URBANITE

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

Deliverable D5.7

URBANITE Ecosystem-v1

Editor(s):	María José López
Responsible Partner:	TECNALIA
Status-Version:	Final - v1.0
Date:	30.06.2021
Distribution level (CO, PU):	PU

Project Number:	GA 870338
Project Title:	URBANITE

Title of Deliverable:	URBANITE Ecosystem-v1
Due Date of Delivery to the EC:	30.06.2021

Workpackage responsible for the Deliverable:	WP5 - URBANITE ecosystem integration and DevOps
Editor(s):	María José López (TECNALIA)
Contributor(s):	Iñaki Etxaniz/Gorka Benguria (TECNALIA) and components' owners.
Reviewer(s):	Massimo Villari, Maria Fazi
Approved by:	All Partners
Recommended/mandatory readers:	WP3, WP4, WP5

Abstract:	Report related to the URBANITE Ecosystem-v1, describing the environment where is deployed and the methods and resources used for the integration of the corresponding components.
Keyword List:	DevOps framework, integration, environments, prototype.
Licensing information:	The license under this component is offered, is not decided yet, and depends on the licences of the different components integrated in it. 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	27/05/2021	First draft version with TOC	TEC
v0.2	18/06/2021	First internal version for checking within TECNALIA integration team	TEC
V0.3	21/06/2021	Version ready for internal review	TEC
V0.4	29/06/2021	Commented by the internal reviewer	AD
V0.5	29/06/2021	Version addressing internal review comments	TEC
V1.0	30/06/2021	Ready for submission	TEC

Table of Contents

List of Figures

List of Tables

Terms and abbreviations

API	Application Programming Interface
DCAT-AP	Data Catalogue vocabulary Application profile for data portals in Europe
DSS	Decision Support System
EC	European Commission
GIS	Geographic Information System
IDM	IDentity Manager
JSON	JavaScript Object Notation
MQTT	Message Queuing Telemetry Transport
OD	Origin/Destination
REST	REpresentational State Transfer
UI	User Interface
XML	eXtensible Markup Language

Executive Summary

This deliverable contains the technical description of the URBANITE Ecosystem in its first initial version as a prototype. The architecture of this version contains the components with a core-functionalities implemented, as well as a set of components that provide the requirements established as functionalities prioritized for this Month 15.

In this document, we detail the initial version of the DevOps framework as part of the integration and continuous delivery strategy [1] planned as the approach to follow for developing and deploying the different software components developing by different partners.

Due to this first version of the URBANITE Ecosystem does not include all the components envisioned in the general URBANITE architecture [2], a reduced version of that general architecture is presented, as well as the list of the requirements that defined the selection of the components involved.

The initial version of the DevOps framework is explained, and the different environments detailed.

Future versions of this document will show the evolution of the Ecosystem, improving the architecture and integrating new components into the system that currently are not in a stable status to be integrated into our current scenario.

1 Introduction

1.1 About this deliverable

This deliverable shows the architecture of the first integrated URBANITE Ecosystem, and the implemented prototype deployed in several environments, including each for the four use cases and another one for integration and general purposes.

The components that are part of this prototype correspond to the first version of the platform, due to month 15.

Furthermore, this document presents the initial approach for the deployment of the DevOps framework, that will be useful for the integration and execution of the platform in further releases of the URBANITE Ecosystem.

1.2 Document structure

The document is structured in four main sections:

- At first, we will introduce the context related to this deliverable, explaining the objectives and the structure of the document.
- In the second section, the main requirements and functionalities covered by this first prototype are presented, including the reduced architecture of the URBANITE Ecosystem-v1.
- In the third section, we will present the installation process of the prototype and its deployment, so anyone can use and test it.
- The fourth section consists on the exposition of the conclusions and the further work to be accomplished in future versions of the URBANITE Ecosystem.

2 Implementation

2.1 Functional description

The URBANITE Ecosystem-v1 is the platform into which the different key results of the project will be integrated. The main objective is to deploy the versions of the components that support the requirements prioritized for this M15 milestone, decided and described in the D5.1 deliverable [3].

The platform provides an entry point to the URBANITE UI from where the users can access the developed functionalities. Four additional entry points are provided, one for each use case, deployed in four different environments apart from the integration one. These dedicated environments for the use cases aim to be a starting way to test and control the URBANITE platform before deploying it in the infrastructure of the municipalities.

So, the **Integration environment** focuses on compiling the code and performing the unit test and integration test reports. This stage also includes the availability of a common storage mechanism for the binaries created, as well as the assets required to deploy the applications (e.g., configuration files, infrastructure-as-code files, deployment scripts), and the use cases environments (in Tecnalia infrastructure) are where the Use Case demos of URBANITE are tried and tested in order to be deployed later, under the control of the final users, most probably into their premises.

Requirements:

The first version of the detailed requirements specification of the URBANITE ecosystem was described in D5.1 deliverable [3] and their prioritization from the point of view of the M15 release of the first prototype of the URBANITE Ecosystem. The platform deployed from the WP5 supports the requirements described on Table 1.

Table 1. List of prioritized requirements for M15 Ecosystem

Req ID	Req. Description	Requirement coverage by the prototype
DH01	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 system), raw data from APIs or data coming from sensors.	Partially covered. Some data sources are connected, but not all. The Data Catalogue contributes to this requirement being able to retrieve METADATA from heterogenous Open Data Portals.
DH02	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.	Partially covered. The architecture of the importer allows for paginated fetching when applicable. However, as of yet there are no importers that make use of this feature.

DH03	Data Harvester should be extensible with new connectors if new, unsupported, data sources are discovered.	Covered. The Data Catalogue contributes to this requirement since it is extensible with new connectors.
DH05	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.	Covered by the scheduling component. The Data Catalogue contributes to this requirement since it is able to synchronize the federated catalogue (collecting metadata) at recurring intervals.
DC01	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.	Partially covered. As of now, there is a transformer for JSON payloads, as well as a component converting Excel files to JSON. More transformers will be added as required.
DC08	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	Not covered. Currently there is no data curation module. Curation will be added at a later point in time.
DF03	The data should be mapped into EU vocabularies	Partially covered. Static values (e.g. data themes) are making use of EU vocabularies, but there is currently no capability to map dynamic values.
DF04	The metadata should be mapped into DCAT-AP metadata	Covered. Metadata is generated in the importing components.
DS02	The data storage component should be able to process and store DCAT-AP compliant metadata.	Covered.
DR01	The data retrieval component must expose API to retrieve and query the data stored in the different repositories	Covered for the following data types: Traffic Flow Observed, Calendar, and Air Quality Observed.
DR02	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.	Covered, the Data Catalogue allows to query in a federated way the DCAT-AP metadata.
DP01	Data projection component will provide dimensionality reduction methods for a better understanding and interpretation of the data.	Not implemented.

DCL01	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.	Not implemented.
SOM01	The Self-Organizing Map will provide the user with a visual topological representation of the data, able to highlight potential clusters.	Not implemented.
TS02	Traffic Simulation component will provide the ability to simulate hypothetical situations and the effects of different measures.	Partially covered. TODO: better UI for creating simulations of hypothetical simulations Currently supported measures: traffic counts (bike, car, PT) per street, emissions per street.
AV01	The component must allow to visualize the analysis results on a combination of map layers, heat maps, traffic flow graphics and other kind of visualization.	Implemented but not integrated yet: support for multiple selectable layers, including heat maps (currently not used), map layers (highlighting of network differences for hypotheticals), traffic flow (per street), emissions (per street). Implemented but will not be integrated: advanced visualizations of the scenario outcomes.
UUI01	The UI must provide uniform access to URBANITE tools and components.	Covered, the UI is integrated with the IDM, Data Catalogue, Bike analysis, Traffic estimation, etc.
UUI02	The UI must be integrated with the DSS visualization capabilities.	Partially covered for the bike analysis and traffic estimation as a DSS.
UUI03	The UI must support different user profiles, offering different functionalities for administrators and final user.	Covered. The UI, through the IDM, manages different roles and the Administration functionalities are provided only for the Admin of the platform.
UUI04	The UI must be responsive to support different types of devices.	Covered. The UI framework itself is responsive and supports different types of devices. The component provided by the partners

		should follow responsiveness in their implementation.

Apart from this prioritization, the prediction component, due to M27 release, has been considered part of this M15 version and its requirement partially covered in these two aspects:

Req ID	Req. Description	Requirement coverage by the prototype
PRED.01	The prediction component will provide an engine to produce prediction for a traffic/mobility variable defined as a time series considering a series of time defined features.	The bike analysis sub-component provides an engine to produce models to compute OD matrixes for bike city services which can consider different features in the computation: the day of the week, hour. In addition, it has the ability to define different zoning for the calculation. The traffic prediction sub-component allows to produce prediction models to compute prediction for the flow of vehicles at the locations of the traffic flow sensors considering; the day of the week, hour. In addition to the raw prediction the models produced are capable to compute an interval of confidence for the generated values.

2.2 Technical description

2.2.1 Continuous integration overview

The technical strategy adopted in the URBANITE development is described in the D5.3 deliverable [1], and it is based on a DevOps approach for the development of all the components.

The different software components that compose this M15 prototype, have been implemented by different partners following different technologies.

As detailed in the Integration Strategy, the DevOps approach will be structured in three environments, as depicted in Figure 2.

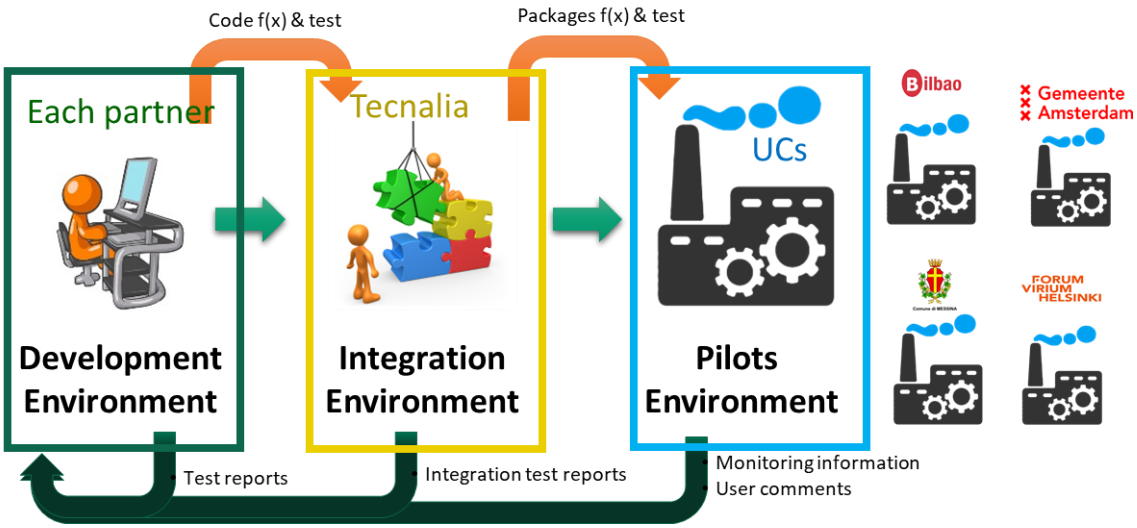

Figure 1. The three environments in URBANITE.

The different environments used for the continuous integration within the URBANITE Ecosystem are depicted in Figure 3, where the blue color is used to indicate the elements available at M15. M15 version does not include the deployment of the URBANITE platform in each municipality (*real Pilots*) but, to provide a way of running particular tests and validations tailored to one municipality, four parallel environments have been set up (*demo Pilots*), one for each city that participates in the use cases.

The workflow is the following: when a developer uploads a new version of its components to the integration environment, the integration process starts compiling the code and testing it in a temporary environment (*Feature branch*). Afterwards, the code is merged in the *develop* environment, where the whole ecosystem is built and tested again. From this environment, in a further step, the developer can promote the code to the *demo Pilots* environments. Also, the integrator can clone the actual version to the Master environment to maintain a stable version accessible, out of the integration ups and downs.

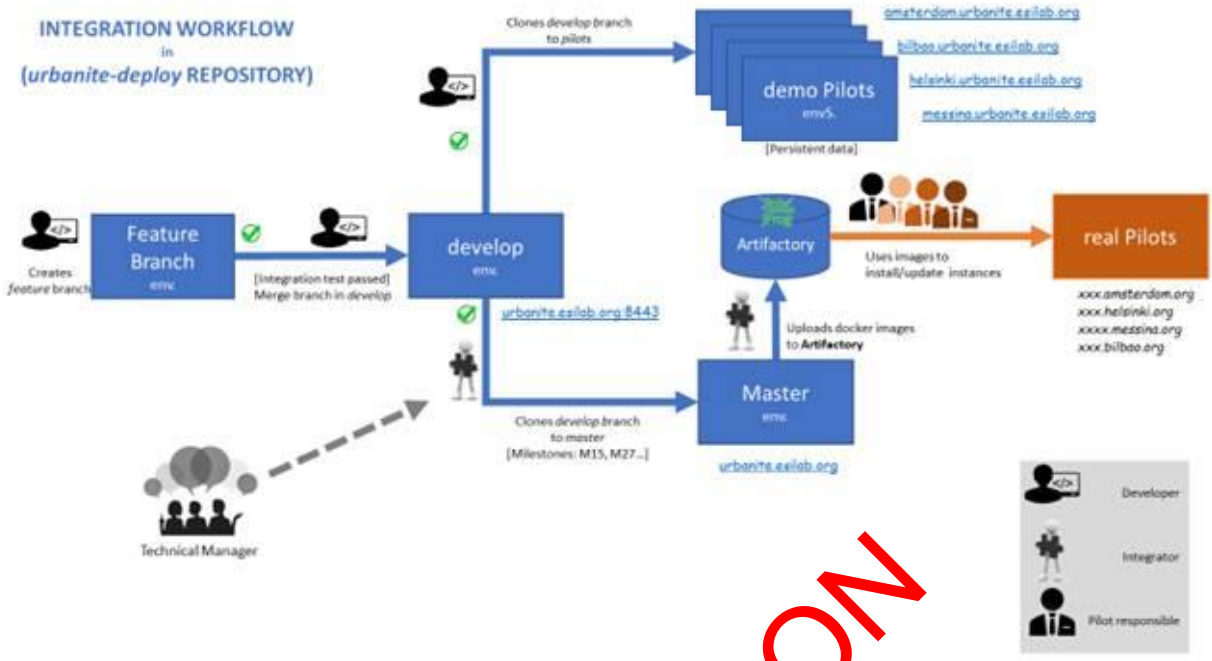

Figure 2.URBANITE integration workflow

A brief description of the environments and its function follows (this can also be consulted in the README.md file of the integration repository[1], where the description of the environments, the components and their access points, and the installations instructions are included).

- **FEATURE BRANCH:** Temporary environment that is created each time a developer wants to integrate a new version of his component. It just checks that the new version of the urbanite platform builds without problems, and is destroyed afterwards.
- **DEVELOP:** Environment that contains the last version of the components running together. Dedicated to test new features, interfaces and communications among components. Available at urbanite.esilab.org:8443.
- **MASTER:** Contains a specific version of the platform, frozen for determined Milestones. Can be accessed at urbanite.esilab.org.
- **DEMO PILOTS:** Four environments, one for each city, where the integrated platform is replicated and adjusted to the characteristics of the use cases. It is a previous step for testing the platform before setting up in the infrastructure of the municipalities:
  - amsterdam.urbanite.esilab.org
  - bilbao.urbanite.esilab.org
  - helsinki.urbanite.esilab.org
  - messina.urbanite.esilab.org
- **REAL PILOTS**: the installation of the platform in each municipality's infrastructure. To be done after the integration phase once a stable version is achieved to test the use cases.

Apart from that, in order to support developers during the integration, we provide:

[1] https://git.code.tecnalia.com/urbanite/private/urbanite-deploy/-/blob/develop/README.md

- Aa **Portainer** [4] instance that allows to access the logs and the console of every container in every environment.
- An **Artifactory** instance to store the images of the containerized components. These images will be used to deploy the final version of the platform in the real Pilots.

2.2.2 Prototype architecture

A partial version of the envisioned URBANITE architecture has been implemented to provide the planned requirements.

The M15 architecture came along from the analysis made by the partners responsible for the components and the prioritization of the interfaces among them. Those interactions defined which components make up this first prototype.

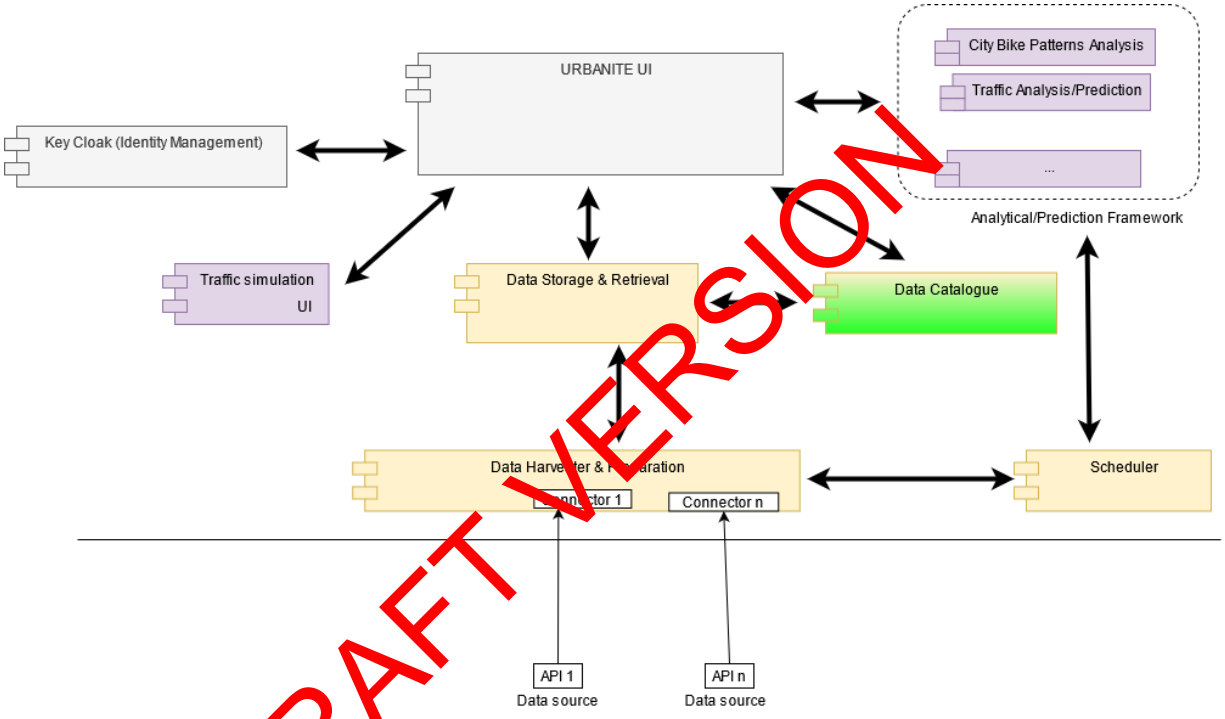

Figure 3.URBANITE Ecosystem-v1 Architecture

A detailed description of these components is included in D5.4 deliverable [2].

The components, during the integration process, have been containerized for a better encapsulation that allows future implantation across different platforms. At the end, each component is composed by one or more Docker containers, and presents a REST interface to the rest of components, if needed.

A diagram with the M15 components and their corresponding containers is shown in the following figure. The URBANITE ecosystem is composed by 9 components that in turn are integrated by a total of 26 multi-layer containers.

The URBANITE UI is the component that provides the graphical User Interface, wrapping the rest of components that provides some graphical interfaces to the final user. The Portainer -integration tool provided to developers to interact with the containers in development and testing- and the Traefik -router that allows publishing the services offered by the different components integrated in the same environment- are utility components provided by the integration environment.

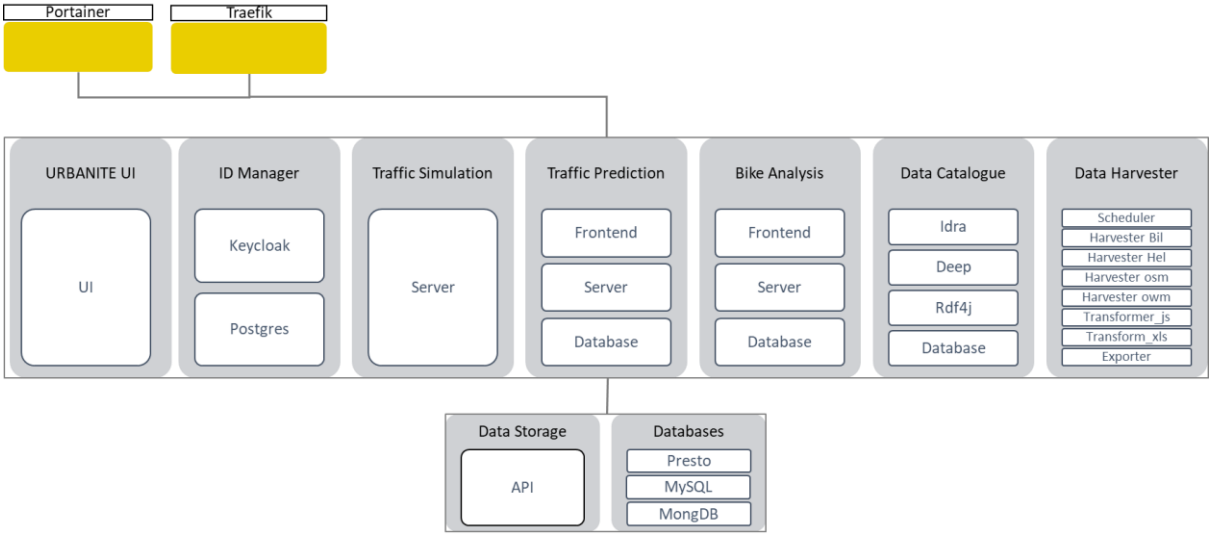

Figure 4.URBANITE Ecosystem-v1 containerized components

2.2.3 Components description

The components that are part of this integrated version of the URBANITE Ecosystem, have been developed by different partners according the requirements established and the status of each work package at this moment of the project.

Each partner has decided the best technologies to use regarding the nature of the component, the interactions among others and the general infrastructure where integrating the whole URBANITE Ecosystem. Most of the services exposed from the components are available as API REST services.

The technologies followed by the URBANITE UI component are the base for creating the rest of the UIs that the components need for interactions with the user. So, a template is created and described in detail in the readme file of the URBANITE UI repository. This template is built starting from NGX-Admin [5], an open source dashboard based on Angular [6], Nebular [7] with Eva Design System [8].

The key functions of these components in this version are as follows:

- URBANITE UI: The entry point to the URBANITE Ecosystem, that allows users to access the functionalities provided by the URBANITE platform at this point of the project.
- Identity Manager (Key Cloak): This component is in charge of securing the access to the other URBANITE's component, whenever security is needed. It is called by other components that interact with the user.
- City Bike Pattern Analysis: This module analyses GPS information related to the mobility of the bikes and transform it in more useful data.
- Traffic Prediction: It performs heuristic prediction for the vehicle flow at a location within the city by the processing of historical values measured by a fixed sensor and other information.
- Traffic Simulation: It offers the simulations of traffic under specified conditions, as proposed mobility policies, different weather conditions, changes to the traffic infrastructure, etc.
- Scheduler: It triggers a pipeline for the harvesting process, downloading data from a list of configured APIs within defined periods of time.

- Data Harvester and Transformation: It is responsible for fetching data from a given API, being the entry point of the data into the pipeline. Then a transformation is done into common models.
- Data Storage and Retrieval: This module stores and retrieves datasets metadata and related data in repositories DCAT-AP compliant metadata and transformed data.
- Data Catalogue: It allows to discover and access the datasets collected and managed by the components of URBANITE Ecosystem for data acquisition, aggregation, and storage.

3 Delivery and usage

3.1 Package information

The structure of the software of each component depends on the technology used by the different partners. Once the versions are uploaded to the gitlab, they are encapsulated as a docker image, as presented in Figure 4.

3.2 Installation instructions

To deploy the URBANITE ecosystem in an easy way we have created a docker compose configuration file, so that the user can install everything in one step that starts the initialization of all the required components in a background task (alternatively, the user can also build the Docker images for each component by separately compiling the respective Docker file included in each module directory).

Installation requirements

- To have Docker tool installed in your machine and accessible.
- To have Git installed.
- We recommend running the URBANITE framework in a powerful machine, because the project is composed by 25+ Docker containers (minimum: 10Gb RAM; 50 GB free storage depending on the datasets used)

Getting started

1. Clone this Git repository in your computer.
2. Navigate to the main root directory of the project
3. Define the required environment variables (see .env) file i.e.
    - export HTTPS_PORT=8443
    - export SERVER_HOST=192.168.56.1.nip.io
    - ...
4. Run in the console the command `docker-compose up`
   This will automatically deploy all the components containers in your *localhost* domain. This deployment may take some minutes.
5. Access to the URBANITE UI web page in http://192.168.56.1.nip.io:8443 with a browser

3.3 User Manual

This prototype is a preliminary version of the URBANITE Ecosystem. Not all the functionalities are implemented and some of the components are not completely accessible.

The entry point to the Ecosystem is the URBANITE UI, in this URL:

https://urbanite.esilab.org

Once the user introduces the credentials:

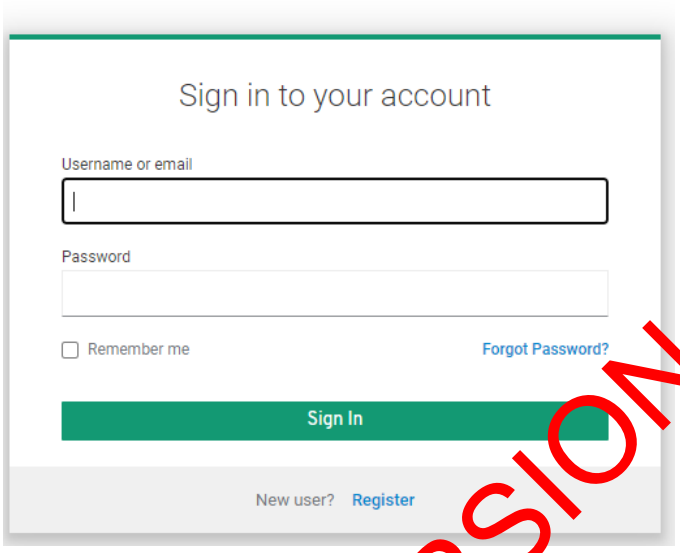

Figure 5.URBANITE login page

The main features provided are placed in the left side of the page:

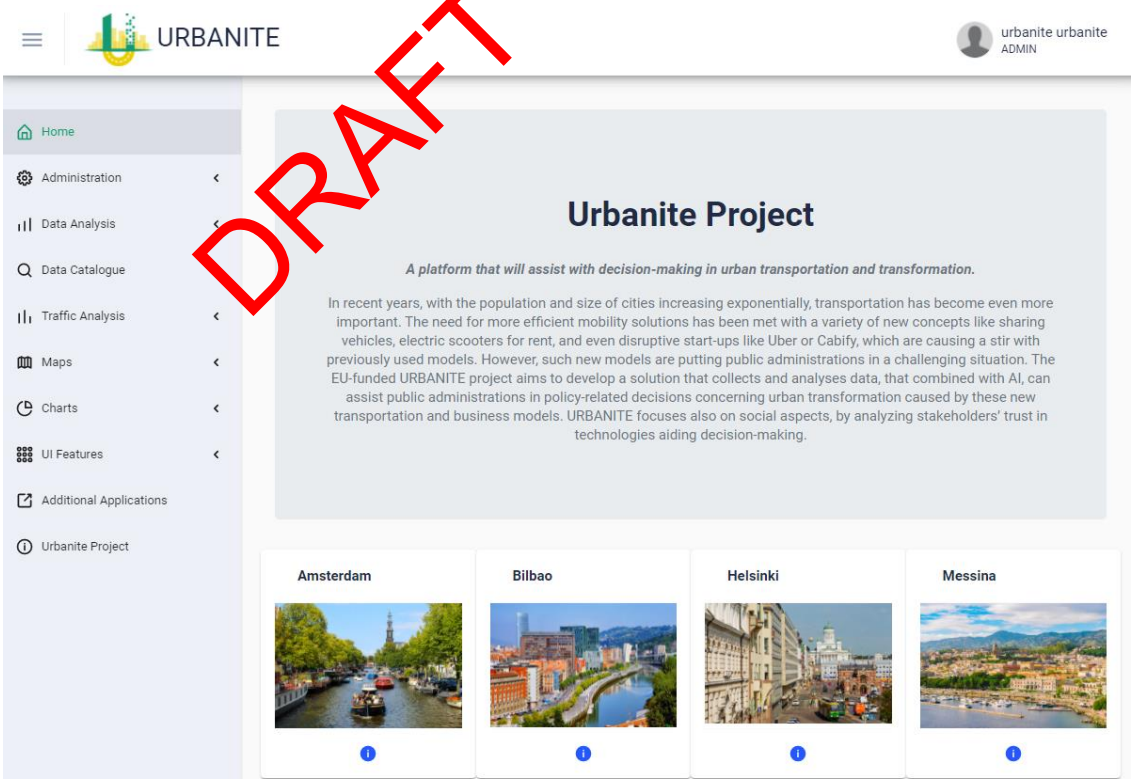

Figure 6.URBANITE UI home page

The integrated functionalities are:

- Administration
- Data Analysis
- Data catalogue
- Traffic Analysis

These functionalities are provided in a very basic stage, according to the status of the project at this point of M15 version.

3.4 Licensing information

The license under which this prototype is delivered is not decided yet. It is a future work to take that decision depending on the licenses of the different components that will be part of the final URBANITE Ecosystem.

3.5 Download

The code is uploaded and available by now in the project GitLab repository:

https://git.code.tecnalia.com/urbanite/releasesThe testable version of this M15 prototype can be checked accessing to the master environment, where a stable version of this prototype is available:

https://urbanite.esilab.org

introducing the previously provided credentials

4 Conclusions

This document contains the report accompanying the initial prototype of the URBANITE Ecosystem-v1, corresponding to the M15 release. It contains the description of the prototype from a functional and technical point of view, considering the preliminary status of the components integrated into it. It also provides some information about how to use it and from where.

This is the first version of the URBANITE Ecosystem due to M15. The next version of the platform (D5.8) will be released in Month 27 following an incremental approach, augmenting the functionalities provided. The related report will describe the new requirements covered, considering the feedback from the use cases at that time.

The license under the URBANITE Ecosystem will be offered, is a decision to take as part of the tasks of the project management and taken by the whole consortium considering the licences of the different components integrated.

5 References

[1] URBANITE Consortium, «D5.3 Integration strategy,» 2020.

[2] URBANITE Consortium, «D5.4 URBANITE Detailed architecture-v1,» 2021.

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

[4] «Portainer,» [En línea]. Available: https://www.portainer.io/.

[5] Akveo, «ngx-admin,» [En línea]. Available: https://akveo.github.io/ngx-admin/.

[6] Google, «Angular,» [En línea]. Available: https://angular.io/.

[7] Akveo, «Nebular,» [En línea]. Available: https://akveo.github.io/nebular/.

[8] Akveo, «Eva Design,» [En línea]. Available: https://eva.design/