MOBIDATALAB

Labs for prototyping future mobility data sharing solutions in the cloud

# D5.1 The Virtual Lab

23/01/2023

Author(s): Anestis MERDANI (AETHON), Alexis EFRAIMIDIS (AETHON)

MobiDataLab is funded by the EU under the H2020 Research and Innovation Programme (grant agreement No 101006879).

# Summary sheet

| | |
|---|---|
| Deliverable Number | D5.1 |
| Deliverable Name | D5.1 The Virtual Lab |
| Full Project Title | MobiDataLab, Labs for prototyping future Mobility Data sharing cloud solutions |
| Responsible Author(s) | Anestis MERDANI (AETHON) |
| Contributing Partner(s) | F6S, HOVE |
| Peer Review | AKKA, HERE |
| Contractual Delivery Date | 30-09-2022 |
| Actual Delivery Date | 27-10-2022 |
| Status | Final |
| Dissemination level | Public |
| Version | V1.0 |
| No. of Pages | **50** |
| WP/Task related to the deliverable | WP5/T5.1 |
| WP/Task responsible | AETHON / AETHON |
| Document ID | MobiDataLab-D5.1-TheVirtualLab-v1.0 |
| Abstract | The Virtual Lab will be a tool for the testing of the MobiDataLab Transport Cloud. It will include functionalities such as forums, polling, live data exchanges, live pair working (e.g., see https://codeshare.io/), etc. |

# Legal Disclaimer

MOBIDATALAB (Grant Agreement No 101006879) is a Research and Innovation Actions project funded by the EU Framework Programme for Research and Innovation Horizon 2020. This document contains information on MOBIDATALAB core activities, findings, and outcomes. The content of this publication is the sole responsibility of the MOBIDATALAB consortium and cannot be considered to reflect the views of the European Commission.

# Project partners

| Organisation | Country | Abbreviation |
|---|---|---|
| AKKA I&S | France | AKKA |
| AETHON SYMVOULI MICHANIKI MONOPROSOPI IKE | Greece | AETHON |
| HOVE | France | HOVE |
| HERE GLOBAL B.V. | Netherlands | HERE |
| F6S NETWORK IRELAND LIMITED | Ireland | F6S |

Funded by the
European Union

# Document history

| Version | Date | Organisation | Main area of changes | Comments |
|---|---|---|---|---|
| 0.1 | 27/10/2021 | AETHON | Table of content | Initial Version |
| 0.2 | 12/09/2022 | AETHON | All | Draft Version |
| 0.3 | 20/09/2022 | AKKA | All | Peer Review |
| 0.4 | 20/09/2022 | HERE | All | Peer Review |
| 0.5 | 23/10/2022 | AETHON | All | Rework |
| 0.6 | 27/10/2022 | AKKA | All | Quality Check |
| 1.0 | 27/10/2022 | AKKA | All | Submission Final Version |

# Executive Summary

The Deliverable D5.1 is the first submitted iteration of the development of the Virtual Lab platform. The Virtual Lab is a tool created to integrate with MobiDataLab Transport Cloud and includes functionalities such as forums, polling, live data exchanges and more. Task 5.1 initiated with a thorough literature review on similar platforms and other EU projects in order to collect the expectations and needs of potentials Virtual Lab users, which serves as the introduction of this deliverable. Following, Deliverable D5.1 describes the components and architecture of the MobiDataLab Virtual Lab. More specifically, technologies used for the development of the platform along with the final architecture scheme followed are presented. The non-functional requirements of the platform are then listed along with the use cases created in formal UML definitions illustrating the flow of activities of users on the platform. Finally, the integration points with MobiDataLab Transport Cloud are preliminarily discussed allowing for future exploitation based on the second iteration of the Transport Cloud prototype expected during the final months of the project.

The Deliverable D5.1 comprises of this report in conjunction with a demonstration (in the format of a screen recording) of the main Virtual Lab flows and features. Task 5.1, and corresponding Deliverable D5.1, focuses on the first complete version of the platform v1.0. The Virtual Lab platform will then be tested by the living labs participants allowing for two consequent updated versions to be developed until the end of the project based on the feedback collected.

Funded by the
European Union

# Table of contents

Funded by the
European Union

# List of figures

# List of tables

Funded by the European Union

# Abbreviations and acronyms

| Abbreviation | Meaning |
|---|---|
| API | Application Programming Interface |
| CSS | Cascading Style Sheets |
| HTTP | Hypertext Transfer Protocol |
| TC | Transport Cloud |
| UI | User Interface |
| UML | Unified Modelling Language |
| UX | User Experience |
| VL | Virtual Lab |
| 3PP | Third Person Parties |
| TC | Transport Cloud |

Funded by the
European Union

# Introduction

## 1.1. Project Overview

1.

There has been an explosion of mobility services and data sharing in recent years. Building on this, the EU-funded MobiDataLab project works to foster the sharing of data amongst transport authorities, operators and other mobility stakeholders in Europe. MobiDataLab develops knowledge as well as a cloud solution aimed at easing the sharing of data. Specifically, the project is based on a continuous co-development of knowledge and technical solutions. It collects and analyses the advice and recommendations of experts and supporting cities, regions, clusters and associations. These actions are assisted by the incremental construction of a cross-thematic knowledge base and a cloud-based service platform, which will improve access and usage of data sharing resources.

## 1.2. Purpose of the deliverable

This deliverable is a report aimed at providing an overview of the Task 5.1, which consists of the conception, definition and design of the Virtual Lab platform. The Virtual Lab is a tool for the testing of the MobiDataLab Transport Cloud. Through the course of the project, three complete versions of the Virtual Lab will be developed. The objective of this deliverable is to describe the components and architecture of the Virtual Lab platform as well as to present Virtual Lab's functional and non-functional requirements. The first complete version v1.0 of the Virtual Lab is being delivered through this report along with a demonstration of platform's features and functionalities.

## 1.3. Intended Audience

The dissemination level of the D5.1 deliverable is 'public' (PU). AETHON as Task 5.1 leader is responsible for it with the contribution of HOVE and F6S. An external review is conducted by members of the Advisory Board. This report is being submitted in conjunction with a demonstration video and corresponds to the development of the first complete version of the Virtual Lab platform. It shall be noted that parts from the demonstration or this report may be disseminated towards the Virtual Lab test users (i.e., Living Labs participants) to assist with its usage. Finally, feedback collected from the Virtual Lab usage by the living labs participants along with improvements and updates incorporated in the two following platform versions (v2.0 & v3.0) will be included in consequent WP5 Deliverables.

## 1.4. Structure of the deliverable and relation with other tasks

The objective of Task 5.1 and corresponding deliverable D5.1 is to *create the Virtual Lab by implementing changes to the Transport Cloud's UI in line with the Living Labs goals to co-create, explore experiment, and evaluate.* As a result, task T4.1 which defines Transport Cloud's architecture strongly influence the development of task 5.1.

At the same time, the Virtual Lab platform will be used and tested in all three living labs instances (#datathon, #hackathon, #codagon). Participants will test (system and acceptance-wise) the platform and a Backlog will be created in the context of Task 5.3. As a result, task 5.3 'Planning and monitoring of the Living Labs' is strongly linked to this task. Deliverable 5.4 'Living Labs execution plan' will set the requirements for the backlog and the feedback collection mechanism and Deliverable D5.5 'Report on Living Labs monitoring' will present collected feedback along with updates and improvements (bug correction) of the Virtual Lab implemented during the course of the three instances and the project.

Tasks 5.4/5.6/5.7 which relate to the planning, execution and monitoring of the three Living Labs are all connected with task 5.1, since the Virtual Lab platform will be used at an accretive way during all x-thons. The reviewing process following each occurrence will produce a list of good practices and lessons learned that will be delivered in corresponding deliverables and will include the aftermath of the Virtual Lab usage in each case.

As in every multidimensional project, so in MobiDataLab there is sometimes a hidden and subtle, interrelation between additional tasks and deliverables. Task 5.1 interacts with task 3.5 'Societal and environmental impacts of the data assessment framework' indirectly through task 5.2 which is about the quantification and measurement of the data exchange culture. During the development of task 3.5 questionnaires were created for the acceptance of the MobiDataLab solutions including the Virtual Lab, as the Transport Cloud's UI. The questionnaires will be executed within task 5.2 before and during the living labs where appropriate actors (i.e., possible MobiDataLab solutions uptakers) assess the non-functional requirements of the Virtual Lab.

Task 2.6, which is about the new use cases and requirements, and task 6.4 which relates to stakeholder coordination both interact with task 5.1. More specifically, the use cases created within 2.6 reflect real-world requirements of a plethora of stakeholders, requirements that will eventually give birth to various challenges that will give form to the actual competitions inserted in the Virtual Lab and solved/explored during the living lab instances. The challenge setters are originated from a big list of stakeholders, a list that is created within task 6.4. Hence the workflow begins with the initial definition of the use cases, depicted in D2.9, continues with the stakeholder engagement process to define actual real life territorial challenges, implemented within T6.4 and concludes on challenge/competition upload on the Virtual Lab platform for the living labs contestants to solve them.

# Literature Review

## 2.1. Social Features

2.

Since the Virtual Lab platform will be an extension of the Transport Cloud focusing on the UI and social aspects of the entire realm of the MobiDataLab solutions, a thorough investigation of platforms which have similar features and goals was executed. Hosting and browsing through challenges and living labs was platform's main priority in order to see how other platforms present this type of information on their respective websites. Most of the platforms presented below, have abundant social features that Virtual Lab is also going to incorporate. These social features are inspired by the social network platform and recreate the tools that bring communities together. Tools such as messaging or forums will be beneficial to the goals of the Virtual Lab and promote harnessing the full advantage of the Transport Cloud in the future by its users.

## 2.2. Platforms like Virtual Lab

In this section, it is represented a mini description of websites that is similar or has common use cases with Virtual Lab. Virtual Lab will adopt the most important and useful features from the similar websites and expand them based on the platform's needs.

- https://www.drivendata.org/about/

This platform hosts data science competitions and has an excellent portfolio of competitions with an apparent and effortless browsing page. Since Virtual Lab is also a platform that will host this type of events, we will try to be inspired by this type of a catalogues for browsing our competitions. The browsing in drivendata website, is a clean and modern way for someone to get an idea of what the events are. Hosting your own competitions and publishing it in Virtual Lab will help with the promotion of initiatives for such type of events in the future.

- https://www.kaggle.com/

This platform has most of the features (primarily social) that Virtual Lab will incorporate. The platform is used for data science competitions but there are many similarities with Mobility competitions. Many of the social features will be implemented and expanded in the Virtual Lab. It is evident, that these features help the competitors to communicate and create discussion on subjects. Posts from users will help the Transport community to exchange ideas about present and future problems in the transport domain. The sharing of the code is also an idea that would benefit Virtual Lab and Hackathons in general.

- https://www.transdairy.net/

Funded by the European Union

Transdairy is a Greek collaborative platform having forums and wikis for seminars and Living Labs. From this platform the forum and wiki tools are very interesting and worth exploring their incorporation in the Virtual Lab. This could result in having a forum that the Transport community share interesting thoughts and create discussions and questions towards other members of the community. A wiki for distinguished themes and concepts is included in Transdairy but it would also be an important addition for the Virtual Lab.

- https://hackathons.copernicus.eu/

The Copernicus Hackathon Programme brings together developers and entrepreneurs by creating Hackathons. Creating a Hackathon and reaching out to people interested is one of the main goals for Virtual lab. The way that Copernicus organises and operates Hackathons has been studied in order to integrate this acknowledgement in the Virtual Lab and the Living Labs which will be created within.

# Platform Technologies and Architecture

## 3.1. Architecture description

3.

Some developers choose monolithic style and others microservices style Most developers choose the architecture style based on the platform’s size and needs. For example, small applications with low volume of data or without any data at all, have no benefit in choosing microservices style. But instead, a larger app can gain many benefits from microservices style. Separation of concerns is the main problem in larger apps and microservices style is the solution of the problem. Virtual lab platform is implemented with microservices architecture as it consists of many services and it is needed to handle each one of them, separately. Also provides many other advantages, like performance, precision, development time, scalability.

**What are microservices?**

Microservice is an independent application/service that runs on its own environment with no dependency from the rest of software. The main difference of microservice architectural and the monolithic style is that the microservice structure treats software development as a single unit. Basically, a microservice is an independent part of a software which has a specific purpose. This purpose could be specific functions or service execution. Also, this type of architecture helps to handle large and complicated apps.

**Usefulness of microservices**

Table 1: Usefulness of microservices depicts the benefits of microservices

*Table 1: Usefulness of microservices*

| Benefit | Description |
|---|---|
| **Scalability** | Scaling a monolithic application is much more difficult to scale as it must be scaled as a complete unit. Microservice architecture gives the opportunity to scale specific services. That means developers focus on specific and independent services rather than the whole application. |
| **Development Time** | This type of architecture provides a better and faster development time because developers are developing a specific service that does not affect the whole application. Also, it provides a better environment for debugging, testing and fixing bugs. |
| **Independent from specific programming languages** | As it was mentioned before, each microservice is an independent service. That means that every developing team which works on a specific service, could choose a different programming language without affecting the other teams. For example, certain features could be implemented with python and others with C++. |

Funded by the European Union

| | |
|---|---|
| **Better deploying flow** | Deploying an application part by part is more manageable than deploying it as a unit since it is easier to fix issues on a specific service rather than to whole app. In addition, only the features that this service provides are affected while the service has production issues. With monolithic architecture, on every update, the whole platform must be redeployed and an insignificant issue on production will affect the whole app. |

## 3.2. Technologies

The next step after creating the platform's architecture, is selecting the appropriate technologies. Choosing technologies might not be effortless as the combination of the technologies is extremely important. Virtual lab for UI uses Reactjs as it is a "component based" framework which allows handling and expanding large apps, faster. The middle layer of the platform is implemented with Nodejs server as it is simple to integrate with Reactjs and with no sql databases such as redis. Additionally, if it is required, it can change in a full stack environment. The last part of the platform (backend) is implemented with python and sql database. Python is a friendly developer language considering that it is simple to learn and understand. Also, it contains a large number of integrated packages which impacts in decreasing the developing time. Also, integrated packages enable the platform to provide more and better features to the users. The combination of the technologies below provides better performance, security, development time.

- Reactjs

Reactjs is a JavaScript library that was created by Jordan Walke and now it is maintained by Meta and many individual contributors (Teams n.d.). It is used for building user interfaces and has many advantages such as it is a component-based library. "Component-based" signifies that, developers can split the app in multiple components and combine them for a bigger result. Splitting an app in multiple small components make the total app more manageable and easier to extend.

Platform's user interface is implemented with Reactjs combining with CSS for styling but also for creating a better user experience.

- Node-js

Nodejs is an open-source server environment that uses JavaScript on the server. The platform uses nodejs server as a middle layer between frontend and backend. The purpose of the server is to implement user authentication and store user's session. Also provides an extra security layer for all HTTP request coming from frontend to backend. Each request will be verified as secure from the nodejs server before redirecting it to backend.

- Python-Flask server-Gunicorn

Platforms' backend is fully implemented with Python, Flask server and Gunicorn server. Platform's backend has two microservices. Each microservice has its own functionalities and it is independent from the other one. The first one is responsible for user actions with his/her profile and the second one is responsible for user actions on living labs. Where it is needed, these two microservices communicate with APIs requests. Flask server is being used in development mode while Gunicorn, is being used in production as the first one is an API framework and not a production server.

- SQL Database

All data coming to or from the platform are stored in SQL database. This choice was made as most of the records are relational.

- Redis

Redis is a no-sql database solution. It is a data structure store which can be used as database, cache, message broker and streaming engine. It is used with the Nodejs server as it stores user's session on its memory which it is faster to retrieve. Platform uses redis to store user's session, as it is better practice to keep user's session data hidden from the browser.

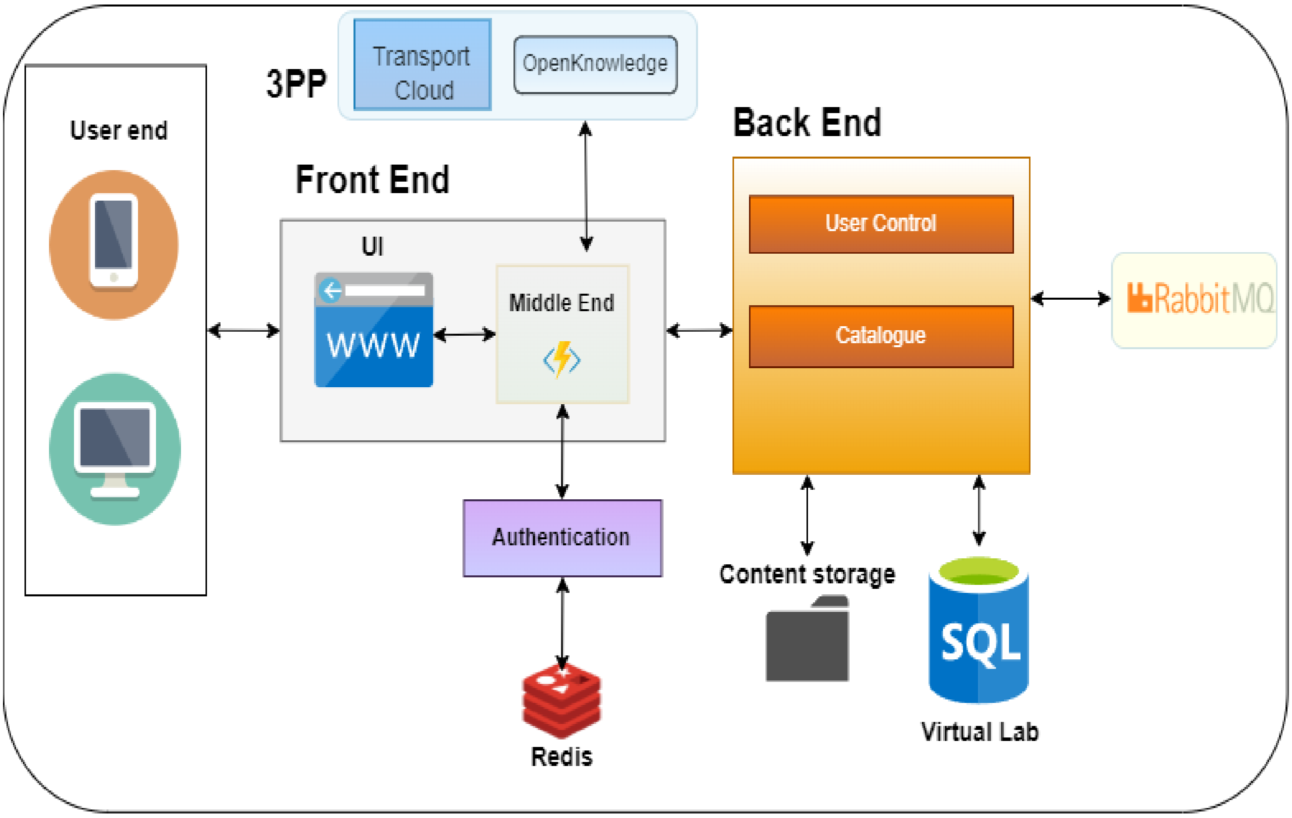
- RabbitMQ

RabbitMQ is a message-broker software where queues are defined, to which application/service to connect in order transfer a message. Each microservice has some periodic tasks or tasks that must run asynchronously. It keeps the message to a queue and then, celery handles the rest of the flow. Celery is used for python web applications. It receives the message from RabbitMQ, and asynchronously executes the task with request-response pattern (Makai n.d.). The flow of request-response pattern is simple to understand. In the first step, producer will send the message to a queue, then the consumer(celery) will execute the task. When the second step is completed, consumer will send a reply message to a different queue and producer will consume it (Comartin 2022).

## 3.3. Architecture Diagram

Based on the necessary requirements mentioned in the previous sections, platform's detailed architecture is presented below. Figure 1: Architecture Diagram provides a graphical representation of the main components and the connections between them.

*Figure 1: Architecture Diagram*



There are two main components, Front-end, Back-end. Each of them has at least one or more sub-components. Every sub-component has its own purpose, and it is independent and autonomous from the other ones.

Front-end has two sub-components, User Interface and Middle-End. The UI is responsible for showing the content (living labs, challenges, datasets, code snippets, posts, user's data) to the user and is the environment where all user actions will take place. The UI communicates with the Middle-End for user authentication. Middle-End is an extra layer of security as the platform does not store user's session on his/her browser but instead, it stores it on another service that is manageable from the platform. The Middle-End uses Redis for data storing. All requests coming from the UI first will be validated from the Middle-End and then will be redirected to the back end. Also, it will be used for 3PP authentication. The detailed connection between Middle-End and Transport Cloud is presented in chapter 5 (Integration Points with Transport Cloud).

Funded by the European Union

Back-end consists of two services, User Control and Catalogue. User control is responsible for user actions with his/her profile data. Catalogue is responsible for user actions with living labs, challenges, solutions, posts, etc. Both services are independent and autonomous from each other. For example, if one of them has production problems, it will not affect the other one. This is the main advantage of the microservice architecture. Also, both services have access to file storage (Content storage), database and RabbitMQ. Content storage stores all files users upload to Virtual Lab, such as profile images. RabbitMQ is used by both services as it allows them to run asynchronous tasks. In many cases, it is needless for a user to wait for a task to be completed.

# 4. Non-Functional Requirements and Use Cases

This section presents the non-functional requirements of the Virtual Lab platform and introduces the main functional requirements the platform satisfies to cover the needs of the use cases introduced below.

## 4.1. Non-Functional Requirements

Virtual Lab is the virtual counterpart of a physical living lab. Hence, the platform must follow all the principles of a physical living lab, such as experiment, evaluate, co-create. Virtual Lab must allow users to discuss their problems publicly, to comment on interesting posts, to vote on solutions and also to have access to tools and datasets. As a result, the platform's non-functional requirements were created based on the aforementioned specifications.

Table 2 summarises the non-functional requirements of the platform.

*Table 2: Non-Functional Requirements*

| Name | Features |
|---|---|
| **Usability** | The platform shall be able to achieve the required functional requirements effectively and time-efficiently. User satisfaction shall be considered and feedback from Living Labs occasions shall be incorporated into the development. |
| **Accessibility** | The platform shall focus on allowing users to easily access information and datasets resulting in promoting active engagement in data sharing. Extra attention shall be given to the integration between Virtual Lab and Transport Cloud, since information accessibility of the first is profoundly linked with the second. |
| **Digital inclusion** | The platform shall promote access and use of digital mobility services and data by vulnerable and excluded users (i.e., 'non-digital natives', people with disabilities). |
| **Engage in co-creation** | The platform shall aim to promote pair-working between users. The pair working shall be: 1) Between various Data Providers & Owners in the co-creation of a Virtual Living Lab, 2) Between Innovators/data users through cooperation during the problem-solving phase and joint submission. |
| **Adaptability** | Since the implementation of the platform will incorporate feedback derived from the Living Labs occasions, the platform shall be easily modified and upgraded based on the given comments and improvement suggestions. |

Funded by the European Union

| | |
|---|---|
| **Performance** | The platform shall have a fast response time as it will be used during real-time activities (i.e. Virtual Living Labs). Response time shall be tracked throughout the project and carefully examined in order not to have extensive waiting periods. |
| **Security** | The platform shall ensure GDPR compliance throughout user's interaction (from filling personal information up to exchanging messages with other users). Special attention shall be given when submitting a solution regarding IPR and user rights. Also, attention will be given at the security of the databases which may include firewalls to prevent unauthorized access and possible malicious actions. |
| **Reliability** | The platform shall be able to operate regardless of the users' number and workload. Efficient data distribution shall be ensured throughout user's interaction. The user shall be informed when completing an asynchronous and heavy task in order to attain reliability as well as user's satisfaction. |
| **Portability – Compatibility** | The platform shall be visible and usable in any digital environment. Since Virtual Lab is promoting remote participation and engagement, users may utilize different devices. |
| **Scalability / Multi-contextuality** | Although the Virtual Lab will initially be used for promoting data sharing for mobility in Europe and tested/improved through the Living Lab instances, the ultimate goal of its implementation is the effective improvement in the culture of data sharing in general. The platform shall be built in a way that could easily be upgraded to include more areas of expertise that would benefit from data sharing (e.g., logistics). |

## 4.2. Use Cases

Table 3 presents platform's use cases along with their description, corresponding actors and the use of 3PP if existing. All the Virtual Lab use cases are then formalised according to the standardised UML model and presented in individual Tables.

*Table 3: Use Cases Description*

| Name | Actors | Description | Use of 3PP |
|---|---|---|---|
| **Create and validate a profile** | All potential regular users | Users will be able to create a profile using their email. After creating the profile, the user will receive a confirmation mail with a link that will validate the credentials. The profile will also distinguish the users in categories as every profile selects a role in subscription form. | X |
| **Log in to a profile** | Subscribed Users | User will be able to log into their profile using a password system and their email. | |
| **Edit Profile** | Subscribed Users | User will be able to edit the profile details defined in the creation. | |
| **Delete Profile** | Subscribed Users | User will be able to delete their profile. | |

Funded by the European Union

| | | | |
|---|---|---|---|
| **Forgot/Reset password** | Subscribed Users | User will be able to reset their password in case their forgot them. | |
| **Publish/Delete/Browse/Edit Code Snippets** | Subscribed users | Users will be able to publish code snippets by creating a post that provides a short description and a way to download the code or a link to a repository. Also browse and download from a list of published code Snippets. | X |
| **Publish/Edit/Delete/Browse text posts** | Subscribed users | Users will be able to create a forum post to initiate a discussion. | |
| **Comments on text posts** | Subscribed users | Users will be able to communicate and share their opinions through comments in text posts. | |
| **Publish /Delete/Browse/Edit dataset posts** | Subscribed users | Users will be able to publish their dataset by creating a dataset post and provide a short description and a way to download the dataset or a link to a repository. Also browse and download from the proper page. | |
| **Publish /Browse/Delete/Edit service posts** | Subscribed users | User will be able to publish their service by creating a service post and provide a short description of how this tool helped him to solve a problem and a link that leads to it. Services post basically represents a tool that helps uses solve a problem. | |
| **Hashtag reference system** | Subscribed users | A system that is used to quick reference other platform elements like, a code snippet, or a dataset, or a text post, or a service | |
| **Notification** | Subscribed users | Users will be notified in case of an event that might be of an interest or in case someone makes an action that affects them, like reference one of their code snippets. Also, in case a referenced code snippet is deleted, the user will be notified that the snippet he/she referenced, was deleted. | |
| **Subscribe in a newsletter** | People interested in the platform | Users will be able to subscribe in platform's newsletter and be notified via mail about new upcoming living labs. | |
| **Insert challenge in a living lab** | Partners and Virtual Lab team | Certain certified users from AETHON team will be able to create/hosts hackathons. These will be showed in the browse hackathons page along with their details about the rewards, locations etc. | |
| **Insert/Edit/Close living labs** | Partners and Virtual Lab team | The first period of virtual lab, only certified users from AETHON team will have the access to host living labs. In the future, probably this will change and give access to other users for hosting a living lab. | |
| **Browse through the current, past, and future living labs** | People interested in the platform | Users will be able to browse through the current, past, and future living labs from a list that also provides a redirect link for the details. (Possible integration with Open Knowledge Database). | X |
| **Submit a solution to a challenge of a living lab** | Subscribed and Certified Users | Users will be able submit a solution to a challenge referencing code snippets and datasets, services, posts and write a small description. In this submission, they will have to | |

Funded by the European Union

| | | | |
|---|---|---|---|
| | | check about the copyrights and approve terms and conditions. | |
| **Voting on a challenge winner** | Subscribed and Certified Users | Users will be able to vote on a challenge solution, to make a winner suggestion. | |
| **Providing feedback** | People interested in platform | Users will be able to provide feedback for bugs they identify during platform usage or suggestions on future developed functionalities. | |
| **Comments on challenges** | Subscribed users | Users will be able to comment on a challenge for some details. | |
| **Publish/Browse/Download/Edit* Datasets (Integration with Transport Cloud)** | Subscribed users | Users will be able to publish and edit their transport data. Also browse the public data of other users and download them. | X |

For better understanding of the use cases, a template was created, Table 4, that contains all the needed information to describe a use case flow.

- **Actor** can be any entity (a human or other external system) that interacts with the app. In our case, both the travellers and the operators.
- **Description** provides a brief explanation of the sequence of actions and the outcome of the use case.
- **Trigger** is what initiated the interaction between the actor and the platform
- **Preconditions** are any conditions that must be true before the use case starts.
- **Postconditions** are the conditions that must be true when the use case ends, no matter which scenario is executed.
- **Normal flow** is a detailed list of actions, which define the interactions between the actor and the platform and lead to achieve the actors’ goal.
- **Alternative flows** are the deviations from the main scenario, meaning that the actor uses different steps from the normal flow, but in the end the original goal is achieved.
- **Exceptions** are anything that lead to not achieving the use case's goal.
- **Frequency** of use is an estimation of the number of times this use case will be performed by the actors.
- **Special requirements** are anything that needs to be explicitly defined for a specific use case.
- **Notes and issues** are any additional comments or issues that need to be solved.

Funded by the European Union

*Table 4: Use Case Template*

| | | | |
|---|---|---|---|
| **Use Case ID:** | | | |
| **Use Case Title:** | | | |
| **Created By:** | | **Last Revision By:** | |
| **Date Created:** | | **Last Revision Date:** | |
| **Actors:** | | | |
| **Description:** | | | |
| **Trigger** | | | |
| **Preconditions:** | | | |
| **Postconditions:** | | | |
| **Normal Flow:** | | | |
| **Alternative Flows:** | | | |
| **Exceptions:** | | | |
| **Frequency of Use:** | | | |
| **Special Requirements:** | | | |
| **Notes and Issues:** | | | |

Funded by the European Union

*Table 5: UC 1 – Create profile*

| | | | |
|---|---|---|---|
| **Use Case ID:** | 1 | | |
| **Use Case Title:** | **Create profile** | | |
| **Created By:** | **Alexis Efraimidis** | **Last Revision By:** | **Anestis Merdani** |
| **Date Created:** | | **Last Revision Date:** | **09/08/2022** |
| **Actors:** | All potential regular users | | |
| **Description:** | Users will be able to create a profile using their email. After creating the profile, the user will receive a confirmation mail with a link that will validate the credentials and the provided email. | | |
| **Trigger** | A user browses the platform and clicks the subscribe button in the home page | | |
| **Preconditions:** | None | | |
| **Postconditions:** | A validated profile in the platform | | |
| **Normal Flow:** | 1. A potential user opens a browser and visits the Virtual Lab home page<br>2. This user visits the subscribe page by clicking the corresponding button in the home page<br>3. The user completes the questions required for proceeding (username, email, password, organization, job title, role, location*)<br>4. The user receives an email in the address that was provided on the previous step to confirm their user credentials and email.<br>5. The user clicks on the link which will direct them to the profile page, logged in and validated<br>6. The user from now on can use the credentials to log in to the platform at any time | | |
| **Alternative Flows:** | Alternative Flow #1<br>1. A profile already uses the provided email<br>2. User continues from step 3 of Normal Flow with a different email<br>Alternative Flow #2<br>1. Password confirmation and password don't match<br>2. User continues from step 3 of Normal flow and changes his/her password so the password confirmation and password match | | |
| **Exceptions:** | • The user has already a profile in the platform with the provided email | | |
| **Frequency of Use:** | Once | | |
| **Special Requirements:** | Password must be at least 8 characters and must include at least one special character | | |
| **Notes and Issues:** | *More details will be requested. These will be defined in a later face | | |

Table 6: UC 2 – Log in

| | | | |
|---|---|---|---|
| **Use Case ID:** | 2 | | |
| **Use Case Title:** | **Log in** | | |
| **Created By:** | **Alexis Efraimidis** | **Last Revision By:** | **Anestis Merdani** |
| **Date Created:** | | **Last Revision Date:** | **09/08/2022** |
| **Actors:** | Subscribed User | | |
| **Description:** | Any subscribed user will be able to log into his/her profile using a password system | | |
| **Trigger** | Clicking the log in button in the platform and inserting his/her credentials | | |
| **Preconditions:** | • A user has a profile in the platform (UC 1 – Create profile)<br>• A user must confirm the profile via the email he/she received (UC 1 – Create profile) | | |
| **Postconditions:** | Having access to all advance features that require a validated profile | | |
| **Normal Flow:** | 1. A user browses the home page of platform and clicks the log in button<br>2. The user is redirected to a page that has a form with text boxes that prompt the user to input of the email and password of their respective profile<br>3. The user inserts the email and password for the profile that he/her has created on UC 1 – Create profile<br>4. If the credentials are correct the user is redirected to the profile page, having all the details about the logged in profile and access to the advanced features of the platform until he/she logs out or closes the browser window (end the session) | | |
| **Alternative Flows:** | Alternative Flow #1<br>8. User inserts wrong credentials (either password, email, or the combination)<br>9. User will see an error message which describes what went wrong<br>10. User continues from step 3 of the Normal flow with the correct credentials | | |
| **Exceptions:** | • User does not have a profile<br>• User hasn't received the validation email | | |
| **Frequency of Use:** | High | | |
| **Special Requirements:** | Insert credentials that were used in profile creation (UC 1 – Create profile) | | |
| **Notes and Issues:** | None | | |

Funded by the European Union

*Table 7: UC 3 – Edit Profile*

| | | | |
|---|---|---|---|
| **Use Case ID:** | **3** | | |
| **Use Case Title:** | **Edit profile** | | |
| **Created By:** | **Alexis EFRAIMIDIS** | **Last Revision By:** | **Anestis MERDANI** |
| **Date Created:** | | **Last Revision Date:** | **09/08/2022** |
| **Actors:** | Subscribed User | | |
| **Description:** | User will be able to log into his/her profile using a password system and edit the profile details defined in the subscription | | |
| **Trigger** | A user makes changes on the profile page and submits them | | |
| **Preconditions:** | • A user has a validated profile in the platform<br>• User must be logged in | | |
| **Postconditions:** | An updated profile with the new details | | |
| **Normal Flow:** | 1. The user is directed to the profile page, having all the details about his/her profile<br>2. The user selects which of the fields with details will be the ones that will perform changes<br>3. The user makes the changes<br>4. The user submits the changes by pressing the submit button on the bottom of the page<br>5. Platform refreshes the page with the new details | | |
| **Alternative Flows:** | Alternative Flow #1<br>5. The new details are not compatible with the required ones<br>6. The user will get an error message that describes what went wrong<br>7. The user continues from step 2 of Normal Flow and insert the details in proper format | | |
| **Exceptions:** | | | |
| **Frequency of Use:** | Medium - Low | | |
| **Special Requirements:** | None | | |
| **Notes and Issues:** | None | | |

*Table 8: UC 4 – Delete Profile*

| | | | |
|---|---|---|---|
| **Use Case ID:** | **4** | | |
| **Use Case Title:** | **Delete Profile** | | |
| **Created By:** | **Alexis EFRAIMIDIS** | **Last Revision By:** | **Anestis MERDANI** |
| **Date Created:** | | **Last Revision Date:** | **09/08/2022** |
| **Actors:** | Subscribed User | | |
| **Description:** | User will be able to log into his/her profile using a password system and delete this profile | | |
| **Trigger** | User deletes his/her profile in the platform | | |
| **Preconditions:** | • A valid profile in platform<br>• User must be logged in | | |
| **Postconditions:** | The user profile will not be available anymore | | |
| **Normal Flow:** | 1. The user is directed to his/her profile<br>2. The user selects the "Delete profile" button<br>3. Platform shows a confirmation message for his/her profile deletion<br>4. The user accepts the confirmation message<br>5. The user is redirected to the platforms home page with no logged in credentials | | |
| **Alternative Flows:** | Alternative Flow #1<br>4. The user rejects the confirmation message<br>5. The user keeps his profile credentials | | |
| **Exceptions:** | | | |
| **Frequency of Use:** | Very Rare | | |
| **Special Requirements:** | None | | |
| **Notes and Issues:** | The user data will be kept for a short number of days for possible account restoration | | |

Funded by the European Union

*Table 9: UC 5 – Forgot/Reset Password*

| | | | |
|---|---|---|---|
| **Use Case ID:** | 5 | | |
| **Use Case Title:** | **Forgot/Reset Password** | | |
| **Created By:** | **Anestis MERDANI** | **Last Revision By:** | |
| **Date Created:** | **09/08/2022** | **Last Revision Date:** | |
| **Actors:** | Subscribed User | | |
| **Description:** | User will be able to reset his/her password in case he/her forgot it | | |
| **Trigger** | A user forgets his/her password | | |
| **Preconditions:** | A valid profile in platform | | |
| **Postconditions:** | The user will be able again to login in with he/her profile. | | |
| **Normal Flow:** | 1. The user is directed to login page<br>2. The user clicks on button forgot password<br>3. The user enters the email of his/her registered profile and presses submit button<br>4. The user will receive an email at his/her email account, that will redirect him/her to reset password page<br>5. The user completes the inputs fields with his/her new password and press submit<br>11. The user is redirected to the platform's login page<br>12. User can login with new password | | |
| **Alternative Flows:** | Alternative Flow #1<br>6. User's password and confirmation password do not match<br>7. User continues from step 5 of the Normal Flow | | |
| **Exceptions:** | The user profile does not exist | | |
| **Frequency of Use:** | Very Rare | | |
| **Special Requirements:** | Password must be at least 8 characters and include at least one special character | | |
| **Notes and Issues:** | | | |

Funded by the European Union

*Table 10: UC 6 – Publish posts (code, dataset, text, service)*

| | | | |
|---|---|---|---|
| **Use Case ID:** | **6** | | |
| **Use Case Title:** | **Publish posts (code, dataset, text, service)** | | |
| **Created By:** | **Alexis EFRAIMIDIS** | **Last Revision By:** | **Anestis MERDANI** |
| **Date Created:** | | **Last Revision Date:** | **09/08/2022** |
| **Actors:** | Subscribed user | | |
| **Description:** | Users will be able to publish posts of any type by creating a post that provides a title, a short description and a way to download the code or a link to a repository or a link to a service | | |
| **Trigger** | The user visits one of the post pages (text, code, dataset, services) and wants to create one or he/she desires to create one, directly from sidebar options | | |
| **Preconditions:** | • User is subscribed<br>• User is logged in | | |
| **Postconditions:** | New post has been created | | |
| **Normal Flow:** | 1. User browses to one of the posts pages (text, code, dataset, service)<br>2. User desires to create a post<br>3. User clicks on button add<br>4. Platform shows a form with the required inputs (title, description, a useful link, references)<br>5. User fills in the form (title, description, useful link)<br>6. User can fill the input of references by referencing an already existing post (with its reference id) that helped him/her to reach a conclusion UC 9 – Hashtag reference system<br>7. User after filling in the form press the button create<br>8. His/her new post has been created | | |
| **Alternative Flows:** | Alternative Flow #1<br>8. User inputs are not in a valid format<br>9. Platform shows an error message and describes to user what went wrong<br>10. User continues from step 4 of Normal Flow<br>Alternative Flow #2<br>5. User closes the form (Step 4 of Normal Flow)<br>6. User continues his/her actions to the platform | | |
| **Exceptions:** | Users without a valid profile in the platform or not logged in with their profile | | |
| **Frequency of Use:** | Frequent | | |
| **Special Requirements:** | None | | |
| **Notes and Issues:** | *This feature is not meant to replace any other code hosting platforms like GitHub but centralize references to code for transport solution and hackathons submissions | | |

Funded by the European Union

*Table 11: UC 7 – Edit posts (code, dataset, text, service)*

| | | | |
|---|---|---|---|
| **Use Case ID:** | **7** | | |
| **Use Case Title:** | **Edit posts (code, dataset, text, service)** | | |
| **Created By:** | **Anestis MERDANI** | **Last Revision By:** | |
| **Date Created:** | | **Last Revision Date:** | |
| **Actors:** | Subscribed user | | |
| **Description:** | User will be able to edit posts after its creation | | |
| **Trigger** | A subscribed user wants to change his/her post | | |
| **Preconditions:** | • User is subscribed<br>• User is logged in<br>• User has at least one created post | | |
| **Postconditions:** | User's post is updated | | |
| **Normal Flow:** | 1. User goes to his/her profile<br>2. User filters his posts based on post type (code, dataset, text, service)<br>3. User finds his/her post that desires to update<br>4. User presses the update button<br>5. Platform shows a form with the post's current values<br>6. User selects the fields he/her wants to edit<br>7. User updates the fields and presses the submit button<br>8. User's post has been updated | | |
| **Alternative Flows:** | Alternative Flow #1<br>7. User inputs are not in a valid format<br>8. Platform shows an error message and describes to user what went wrong<br>9. User continues from step 6 of Normal Flow<br>Alternative Flow #2<br>6. User closes the form (Step 5 of Normal Flow)<br>7. User continues his/her actions to the platform | | |
| **Exceptions:** | Users without a valid profile in the platform or not logged in with their profile | | |
| **Frequency of Use:** | Frequent | | |
| **Special Requirements:** | None | | |
| **Notes and Issues:** | *This feature is not meant to replace any other code hosting platforms like GitHub but centralize references to code for transport solution and hackathons submissions | | |

Funded by the European Union

*Table 12: UC 8 – Delete posts (code, dataset, text, service)*

| | | | |
|---|---|---|---|
| **Use Case ID:** | **8** | | |
| **Use Case Title:** | **Delete posts (code, dataset, service, text)** | | |
| **Created By:** | **Anestis MERDANI** | **Last Revision By:** | |
| **Date Created:** | **09/08/2022** | **Last Revision Date:** | |
| **Actors:** | Subscribed users | | |
| **Description:** | Users will be able to delete a post after its creation | | |
| **Trigger** | User desires to delete a post (code, service, dataset, text) | | |
| **Preconditions:** | • User has profile<br>• User is logged in<br>• User has at least one created post | | |
| **Postconditions:** | User's post has been deleted | | |
| **Normal Flow:** | 1. User goes to his/her profile<br>2. User find his/her post<br>3. User presses the delete button<br>4. Platform shows a confirmation message and asks the user if he/she is sure of this action<br>5. Users accept the confirmation message<br>6. User's post has been deleted | | |
| **Alternative Flows:** | Alternative Flow #1<br>5. User rejects the confirmation message<br>6. User continues his/her experience in the platform | | |
| **Exceptions:** | Users without a valid profile in the platform or not logged in their profile | | |
| **Frequency of Use:** | Frequent | | |
| **Special Requirements:** | Understanding the hashtag concept | | |
| **Notes and Issues:** | None | | |

Funded by the European Union

*Table 13: UC 9 – Hashtag reference system*

| | | | |
|---|---|---|---|
| **Use Case ID:** | **9** | | |
| **Use Case Title:** | **Hashtag reference system** | | |
| **Created By:** | **Anestis MERDANI** | **Last Revision By:** | |
| **Date Created:** | **09/08/2022** | **Last Revision Date:** | |
| **Actors:** | Subscribed users | | |
| **Description:** | Users will be able to reference other posts that helped them to reach a conclusion | | |
| **Trigger** | User desires to reference another post | | |
| **Preconditions:** | • User has profile<br>• User is logged in<br>• User has at least one created post<br>• Post that will be referenced must not be deleted | | |
| **Postconditions:** | Post references has been inserted | | |
| **Normal Flow:** | 1. User creates a code post Table 10: UC 6 – Publish posts (code, dataset, text, service)<br>2. User at reference input wants to reference a dataset that helped him<br>3. User finds the reference id of that dataset post<br>4. User inserts that id in the reference input<br>5. User selects the reference result<br>6. User presses submit button<br>7. User has successfully referenced another post | | |
| **Alternative Flows:** | Alternative Flow #1<br>5. The platform shows that reference id has no result because the post has been deleted<br>6. Users finds another post to reference or continues without references | | |
| **Exceptions:** | Users without a valid profile in the platform or not logged in with their profile | | |
| **Frequency of Use:** | Frequent | | |
| **Special Requirements:** | Understanding the hashtag concept | | |
| **Notes and Issues:** | None | | |

*Table 14: UC 10 – Notifications*

| | | | |
|---|---|---|---|
| **Use Case ID:** | **10** | | |
| **Use Case Title:** | **Notifications** | | |
| **Created By:** | **Alexis EFRAIMIDIS** | **Last Revision By:** | **Anestis MERDANI** |
| **Date Created:** | | **Last Revision Date:** | **09/08/2022** |
| **Actors:** | Subscribed users | | |
| **Description:** | Users will be able to be notified in case someone makes an action that affects them, like reference one of their code snippets. Also, in case a referenced post is deleted, the user will be notified that the post he/she referenced was deleted | | |
| **Trigger** | User wants to be notified in case someone makes an action that affects them | | |
| **Preconditions:** | • User has profile<br>• User is logged in | | |
| **Postconditions:** | User will receive a notification | | |
| **Normal Flow:** | 1. User2 creates a post<br>2. User2 references a post from User<br>3. User will receive a notification | | |
| **Alternative Flows:** | Alternative Flow #1<br>1. User creates a post<br>2. User references a post from User2<br>3. User2 deletes his post<br>4. User will receive a notification that the post he referenced has been deleted<br>Alternative Flow #2<br>1. User creates a post type text<br>2. User2 makes a comment on that post<br>3. User will receive a notification that someone made a comment on his/her post<br>Alternative Flow #3<br>1. User creates a challenge<br>2. User2 comments on the challenge for details<br>3. User will receive a notification that someone commented on his/her challenge | | |
| **Exceptions:** | Users without a valid profile in the platform or not logged in with their profile | | |
| **Frequency of Use:** | Frequent | | |
| **Special Requirements:** | | | |
| **Notes and Issues:** | | | |

Funded by the European Union

*Table 15: UC 11 – Subscribe to newsletter*

| | | | |
|---|---|---|---|
| **Use Case ID:** | **11** | | |
| **Use Case Title:** | **Subscribe to newsletter** | | |
| **Created By:** | **Alexis EFRAIMIDIS** | **Last Revision By:** | **Anestis MERDANI** |
| **Date Created:** | | **Last Revision Date:** | **09/08/2022** |
| **Actors:** | Subscribed/Not subscribed users | | |
| **Description:** | Users will be able to subscribe in a newsletter so they can be notified in case of a new living lab or in case of a new challenge | | |
| **Trigger** | Any person browsing through platform can click on subscribe to the newsletter | | |
| **Preconditions:** | Users must have an email account | | |
| **Postconditions:** | The users will receive emails with some important information about the platform or about new challenges and living labs | | |
| **Normal Flow:** | 1. User opens the platform<br>2. User clicks the button newsletter on the navigation bar<br>3. Platform shows a form with an input field (email)<br>4. User completes the field and submits the form<br>5. User has successfully subscribed to the platform's newsletter<br>6. User will receive an email that includes newsletter subscription and a unsubscribe link in case the user desires to unsubscribe from newsletter. | | |
| **Alternative Flows:** | | | |
| **Exceptions:** | Users without a valid profile in the platform or not logged in with their profile | | |
| **Frequency of Use:** | Frequent | | |
| **Special Requirements:** | User must own an email account | | |
| **Notes and Issues:** | | | |

Funded by the European Union

*Table 16: UC 12 – Host a living lab*

| | | | |
|---|---|---|---|
| **Use Case ID:** | 12 | | |
| **Use Case Title:** | **Host a living lab** | | |
| **Created By:** | **Anestis MERDANI** | **Last Revision By:** | |
| **Date Created:** | **09/08/2022** | **Last Revision Date:** | |
| **Actors:** | Subscribed users with profile role “admin” | | |
| **Description:** | Users will be able to host a living lab | | |
| **Trigger** | User clicks the host a living lab button | | |
| **Preconditions:** | • User has profile<br>• User is logged in<br>• User has the right to host a living lab (in the first period of the platform, only admin users will have the access to host living labs. In the future, probably this will change and give access to other users for hosting a living lab.) | | |
| **Postconditions:** | User is hosting a living lab | | |
| **Normal Flow:** | 1. User is browsing through current living labs page<br>2. User wants to host a living lab<br>3. User clicks host a living lab button<br>4. User will be redirected to living lab’s form<br>5. User completes the form<br>6. User presses the submit button<br>7. User is hosting a living lab | | |
| **Alternative Flows:** | Alternative Flow #1<br>7. User has not the rights of hosting a living lab<br>8. User closes the form and continues his/her experience to the platform<br>Alternative Flow #2<br>7. User does not have access to host a living lab<br>8. User makes a request for updating his/her role<br>9. The platform accepts his request<br>10. Users continues from step 3 of Normal Flow | | |
| **Exceptions:** | Users without a valid profile in the platform or not logged in with their profile or users with not rights for this action. | | |
| **Frequency of Use:** | Frequent | | |
| **Special Requirements:** | User must have the rights of hosting a living lab. These rights are based on user roles at profile creation. | | |
| **Notes and Issues:** | | | |

Funded by the European Union

*Table 17: UC 13 – Edit a living lab*

| | | | |
|---|---|---|---|
| **Use Case ID:** | 13 | | |
| **Use Case Title:** | **Edit a living lab** | | |
| **Created By:** | **Anestis MERDANI** | **Last Revision By:** | |
| **Date Created:** | **09/08/2022** | **Last Revision Date:** | |
| **Actors:** | Subscribed users with profile role “admin” | | |
| **Description:** | Users will be able edit an existing living lab | | |
| **Trigger** | User clicks the edit living lab button | | |
| **Preconditions:** | • User has profile<br>• User is logged in<br>• User hosts at least one living lab | | |
| **Postconditions:** | User’s living lab has been updated | | |
| **Normal Flow:** | 1. User navigates to his/her profile<br>2. User find his/her living lab<br>3. User presses the edit button<br>4. User will be navigated to the edit living lab page, input fields will be completed with the current values<br>5. User updates the fields<br>6. User presses the submit button<br>7. User’s living lab has been successfully updated | | |
| **Alternative Flows:** | Alternative Flow #1<br>7. User’s updated fields are not in proper format<br>8. The platform shows a message which describes to user what went wrong<br>9. User updates the input fields in the proper format<br>10. User continues from step 6 of Normal Flow | | |
| **Exceptions:** | Users without a valid profile in the platform or not logged in with their profile | | |
| **Frequency of Use:** | Frequent | | |
| **Special Requirements:** | User must have the rights of hosting a living lab. These rights are based on user roles at profile creation. | | |
| **Notes and Issues:** | | | |

*Table 18: UC 14 – Create a challenge*

| | | | |
|---|---|---|---|
| **Use Case ID:** | 14 | | |
| **Use Case Title:** | **Create a challenge** | | |
| **Created By:** | **Anestis MERDANI** | **Last Revision By:** | |
| **Date Created:** | **10/08/2022** | **Last Revision Date:** | |
| **Actors:** | Subscribed users with profile role "challenger" | | |
| **Description:** | User will be able to create challenges in every available living lab | | |
| **Trigger** | User clicks insert challenge button on living labs details page | | |
| **Preconditions:** | • User has profile<br>• User is logged in<br>• At least one living lab must be available<br>• User has the rights to create a challenge (The first period of the platform, only users with role challenger will have the access to create challenges. In the future, probably this will change and give access also to other users.) | | |
| **Postconditions:** | User creates his own challenge in a living lab | | |
| **Normal Flow:** | 1. User desires to create a challenge<br>2. User navigates to living labs page<br>3. User finds the proper living lab for his challenge<br>4. User clicks on the living lab<br>5. User navigates to living labs details page<br>6. User clicks the insert challenge button<br>7. Platform shows a form with the required inputs<br>8. User completes the form<br>9. User presses the submit button<br>10. User's new challenge has been created | | |
| **Alternative Flows:** | Alternative Flow #1<br>10. User inputs are not in the requested format<br>11. Platform shows a message which describes to the user about the proper formats<br>12. User continues from step 8 of the Normal Flow with the requested formats<br>Alternative Flow #2<br>10. User does not have access in challenge creation<br>11. The platform shows a message which describe that his/her current profile role has no access for this action<br>12. User make a request for updating his profile role and continues from step 8 of Normal Flow or closes the form and continues his navigation to the platform | | |
| **Exceptions:** | Users without a valid profile in the platform or not logged in with their profile or users with not rights for this action. | | |
| **Frequency of Use:** | Frequent | | |

Funded by the European Union

| | |
|---|---|
| **Special Requirements:** | User must have the rights of creating a challenge. These rights are based on user roles at profile creation. |
| **Notes and Issues:** | |

*Table 19: UC 15 – Edit a challenge*

| **Use Case ID:** | 15 | | |
|---|---|---|---|
| **Use Case Title:** | **Edit a challenge** | | |
| **Created By:** | **Anestis MERDANI** | **Last Revision By:** | |
| **Date Created:** | **10/08/2022** | **Last Revision Date:** | |
| **Actors:** | Subscribed users with profile role "challenger" | | |
| **Description:** | Users will be able edit an existing challenge | | |
| **Trigger** | User clicks the edit button of a challenge on his/her profile | | |
| **Preconditions:** | • User has profile<br>• User is logged in<br>• User has at least one challenge | | |
| **Postconditions:** | User's challenge is updated | | |
| **Normal Flow:** | 1. User navigates to his/her profile<br>2. User finds his/her challenge<br>3. User presses the edit button<br>4. Platform navigates the user to the edit challenge page, where input fields will be completed with the current values<br>5. User updates the fields<br>6. User presses the submit button<br>7. User's challenge has been successfully updated | | |
| **Alternative Flows:** | Alternative Flow #1<br>7. User's updated fields are not in proper format<br>8. The platform shows a message which describes to the user what went wrong<br>9. User updates the input fields in the proper format<br>10. User continues from step 6 of Normal Flow | | |
| **Exceptions:** | Users without a valid profile in the platform or not logged in with their profile | | |
| **Frequency of Use:** | Frequent | | |
| **Special Requirements:** | | | |
| **Notes and Issues:** | | | |

Funded by the European Union

*Table 20: UC 16 – Delete a challenge/ living lab*

| | | | |
|---|---|---|---|
| **Use Case ID:** | 16 | | |
| **Use Case Title:** | **Delete challenge/ living lab** | | |
| **Created By:** | **Anestis MERDANI** | **Last Revision By:** | |
| **Date Created:** | **10/08/2022** | **Last Revision Date:** | |
| **Actors:** | Subscribed users | | |
| **Description:** | Users will be able edit an existing living lab / challenge | | |
| **Trigger** | User clicks the delete button of a challenge/ living lab on his/her profile | | |
| **Preconditions:** | • User has profile<br>• User is logged in<br>• User has at least one challenge/living lab | | |
| **Postconditions:** | User's challenge/living lab is deleted | | |
| **Normal Flow:** | 1. User navigates to his/her profile<br>2. User finds his/her challenge/living lab<br>3. User presses the delete button<br>4. Platform shows a confirmation message for this action<br>5. User accepts the message<br>6. Platform shows a message that user's challenge/ living lab has been successfully deleted | | |
| **Alternative Flows:** | Alternative Flow #1<br>5. User rejects the confirmation message<br>6. User continues his/her navigation to the platform | | |
| **Exceptions:** | Users without a valid profile in the platform or not logged in with their profile | | |
| **Frequency of Use:** | Frequent | | |
| **Special Requirements:** | | | |
| **Notes and Issues:** | | | |

Funded by the European Union

*Table 21: UC 17 – Submit solution on a challenge*

| | | | |
|---|---|---|---|
| **Use Case ID:** | 17 | | |
| **Use Case Title:** | **Submit solution on a challenge** | | |
| **Created By:** | **Anestis MERDANI** | **Last Revision By:** | |
| **Date Created:** | **10/08/2022** | **Last Revision Date:** | |
| **Actors:** | Subscribed users with profile role “contestant” | | |
| **Description:** | Users will profile role “contestant” will be able to submit solution on a challenge | | |
| **Trigger** | User clicks the submit solution button on challenge details page | | |
| **Preconditions:** | • User has profile<br>• User is logged in | | |
| **Postconditions:** | User’s solution has been submitted | | |
| **Normal Flow:** | 1. User navigates to challenge details page<br>2. User clicks the submit solution button<br>3. Platform navigates the user to solution form<br>4. User completes the form input fields<br>5. User presses the submit button<br>6. Platform shows him/her a message that his/her solution was successfully submitted | | |
| **Alternative Flows:** | Alternative Flow #1<br>6. Platform shows the user an error message, that his/her inputs are not in the proper format<br>7. User continues from step 4 of Normal Flow with the requested formats | | |
| **Exceptions:** | Users without a valid profile in the platform or not logged in with their profile | | |
| **Frequency of Use:** | Frequent | | |
| **Special Requirements:** | User’s profile role must be contestant | | |
| **Notes and Issues:** | | | |

Funded by the European Union

*Table 22: UC 18 – Vote on a solution*

| | | | |
|---|---|---|---|
| **Use Case ID:** | 18 | | |
| **Use Case Title:** | **Vote on a solution** | | |
| **Created By:** | **Anestis MERDANI** | **Last Revision By:** | |
| **Date Created:** | **10/08/2022** | **Last Revision Date:** | |
| **Actors:** | Subscribed users | | |
| **Description:** | Users will be able to vote on a challenge's solution, to make a winner suggestion. | | |
| **Trigger** | User clicks the vote button on solution details page | | |
| **Preconditions:** | • User has profile<br>• User is logged in | | |
| **Postconditions:** | User's vote on the solution has been submitted | | |
| **Normal Flow:** | 1. User navigates to challenge's solutions<br>2. User reads some of the solutions<br>3. User finds a solution that he/she thinks is the best<br>4. User clicks the vote button on solution details page<br>5. Flatform shows to the user that his/her vote has been submitted | | |
| **Alternative Flows:** | | | |
| **Exceptions:** | Users without a valid profile in the platform or not logged in with their profile | | |
| **Frequency of Use:** | Frequent | | |
| **Special Requirements:** | | | |
| **Notes and Issues:** | | | |

Funded by the European Union

*Table 23: UC 19 – Feedback*

| | | | |
|---|---|---|---|
| **Use Case ID:** | 19 | | |
| **Use Case Title:** | **Feedback** | | |
| **Created By:** | **Anestis MERDANI** | **Last Revision By:** | |
| **Date Created:** | **30/08/2022** | **Last Revision Date:** | |
| **Actors:** | All users | | |
| **Description:** | User will be able to provide feedback for bugs they identify during platform usage or suggestions on future developed functionalities | | |
| **Trigger** | User clicks the feedback button. Button is available on every page | | |
| **Preconditions:** | | | |
| **Postconditions:** | Platform receives the feedback message from user | | |
| **Normal Flow:** | 1. User wants to send a feedback message<br>2. User clicks the feedback button<br>3. Platform shows a form<br>4. User selects if the feedback message is for a bug report or a new suggested feature<br>5. User completes the description input<br>6. User submits the form<br>7. Platform receives his/her feedback message | | |
| **Alternative Flows:** | | | |
| **Exceptions:** | Users without a valid profile in the platform or not logged in with their profile | | |
| **Frequency of Use:** | Rare | | |
| **Special Requirements:** | | | |
| **Notes and Issues:** | | | |

Funded by the European Union

*Table 24: UC 20 – Comments*

| | | | |
|---|---|---|---|
| **Use Case ID:** | **20** | | |
| **Use Case Title:** | **Comments** | | |
| **Created By:** | **Anestis MERDANI** | **Last Revision By:** | |
| **Date Created:** | **30/08/2022** | **Last Revision Date:** | |
| **Actors:** | Subscribed users | | |
| **Description:** | User will be able to comment on a challenge for some details or on text posts and have a discussion like a forum. | | |
| **Trigger** | User wants to comment on a challenge for details or on a text post | | |
| **Preconditions:** | • User must have a valid profile<br>• User must be logged in | | |
| **Postconditions:** | User has made a comment | | |
| **Normal Flow:** | 1. User wants details for a challenge<br>2. User navigates to challenge details and goes to discussion section<br>3. Platform shows the comment input field<br>4. User inserts his/her text and press button submit<br>5. User's comment has been successfully created | | |
| **Alternative Flows:** | Alternative Flow #1<br>1. User checks the available text posts<br>2. User finds a text post that is interested<br>3. User navigates on text post details<br>4. Platform shows the details and a comment input field<br>5. User inserts his/her text into the input field and press submit button<br>6. User's comment has been successfully created | | |
| **Exceptions:** | Users without a valid profile in the platform or not logged in with their profile | | |
| **Frequency of Use:** | Frequent | | |
| **Special Requirements:** | | | |
| **Notes and Issues:** | | | |

Funded by the European Union

# 5. Integration Points with Transport Cloud

The Transport Cloud is primarily designed to demonstrate and offer solutions to reduce and, in some cases, remove current technical limitations identified as barriers to data sharing and reuse. It will demonstrate a cloud-based prototype platform for sharing transport data, accessible to interested mobility actors. It is designed according to federated cloud principles. The MobiDataLab platform will highlight how to facilitate access to mobility data, in an open, interoperable, and privacy preserving way, developing open tools, and making them available (D4.1). (Salvatore TRANI 2022)

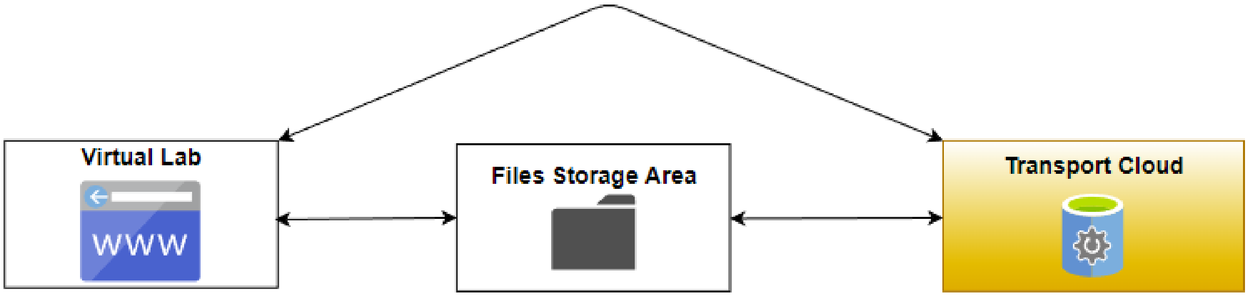


*Figure 2: Generic Use Case diagram of integration with Transport Cloud*

In many use cases (living labs) of the platform, user will have to upload or download transport or any kind of data-metadata in order to submit a solution on a specific challenge. The diagram above shows, how user's data or metadata will be stored or retrieved. Based on the diagram, there are two possible ways of integrating Virtual Lab with Transport Cloud.

The first one is based on a public/shared file storage area. In this area each service will store data-metadata with a specific name convention. Hence, if it is required, the other service can retrieve these data-metadata and provide them to the user.

*Table 25: UC 21- Integration with TC (1)*

| | | | |
|---|---|---|---|
| **Use Case ID:** | **21** | | |
| **Use Case Title:** | **Integration with Transport Cloud based on public/shared file storage** | | |
| **Created By:** | **Anestis MERDANI** | **Last Revision By:** | |
| **Date Created:** | **2/9/2022** | **Last Revision Date:** | |
| **Actors:** | User, Virtual Lab, Transport Cloud | | |
| **Description:** | User will be able to upload or download a dataset. | | |

Funded by the European Union

| | |
|---|---|
| **Trigger** | User desires to download or upload a dataset. |
| **Preconditions:** | User must have a valid profile |
| **Postconditions:** | User has uploaded or downloaded a dataset |
| **Normal Flow:** | 1. User participates in a challenge<br>2. User clicks on the preview(dataset) button<br>3. Platform retrieves the dataset from the shared file store, which was stored from Transport Cloud or Virtual Lab, with a specific name convention.<br>4. User downloads the returned dataset |
| **Alternative Flows:** | Alternative Flow #1<br>1. User desires to create/share a dataset on Virtual Lab<br>2. User inserts his/her dataset to the Virtual lab's form<br>3. User submits the form<br>4. Virtual lab stores it, with a specific name convention in the shared file storage<br>5. Transport Cloud retrieves the data from the shared file storage |
| **Exceptions:** | Users without a valid profile in the platform or not logged in with their profile |
| **Frequency of Use:** | Frequent |
| **Special Requirements:** | |
| **Notes and Issues:** | |

The second one is based on a direct communication between Transport Cloud and Virtual Lab through APIs. This way requires from both services, to have at least one open API. With an open API, Virtual lab can upload and retrieve data or metadata directly from Transport Cloud.

*Table 26: UC 22 - Integration with TC (2)*

| **Use Case ID:** | **22** | | |
|---|---|---|---|
| **Use Case Title:** | **Integration with Transport Cloud via a direct API** | | |
| **Created By:** | **Anestis MERDANI** | **Last Revision By:** | |
| **Date Created:** | **2/9/2022** | **Last Revision Date:** | |
| **Actors:** | User, Virtual Lab, Transport Cloud | | |
| **Description:** | User will be able to upload or download a dataset. | | |
| **Trigger** | User desires to download or upload a dataset. | | |
| **Preconditions:** | User must have a valid profile | | |
| **Postconditions:** | User has uploaded or downloaded a dataset | | |
| **Normal Flow:** | 1. User participates in a challenge<br>2. User clicks on the preview(dataset) button | | |

Funded by the European Union

| | |
|---|---|
| | 3. Virtual Lab retrieves the dataset directly from Transport Cloud with an API request<br>4. Virtual lab shows the dataset to the user<br>5. User downloads the given dataset |
| **Alternative Flows:** | Alternative Flow #1<br>1. User desires to create a dataset on Virtual Lab<br>2. User inserts his/her dataset to the Virtual Lab’s form<br>3. User submits the form<br>4. Virtual lab renames the dataset to a specific name convention<br>5. Virtual lab sends the dataset to Transport Cloud via an API request<br>6. Transport Cloud stores the given dataset<br>7. Dataset can be retrieved directly from Transport Cloud |
| **Exceptions:** | Users without a valid profile in the platform or not logged in with their profile |
| **Frequency of Use:** | Frequent |
| **Special Requirements:** | |
| **Notes and Issues:** | |

# 6. Future Work

During the course of the MobiDataLab project three consequent and gradual versions of the Virtual Lab platform will be developed. This deliverable along with its demonstrator focuses on the development of the first and complete version v1.0.

Following versions of the Virtual Lab will include improvements on features and functionalities (bugs correction) based on the Backlog created and monitored through Task T5.3. Feedback collection during the Virtual Lab usage will also include suggestions on additional platform functionalities. Additional functionalities that will be collected along with further functionalities initially defined in the project proposal process but weren't included in the first version will be incorporated in the future ones.

Additional potential functionalities that can be further explored and incorporated into future platform versions are presented below in Table 27.

*Table 27: Additional Functionalities for future platform versions*

| Name | Actors | Description | Use of 3PP |
|---|---|---|---|
| **Log in through google or F6S account** | Users | Users will be able to log in to the platform via 3PP like google or F6S and not create account manually. | X |
| **Live messaging** | Users | Users will be able to communicate with each other via personal messages. Also, users will be able to create chat groups to share opinions and thoughts. | |
| **Live code pair working** | Users | This feature will allow users to work simultaneously and live on a specific code file. Users will be in groups of 2 or 3. | |
| **Video conferencing for full remote Living Lab implementation** | Users | Platform will allow organizations or users to communicate via video calls. This feature could make living labs fully remote. | |
| **Certificates** | Platform | In the future, platform may provide certificates on challenges winners. This could motivate a lot of users to be more active on participating to platform's challenges. | |
| **Ranking system** | Users | Platform will rank all users based on their activities (participating on challenges, winning challenges, creating challenges.). The more active a user is, the more his/her ranking will grow along with his visibility on the platform. | |
| **Challenges reminder** | Users | Every user will be able to click the reminder button on challenge details, so when the starting date approaches, the platform will | |

| | | | |
|---|---|---|---|
| | | remind him/her about the approaching challenge with a notification or an email. | |
| **Cooperation between two or more organizations on hosting a living lab** | Users | Platform will give the opportunity to organizations to recruit other entities and cooperate towards hosting a living lab together. That means that one living lab could be hosted by more than one organization. | |

Funded by the European Union

# 7. Conclusions

The present document reports on the components and architecture of the MobiDataLab Virtual Lab. The Virtual Lab is a tool created for the testing of the MobiDataLab Transport Cloud and includes functionalities such as forums, polling, live data exchanges and more. The report includes functional, non-functional requirements as wells as all version v1.0 use cases of the platform. The integration between the Virtual Lab and the Transport Cloud is also discussed allowing for further investigation through the course of the project.

Feedback from the usage and testing of the Virtual Lab by the Living Labs participants will be collected incorporated as improvements and updates to the two future versions of the platform until the end of the project. While the first version of the platform is complete in the sense that it covers all major functionalities envisioned (e.g., hosting a living lab, creating a challenge, submitting a solution, access/exchange/reuse/enrich mobility data) it will be further updated and extended as work on the project progresses and intricacies of each living lab execution become apparent.

In the forthcoming period, the report will serve as the guide for further development and update incorporated in future Virtual Lab versions but also as a documentation material for users.

# 8. References

*About.* s.d. https://www.drivendata.org/about/.

*cloudamqp.* s.d. https://www.cloudamqp.com/blog/part1-rabbitmq-for-beginners-what-is-rabbitmq.html (accès le 08 31, 2022).

Comartin, Derek. *asynchronous-request-response-pattern-for-non-blocking-workflows.* codeopinion. 13 04 2022. https://codeopinion.com/asynchronous-request-response-pattern-for-non-blocking-workflows/ (accès le 09 01, 2022).

*Landing.* s.d. https://www.kaggle.com/.

*Landing.* s.d. https://hackathons.copernicus.eu/ (accès le 08 01, 2022).

Makai, Matt. *Celery.* Fullstackpython. s.d. https://www.fullstackpython.com/celery.html (accès le 09 01, 2022).

Salvatore TRANI, Alberto BLANCO JUSTICIA, Thierry CHEVALLIER, Didier de RYCK, Johannes LAUER, Francesco LETTICH, Sorel SIGHOKO. «D4.1 Transport Cloud Architecture Dossier V1.» 2022.

*Teams.* Reactjs. s.d. https://reactjs.org/community/team.html (accès le 09 01, 2022).

*transdiary.* s.d. https://www.transdairy.net/ (accès le 01 04, 2022).

*what-are-microservices.* s.d. https://www.logicmonitor.com/blog/what-are-microservices.

Funded by the
European Union

# MobiDataLab consortium

The consortium of MobiDataLab consists of 10 partners with multidisciplinary and complementary competencies. This includes leading universities, networks and industry sector specialists.

AKKA AETHON ENGINEERING F6S here

ICOOR Kisio KU LEUVEN POLIS UNIVERSITAT ROVIRA i VIRGILI

@MobiDataLab
#MobiDataLab

in https://www.linkedin.com/company/mobidatalab

For further information please visit **www.mobidatalab.eu**

MobiDataLab is co-funded by the EU under the H2020 Research and Innovation Programme (grant agreement No 101006879).

The content of this document reflects solely the views of its authors. The European Commission is not liable for any use that may be made of the information contained therein. The MobiDataLab consortium members shall have no liability for damages of any kind that may result from the use of these materials.