URBANITE Ecosystem: Integration and DevOps

María José López†
ICT Division
TECNALIA, Basque Research
and Technology Alliance (BRTA)
Spain
mariajose.lopez@tecnalia.com

Iñaki Etxaniz
ICT Division
TECNALIA, Basque Research
and Technology Alliance (BRTA)
Spain
inaki.etxaniz@tecnalia.com

Giuseppe Ciulla
Research & Development
Laboratory
Engineering Ingegneria
Informatica
Palermo, Italy
giuseppe.ciulla@eng.it

ABSTRACT

URBANITE is a collaborative research and innovation project whose outcomes are mainly software based. These outcomes will be implemented in a collaborative manner by different development teams from different partners. In order to manage the development environments, and the integration of the different software components in on time releases, the proper DevOps strategy and processes has been defined and set up.

This paper describes the URBANITE integrated architecture at month 12, with a theoretical vision of the URBANITE system that will cover all the functional and non-functional initial requirements set by the technical work-packages considering the social perspective and the input of the use cases.

The definition of the interactions among components is shown through the specification of the interfaces, considering the dataflows envisioned for meeting the needs of the different stakeholders. Different tools, environments and strategies envisioned for the management of the development, integration and validation stages of the software components to be implemented during the life cycle of the project are described as part of the integration strategy.

KEYWORDS

DevOps, Integration, Ecosystem, Requirements, Architecture, Prototype.

1 REQUIREMENTS

The process for setting up the URBANITE Ecosystem receives inputs from the rest of technical components, related to the data management and simulation processes, regarding to:

- Technical (software) requirements, expressing both functionality needs and non-functionality aspects.
- Architectural structure and configuration of the components implemented in different work packages.
- And about how to integrate them into the overall URBANITE UI.

Also the Use Cases propose functionalities for the URBANITE ecosystem, that will be considered for being part of the URBANITE ecosystem and prioritize for their implementation.

This platform will comprise the URBANITE components implemented, Key Results or KR from now on, (Key Result KR1-Virtual Social Policy Lab, KR3-Data Management Platform, KR4- Algorithms) and their integration in the URBANITE ecosystem (KR5).

The elicitation of the first version of the functional and non-functional requirements for the URBANITE ecosystem and the related components is described as an iterative process where both the technology providers and use case providers' have participated.

For the functional requirements a combined approach has been followed: 1) a top down approach led by the technology provider partners, who have defined the first set of functional requirements and 2) a bottom up approach where the needs of the Use Cases have been monitored and UC initial requirements have been extracted. For the Non-Functional Requirements, these have been detailed per component, including relevant aspects, such as performance, usability or resources needs for deployment.

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

The URBANITE ecosystem will include all the components for data management, analysis and support to the decision making that are going to be created/developed/implemented in the context of the URBANITE project. The first version of the requirements will be updated in further reports and analysis.

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

- **Requirements coming from the URBANITE action specification**: These requirements cover the functionalities described in the URBANITE action specification. The first version of these requirements has been described by the Technology providers partners (Fraunhofer, Tecnalia, JSI and Engineering Ingegneria Informatica) based on the URBANITE approach and high-level architecture description included in the URBANITE action.

- **Requirements coming from the Use Cases**: The Use Cases proposed functionalities for the URBANITE ecosystem, so that the features offered can cover their needs.
- **Requirements coming from the co-creation sessions (SoPoLabs)**: It is expected that some requirements may be derived from the SoPolabs that will be conducted in the context of WP2. If relevant these requirements will be considered for the URBANITE ecosystem and prioritize for their implementation.
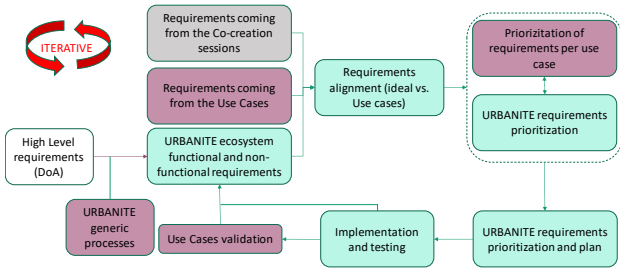

Figure 1: URBANITE process for requirements gathering and prioritization

The different users of the URBANITE platform to perform any of the previously described are:

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

2 ARCHITECTURE

The detailed description of the entire global architecture of the URBANITE ecosystem as a general representation of it, is in its first version and can evolve following the needs of the project. Structural and behavioural analysis of each component of the architecture was performed identifying interactions and dependencies among them.

Three layers of components can be observed and identified by colours:

- Yellow components are those that manage the data, and implemented withing the WP3
- The purple ones are dedicated to the simulation and analysis of the data ingested to the system for the yellow ones.
- And the grey components are those related to the UI, as the entry point to the platform and for user management.
- There is also a green component considered as a repository of the datasets stored by the data management layer.
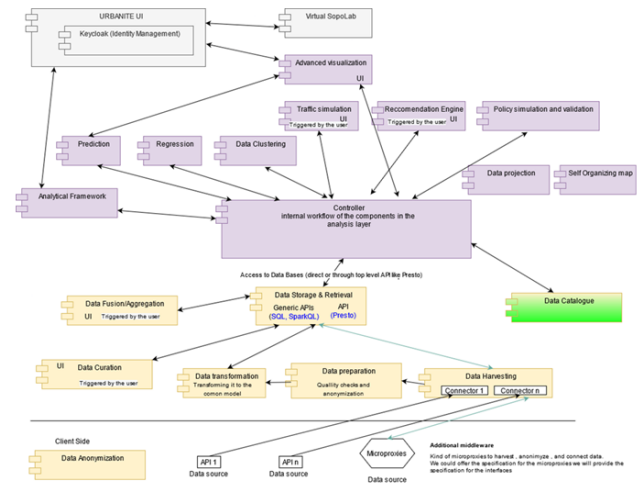

Figure 2: First version of URBANITE architecture

In the Month 15 of the project, the architecture is a reduced version of the overall URBANITE architecture and consists of the components that provide the functionalities designed covering the current version of the requirements.

The components are:

- 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 into 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.
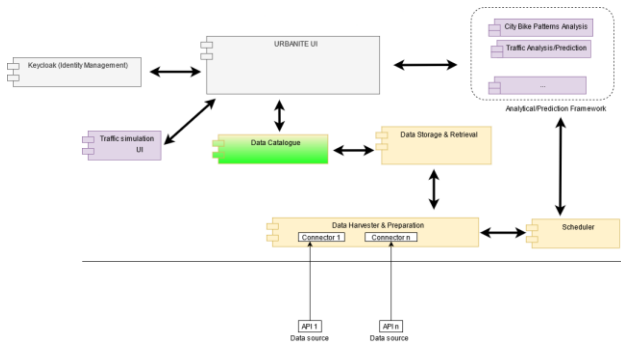
Figure 3: Month 15 version of URBANITE architecture

3 INTEGRATION AND DEVOPS STRATEGY

This section presents the infrastructure and tools planned to be used internally for the development and operation. The DevOps approach requires the set-up of a development and delivery pipeline, that consists in the stages an application goes through from development through production.

The URBANITE iterative and incremental approach mandates the adoption of a development and deployment process able to fully support it. That is why the project will adopt a DevOps approach in the development of all KRs. DevOps integrates development and operations into a single-minded entity with common goals: high-quality software, faster releases and improved users' satisfaction. DevOps also incorporates a number of agile principles, methods, and practices such as continuous delivery, continuous integration, and collaboration [1].

The different KRs, which are the outcomes of URBANITE [2], are composed of several software components that will be implemented by different partners following different technologies.

In URBANITE, the DevOps approach will be structured in three environments as depicted in the figure.
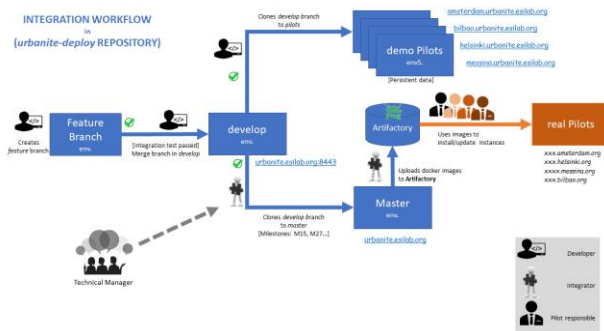

Figure 4: Continuous Integration and DevOps approach

The description of the environments that are part of the integration system:

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.
MASTER: Contains a specific version of the platform, frozen for specific Milestones.
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.
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:
A Portainer 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.

4 URBANITE ECOSYSTEM

The main result of the URBANITE project is the URBANITE Ecosystem and aggregates all aspects of the project, namely the citizen participation, both social (citizen participation, attitude and trust in disruptive technologies, co-creation) and technical aspects (data management platform, algorithms and so on).

The URBANITE UI is an integration framework at User Interface level.
The integration strategy provides different approaches that can be followed:
1. **External component integration**
   Iframe: the external application is included in the UI through an iframe
   External link: the application is referenced in a dedicated section of the UI, and a specific link is provided to the user
2. **Template component integration:** the external application, that must provide a set of REST APIs for developing a specific component included in the UI.

The URBANITE UI is an Angular application built taking advantage of Nebular, ngx-admin frameworks and Eva Design System. With the addition of some of the most popular front-end libraries and packages.
The access to the Urbanite UI is provided through the Urbanite's Identity Manager component (an instance of KeyCloak whose theme is customized following Urbanite's colour palette).
The Urbanite UI provides Role-Based access to specific functionalities following the IDM returned role(s) of the user
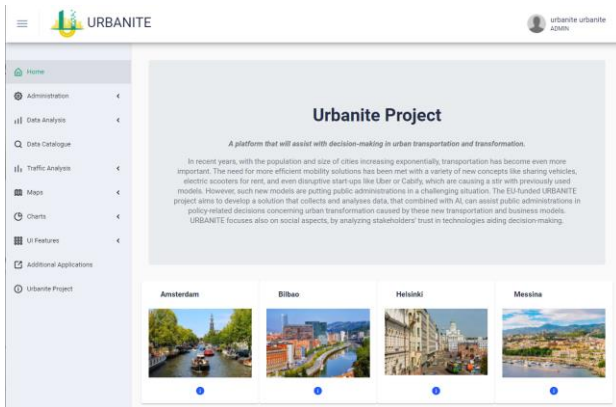
Figure 5: M15 Component's integration REST APIs based

The figure above shows is the URBANITE UI, integrating the components implemented for the M15 version of the prototype, covering the requirements and the functionalities provided by the implementation planned at this point.

The left part of the page includes the available options. Some of them are general utilities, and other are functionalities implemented within the different technical work packages.

- The Home page offers descriptions of the four municipalities and the basis of the URBANITE project. There is an additional information section for each description that allows to extend the details of the selected city.
- The Administration, Data Analysis, Data Catalogue and Traffic Analysis are specific sections that provide services related to the data of the different municipalities.

- Maps: where are two examples of how a developer can build and manage maps, using the libraries provided by this UI.
- Charts: this option displays three possible library alternatives provided by the Urbanite UI to build bars, pies and line charts.
- UI Features: about style examples as colors, icons, typography, and the grid system that should be used for implementing pages to provide responsiveness.
- Additional Applications: is a section where external links to other applications can be added through the URBANITE UI configuration file. For instance, the forum page is linked.
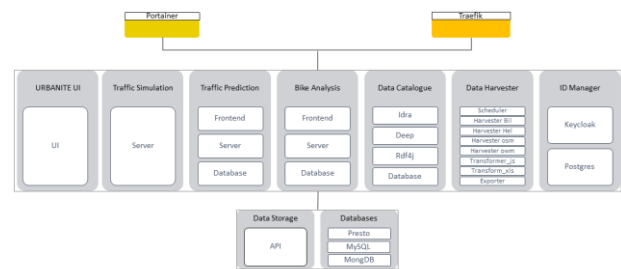- And the Urbanite Project where included information about the objectives of the project.

Figure 6: URBANIITE Ecosystem v1

The Figure 6 describes the schema that supports the before explained prototype.

REFERENCES

[1] New Relic, ""Navigating DevOps - What it is and why it matters to you and your business"," New Relic, 2014.

[2] URBANITE Consortium, "URBANITE Annex 1 - Description of Action," 2019.