data retrieval				
DR.01	03,04,05	01,02,03,05,06,07,09	01,03,0409,10,11	01, 08
DR.02	(3)	-	-	-
Data projection				
DP.01	-	-	-	-

Data clustering				
DCL.01	03,04, 10	-	01, 09,10,11	08
Self-Organizing Map				
DCL.01	-	01	01	08
Regression				
REG.01	-	-	-	-
Traffic simulation				
TFS.01	01, 03,04,05-	01,07	03,04, 09,10,11	01
TFS.02	01, 03,04,05	01,07	03,04, 09,10,11	01
Advanced Visualisation				
AV.01	-	-	-	-
AV.02	01,03,04,05	01,02,03,05,06, 07	03,04,09,10,11	01
AV.03	01,02,03,04,0510	01,02,03,05,06, 07	03,04,09,10,11	01
Recommendation Engine				
RE.01				
RE.02	-	-	09	01
RE.03	-	-	11	01
Prediction				
PRED.01	-	-	09	01
Policy Simulation and validation				
PSV.01	01, 03,04,05	-	03,04,09,10,	01
PSV.02	-	-	03,04, 11	01
PSV.03	-	04	03,04,11	01
PSV.04	-	-	03,04,11	01
PSV.05	-	-	03,04 ,11	01
Urbanite UI				
UUI.01	07	05	08	02,09
UUI.02	-	-	-	-
UUI.03	-	-	-	-
UUI.04	-	-	-	03,04

UUI.05	-	-	02	
Requirements not covered				
NO-REQ	06 Wizard for charts 08 Automatized anal. 09 Download data	08 Allow users to exchange info	06 Enrich charts, graphs 07 Allow users to share layers, charts…	05 Enrich published datasets & results

5.3 Requirements prioritization matrix

For the prioritization of the requirements in URBANITE, several aspects have been considered:

- Baseline functionalities will be implemented in the initial versions, while more complex functionalities depending on the basic ones will be implemented in subsequent releases. In the case of the Virtual SoPoLab a complete version needs to be implemented for M12 and integrated into the ecosystem in M15 as this component will support the SopoLab sessions from WP2. The Virtual SoPoLab may be updated with new functionalities from requirements arisen during the usage of the component in the actual SoPoLab sessions.
- Functionalities which are core for the use cases have been prioritized for the initial releases, based on the use cases needs, defined through working sessions with the use cases in WP6. This prioritization of the functionalities will be accordingly updated as the description and needs of the use cases evolve.
- Functionalities with strong dependencies with other tools (i.e. interfaces related functionalities) will be implemented in early stages of the project so that the integrated framework can be built up and the components can test the dependencies with other components. The objective is to have a reference implementation of the URBANITE ecosystem with baseline functionalities but supporting the whole data workflow in the ecosystem, from the data harvesting to the visualization of the data analysis or data simulation.

The initial prioritization of the URBANITE functional requirements is shown in the next table.:

Table 8. Functional Requirements prioritization matrix

Virtual SoPoLab	**M15**	**M27**	**M33**
VSPL.01	X		
VSPL.02	X		
VSPL.03	X		
VSPL.04	X		
VSPL.05	X		
VSPL.06	X		
VSPL.07	X		
VSPL.08	X		
VSPL.09	X		
VSPL.10	X		
Data Harvesting	**M15**	**M27**	**M33**
DH.01	X		
DH.02	X		

DH.03	X		
DH.04		X	
DH.05	X		
Data Curation	**M15**	**M27**	**M33**
DC.01	X		
DC.02			X
DC.03			X
DC.04		X	
DC.05		X	
DC.06		X	
DC.07			X
DC.08	X		
Data fusion/aggregation	**M15**	**M27**	**M33**
DF.01			X
DF.02		X	
DF.03	X		
DF.04	X		
Data storage	**M15**	**M27**	**M33**
DS.01			X
DS.02	X		
DS.03	X		
DS.04	X		
Data projection	**M15**	**M27**	**M33**
DP.01	X		
Data clustering	**M15**	**M27**	**M33**
DCL.01			
Self-Organizing Map	**M15**	**M27**	**M33**
DCL.01	X		
Regression	**M15**	**M27**	**M33**
REG.01		X	
Traffic simulation	**M15**	**M27**	**M33**
TFS.01		X	
TFS.02	X		
Advanced Visualisation	**M15**	**M27**	**M33**
AV.01	X		
AV.02	X		
AV.03		X	
Recommendation Engine	**M15**	**M27**	**M33**
RE.01		X	
RE.02		X	
RE.03		X	

Prediction	**M15**	**M27**	**M33**
PRED.01		X	
Policy Simulation and validation	**M15**	**M27**	**M33**
PSV.01		X	
PSV.02		X	
PSV.03		X	
PSV.04		X	
PSV.05		X	
URBANITE UI	**M15**	**M27**	**M33**
UUI.01[3]	X	X	X
UUI.02[4]	X	X	X
UUI.03	X		
UUI.04	X		
UUI.05		X	

[3]Each partner providing components with UI should adapt the specific UI to the URBANITE one. Different UIs from different components will be added and updated for each version of the Ecosystem

[4] The planification of this requirement is still under discussion and it will be updated for the next version of the requirements elicitation

6 Conclusions

This document has presented the requirements elicited for the URBANITE ecosystem requirements, as well as the methodology used for its gathering and prioritization.

In URBANITE, the requirements elicitation is an iterative process. Moreover, it results from a mixture of a top-down and a bottom-up processes, where technology providers and use case providers' requirements come together into a common set, which are shown in this document.

For this project, requirements have been differentiated into functional requirements, use cases requirements, and non-functional requirements.

For the functional requirements, a total of 54 functional requirements have been elicited and grouped into 16 functional components, covering all the expected functionalities. This collection of FRs will be updated in accordance with the process presented in the current document.

For the use cases requirements, an initial approximation to the use cases has been made through two main discussion topics. First the analysis of the URBANITE generic processes and actors and its alignment with respect to the first version of the UCs scenarios and UCs users' profiles. Second the alignment of the first set of UCs requirements with the initial version of URBANITE components functional requirements. This alignment will be updated in the next version of this document (D5.2) also considering the updates on the UCs to be delivered in D6.2.

In the case of non-functional requirements, a distinction between performance, security, deployment and availability has been made. In this context, these Non-functional properties have been studied per URBANITE component, providing an overview of the most relevant NFR to be considered and tackled per component and in general, in the URBANITE ecosystem.

The document also includes the benchmarking of the potential tools that can support (totally/partially) the functionalities of the URBANITE systems and that were proposed by the technology partners to serve as baseline technologies to build the URBANITE components conforming the ecosystem.

The document concludes a set of traceability matrices showing: 1) the alignment of the elicited requirements with respect to the URBANITE Key Results; 2) the alignment of which requirement will implement which use case one, and finally; 3) the prioritization matrix, which reflects which requirement will be implemented in which iteration of the URBANITE work plan.

Future versions of this document will include a review of these requirements as well as potential new ones that could arise through the functional validation, use case validation activities, or SoPoLab sessions.

7 References

[1] URBANITE, “URBANITE DoA,” 2019.

[2] URBANITE consortium, “URBANITE DoA,” 2020.

[3] URBANITE, “D6.1 URBANITE use cases requirements and evaluation methodology,” 2021.

[4] S. Madsen, “How to Prioritize with the MoSCoW Technique,” October 2017. [Online]. Available: https://www.projectmanager.com/training/prioritize-moscow-technique . [Accessed March 2018].

APPENDIX: Tools description

Tool 1: ARX

Tool 1	
Name	ARX-Data Anonymization Tool
Main functionalities	ARX is an open source software for anonymizing sensitive personal data. It supports: • Risk-based anonymization using super-population models, strict-average risk and k-map • Syntactic privacy models, such as k-anonymity, ℓ-diversity, t-closeness, δ-disclosure privacy and δ-presence • Semantic privacy models, such as (ε, δ)-differential privacy • Methods for optimizing the profitability of data publishing based on monetary cost-benefit analyses • Data transformation with generalization, suppression, micro-aggregation and top/bottom-coding as well as global and local recoding • Methods for analyzing data utility • Methods for analyzing re-identification risks It is available as a standalone anonymization tool and as a software library with an API that delivers data anonymization capabilities to any Java program.
Integration aspects in URBANITE	The ARX tool could be integrated as a third-party tool in the URBANITE ecosystem (including UI) or as a library through an API
Type of Tool	Standalone desktop tool + Software Library
License	Apache 2
Technology	Java
Related core service component	
Related sub-component	Anonymization Engine
Related KR	KR3
URBANITE partner	TECNALIA

Tool 2: Piveau *consus*

Tool 2	
Name	Piveau *consus*
Main functionalities	*consus* is a collection of services aimed at retrieving data from data sources on the web (i.e. harvesting). Each service runs on the JVM. Deployment of the services in dedicated Docker containers is supported. The stack supports: • Downloading data from an increasing number of different types of APIs, for example, CKAN, Drupal, or SPARQL • Transforming of JSON and XML data via custom JavaScript and XSLT scripts respectively. These scripts can be managed using Git • Compatibility with established data storage systems like Triple-Stores (linked data) and Hadoop clusters (big data) • Logging to Logstash for use in conjunction with the ELK stack • Scheduling
Integration aspects in URBANITE	Consus consist of multiple docker microservices exposing REST APIs. Can be used with either docker-compose or platforms like RedHat Openshift. In order to receive notifications, a mail backend has to be set up.
Type of Tool	Web service
License	Apache 2.0
Technology	Java/Kotlin, JavaScript, XSLT
Related core service component	• Piveau *Hub* (data management) • Piveau *Incognito* (anonymizer) • Piveau *UI* (visualization)
Related sub-component	• Scheduling: Fetch data from APIs in recurring intervals • Transformation: Restructuring data in itself or an entirely new format to achieve compatibility with the data storage solution
Related KR	KR3
URBANITE partner	Fraunhofer

Tool 3: Piveau *hub*

Tool 3	
Name	Piveau *hub*
Main functionalities	*hub* is the central data management component of the Piveau stack. It provides APIs to which the *consus* services can push data to. The *hub* handles consistently storing the data to either a triple store or Hadoop cluster. The stored data can also be retrieved in various formats via API.
Integration aspects in URBANITE	The hub is a single web service exposing various REST APIs. It relies on Elasticsearch and OpenLink Virtuoso and server as a single-point-of-access for both. Runs in a docker container
Type of Tool	Web service
License	Apache 2.0
Technology	Java/Kotlin, Elasticsearch
Related core service component	• Piveau *Consus* (data harvesting) • Piveau *Incognito* (anonymizer) • Piveau *UI* (visualization)
Related sub-component	
Related KR	KR3
URBANITE partner	Fraunhofer

Tool 4: Piveau *ui*

Tool 4	
Name	Piveau *ui*
Main functionalities	The *ui* serves as the central interface to the hub. It allows searching for datasets by facets like keywords and format as well as via custom search terms. Key metadata of the datasets and their distributions is shown in a dedicated view.
Integration aspects in URBANITE	Docker webservice making requires Hub, all other components are optional.
Type of Tool	Web service
License	Apache 2.0
Technology	Vue.js
Related core service component	• Piveau *Hub* (data storage) • Piveau *Incognito* (anonymizer) • Piveau *Consus* (data harvesting)
Related sub-component	
Related KR	KR3
URBANITE partner	Fraunhofer

Tool 5: Piveau *incognito*

Tool 5	
Name	Piveau *incognito*
Main functionalities	The *incognito* anonymization service is capable of transforming large datasets in conformity with data protection requirements for further data analysis. *Incognito* integrates the ARX anonymization library, which provides essential anonymization methods to categorize and modify personal data. The anonymization process requires a configuration file (JSON) containing necessary information about the datasets as well as the required anonymization method, which depends on the different privacy model ARX provides.
Integration aspects in URBANITE	Standalone web service. Can be integrated into the pipe concept employed for Consus. Thereby, no sensitive information is stored in the hub, only the anonymized data.
Type of Tool	Web service
License	Apache 2.0
Technology	Java/Kotlin, ARX
Related core service component	• Piveau *Hub* (data storage) • Piveau *UI* (visualization) • Piveau *Consus* (data harvesting)
Related sub-component	
Related KR	KR3
URBANITE partner	Fraunhofer

Tool 6: SUMO

Tool 6	
Name	Simulation of Urban MObility SUMO
Main functionalities	Traffic simulator and toolchain: ● Microscopic simulation of traffic ● Agent-based simulation ● Generate a graph of roads from OpenStreetMap data ● Allows interaction with simulation on-line ● Provides basic visualization ● allows interaction using Python ● CLI integration ● GUI available
Integration aspects in URBANITE	It could be integrated as a third-party tool in the URBANITE ecosystem (including UI)
Type of Tool	Standalone
License	Eclipse Public License Version 2
Technology	C++ and Python,
Related core service component	URBANITE Algorithms
Related sub-component	Traffic Simulation
Related KR	KR4
URBANITE partner	JSI

Tool 7: MATSim

Tool 7	
Name	Multi-Agent Transport Simulation Toolkit
Main functionalities	Framework for implementing large-scale agent-based transport simulations: ● Microscopic simulation of traffic ● Agent-based simulation ● Generate a graph of roads from OpenStreetMap data
Integration aspects in URBANITE	Integrated as part of back-end. GUI for supported functionalities will be developed.
Type of Tool	Framework
License	GNU General Public License 2
Technology	Java framework
Related core service component	URBANITE Algorithms
Related sub-component	Traffic Simulation
Related KR	KR4
URBANITE partner	JSI

Tool 8: WEKA

Tool 8	
Name	WEKA
Main functionalities	Workbench for machine learning: ● Automatically generate the best model for a dataset ● Clustering ● Classification and regression ● Attribute selection ● Targeted Projection Pursuit ● Deep learning ● GUI available ● CLI integration
Integration aspects in URBANITE	It can be Integrated as part of the back-end. GUI for supported functionalities will be developed.
Type of Tool	Standalone
License	GNU General Public License
Technology	Java
Related core service component	URBANITE Algorithms
Related sub-component	● Visualizations component ● Prediction component ● Self-organizing map ● Projection component ● Clustering component
Related KR	KR4
URBANITE partner	JSI

Tool 9: Orange 3

Tool 9	
Name	Orange 3
Main functionalities	Machine learning and visualizations ● Simple visualizations ● CN2 Rule Viewer, Heatmap, Plots, Distributions, Tree viewer (clustering etc.)
Integration aspects in URBANITE	It could be integrated as a third-party tool in the URBANITE ecosystem (including UI)
Type of Tool	Desktop
License	GNU General Public License 3.0
Technology	Python
Related core service component	URBANITE Algorithms
Related sub-component	● Visualizations component ● Projection component ● Clustering component ● Prediction component ● Self-organizing map
Related KR	KR4
URBANITE partner	JSI

Tool 10: Idra

Tool 10	
Name	Idra
Main functionalities	A web application developed in FESTIVAL H2020 project, able to federate existing Open Data Management Systems (ODMS) based on different technologies (CKAN, DKAN, Socrata, Orion Context Broker, etc.). It uniforms representation of collected open datasets' metadata according to DCAT-AP standard and provides a set of RESTful APIs to be used by third party applications. Idra also provides a unique point of access to federated ODMS.
Integration aspects in URBANITE	The metadata (DCAT-AP) of the datasets stored within Idra will be used for building on the flight visualizations (starting from the distribution referenced in the metadata) that will be used into the URBANITE UI component. Moreover, Idra maps heterogeneous metadata to DCAT-AP standard.
Type of Tool	Web application composed by a back-end core module exposing RESTFul APIs and font-end catalogue.
License	AGPL-3.0
Technology	Java, Angular, SPARQL, Apache Jena, RDF4j
Related core service component	URBANITE Data Management Platform
Related sub-component	• Data Harvester • Data Hub • Scheduler • DCAT-AP MetaData Mapping
Related KR	KR3
URBANITE partner	ENG

Tool 11: DataModel Mapper

Tool 11	
Name	DataModel Mapper
Main functionalities	The Data Model Mapper tool enables to convert several file types (e.g. CSV, JSON, GeoJson) to the different Data Models defined both by FIWARE and the SynchroniCity Project. The files in input can contain either rows, JSON objects or GeoJSON Features, each of them representing an object to be mapped to an NGSI entity, according to the selected Data Model. The input data are validated against the JSON schema corresponding to a target Data Model and, further, the validated and converted data are sent to the FIWARE Orion Context Broker and/or to a local file.
Integration aspects in URBANITE	DataModel Mapper converts input data (e.g. CVS files) into NGSI Entities. This NGSI Entities to be stored into the Data Storage component. (to be updated to manage and produce NGSI-LD Entities)
Type of Tool	Can be used as a Server application exposing RESTFul APIs or as a standalone application through CLI (Command Line Interface).
License	AGPL-3.0
Technology	NodeJS
Related core service component	URBANITE Data Management Platform
Related sub-component	• Data Validation • Data Mapping • Data Transformation
Related KR	KR3
URBANITE partner	ENG

Tool 12: CaPe

Tool 12	
Name	CaPe
Main functionalities	CaPe provides an Open Source ICT suite for a consent-based, user centric personal data management & GDPR compliant enabling: i) the data subjects to manage, trace their own data and its associated consent ii) the data controllers to use consent to data sharing among digital services using personal data and meet the GDPR requirements CaPe reduces the burden organisations face in consent management by managing the entire consent life cycle at inter and intra-organizational level and Increases consumers' confidence by applying an innovative approach to consent management that puts individuals at the centre offering self-service transparency tools.
Integration aspects in URBANITE	Not yet defined
Type of Tool	Microservices based application exposing RESTFul APIs and a dashboard front-end component.
License	MIT
Technology	Java, Angular, MongoDB
Related core service component	
Related sub-component	
Related KR	KR3
URBANITE partner	ENG

Tool 13: Digital Enabler Platform

Tool 13	
Name	Digital Enabler Platform
Main functionalities	The Digital Enabler platform (IDE) is an Internet of Everything platform, powered (IoE) by FIWARE (IoE), to crawling, collecting, integrating, analyze and rendering scattered data coming from heterogeneous data providers, including sensors, manufacturing systems and machines. It enables multi-domain data integration, harmonization and multi-device interoperability.
Integration aspects in URBANITE	DE can be used to access data from heterogeneous data sources, to aggregate such data and to build visualization over the aggregated and/or the row data.
Type of Tool	Web/Server application
License	AGPL3
Technology	FIWARE, Java
Related core service component	URBANITE Data Management Platform URBANITE Advanced Visualization
Related sub-component	Data Harvesting Data Visualization Data Aggregation Scheduling Data Mapping
Related KR	KR3
URBANITE partner	ENG

Tool 14: Micro Proxies

Tool 14	
Name	Micro Proxies
Main functionalities	"Micro Proxies" consist of a set of specification and reference implementation of customizable technological element aiming to enable interaction with legacy systems and to manage access to provided data; Micro Proxies can be adopted to prevent unauthorized access, to deal with interoperability aspects, to prevent the communication of a specific set of data (e.g. private or sensitive data), etc.
Integration aspects in URBANITE	Micro-proxies can act as a middleware for legacy or external system and URBANITE Platform to perform an adaptation of data at the premise level.
Type of Tool	Specifications of Web Services (with reference implementation)
License	AGPL3
Technology	Java
Related core service component	URBANITE Data Management Platform
Related sub-component	Data Harvesting
Related KR	KR3
URBANITE partner	ENG

Tool 15: Decidim

Tool 15	
Name	Decidim[5]
Main functionalities	Decidim is an open-source participatory democracy platform supported by a strong community and used by several cities, regions and organizations. Within Decidim, the administrator can configure bodies such as Municipal Councils and City Assemblies thanks to the definition of the so-called Assemblies. Within an Assembly, the administrator can manage and configure several useful components that are further be used by citizens to create debates and/or propose ideas and challenges.
Integration aspects in URBANITE	Integration through the UI (lightweight integration).
Type of Tool	Web Application
License	AGPL-3.0
Technology	Ruby on Rails
Related core service component	
Related sub-component	URBANITE Forum – VSPL (Virtual SoPo Lab)
Related KR	KR1
URBANITE partner	ENG

[5] https://decidim.org/