URBANITE

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

Deliverable D5.5

URBANITE Detailed architecture-v2

Editor(s):	Maria José López
Responsible Partner:	Tecnalia
Status-Version:	Final
Date:	30.03.2022
Distribution level (CO, PU):	PU

Project Number:	GA 870338
Project Title:	URBANITE

Title of Deliverable:	URBANITE Detailed architecture-v2
Due Date of Delivery to the EC:	31/03/2022

Workpackage responsible for the Deliverable:	WP5 - URBANITE ecosystem integration and DevOps
Editor(s):	Tecnalia
Contributor(s):	Maria José López, Sonia Bilbao, Sergio Campos, Iñaki Olabarrieta (Tecnalia), Torben Jastrow (FhG), Maj Smerkol (JSI), Giuseppe Ciulla (ENG)
Reviewer(s):	Keye West (WAAG)
Approved by:	All Partners
Recommended/mandatory readers:	WP2, WP3, WP4, WP5, WP6

Abstract:	This document contains the detailed design of URBANITE: its components, modules, and. This deliverable is the result of Task 5.2.
Keyword List:	Architecture, Development environment, integration, testing, requirements.
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/
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	25/02/2022	First draft version with TOC	Tecnalia
V0.2	11/03/2022	Contributions about WP3	Tecnalia & FhG
V0.3	17/03/2022	Contributions to WP4 analytical framework section	Tecnalia
V0.4	17/03/2022	Contributions to Virtual SoPoLab section	ENG
V0.5	18/03/2022	Contributions to Controller and Identity and Authorization sections	ENG
V0.6	19/03/2022	Contributions to Data Catalogue and UUI sections	ENG
V0.7	24/03/2022	Contributions about simulation and recommendation section	JSI
V0.8	25/03/2022	General sections ready for internal review	Tecnalia
V1.0	30/03/2022	Final version of submission	Tecnalia

Table of Contents

List of Figures

List of Tables

Terms and abbreviations

API	Application programming Interfaces
CSV	Comma Separated Values
DCAT	Data CATalogue
DCAT-AP	DCAT Application Profile
GDPR	General Data Protection Regulation
GPS	Global Positioning System
HP	High Performance
JSON	JavaScript Object Notation
JSON-LD	JSON for Linking Data
KPI	Key Performance Indicators
MADA	Multi-Attribute Decision Analysis
NoSQL	No Structured Query Language
OD	Origin Destination
REST	REpresentational State Transfer
SAML	Security Assertion Markup Language
SQL	Structured Query Language
TSDB	Time Series Data Base
UI	User Interface
UUI	URBANITE User Interface
VSPL	Virtual SopoLab
WP	Work Package
XSL	eXtensible Stylesheet Language

Executive Summary

This document contains the description of the URBANITE integrated architecture to be released in month 27. The schema reflects the evolution of the components from the previous version of this deliverable and the modifications arisen from the different use cases definition and the first approaches implemented in each of them.

The general schema of the architecture has been reorganized considering the final version of the D5.2 deliverable [1], where the requirements were confirmed and adjusted to the use cases revision. The work done within the context of WP has also been considered.

The previous version of this deliverable [2] gathered the details about each technical component as well as the interaction with the rest of the architecture. In order to avoid duplicated content, the strategy for this document has been focused on the evolution of those components and the interfaces reflected in its previous version [2].

The components that are part of the URBANITE architecture are developed within the technical work packages WP2, WP3, WP4 and WP5, supporting the functionalities defined by the use cases and answering the issues that arose in the evaluation made by them.

The documentation about each component includes in detail the structure and motivation of the different features provided by them, so the dedicated section in this document could refer to them and not duplicate information and details.

The sections in this document are organized to follow the layers depicted in the general schema, validated along with the project. The next release of the URBANITE Ecosystem is planned for month 27, and although the status of the platform is stable and no major modifications are planned, it can suffer minor changes to adapt or to any detected condition until the end of the project.

The general URBANITE UI integrates all the layers mentioned above, allowing a fluid use of the platform, and managing the identities and authorizations, making this ecosystem a safe and secure context for the different stakeholders, as well as some additional features related to the customization of the dashboard and provided during the last review of the project.

The process of integration has been improved as well as different integrations have been performed for ensuring smooth interoperability of the different components covering the entire data processing chain and implementing the orchestration of all the components and services as an URBANITE controller module.

This is the last deliverable about the URBANITE architecture; however, if any change is required once this document is submitted, the document accompanying the last version release will show the final and deployed schema and its final features.

1 Introduction

1.1 About this deliverable

This deliverable is the last and final release of the detailed design of the URBANITE architecture. In order to make it more readable, a resume of the main functionalities of each component has been included and an exposition of the evolution followed from the status of the components in the first version of the deliverable. Any important aspect has been highlighted as part of the reasoning regarding its evolution.

The document reflects the work done in the T5.2 and T5.4 tasks and reflects the technical specification of the second prototypes of the components implemented within the context of WP3 and WP4, as well as the feedback and comments provided by the use cases and the sessions organized as part of the SoPoLab sessions with WP2.

The URBANITE User Interface is an important part of the architecture since it is the contact point between the users and the ecosystem, as well as the final visualisation of all the analysis that the platform allows in order to support the decisions to be made by those users. This UI will be developed using responsive web technologies that will bring a good user experience.

1.2 Document structure

The structure of the document aims to offer a clear understanding of the architecture that supports the URBANITE Ecosystem, including all the specific characteristics of each pilot environment.

There are some main sections grouping the components following the type of functions they will provide as layers. As subsections, the individual components are presented, explaining their main functionality and the evolution from the last architectural schema submitted in the previous version of this deliverable.

The mentioned structure is as follow:

- Section 2: An overview of the general schema of the integrated architecture, as well as the schema for the different pilots.
- Section 3: The main section grouping the components following the layers dedicated to the same subject or general role they play. Inside these sections, six subsections with two parts regarding the main description and the evolution.
    - 3.1: A description of the data management layer or platform, describing the components related to the acquisition, aggregation, and storage of the data, corresponding to the process in which the data are harvested and finally stored in a formalized way. These components are Data Harvesting, Curation, Preparation, Transformation, Anonymization, Data Fusion/Aggregation, Data Storage & Retrieval and Data Catalogue.
    - 3.2: The Data analysis, simulation and recommendation layer or platform components regarding to the tools to perform the analysis and simulations that are the core of the Ecosystem. The Advanced Visualisation, the Exploratory data Analysis, the Traffic simulation, and the Analytical framework are the components that provide the intelligence of the platform in order to offer the analysis and processes needed for decision support.
    - 3.3: SoPoLab. Description of this Digital virtual space, for sharing experiences of different policy domains.

    - 3.4: The integrated URBANITE UI, for describing the component that provides access to the URBANITE technical tools offered by the rest of the components, and the options for customizing the dashboard for a better user experience.
    - 3.5: The Identity/Authorization Management, for securing the access to the other URBANITE's components, resources, and services.
- Section 4: Conclusions related to the final status of the architecture and the work until the end of the project.
- Section 5: References made in the content of the document.

2 Overview of the URBANITE integrated conceptual architecture

2.1 URBANITE ecosystem generic architecture

The general schema of the URBANITE architecture is depicted in Figure 1.

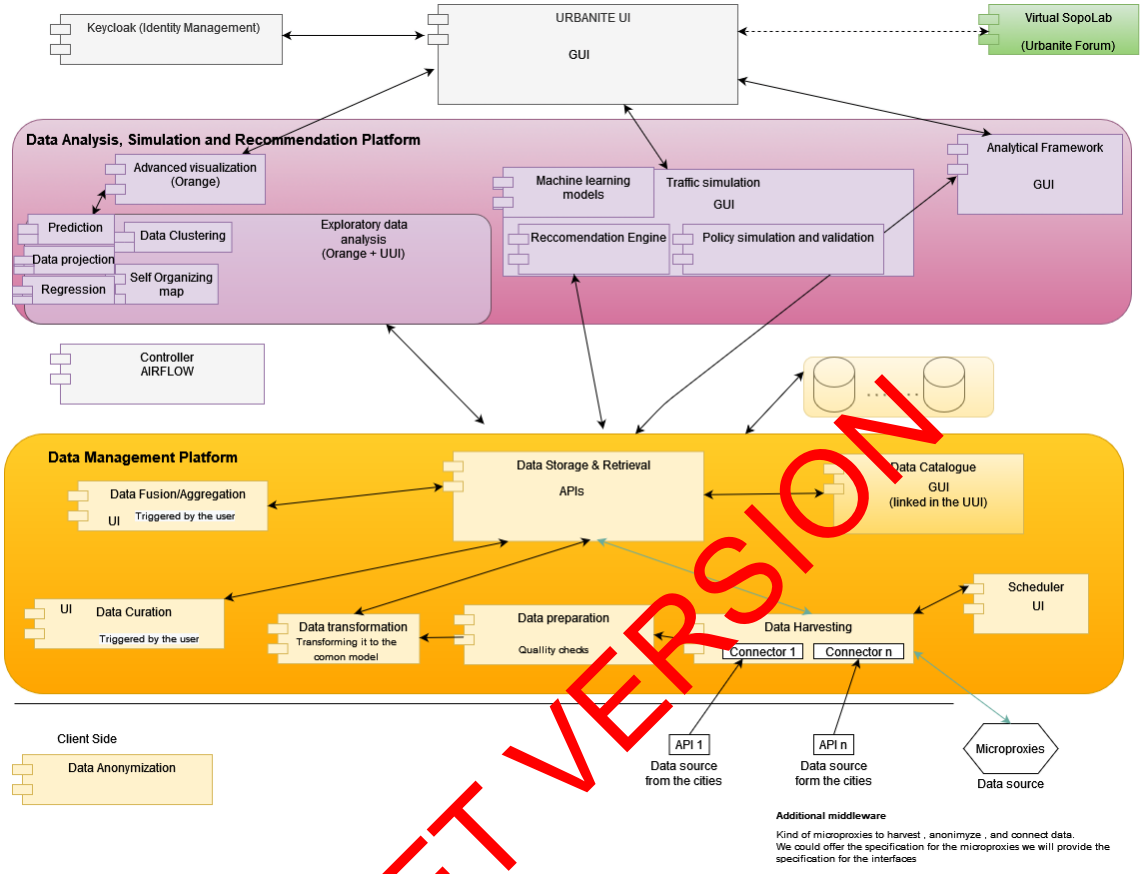

Figure 1. URBANITE general architecture schema

The main variations to the previous schema in deliverable D5.4 [2] are those related to the organization of the different components in the Data Analysis Simulation and Recommendation platform. The final definition of the requirements, made in the D5.2 deliverable [1] meant a reorganization of the components regarding the nature of the different elements depicted in the previous architecture. So, some of the previously considered components such as Prediction, regression, Data Projection, Data clustering and self organizing maps, where then considered as tools for implementing the real functionalities to be provided as an added value to the use cases. The big component then is named Exploratory data analysis. Same process was followed with the now considered tools machine learning models, recommendation engine and policy simulation and validation; those became parts of the bigger component Traffic Simulation.

2.2 URBANITE ecosystem pilots architecture

The technical implementation of the use cases, described in the D6.2 deliverable [3], defines different components for each environment with a basic structure. The sources of data, the analysis, and the simulations available are specific developments for each use case.

The schema reflects the environment that supports the available scenarios defined in each use case and for each city.

The corresponding analysis indicated in the different analytical Frameworks as well as the simulations of the Traffic Simulation group, are described in section 3.2 of this document and in the D6.2 deliverable [3] in more detail.

The pictures above reflect the general schema for the different environments, and show the commonality shared for all of them. Nevertheless, those are not the final picture so they can be adapted to the needs of every use case until they are deployed.

The elements highlighted in blue are the specific elements developed within the context of every use case. Notice that in the Amsterdam pilot, there are no analysis developed yet.

Amsterdam pilot.

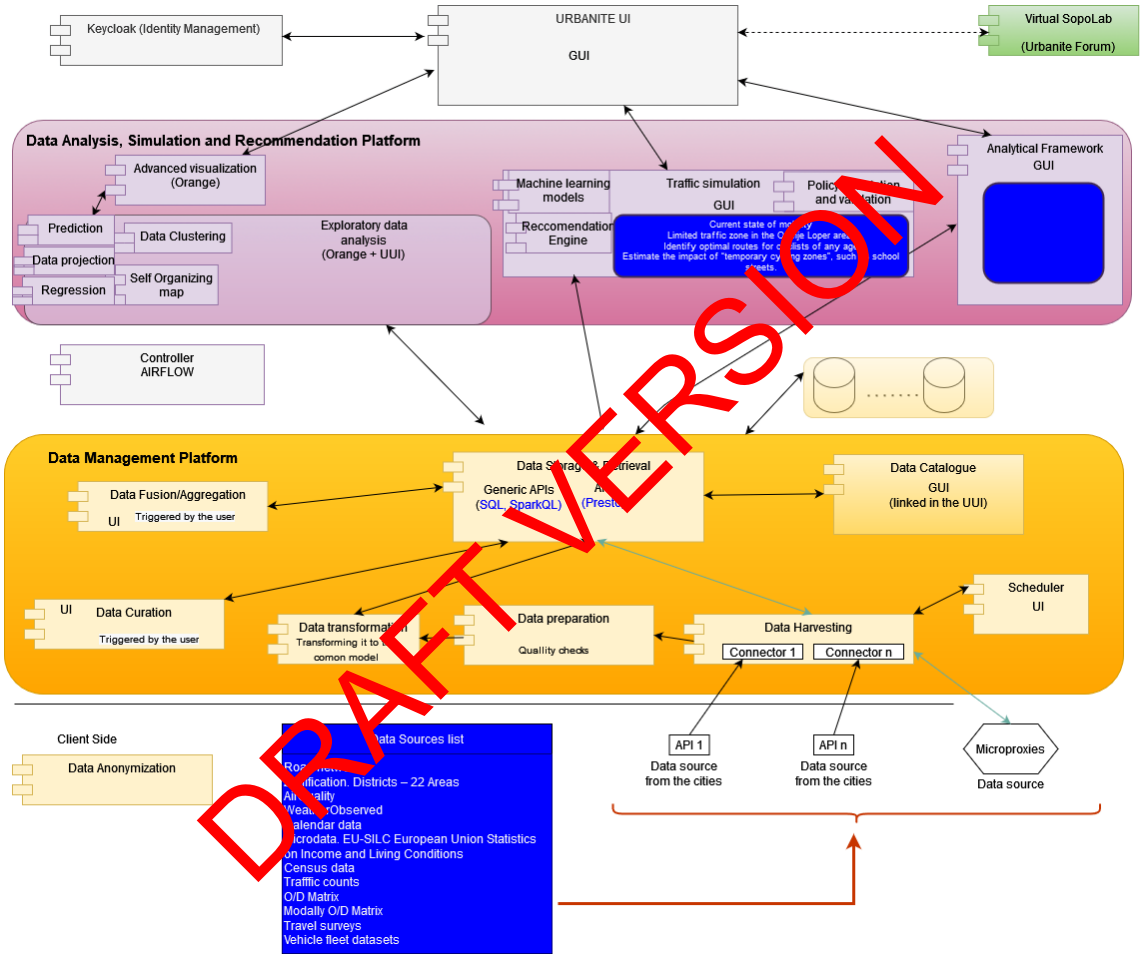

Figure 2. URBANITE general architecture schema supporting AMSTERDAM use case

Bilbao pilot.

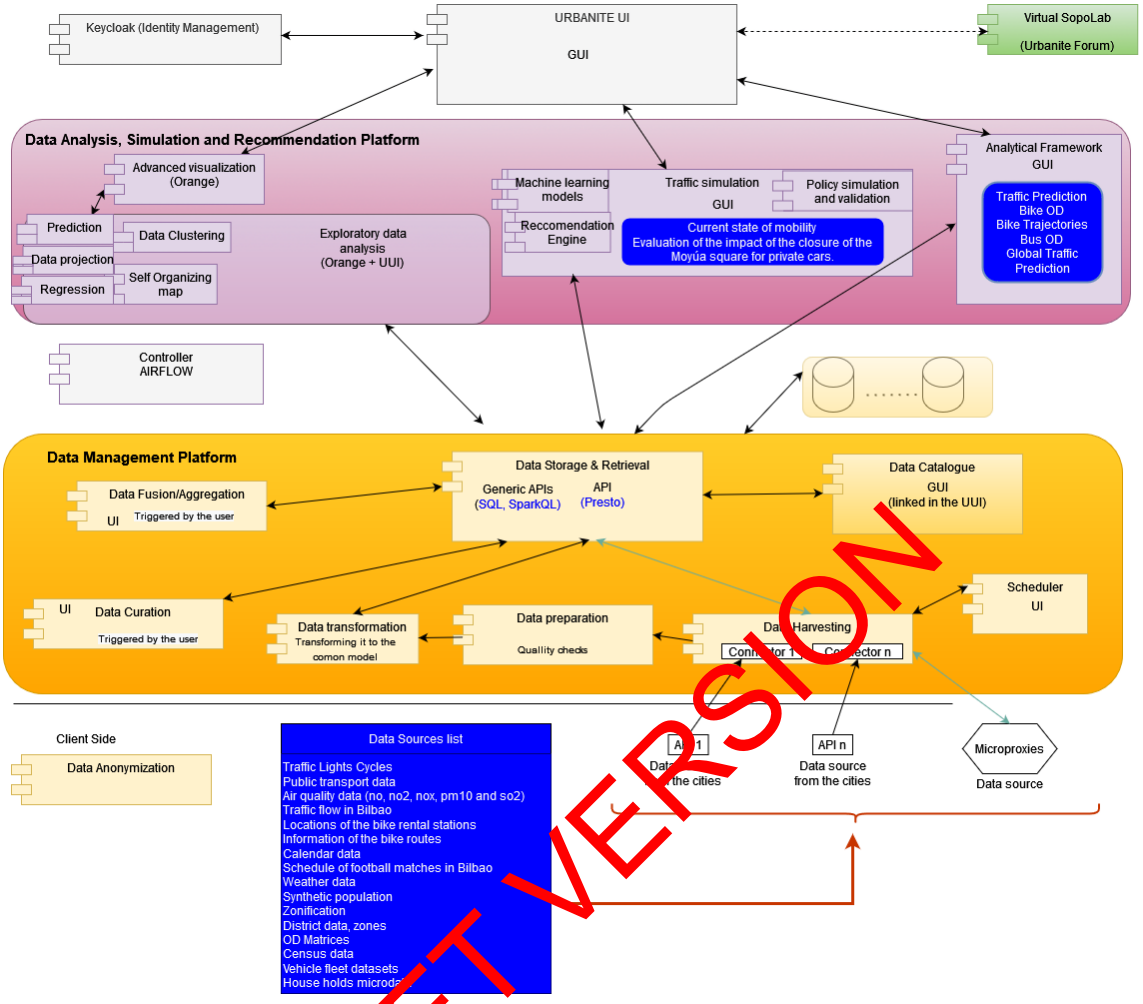

Figure 3. URBANITE general architecture schema supporting BILBAO use case

Helsinki pilot.

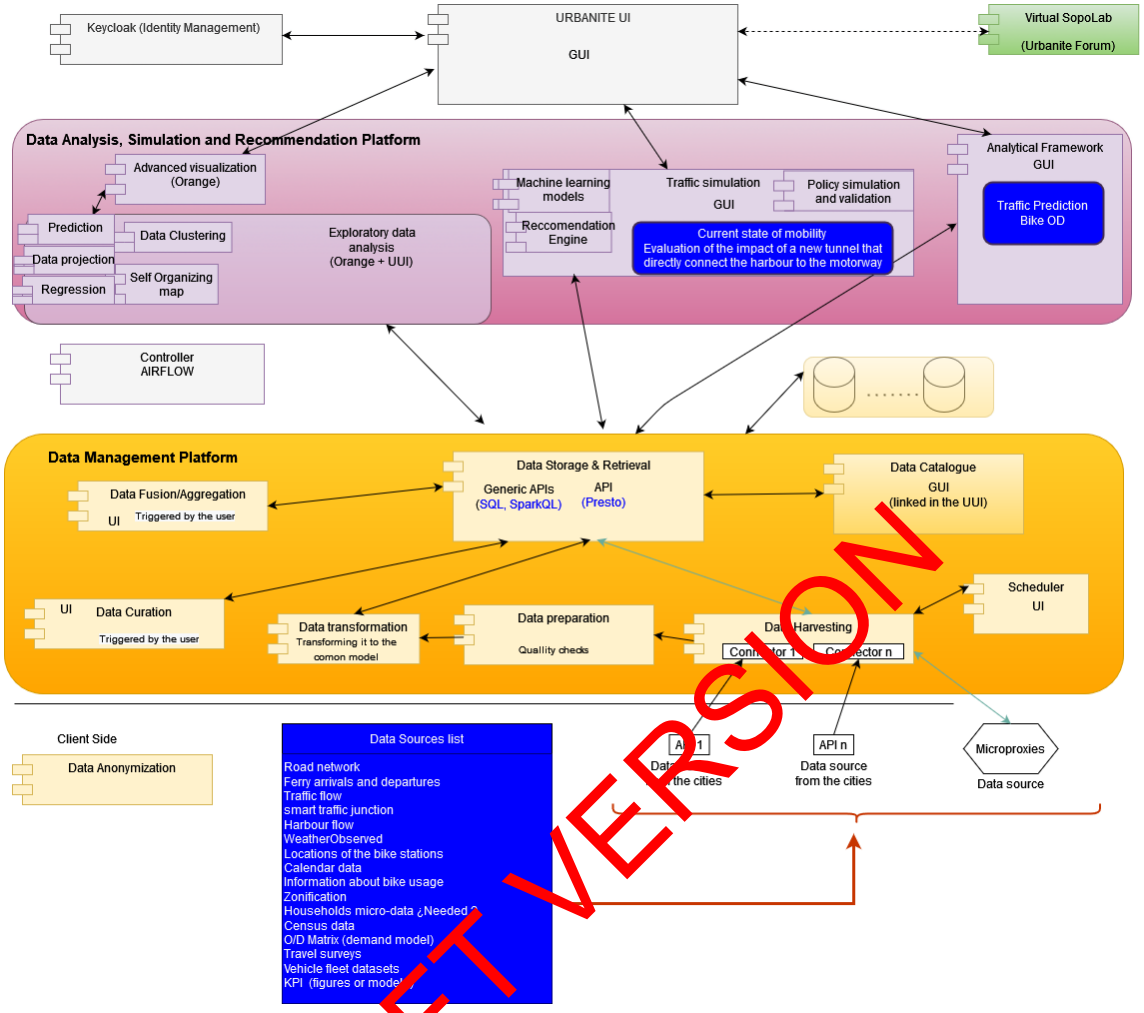

Figure 4. URBANITE general architecture schema supporting HELSINKI use case

Messina pilot.

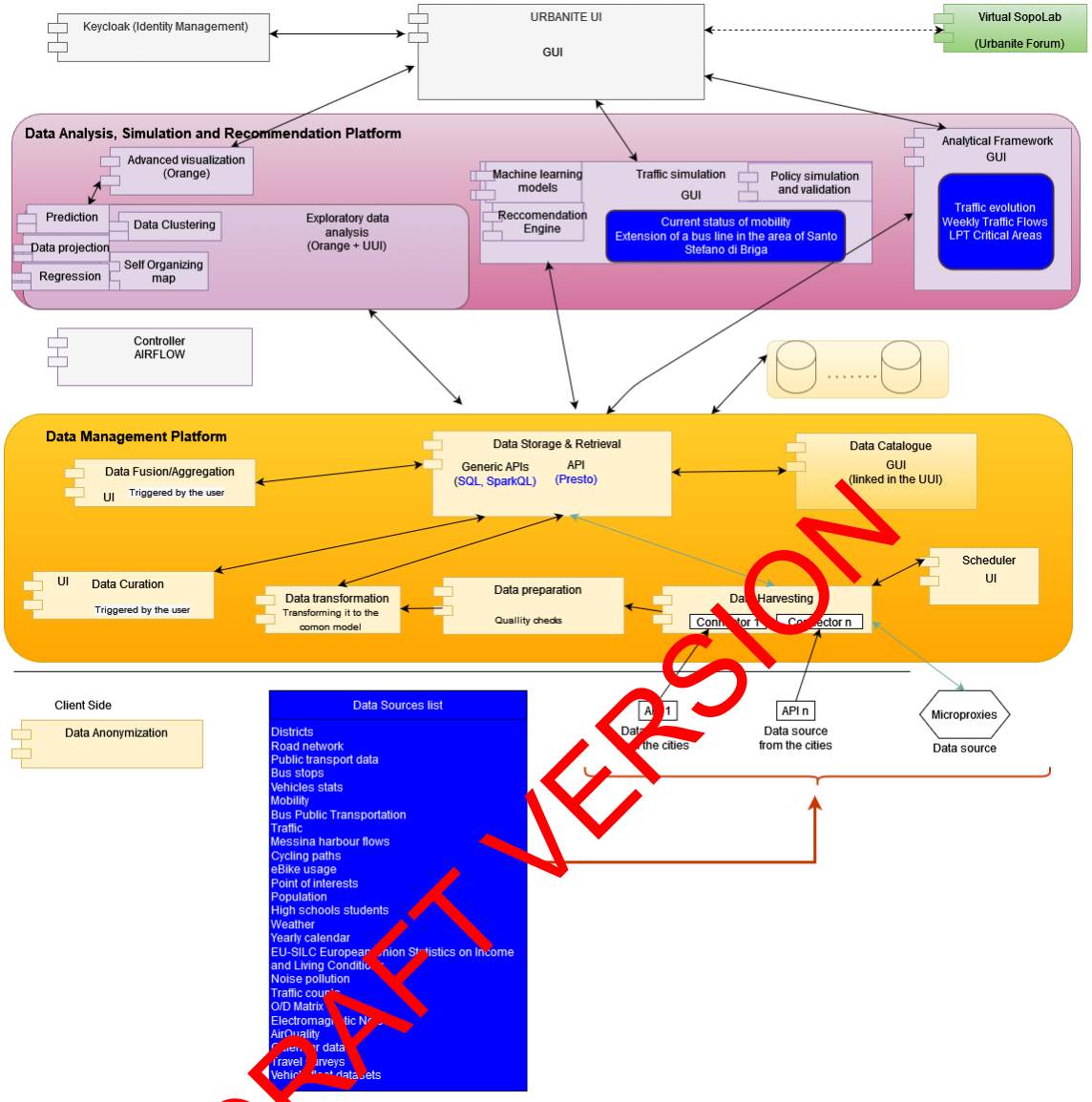

Figure 5. URBANITE general architecture schema supporting MESSINA use case

3 URBANITE ecosystem architecture detailed design

The following sections describe the technical aspects of the different components as well as the interactions among them. The structure of the layers and components is taken from the last architectural schema and included in the D5.2 deliverable.

Considering that this is the second and final version of the document, and to avoid the duplication of the definitions of every component, this section gathers the changes and final status of the architecture and components. If no changes have been made, then the previous version of the deliverable will be referenced.

3.1 URBANITE Data Management Layer

The Data Management Platform stands for a variety of distinct software components that work together to deliver the key functionalities that are data harvesting, data preparation/transformation/curation/anonymisation, and data aggregation, fusion, and storage. Detailed information on these components is available in three deliverables D3.2 [4], D3.5 [5], and D3.7 [6]. The Data Management Layer follows a microservice architecture.

The harvesting modules and connectors provide the means to import (i.e. download) data and metadata from endpoints on the web. All these different kinds of data and metadata then need to be checked, cleaned and harmonised for further processing (see D3.5 [5]). This is achieved by data preparation and subsequent transformation steps, as well as curation. Once the data and metadata are brought into a common format, they need to be stored in dedicated databases (see D3.7 [6]).

Additionally, the (meta-)data needs to be downloaded in regular intervals to account for changes thereof. Managing these intervals is the responsibility of the Scheduler, which triggers the data importers, which in turn download the data.

The architecture of the Data Management Layer has not changed from version 1. However, the Data Storage & Retrieval Component has been enhanced with new repositories to improve performance as will be explained in section 3.1.4.

3.1.1 Data Harvesting

3.1.1.1 Main functionality

Harvesting refers to the subset of steps from the import of data to the export into a data store. In URBANITE, this harvesting process has been implemented using a pipeline, i.e., a chain of processing components arranged so that the output of each component is the input of the next. The pipeline has been developed using the open-source solution named Piveau Pipe Concept.

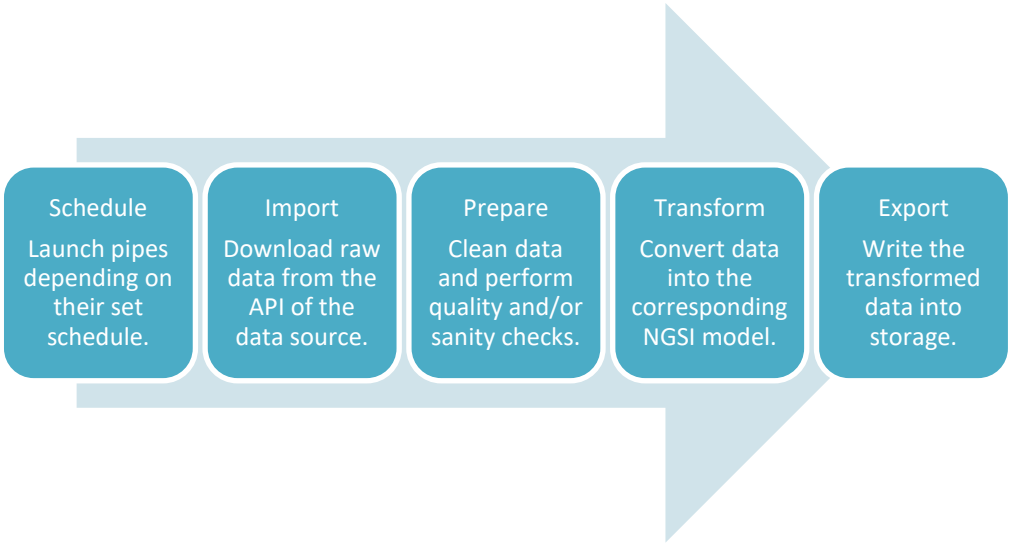

Figure 6: URBANITE data harvesting implemented using the Piveau Pipeline concept

3.1.1.2 Evolution and final status

There have been no technical updates or modifications with respect to the content described in"D5.4 URBANITE Detailed architecture v1" [2]. For a detailed description of the implementation, please refer to "D3.2 Data Harvesting Module and Connectors Implementation v1.0" [4].

3.1.2 Data Curation, Preparation, Transformation and Anonymization

These software components are related to the tasks of data manipulation between initial harvesting and storage. This includes, but is not limited to, the steps of data anonymisation, preparation, transformation, and curation. Depending on the nature and quality of the harvested data none, some, or all of these steps could be necessary.

3.1.2.1 Main functionality

Data anonymisation aims to address privacy protection by removing personally identifiable information from data sets, so that the people whom the data describe remain anonymous (this is compliant with GDPR regulations).

Data Preparation refers to the process of ensuring a certain level of (meta-) data quality. This includes detecting and removing false/implausible data, for example. Validating against a given specific schema could be one way of achieving this.

Before data is used by the algorithms and simulation models, we need to ensure that it meets certain quality criteria. Some of these quality checks are done in the pipeline process, while others are done over already stored data. The quality aspects analyzed in URBANITE are the following: accuracy, completeness, consistency and precision.

Data Transformation is the conversion from one format into another, without altering the (meta-) data's semantics. The following transformations have been developed: JSON to JSON, CSV to JSON and XLS(X) to JSON. The transformed data structures are compliant with the URBANITE common data models based on FIWARE Smart Models.

Finally, data curation is considered the maintenance and enrichment of data after the previous steps have been completed. Data curation has been focused on cleaning trajectory data based on GPS measurements. These data contain noise which implies that the obtained measurements

do not match exactly with the real positions of the sensors. Map-Matching algorithms are used not only to correct the measurement noise but also to reconstruct the intermediate points, producing a complete set of locations between the origin and the final destination of the trajectory.

3.1.2.2 Evolution and final status

For a detailed description of the implementation of each of the components, please refer to "D3.5 Data Curation Module Implementation v1.0" [5].

3.1.3 Data Fusion/Aggregation

Before data is processed by the analysis and prediction algorithms, it is sometimes necessary to either aggregate or fuse datasets.

3.1.3.1 Main functionality

Data aggregation is the process of gathering data and presenting it in a summarised format. It can be used to hide personal information, or to provide information in a synthetic form.

Data fusion is the process of integrating multiple data sources to produce more consistent, accurate, helpful 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.

3.1.3.2 Evolution and final status

Currently, two components use the data aggregation and fusion functionalities, i.e. the Bike Analysis component and the Traffic Prediction component.

The Bike Analysis component uses bike aggregated data. Bike rentals are aggregated both in time and in space. For time aggregations, one-hour time slots are considered, whereas, for space aggregations, the districts within the cities are used.

In addition, the Traffic Prediction component uses traffic flow aggregated data. In this case, traffic counts at every inductive loop are aggregated only in time and within a period of 5 minutes.

Moreover, these two components also need meteorological information, events and calendar data for their analysis and predictions. However, each type of data is harvested with different frequencies: meteorological data is available every hour, events occur at specific moments in time, and calendar data is available daily. The fusion of all these data types implies aligning the information at a given instant in time.

In the case of the Bike Analysis component, all the sources are fused to hourly frequency, whereas for the Traffic Prediction component, all the sources are fused to a five minutes frequency.

For a detailed description of the implementation of each of the components, please refer to "D3.7 Data aggregation and storage module implementation v1.0" [6].

3.1.4 Data Storage & retrieval

The Data Storage & Retrieval component provides the means to store and retrieve datasets metadata and related data.

The general description of these functions is described in the previous version of this document [2] and in the D3.7 document [6].

3.1.4.1 Main functionality

As mentioned before in the specific deliverable [6], the chosen Data Storage system is a combination of SQL (MySQL) and NoSQL (MongoDB) databases. The data is stored in the databases according to the models described in [6] and formatted in JSON-LD format as key-values.

The main technical structure of these functionalities remains as described in D3.7 and in the previous version of the architecture [2], although some modifications are being made in order to improve the behaviour of some components when retrieving specific kinds of data.

3.1.4.2 Evolution and final status

The evolution due to performance issues can be observed in the picture below:

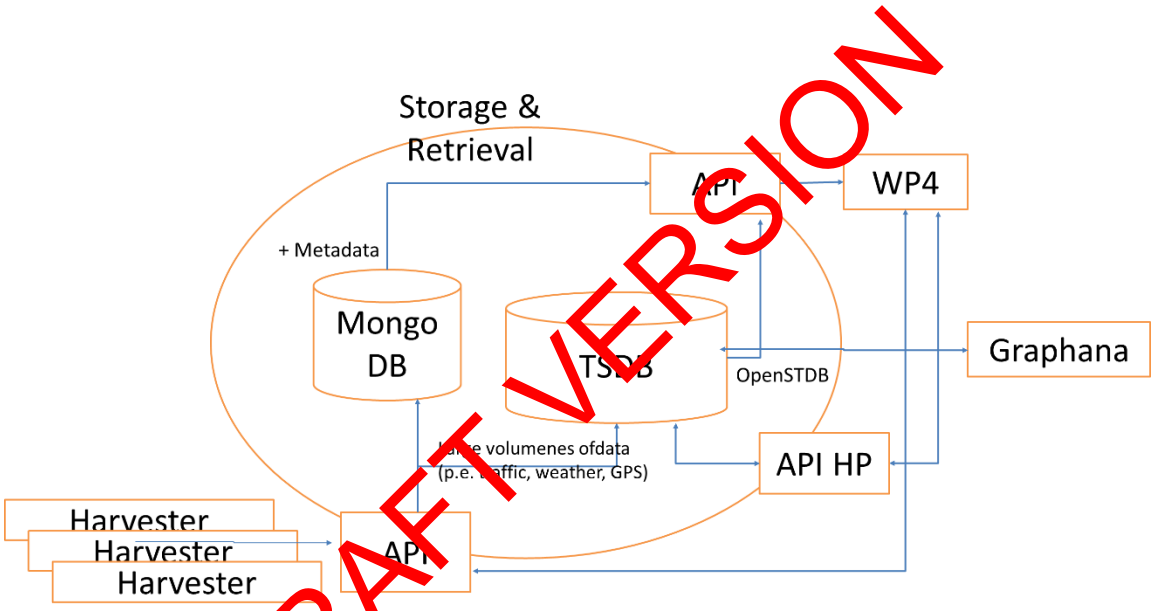

Figure 7: URBANITE data Storage & Retrieval organisation

In order to improve the response of the Storage & retrieval System when time series are handled, a new database has been included in the schema:

- Create a complementary database for time series for timed data (e.g. traffic data) (OpenSTDB)
- Implement 2 APIs, one for insertion and one for query
    - In the insert, record data + metadata in MongoBD or, if they are traffic, only in STDB (without metadata).
    - In the query, if it's not about traffic, you run Mongo, which has all the info. If it is traffic, you retrieve it from OpenSTDB and append the metadata part to it (directly in the code)
- OpenSTDB API is also available

- Relatively easy aggregations.
- No impact in harvesters.
- Gaps in the data, to address once the above is resolved.
- Easy link to Graphana visualisations..

The final schema of the databases system will be described in the document accompanying the next URBANITE Ecosystem release due to June 2022.

The expertise of partner AD (Alma Digit) with MongoDB supports the decision for improving this system. AD has been working on internal aspects of their tasks in order to implement the best structure in the MongoDB, considering the response needed by their modules.

3.1.5 Data Catalogue

The Data Catalogue provides the functionalities to discover and access the datasets managed by Data Storage and Retrieval component. It is based on Idra, an open-source tool primarily able to federate open-data management systems based on heterogeneous technologies. To this aim, Idra is based on a modular architecture that leverages the concept of "connector" to interact with the federated (open) data management systems. Idra can be easily extended with new classes of connectors able to interact with systems based on new technologies. Idra uniforms information is retrieved through the connectors (metadata of datasets) according to DCAT-AP format) and provides a set of RESTful APIs to be used by third party applications.

3.1.5.1 Main functionality

The main functionalities of the Data Catalogue are related to the discovery of datasets managed by the Data Storage and Retrieval; to this aim, Idra has been extended with a new connector able to interact with the API exposed by the Data Storage and Retrieval. Furthermore, it is able to schedule the update period to check the availability of new datasets or updates related to datasets already available.

3.1.5.2 Evolution and final status

Many improvements have been performed to support the different data introduced into the Ecosystem and more changes can arise from the use cases and the different necessities of future releases; therefore in this section, the current status of the Data Catalogue is presented.

The administration page of the Data Catalogue allows to access the list of federated data management systems.

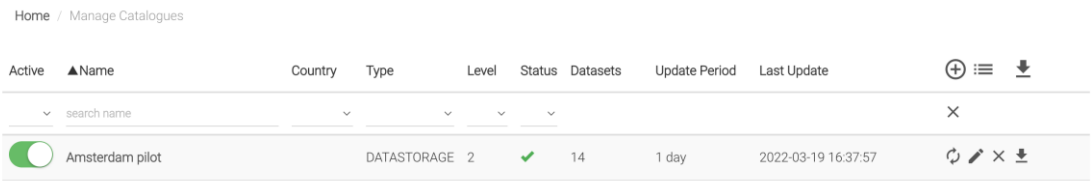

Figure 8 Data Catalogue – list of federated data management systems

From this page, the user can

- Add/Edit/Delete a data management system.
- Activate/Deactivate a data management system.

- Start the synchronization of a data management system.
- Download a data management system dump or the entire federation dump with DCAT-AP profile.

The figure depicts these functionalities

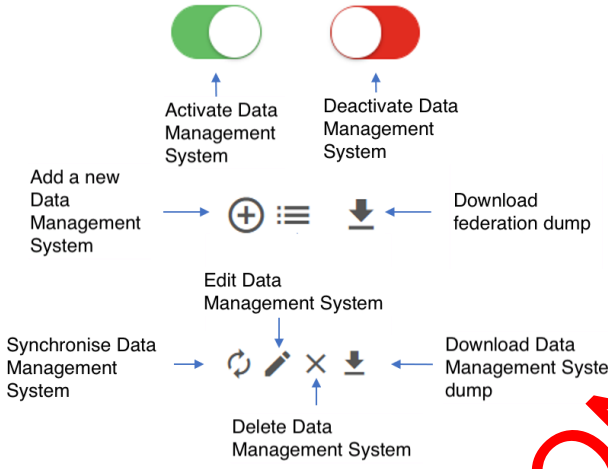

Figure 9 Data Catalogue – users actions on federated data management systems

By clicking on the button ⊕ the the Data Catalogue component provides the form to add a nre Data Management System.

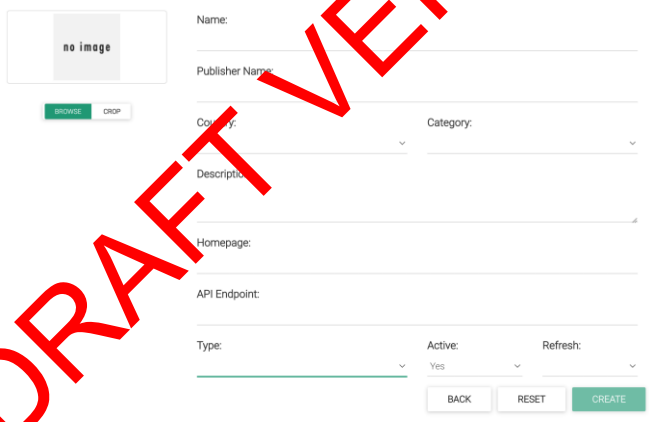

Figure 10 Data Catalogue – edit form of a federated data management system

From this form, the user can insert the information related to the new data management system

- Name: The name of the data management system.
- Publisher Name: The publisher name of the data management system.
- Country: The country of the data management system, if any.
- Category: The category of the data management system (e.g. Municipality, Private institution).
- Description: A description of the data management system.
- Homepage: The homepage of the data management system.
- API Endpoint: The endpoint of its REST APIs, if any.
- Type: The type of the data management system.

- Active: Through this field, if true, the metadata of the datasets are retrieved from the data management system; if false, the data management system is federated, but its dataset's metadata are not retrieved.
- Refresh: This parameter is used to set the synchronisation period of the data management system (e.g. one hour, one day, one week).

Concerning the searching functionalities, these give access to the list of available datasets and allows to search among them by inputting keyword and/or selecting tags reported in the left column of the UI.

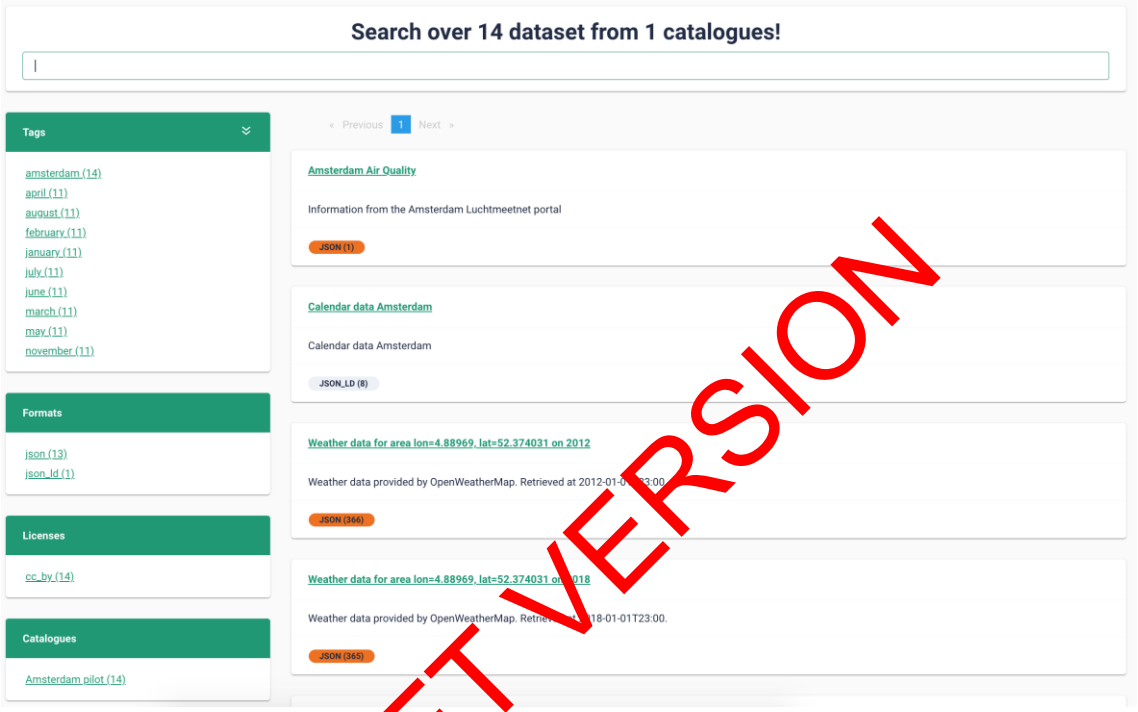

Figure 11 Data Catalogue – list of available datasets

The use can access the detail of a dataset by clicking on its title

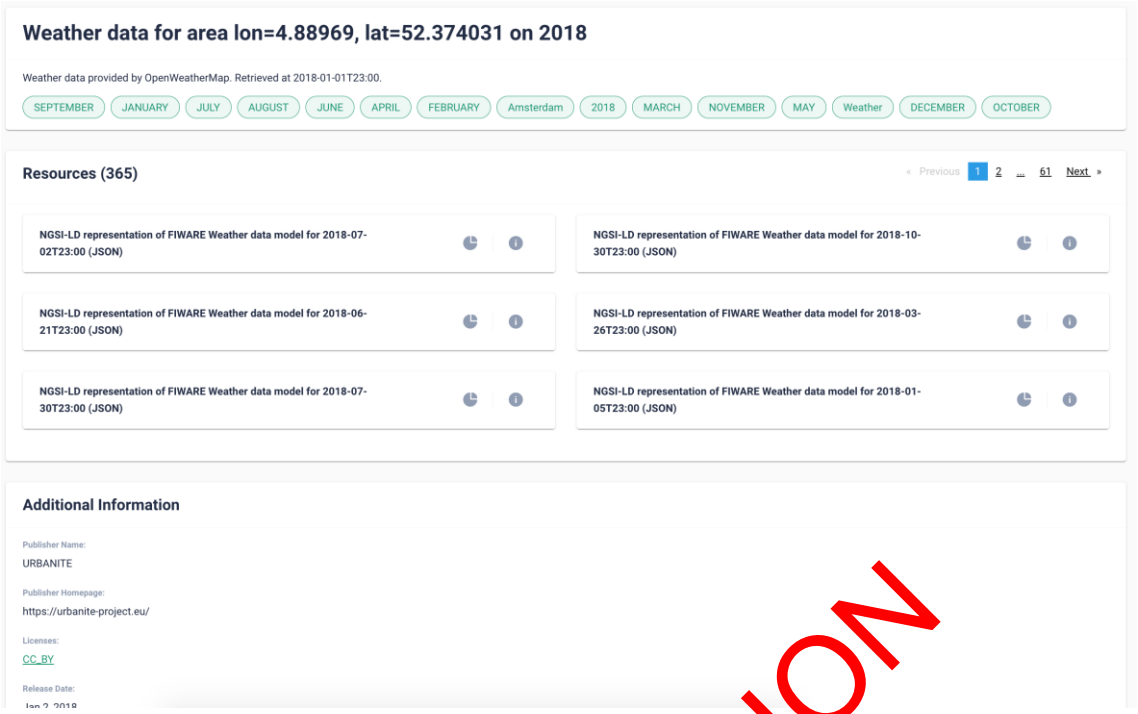

Figure 12 Data Catalogue – details of a dataset

Among the details, the user can find the general description of the dataset, its associated keywords, and the list of associated resources (the concrete data).

For each resource, the user can perform three operations

- Access the details of a resource and the download links (by clicking on the button)
- Access available charts created from the resource (by clicking on the button ; this button appears only if at least one chart is available)
- Create a chart from the resource (by clicking on the button)

Starting the creation of a chart, the Data Catalogue opens a wizard that guides the users in doing this operation.

First, the user selects the fields of interest of the selected resource.

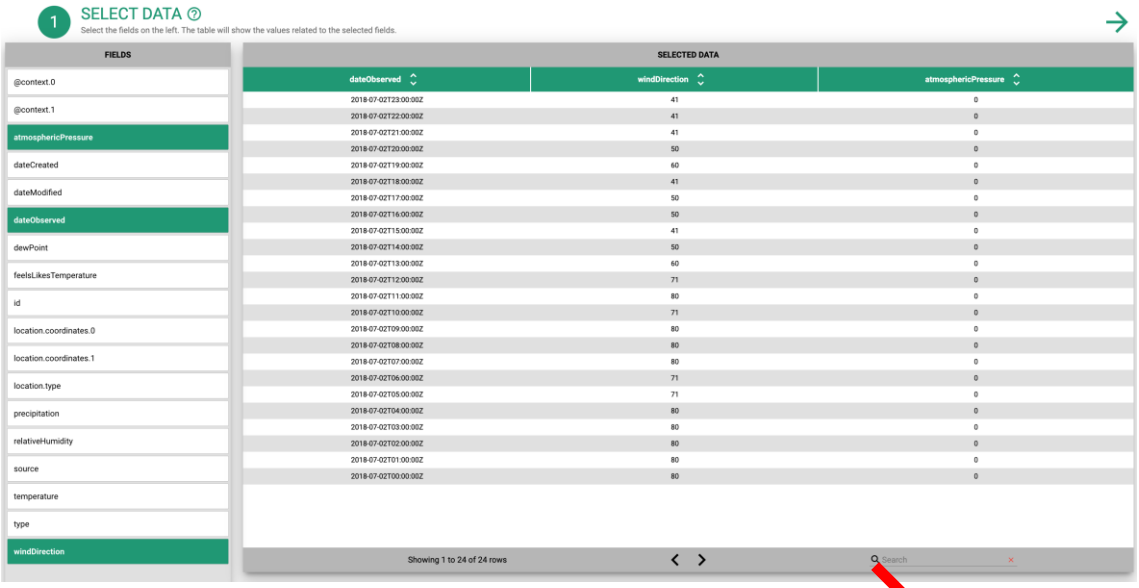

Figure 13 Data Catalogue – chart wizard; first step "fields selection"

Second, the users select the desired type of chart among different options (bar chart, line chart, spider charts, map, etc.) and associate the fields previously selected to the parameters of the chosen chart. By doing so, the wizard generates a preview of the chart.

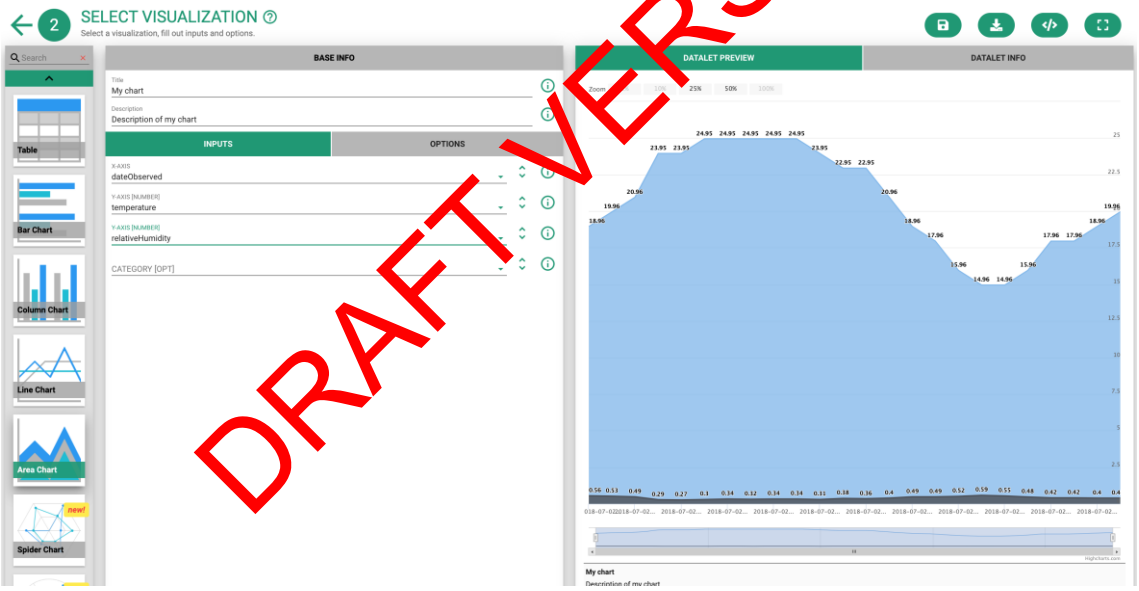

Figure 14 Data Catalogue – chart wizard; second step "chart definition"

3.2 URBANITE data analysis, simulation, and recommendation layer

3.2.1 Advanced Visualisation

The advanced visualisation component includes several visualisations developed for the UUI and the advanced visualisations provided as developed widgets for the external tool Orange[1], a data

[1] https://pypi.org/project/Orange3/

visualization, *machine learning* and data mining toolkit [7]. These are presented in the next paragraphs.

The visualisation components, developed for the UUI, include:

- Map based visualisations. These present data as layers on the map of the city. Several layers have been developed. Most of the KPIs of all four pilots are presented as map layers over the road network.
- Animated vehicle movements. The first version of a map layer showing simulated vehicle movements as an animation was developed and it allows an intuitive overview of the traffic simulation.
- Policy comparison visualisations, including 2D and 3D charts and graphs, have been improved.
- Decision support system visualisations have been added, visualizing the decision models in detail and the results of comparisons.

The widgets, developed for Orange, include:

- A group of widgets for data access, a generic one for accessing the URBANITE data platform, a widget for traffic data and a widget for accessing the data required for simulation calibration.
- Several widgets, developed for interaction and analysis of traffic-related data specifically.

These widgets are made compatible with existing ones. This enables the use of preexisting widgets on URBANITE data. A large ecosystem exists of widgets for Orange, that can be used on the harvested data.

3.2.1.1 Main functionality

The main functionality of the advanced visualisation component is to visualise the traffic data, traffic simulations related data and their results. The map-based visualisations present in an intuitive and clutter free way the local values of KPIs, traffic flows, congestions, etc. These can be moved, zoomed, and show detailed values when selected. An animated visualisation of simulated vehicle movements allows for a simple and intuitive overview of the simulation results.

The widgets developed for Orange main functionality is visualising the data and analysis, allowing interaction with intermediate and final results. Custom and predefined analysis pipelines can be built and run by the users, exposing a powerful data analysis toolbox.

3.2.1.2 Evolution and final status

The initial visualisations were limited in scope due to the initial status of other modules. Since then other modules, such as the traffic simulation module and decision support system, have been further developed. The related visualisation has been developed, including:

- New version of the map based KPI visualisations.
- Improvements to the visualisation of vehicle movements.
- New version of the scenario comparison visualisation and addition of several other visualisations for the decision support system, including comparisons of specific KPIs and the decision models in high detail.

In the last year, the Orange widgets described in this section have been developed. These have been developed to support interactive data analysis and will be further improved until the end of task T4.1.

3.2.2 Exploratory Data Analysis

The exploratory data analysis module includes the components for prediction and regression, self organising map, clustering, and projection. The module's main function is to provide interactive and visualisation supported data exploration and analysis. Each of the components of the module provides functionalities that together present the user with a powerful data analysis toolkit.

The modules were developed during the last year, after considering several possible solutions and tools. The Orange framework was chosen for its simplicity and intuitiveness while providing a robust set of tools allowing for complex analyses. The widgets were developed to enable the use of this toolset.

3.2.2.1 Main functionality

The primary role of the exploratory data analysis modules is to enable users to interact with the data using methods for data clustering, projection, prediction, and regression as well as self-organising maps.

The clustering methods are used within the URBANITE ecosystem to create different traffic simulation calibration datasets that represent different types of situations. Examples include traffic patterns on a rainy day, sunny days, weekends, or workdays.

The projection methods are used to reduce the dimensionality of machine learning modules, used by the URBANITE decision support system, among other uses. This allows to build models faster at little expense of accuracy, often preventing overfit. These methods are also used to visualise high dimensional data.

Self-organising maps visualise data in a way that keeps more similar data points close together. They can be used to visually cluster data and understand which attributes distinguish between different subsets.

Regression and prediction methods allow the estimation of expected future values of some variable or variables, and classification is a related process that categorises the input values into predefined categories. These methods are used to estimate traffic counts based on some attributes, e.g. weather and time.

These methods are commonly used together to perform certain tasks. The URBANITE traffic simulation module uses clustering and classification to build training datasets. Classification and prediction methods are used in the URBANITE decision support system. The module developed presents the users with the capability to perform a similar analysis.

3.2.2.2 Evolution and final status

The exploratory data analysis module was designed after considering several alternative possible solutions. These included uses of different tools to provide the required functionality, such as WEKA[2] [7] and Orange, or developing the functionality from scratch. Orange was chosen,

[2] https://www.cs.waikato.ac.nz/~ml/weka/

and the designed exploratory data analysis module was considered by the consortium. The module was developed recently and will be further improved until the end of task T4.1.

3.2.3 Traffic Simulation

The traffic simulation module implements the policy simulation and validation, recommendation engine and utilizes machine learning models. The policy simulation and validation engine is built on top of microscopic traffic simulation, components for policy representation and algorithms for KPI estimation. Traffic simulations have been upgraded in the last year, including improved population models and travel demand models, the addition of calibration algorithms and KPI calculation algorithms. Specifically, for the Helsinki and Messina use cases, travel demand models have been improved to include the harbor-based traffic. In the last year, some of the scenarios for simulations have changed, as have some of the KPIs required, which is reflected in our work. For each of the pilot cities, specific simulations were built and improved upon by including more data harvested.

3.2.3.1 Main functionality

The main functionality of the traffic simulation module is to represent the proposed policies as simulations and simulate both the baseline and proposed scenarios. These can be analysed and further processed by the decision support system. The performed simulations are used by machine learning models that learn the connections between traffic patterns and the represented policies. Parts of the traffic simulation system are:

- Microscopic multi-modal traffic simulation. Some of the functionality was extended, such as public transport simulation, parking, and air pollutant emission estimation.
    - Traffic simulation calibration, with calibration data sets created for each pilot. These can also be created using the exploratory data analysis module.
    - Population model defined for each pilot. Three different methods of population modelling are used in different pilots.
    - The travel demand model is defined for each pilot. Two different travel demand modelling methods are used in the pilots and others.
- Recommendation support system using multi-attribute decision analysis (MADA) methodology to analyse and compare different policy proposals.
    - KPIs were defined for each pilot and developed based on simulation results.
    - Decision models were created for each pilot, utilising the estimated KPIs.
    - FEXi, an open source MADA toolkit was integrated.
- Machine learning modules used the simulation results, outcomes of the recommendation support system and encoded proposed policies to learn connections between policy and traffic patterns.

3.2.3.2 Evolution and final status

The first version of the traffic simulation module supported basic traffic simulation and some KPI estimations. The population models and travel demand models have been upgraded and additional data included. At the same time, the supported simulation capabilities were expanded. The simulations are still being improved upon and several additional data sources will be included when access is granted.

We have improved upon the initial decision support system. Visualisations of decision models, KPI values and alternative comparisons have been developed.

The machine learning modules utilise the simulation results and decision support system outcomes. The initial version has been developed in the last year and will be further improved upon.

3.2.4 Analytical Framework

Within the context of the WP4 and as an analysis for helping the end user to make decisions in order to make the policies needed to improve the aspects considered in each pilot, some modules have been developed.

The details about these modules are described in the corresponding deliverables of the WP4 [8] and in the one which describes the implementation of the use cases [3].

3.2.4.1 Traffic Prediction

3.2.4.1.1 Main functionality

This module performs heuristic prediction for the traffic flow at a given sensor location within the city. It has the capability of performing prediction for the following 24 hours or 7 days considering the following variables:

- Calendar data
- Holiday data
- Weather data
- Event data: football games (Bilbao), ferry arrivals (Helsinki)

In addition to the prediction of the flow, the uncertainty of the predicted value is also provided by the algorithm. This module is implemented for the Bilbao and Helsinki use cases.

3.2.4.1.2 Evolution and final status

A major architectural change has been performed in this module; in the previous version, the module acquired all the historical data from an internal database. In the current version, the data is stored in a binary file (pickle format) where the data has been aggregated and fused with other data ready to be used by the analytical processes. Before deciding on choosing this binary file to store the data within the module it was considered to get the data directly from storage and to perform the aggregation and fusion direction at the beginning of the processing stage. These two steps turned out to be too demanding to be performed in real time and it was decided to perform these steps beforehand. In the final version of the module, this module is planned to have a mechanism to update this file in an automatic way without having to do the aggregation and fusion manually.

From the algorithmic point of view, it was decided to settle to use Random Forest due to its flexibility and robustness and the “month” was removed from the list of features considered because it was feared that data might not have enough depth and with the hope that its dependence is encoded in the rest of the features, specifically the weather variables.

3.2.4.2 Bike Analysis

3.2.4.2.1 Main functionality

This module computes the most likely values for the origin-destination matrix as a function of time for the following 24 hours or 7 days, considering the same variables as in the Traffic Prediction module. This module is implemented for the Bilbao and Helsinki use cases.

3.2.4.2.2 Evolution and final status

The same architectural change as in the Traffic Prediction module has been performed for the same reasons. Random Forest also has been chosen as the algorithm by default to compute the estimations.

3.2.4.3 Bike Trajectories

3.2.4.3.1 Main functionality

This module allows to visualise the most common points for the trajectories of bicycles rented from the Bilbao bike city service at a given date.

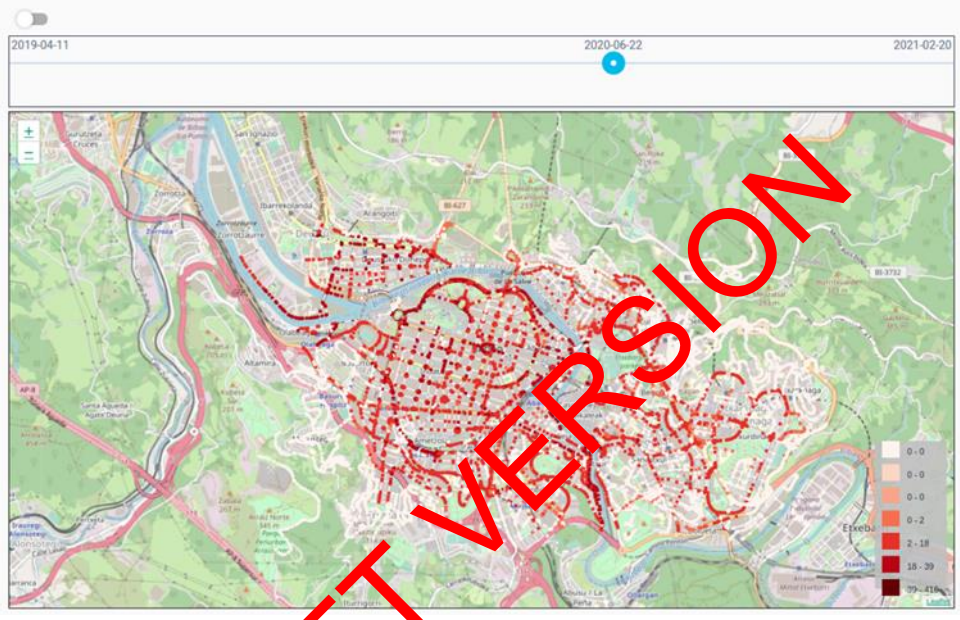

Figure 15.: Image of the result provided by the Bike Trajectories Module

Each of the points within the road network is coloured depending on the number of trajectories that visit that given point on the chosen day.

3.2.4.3.2 Evolution and final status

Cleaning the trajectories to obtain a set of points from the road network is a very demanding process that needs to be performed in advance. The current version of the module has all the dates within the history of the dataset pre-processed. The final version will be incorporate a mechanism to update the available days as the data set of trajectories increases within the URBANITE storage.

3.2.4.4 Bus OD

3.2.4.4.1 Main functionality

This module's goal is to compute the origin-destination matrix for the bus service in the use case of Bilbao. It has two different modes of operation:

- Per Line mode: This mode computes the number of people that leaves the bus at every stop for a given bus line and a given entrance stop.

- Per Trip mode: This mode computes the number of people that finish their trip at every stop, starting from a given starting stop independently of the bus line or lines (considering possible transfers) chosen for the trip.

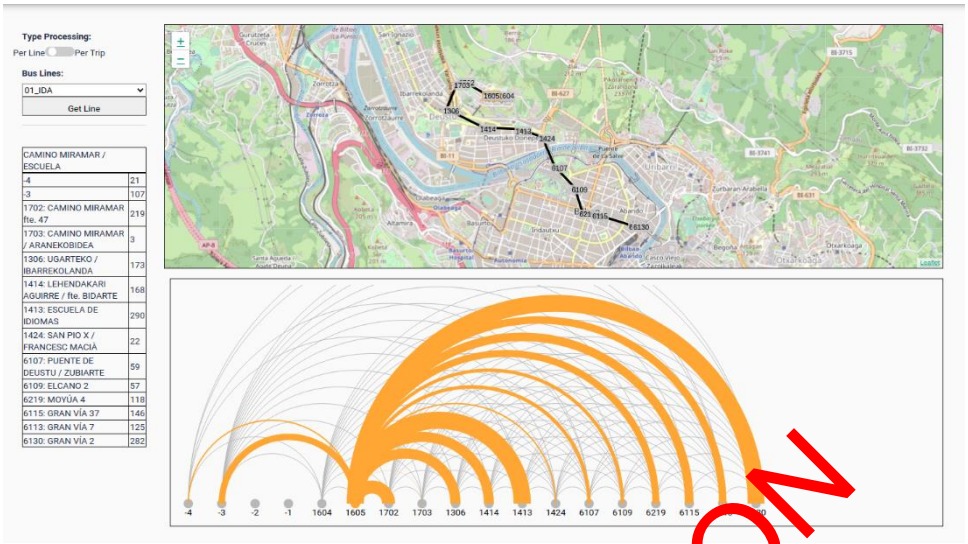

Figure 16.: Image of the result provided by the Bike Trajectories Module (per line mode)

3.2.4.4.2 Evolution and final status

This is a completely new module that was not on the platform in the previous release. At the time this document is being written, there is only one month of data available that is directly pre-processed within the module. The final version of this module will be able to process data from the storage periodically to include these new data in the computation of the origin-destination matrices.

3.2.4.5 Global Traffic Prediction

3.2.4.5.1 Main functionality

This module allows to visualise the results of the traffic prediction module on the per city basis instead of per location basis. In the contraposition of the Traffic Prediction module, the features can be freely chosen by a set of controls that allows the user to visualize the state of traffic in the whole city given a set of features. This module relies completely on the traffic prediction module. This module is implemented for the Bilbao and Helsinki use cases.

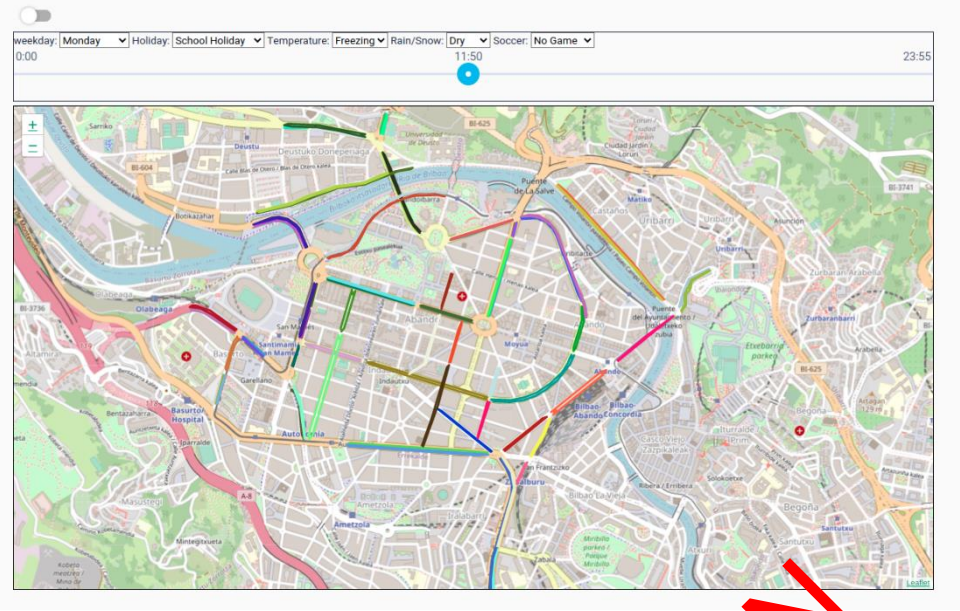

Figure 17.: Image of the Global Traffic Prediction Prototype

The above image presents the prototype for this module, in which the different features can be chosen among then: the hour of day, day of the week, weather characteristics and events.

3.2.4.5.2 Evolution and final status

The visualisation side of the module is ready for the moment when the Traffic Prediction module is complete, and it can feed this one.

3.2.4.6 Traffic Evolution, Weekly Traffic Flows and LPT Critical areas

3.2.4.6.1 Main functionality

These are not formal modules, but visualisations built leveraging the URBANITE UI integrated libraries and data from Messina use case. The definitions of them are described in D6.2 [3].

All of them present information in a dashboard built upon a particular kind of information:

- Traffic Evolution: suburban roads that connect the city of Messina with the surrounding towns
- Weekly Traffic Flows: same information as the previous one but under a different perspective. In addition, it allows the user to select two different roads to compare.
- LPT Critical areas: Bus Public Transportation dataset. The user can identify the areas of the city of Messina where public transportation vehicles (i.e. busses) stop exceeding a certain time in a specific observation period.

3.2.4.6.2 Evolution and final status

The current status of these visualisations is included in detail in the corresponding deliverable D6.2 [3] as well as the work planned for them until the end of the project.

The pictures above reflect the aspect of the main page of the different dashboards:

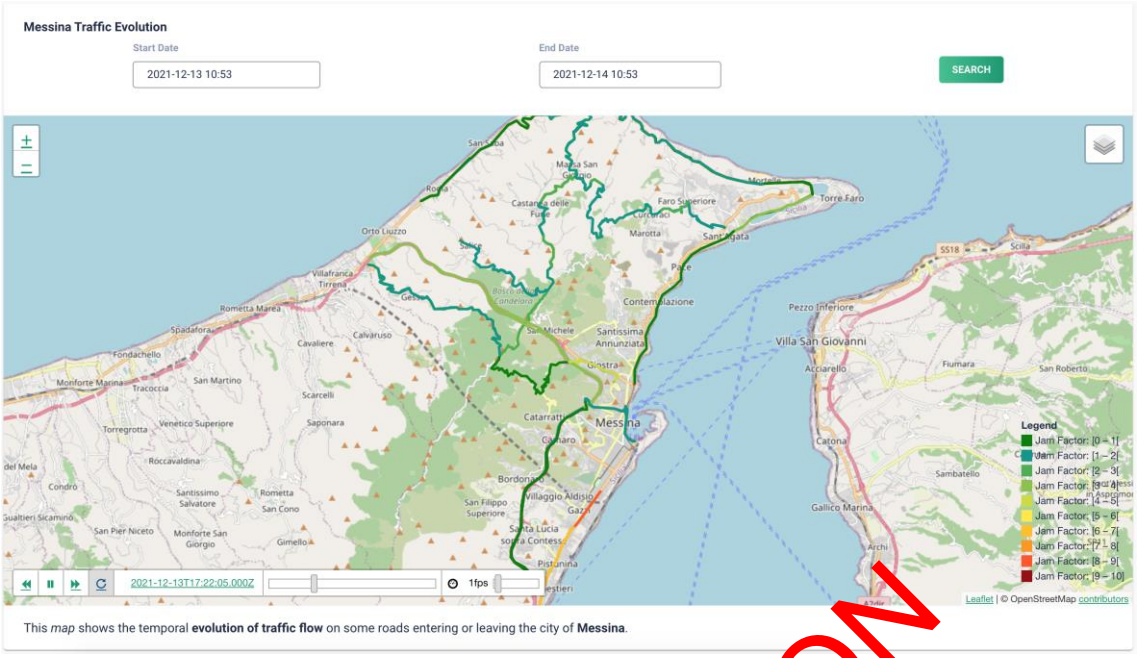

Figure 18 Messina traffic evolution dashboard.

Figure 19 Messina weekly traffic flows dashboard.

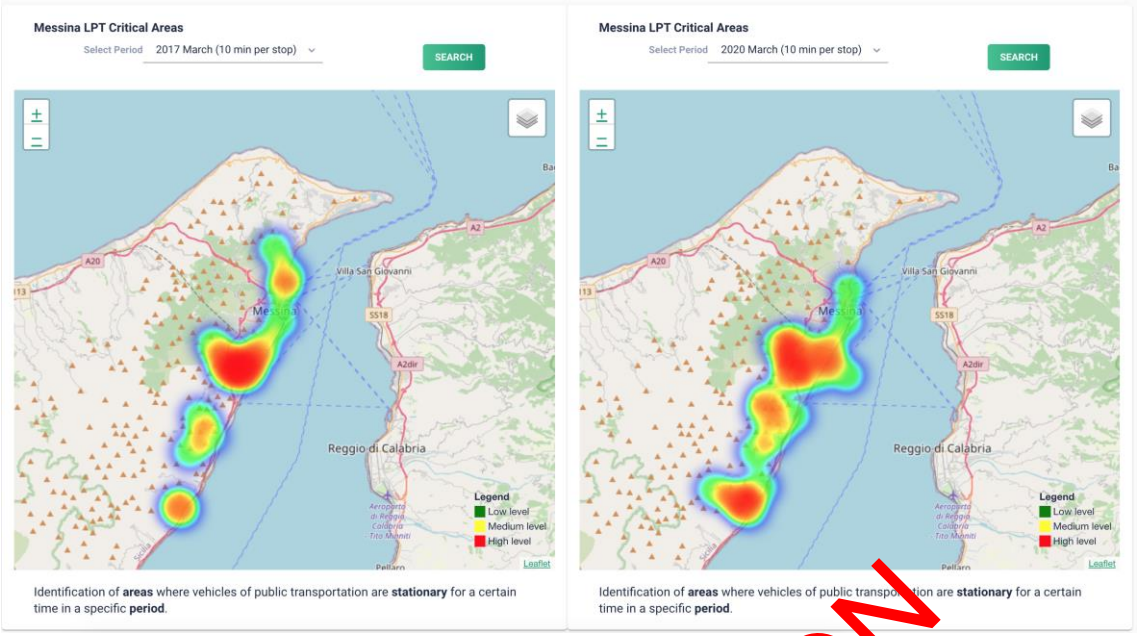

Figure 20 Messina LPT critical areas dashboard.

3.3 URBANITE virtual SoPoLab

The Virtual SoPoLab (VSPL), renamed URBANITE Forum[3] for the end users, is a web-based tool that offers a virtual environment where citizens can discuss needs and problems and propose challenges and solutions following a co-creation approach. VSPL is based on Decidim[4], an open-source platform (released under AGPL-3.0 license) for participatory democracy. Decidim is supported by a strong community and has already adopted by different cities, regions, and organizations[5].

3.3.1 Main functionality

The main aim of the VSPL is to enable and facilitate on-line collaboration among its users, following co-creation principles. Through the VSPL, the users are able to report needs and problems, launch challenges (e.g. to solve the reported needs/problems), and propose possible solutions to address them.

To do so, the VSLP allows to evaluate of proposed solutions and select the best ones that can be so further refined and implemented, thanks to the possibility to suggest refinements/improvements for selected solutions.

To improve participation and support interactions among the users, the VSPL allows to host different resources, such as documents reporting guidelines, methodologies, and best practices, and to exchange of information between the users.

3.3.2 Evolution and final status

This section offers an overview of the current status of the VSPL.

[3] https://forum.urbanite-project.eu/

[4] https://decidim.org/

[5] https://decidim.org/usedby/

The VSPL is mainly based on the Assembly module provided by Decidim; within Decidim, an Assembly is a virtual space that can represent bodies such as Municipal Councils or City Assemblies.

Assemblies can include different components (Table 1), the building blocks enabling participatory processes.

Table 1: Overview of main components for assemblies.

Component	Distribution
Accountability	This component provides the level of execution of the results of the assembly.
Blog	This component is used by the administrator to provide useful insights about the assembly.
Budgets	This component provides the means to create participatory budgeting processes.
Debates	This component act as a *forum* allowing users to open questions or discussions within the assembly.
Meetings	This component is used to schedule online or in-person meetings.
Page	This component allows to add informative pages with[illegible] the assembly.
Proposals	This component provides the means to create ideas and challenges, the so-called **proposals**, within the assembly. Proposals will be further evaluated by the administrators.
Sortitions	This component allows to randomly select a group of users from the proposals to create a pool of candidates to build, for instance, a committee
Surveys	This component allows to create surveys.

Assemblies can also contain attachments (managed by a user with an administrator role, which can upload files or create folders to group files).

Comments provided by the users in the assemblies can be moderated. Users can report comments. All the reports provided are listed in the *Moderations* section of the VSPL. The administrator is in charge of evaluating the report and choosing between hiding the content or deleting the complaint.

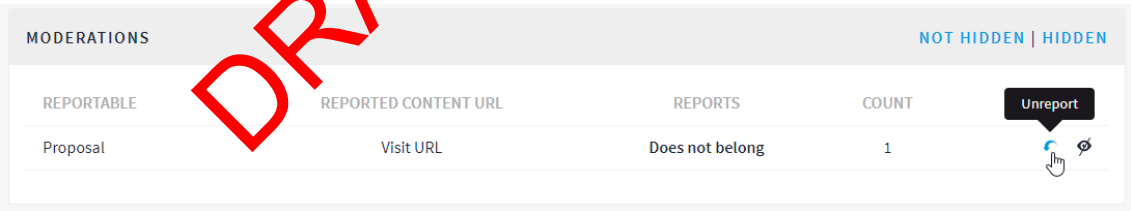

Figure 21 Virtual SoPoLab – moderate content

Once a user accesses an assembly, its general information is displayed along with the related documents and folders attached (the administrator of the assembly can upload file or can create folders to group files).

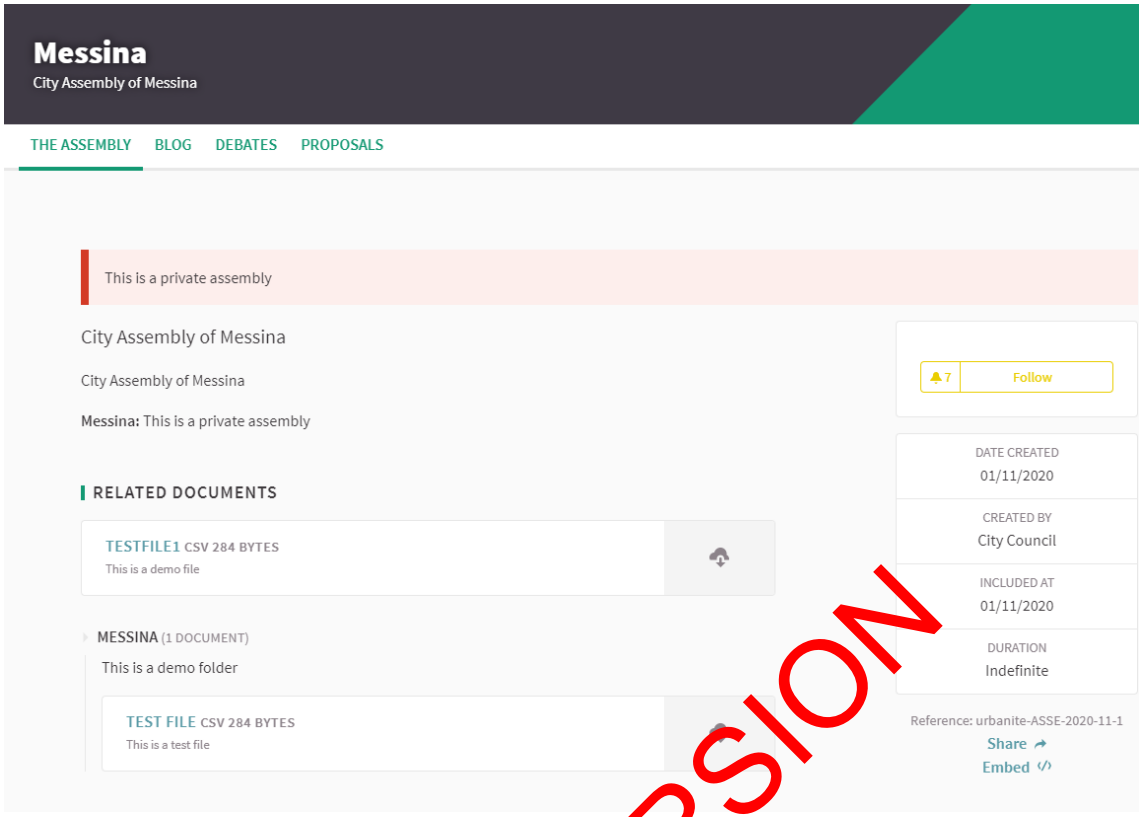

Figure 22 Virtual SoPoLab – general information of an assembly

The navbar contains the enabled and configured components.

Debates

To access the debates, the user clicks the Debates link in the assembly's navbar. To contribute to a debate, the user can comment on it through the dedicated comment section.

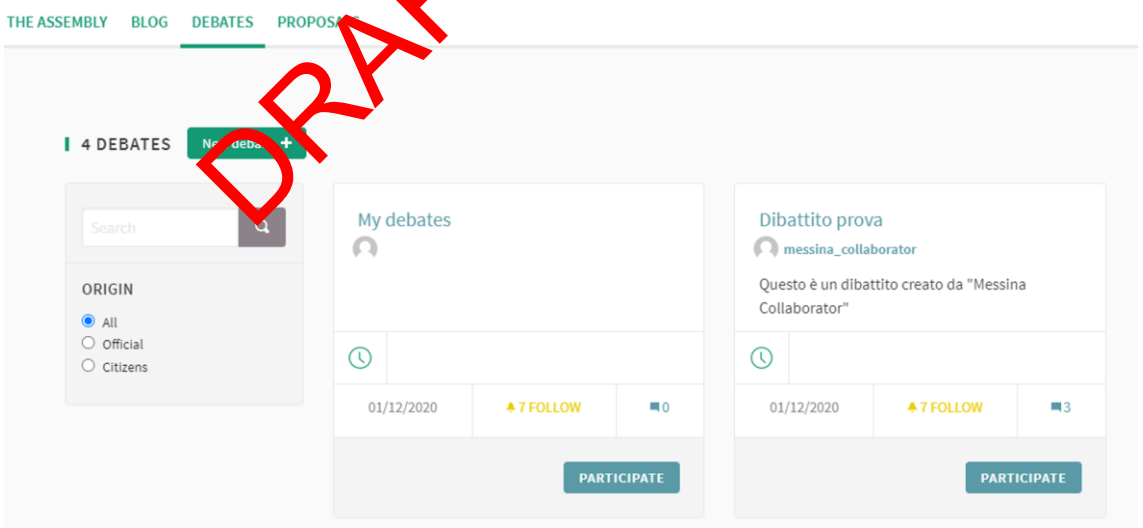

Figure 23 Virtual SoPoLab – debates

The user can access the details of a debate by clicking on the *Participate* button

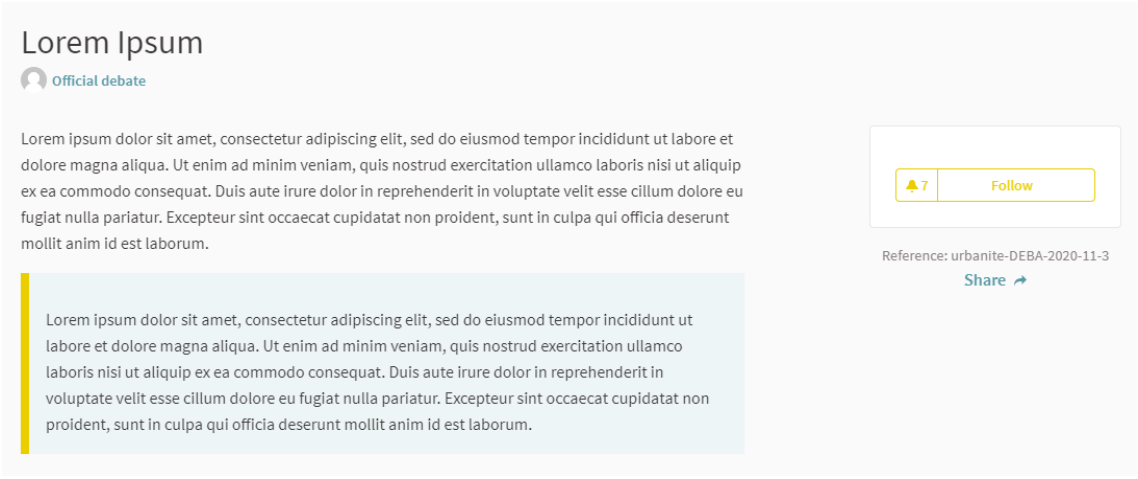

Figure 24 Virtual SoPoLab – details of a debate

The user can create new debates by clicking on the *New Debate* button; the user is prompted with a form to provide the title and the content of the new debate.

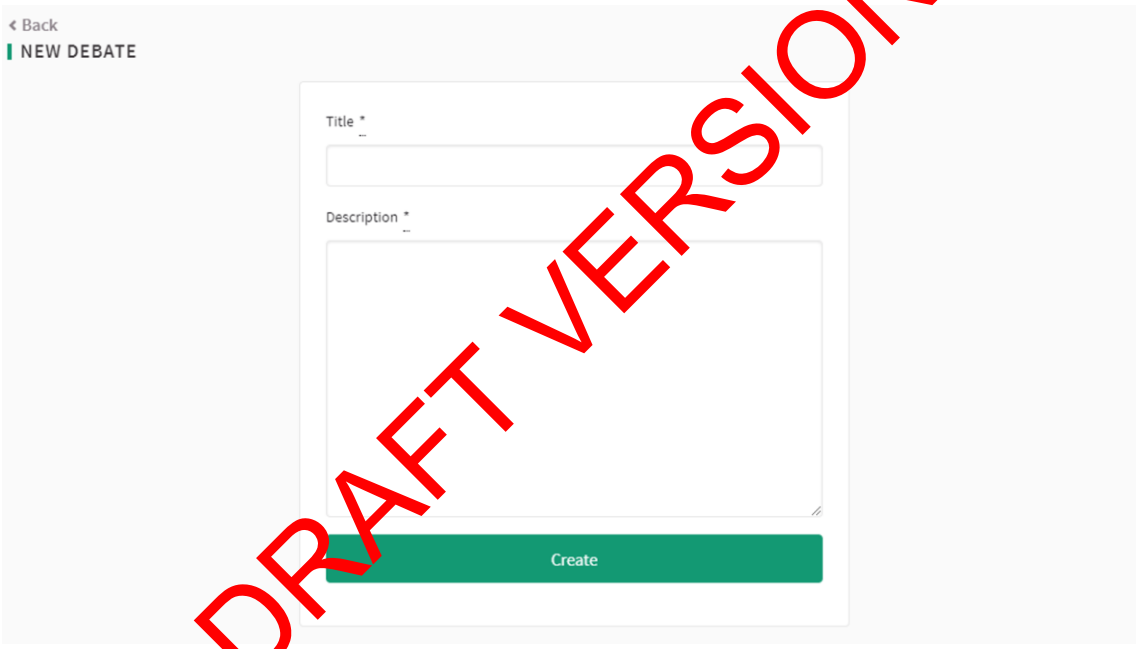

Figure 25 Virtual SoPoLab – new debate

Proposals

To access the proposals, the user clicks the Proposals link in the assembly's navbar. The user is provided with the available proposals and can access their details by clicking on their title.

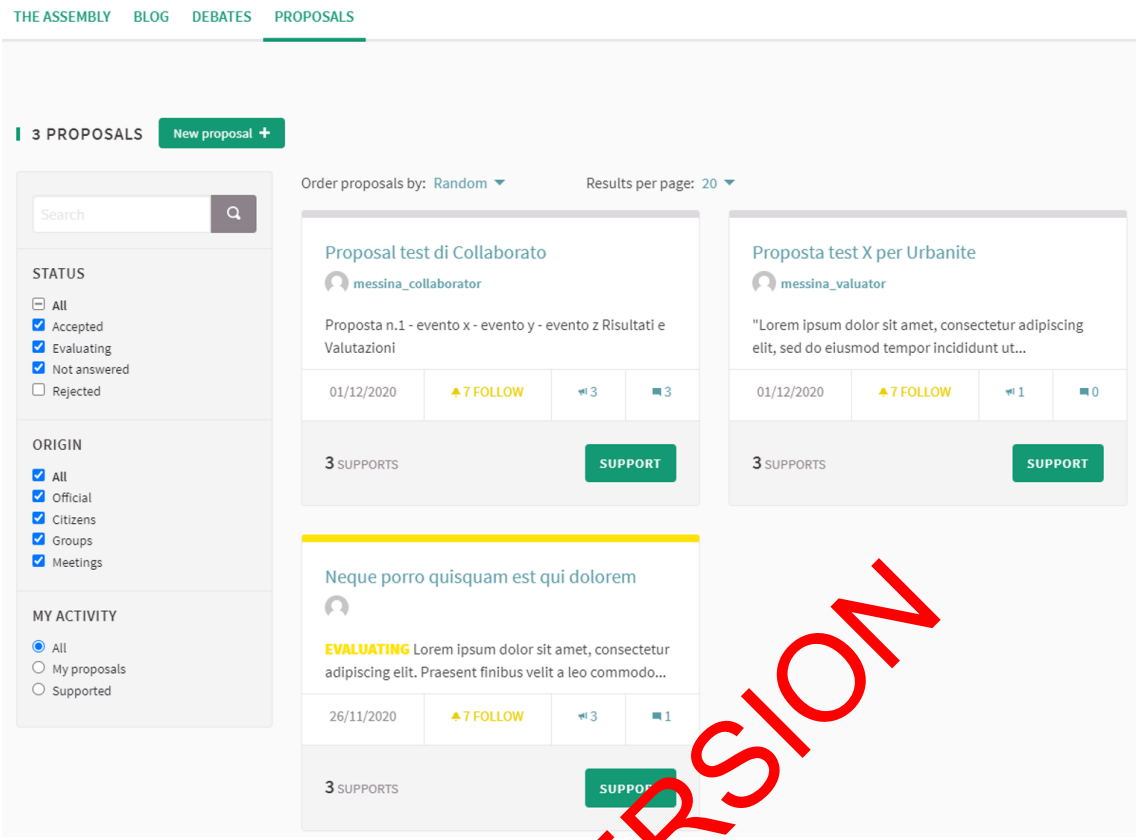

Figure 26 Virtual SCPoLab – proposals

The user can also support a proposal by clicking on the *Support* button or comment a proposal through the dedicated comment section.

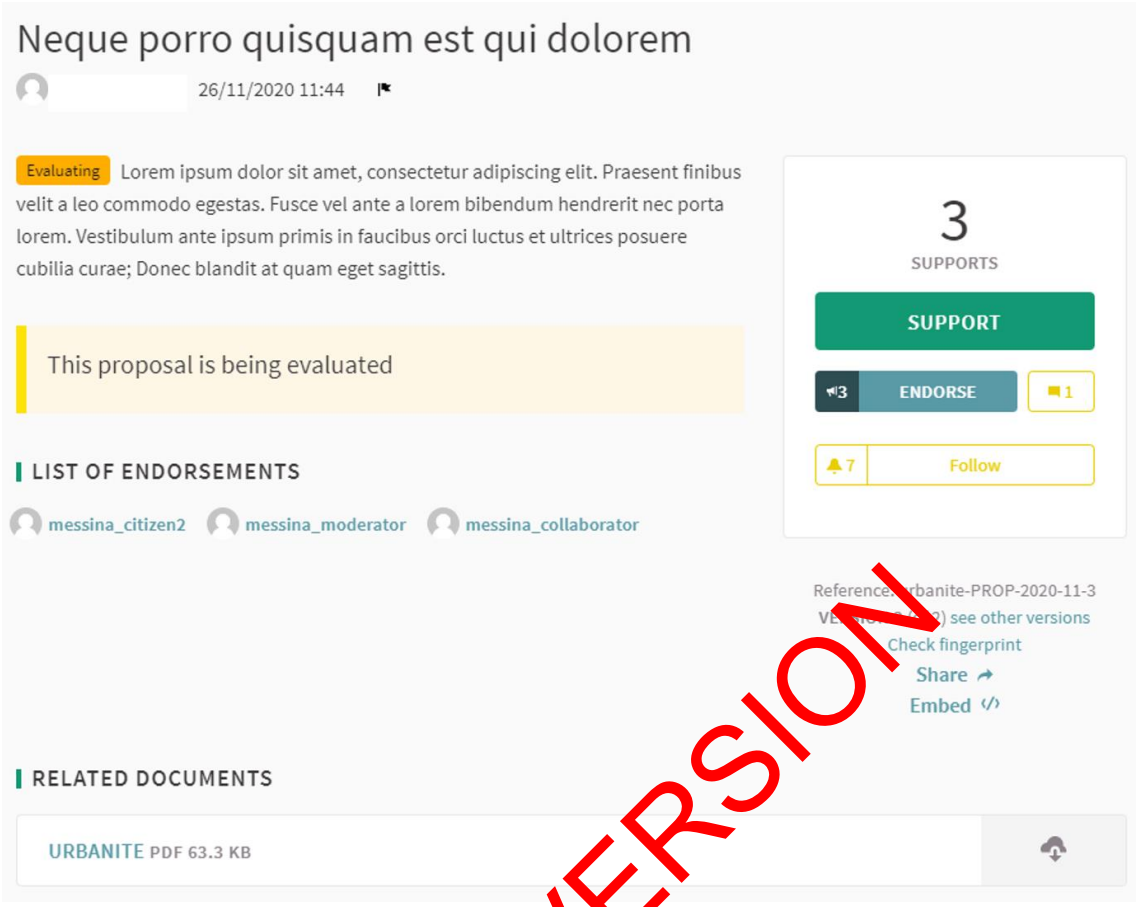

Figure 27 Virtual SoPoLab – details of a proposals

Through the *New Proposal* button, the user can create new proposals by providing the required information through a dedicated form.

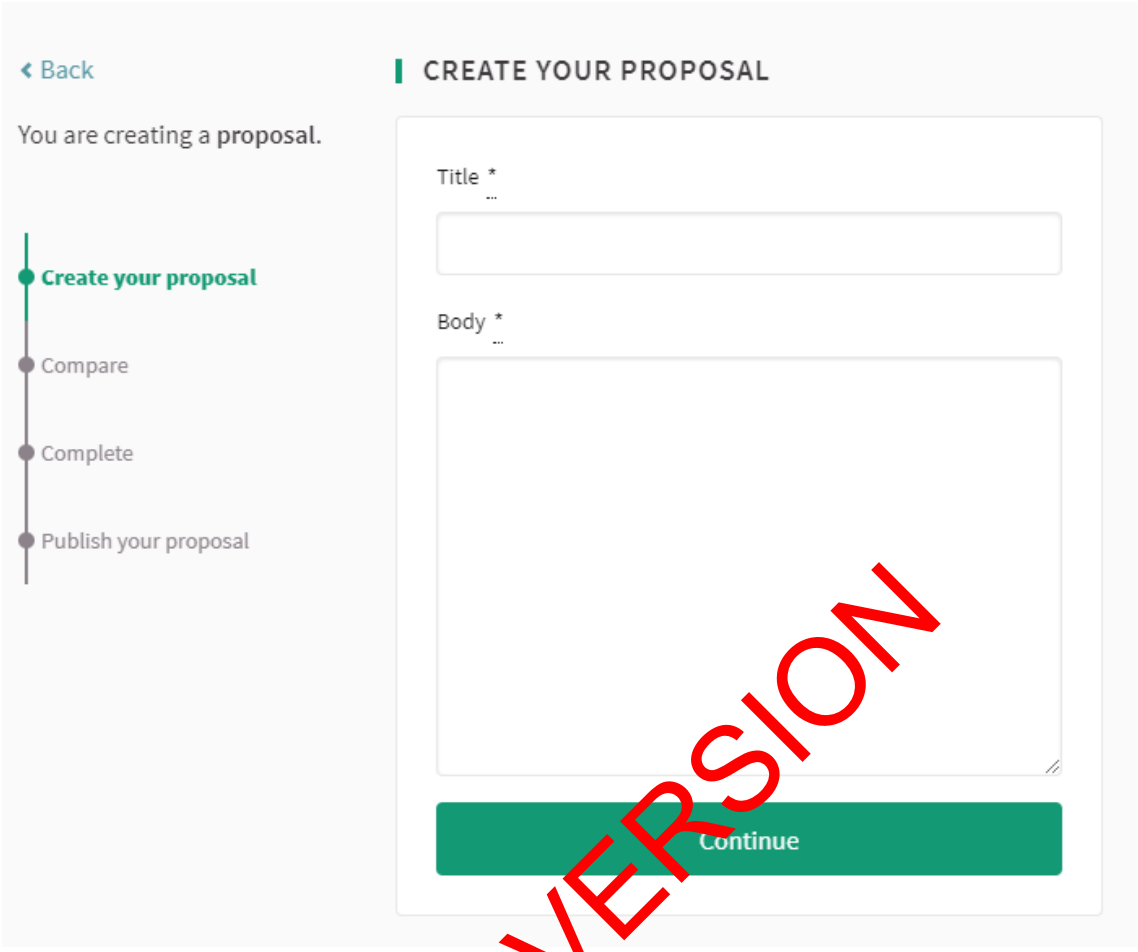

Figure 28 Virtual SoPolis – create a proposal

3.4 Integrated URBANITE UI

The URBANITE UI is the main interaction point between the URBANITE Platform and the end users. It is conceived as an integration framework at the UI level, and acts as a wrapper of the different components of the platform.

For this purpose, the URBANITE UI is accompanied by a UI template (to facilitate the development of new parts of the UI), integration strategies, a CSS[6], and in general, a complete design system (to guide the integration of an existing UI into the URBANITE UI).

A design system is a collection of reusable components with a standard style, that also provides best practices to build a scalable UI. The design system of the URBANITE UI is based on Eva Design theme, which has been customized starting from the colours of the URBANITE Logo and adopts the Roboto font. The main colour is green (hex color #139973).

From a technical perspective, the URBANITE UI is based on different tools and technologies; these are summarized in the *Table 2*

Table 2: Overview of tools and technologies for the URBANITE UI.

Name	Description

[6] Cascading Style Sheets

Angular	This is a JavaScript front-end framework. It is used to build efficient single-page applications.
Nebular	This is a library for Angular that offers reusable components and features to simplify the development of Angular based applications. It is based on Eva Design System.
ngx-admin	This is a dashboard based on Angular and Nebular.
Leaflet	This is open-source JavaScript library for the management of interactive maps.
ngx-leaflet	This is a library for Angular that provides components that allows to integrate Leaflet into Angular based applications.
Chart.js	This is open-source JavaScript library that provides features for creating and managing charts.
angular2-chartjs	This library for Angular that provides components that allows to integrate the Chart.js library into Angular applications.
Bootstrap	This is a front-end UI toolkit that allows to build a responsive UI.
Fontawesome	This is an icon set that provides several icons ready to be used in a UI.
Angular Material	This library for Angular that provides Material Design components for Angular based applications.

3.4.1 Main functionality

The URBANITE UI is composed of three main elements: a central panel that provides the user interface of the accessed functionality, a left column that provides the menu of the available functionalities, and a top bar that provides a button to resize the left columns.

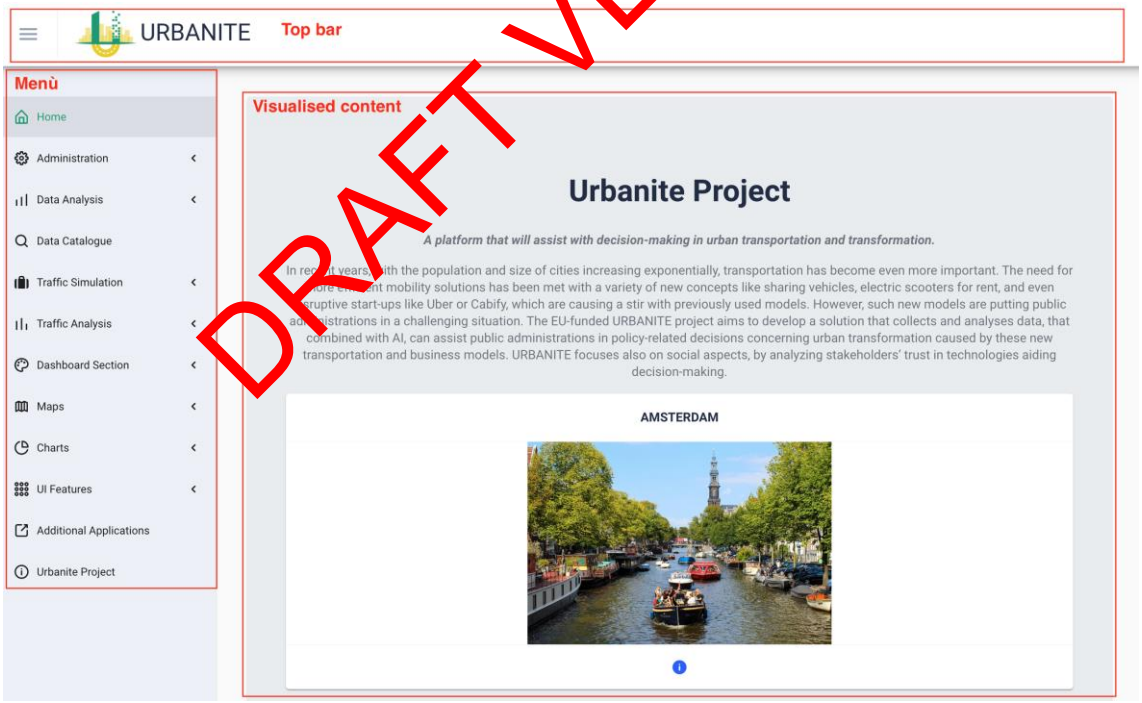

Figure 29 URBANITE UI – main elements

The functionalities reported in the menu on the left column can be customized.

3.4.2 Evolution and final status

Since the UUI is the entry point to the different functionalities provided by the URBANITE Ecosystem, its development supports the evolution of the rest of the components and functions that the whole platform gathers, for all of the use cases.

This section reflects the current status of this User Interface, although it can be modified to cover new requirements arising from the evaluations and deployments of the pilots.

When accessing the main screen of the URBANITE Ecosystem, the default menu includes the functionalities reported in the Table 3 :

Table 3: Overview of default functionalities of the menu of the URBANITE UI.

Menù item	Distribution
Home	Provides a link to the main page of the URBANITE UI
Administration	Provides access to the Administration tools of the URBANTE Platform, as for the management of the *Data Catalogue* and the *Identity/Authorization Management*.
Data Analysis	Provides access to the data analysis tools of the URBANITE Platform, as the Traffic Prediction, Bike Analysis, Bike Trajectories, Bus OD, and Global Traffic Prediction (described in section 3.2.4)
Data Catalogue	Provides access to the functionalities for the end-user offered by the Data Catalogue, mainly the search of the datasets managed by the URBANITE Platform and the creation of charts and graphs from them.
Traffic Simulation	Provides the access to the simulation capabilities offered by the Traffic Simulation
Dashboard Section	Provides access to the functionalities for the management of customs dashboard (create, modify, delete, share and clone), as well as the the management of customs menu items.
Maps	Provides a sample map to demonstrate the usage of the Leaflet and ngx-leaflet libraries
Charts	Provides sample charts to demonstrate the usage of Chart.js and angular2-chartjs libraries
UI Features	Provides sample templated for the organization of a UI.
Additional Applications	Provide an example of link to an external application.
Urbanite Project	Provides information about the URBANITE project.

Dashboard Section

Within the Dashboard Section, it is possible to access three functionalities

- Manage Dashboard Pages: this functionality allows the user to create, edit, delete, and share custom dashboards.
- Menage Menu Blocks: this functionality allows the user to create, edit and delete custom menu items on the left column.
- Clone Dashboard: this functionality allows the user to clone a dashboard shared by another user or a dashboard already created by the user.

Concerning the Manage Dashboard Pages, the UI first provides the list of existing dashboards

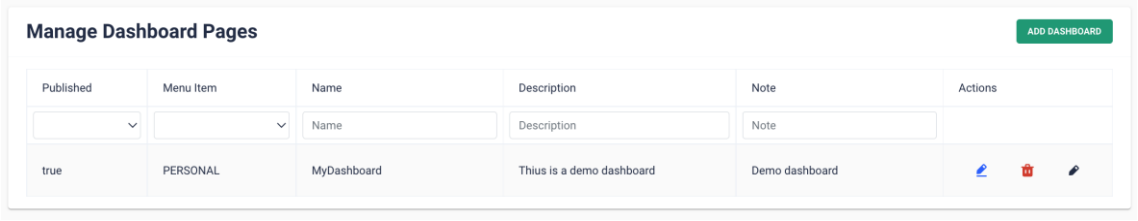

Figure 30 URBANITE UI – dashboards list

From this view, it is possible to create a new dashboard (by clicking on the button ADD DASHBOARD) or act on an existing one, by clicking on the button reported under the Acton column.

- The blue edit button allows to edit the metadata of a dashboard.
- The black edit button allows to edit the elements of a dashboard
- The red delete button allows to delete a dashboard

The UI to edit the metadata of a dashboard is represented in the figure below

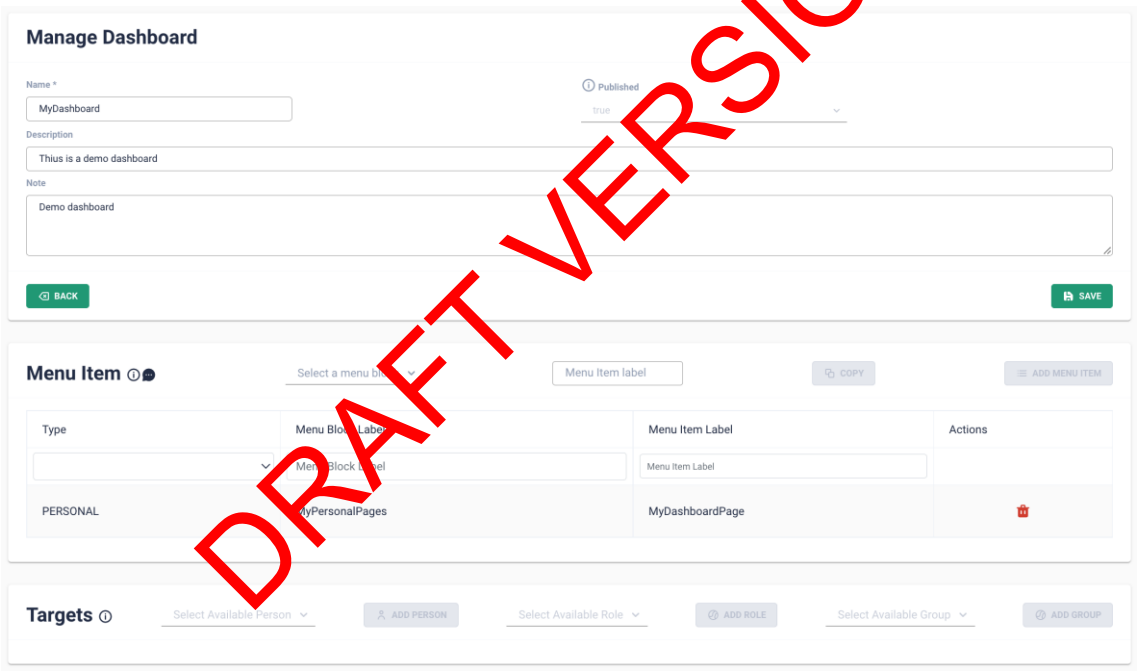

Figure 31 URBANITE UI – metadata of a dashboard

Within the "Targets" section, it is possible to select the user to share with the dashboard. In detail, it is possible to select a single user, a specific role (in this case, the dashboard is shared with all users associated with the selected role), or a specific group (in this case, the dashboard is shared with all the users belonging to the selected group). For this purpose, this specific functionality is integrated with the Identity/Authorization Management component.

The UI to edit the element of a dashboard is represented in the figure below.

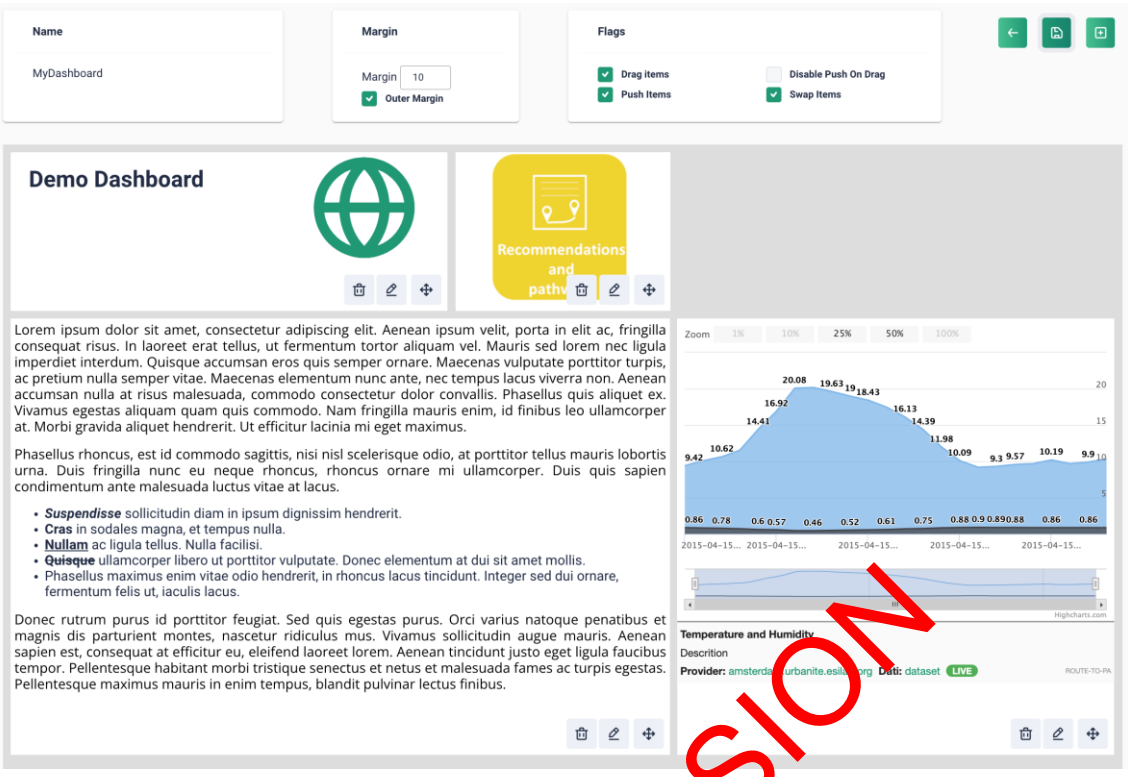

Figure 32 URBANITE UI - elements of a dashboard

Elements can be added by clicking on the button . Each element can be freely located in the dashboard and configured to include different kind of resources among

- Text: allows to insert free text through a text editor
- Image: allows to upload an image
- Chart: allow to include an existing chart created through the Data Catalogue
- Iframe: allow to include an iframe to integrate an existing web page
- Summary: allow to include basic information such as title, subtitle, etc.

Concerning the Menage Menu Blocks, the UI first provides the list of existing menu items

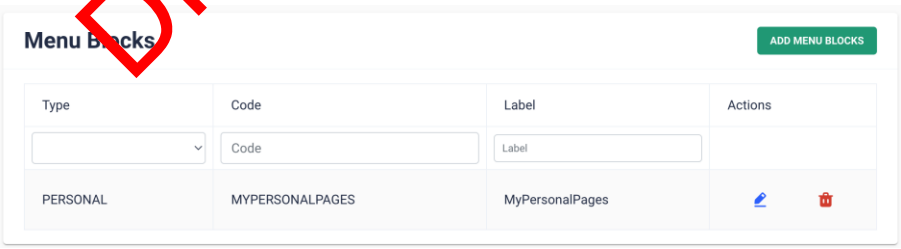

Figure 33 URBANITE UI - menu items list

From this view, it is possible to create a new menu item (by clicking on the button ADD MENU BLOCKS) or to act on an existing one, by clicking on the button reported under the Acton column.

- The blue edit button allows to edit the metadata of a menu item.
- The red delete button allows to delete a menu item

The UI to edit the metadata of a menu item is represented in the figure below.

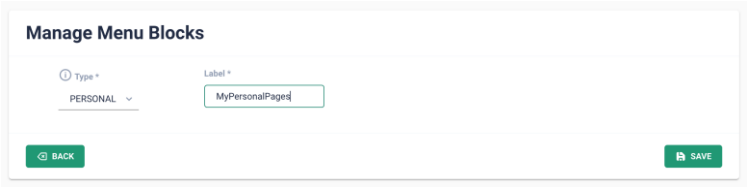

Figure 34 URBANITE UI - metadata of a menu item

Concerning the Clone Dashboard, the UI provide a wizard to clone available dashboards.

First, the user selects the dashboard to clone, choosing it from a dropdown menu.

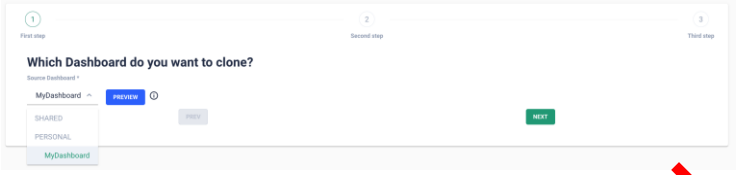

Figure 35 URBANITE UI - clone dashboards; first step "select a dashboard to clone"

By clicking on the button "Next", the user is guided to the second step, to input the title of the cloned dashboard.

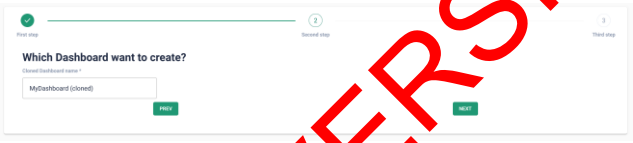

Figure 36 URBANITE UI - clone dashboards; second step "title of the dashboard"

By clicking on the button "Next", the user is guided to the third step, where the user has to confirm the cloning operation.

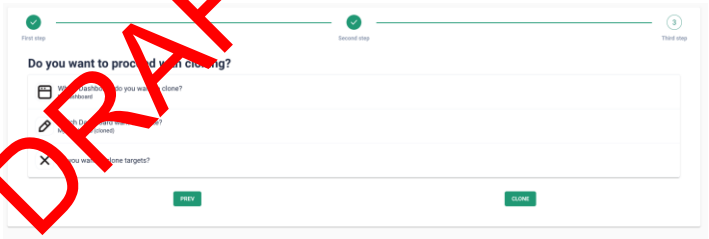

Figure 37 URBANITE UI - clone dashboards; third step "confirm"

The user can confirm the operation by clicking the button "CLONE". Doing so, the dashboard is cloned and made available.

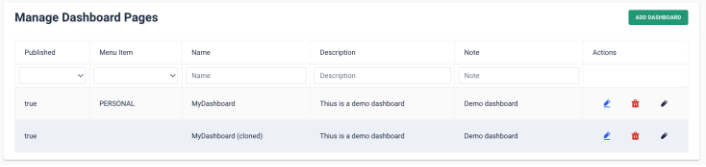

Figure 38 URBANITE UI - clone dashboards; first step "select a dashboard to clone"

3.5 Identity/Authorization Management

The Identity/Authorization Management implements the security functionalities of the URBANITE, securing access to the other components of the platform. It manages the procedures for the authentication and authorization of the users.

This component is based on the open-source tool Keycloak, an Identity and Access Management tool, that leverages well-adopted standard protocols (such as OpenID Connect, Oauth2 and SAML) and offers functionalities such as the single-sing on.

Furthermore, it allows the management of users and trusted applications (e.g. register, modify, or delete users and applications).

3.5.1 Main functionality

The Identity/Authorization Management offers a login page that is integrated into the general URBANITE UI.

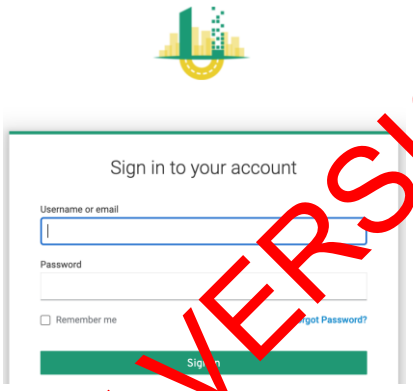

Figure 39 Identity/Authorization Management – login form

Furthermore, it includes an administration console from which it is possible to configure the tools and manage its behaviour, such as the configuration of realms, the registration of users and client applications, the management of their roles and the assignments to the users.

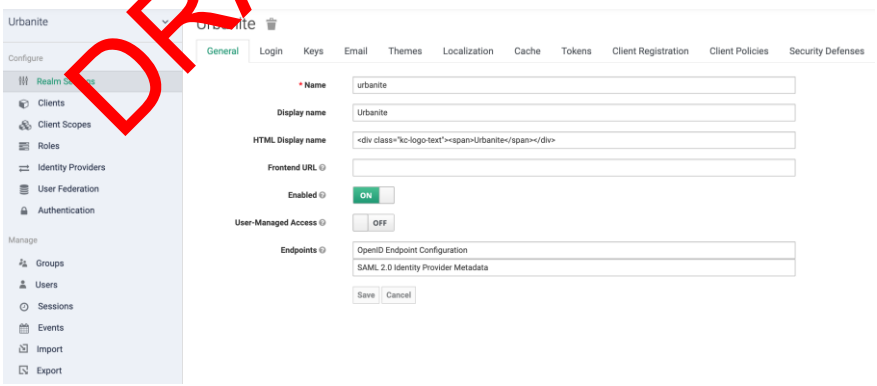

Figure 40 Identity/Authorization Management – administration console

3.5.2 Evolution and final status

No major changes have been made in this component.

The Identity/Authorization Management includes an administration console from which it is possible to configure the tools and manage its behaviour, such as the configuration of realms,

the registration of users and client applications, the management of their roles and the assignments to the users.

The Identity/Authorization Management can also acts as an Identity Broker since it allows the connection of different identity providers, such as social networks (e.g. Facebook, Google, Twitter, etc.) or custom identity providers based on supported protocols already mentioned, or to federate external user identity databases such as LDAP and Active Directory.

3.6 Controller

The Controller offers the possibility to orchestrate different kinds of tasks within the URBANITE Platform, that involve the components devoted to the analysis and processing of data.

It is based on Apache Airflow, an open-source tool that allows defining workflows following the principle of "configuration as code". In Airflow, each workflow is defined in the Python programming language.

The Controller, through the defined workflows, is able to interact with the components of the URBANITE Ecosystem.

3.6.1 Main functionality

The main functionality offered by the Controller is the management and execution of the workflows orchestrating the different steps in implementing a data processing pipeline.

The controller offers the possibility to manage and monitor the available workflow through its user interface. the Each workflow. For each workflow, the user has the chance to check if the related tasks succeeded, failed, or are still running.

3.6.2 Evolution and final status

The status of the Controller component remains similar that the one reported in the last version of this deliverable [2].

The current status of these components is described in this section.

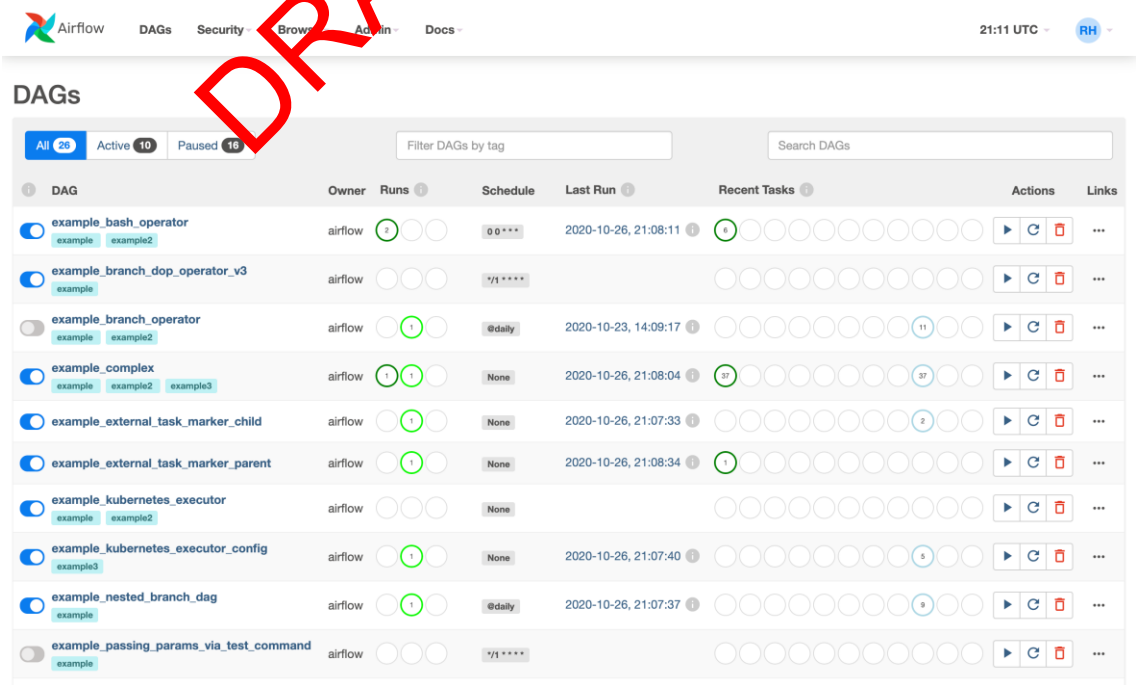

Figure 41 Controller – list of workflows

For each workflow, it is possible to access the relative details and views.

- Tree View: offers a tree representation of the workflow and of its steps that allows to easily identify the blocking ones.
- Graph View: offers a comprehensive view of the workflow, its dependencies and the current status for specific executions.
- Calendar View: offers an overview of your entire history of the workflow (months, or even years); this view allows to be aware of the overall success/failure rate of the workflow executions.
- Variable View: offers the possibility to access the key-value pair of a variable used during the execution of the workflows, allowing the user to create, edit or delete them. Value of a variable is hidden if the key contains keywords as “password”, “secret”, “passwd”, “authorization”, “api_key”, “apikey”, “access_token”).
- Gantt Chart View: offers the possibility to analyze task duration and possible overlap; this view allows to identify potential bottlenecks of the workflow.
- Task Duration View: offers the possibility to access information concerning the duration of the different tasks of a workflow over the past N executions; this view allows the user to identify the most time-consuming tasks.
- Code View: offers the possibility to access the source code of a workflow.

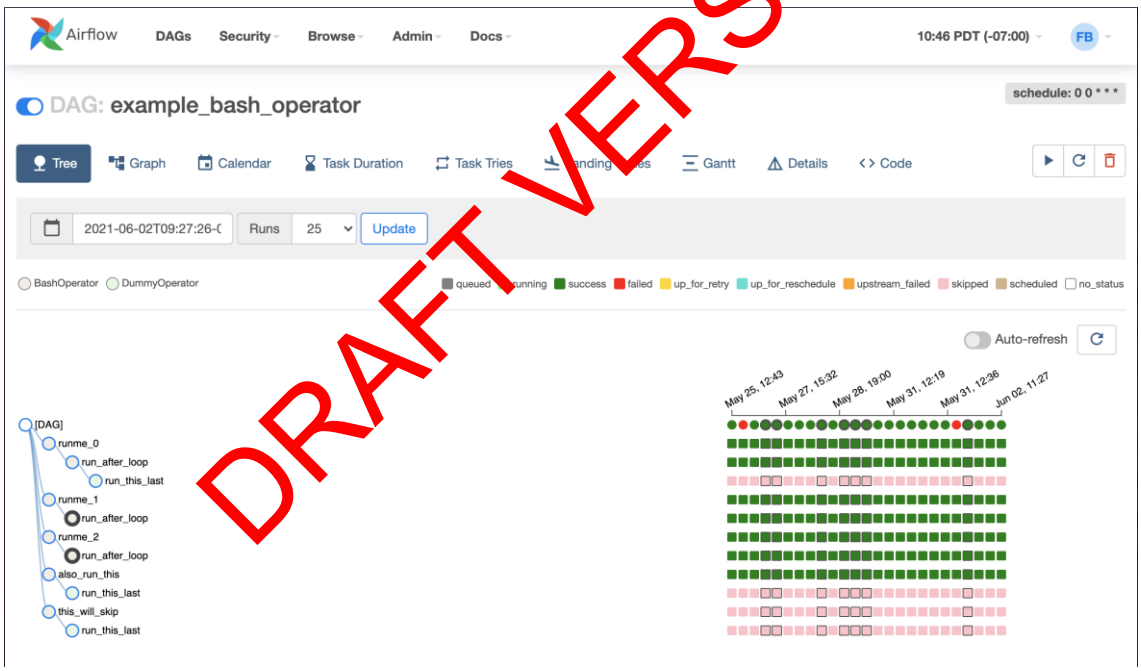

Figure 42 Controller – Tree view

4 Conclusions

This document provides a description of the entire global architecture of the URBANITE ecosystem and gathers the modifications made to the previous schema of the architecture, resulted as consequences of the evolution of the project since then.

The experience of releasing the first integrated Ecosystem in June 2021 contributed to the changes made to the schema as well as the different reviews suffered for the technical components to reflect the technical and functional requirements final version.

Considering this is the second version of a deliverable describing the architecture and that the final release of the URBANITE integrated platform has not been deployed yet, more modifications could be implemented in order to improve the behaviour of the Ecosystem in each of the pilots. Those modifications and the final status of the architecture will be described in every document associated with each of the further releases of the integrated platform, which are D5.8 and D5.9 URBANITE Ecosystems due in June and December 2022 respectively.

This schema is driven by the layers or in-platforms organization made when revising the requirements for making them final. That work was gathered in the D5.2 deliverable [1].

The deployment of this URBANITE Ecosystem in any of its versions will follow the DevOps methods and mechanisms described in the D5.6 deliverable [4], made within this same work package.

This document is the last of a series of two which describe the structure defined for the URBANITE Ecosystem technical and functional organization. No further modules will be developed, or they will be minor changes to the existing ones.

The details of every technical component are described in the corresponding technical documents that accompany the final versions of them.

5 References

[1] URBANITE Consortium, «D5.2 Detailed requirements specification,» 2021.

[2] URBANITE Consortium, «D5.4 URBANITE detailed architecture,» 2021.

[3] URBANITE Consortium, «D6.2 URBANITE Use cases implementation».

[4] URBANITE Consortium, «D3.2 Data Harvesting module and connectors implementation,» 2021.

[5] URBANITE Consortium, «D3.5 Data Curation module implementation,» 2021.

[6] URBANITE Consortium, «D3.7 Data Aggregation and Storage module implementation,» 2021.

[7] URBANITE Consortium, «D5.1 Detailed requirements specification,» 2020.

[8] URBANITE Consortium, «D4.2 Implementation of Strategies and algorithms for data modelling and visualizations,» 2022.

[9] URBANITE Consortium, «D5.1 Detailed requirements specification-v1,» 2020.

[10] URBANITE Consortium, «D5.6 URBANITE DevOps Infrastructre,» 2021.