

H2020-SC6-GOVERNANCE-2018-2019-2020

DT-GOVERNANCE-05-2018-2019-2020



D5.3: ACROSS Platform Prototype and Applications - Initial

Project Reference No	959157 — ACROSS — H2020-SC6-GOVERNANCE-2018-2019-2020
Deliverable	D5.3: ACROSS Platform Prototype and Applications - Initial
Work package	WP5: ACROSS Applications and Platform Integration
Nature	Other
Dissemination Level	Public
Date	31/7/2022
Status	Final
Editor(s)	Anestis Sidiropoulos (ATC)
Contributor(s)	Vincenzo Savarino (ENG), Marina Klitsi (ATC), Anna Opaska (FHG), Petros Christopoulos (GRNET)
Reviewer(s)	ENG, GRNET
Document description	This deliverable describes the first working prototype of the ACROSS platform.



About

The project is co-funded by the European Commission's Horizon 2020 research and innovation framework programme. Spanning through three years, ACROSS consists of a consortium of 10 partners from 7 countries: Athens Technology Center (coordinator), Tecnalia, Dataport, Engineering, Fraunhofer, GRNET, TimeLex, The Lisbon Council, Waag and VARAM.

DISCLAIMER

The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Commission. The Commission does not guarantee the accuracy of the data included in this study. Neither the Commission nor any person acting on the Commission's behalf may be held responsible for the use, which may be made of the information contained therein.

© 2021 – European Union. All rights reserved. Certain parts are licensed under conditions to the EU.



Document Revision History

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
V0.1	17/06/2022	First skeleton of the deliverable and TOC	ATC
V0.2	10/07/2022	Initial Contributions	ATC, ENG
V0.3	15/07/2022	First Draft Version	ATC
V0.4	18/07/2022	Updated version sent for Internal Revision	ATC
V0.5	27/07/2022	Revision	ENG, GRNET
V0.6	28/07/2022	Updated version based on the comments provided	ATC
V1.0	31/7/2022	Final version	ATC



Executive Summary

This document describes the first prototype of the ACROSS platform. This is a first working version of the prototype that offers an initial set of the expected functionalities and will act as the test-bed for the ACROSS stakeholders to experience with this initial set of ACROSS provisions and assess the concepts and knowledge conveyed by the project.

The deliverable is describing the platform from two different aspects; from the internal point of view, in which the functional and architectural structure is presented briefly, and from the end-user point of view, in which the external appearance, look-and-feel and the overall high level functionalities and interaction potentials are explained. Moreover, the document provides an overview of the current functionalities exposing every available feature so far.

The ACROSS Platform, far from being a simple container for the individual modules, is a coherent application, where several different components reside and collaborate in harmony. The first prototype is a proof of concept to the ACROSS end users. It encapsulates most of the underlying technologies and gives a clear and easy to use graphical interface. The benefit of the current architecture is that any additional functionality can be wrapped into a separate component and be added to the platform, provided that it abides by the basic communication standards exposed by the ACROSS platform architecture. The scope of this practice is to enable future extensions of the platform to arising functionalities, which may maximise the potentials for further exploitation and adoption of the platform beyond the project end.



Table of Contents

1	INTRODUCTION	1
1.1	PURPOSE AND SCOPE	1
1.2	APPROACH FOR WORK PACKAGE AND RELATION TO OTHER WORK PACKAGES AND DELIVERABLES	1
1.3	METHODOLOGY AND STRUCTURE OF THE DELIVERABLE	1
2	SOFTWARE AND SYSTEM DESIGN PRINCIPLES	2
2.1	OPERATIONAL ENVIRONMENT	2
2.2	SYSTEM ARCHITECTURE OVERVIEW	4
2.3	COMPONENT INTERACTION	5
3	COMPONENTS INTEGRATED FOR THE INITIAL VERSION	7
3.1	SCENARIO IMPLEMENTATION	7
3.2	COMPONENTS INTEGRATION	13
3.2.1	<i>Service Catalogue</i>	14
3.2.2	<i>User Journey Service Engine</i>	14
4	CONCLUSIONS	16
5	REFERENCES	17

List of Figures

FIGURE 1: SYSTEM ARCHITECTURE	4
FIGURE 2: COMPONENTS INTERACTION	5
FIGURE 3: SERVICE INFORMATION SECTIONS INCLUDED IN THE SERVICE CATALOGUE.....	8
FIGURE 4: USER INTERFACE OF THE UJMT.....	8
FIGURE 5: SERVICE SELECTION FOR AN ACTION IN THE UJMT	9
FIGURE 6: GENERATED BPMN DIAGRAM	9
FIGURE 7: LATVIA-TO-GERMANY-TO-STUDY JOURNEY CREATION	10
FIGURE 8: TRANSPARENCY DASHBOARD REDIRECTION.....	10
FIGURE 9: TRANSPARENCY DASHBOARD	10
FIGURE 10: ACROSS JOURNEY STEP PROGRESS	11
FIGURE 11: ACROSS JOURNEY STATUS	11



FIGURE 12: VIRTUAL ASSISTANT USER INTERFACE..... 12
FIGURE 13: SERVICE CATALOGUE APIS 14
FIGURE 14: UJSE APIS 15

List of Terms and Abbreviations

Abbreviation	Definition
UJMT	User Journey Modelling Tool
UJSE	User Journey Service Engine
IMS	Identity Management System
CI/CD	Continuous Integration and Continuous Deployment
K8S	Kubernetes
SSO	Single Sign On
BPMN	Business Process Model and Notation



1 Introduction

1.1 Purpose and Scope

This deliverable aims to provide an overview of the initial release of the ACROSS prototype. To this end the report describes shortly the implementation aspects as well as the current functionality.

1.2 Approach for Work Package and Relation to other Work Packages and Deliverables

The WP5: ACROSS Application and Platform Integration aims at providing the implementation aspects for the delivery of the ACROSS components integration in a unified platform. The design of the ACROSS platform is driven from the user requirements definition and the technical specifications as delivered by the WP6: Use cases deployment, evaluation & impact assessment, WP3: ACROSS Data Governance Framework and WP4: ACROSS Modules Setup.

Following the implementation of the individual modules in WP3: ACROSS Data Governance Framework and WP4: ACROSS Modules Setup, this WP delivers an integrated view of the ACROSS platform to act as the test-bed for setting the ACROSS pilots in WP6: Use cases deployment, evaluation & impact assessment and validating the results in real life scenarios. The content presented in this deliverable is subject to refinements and modifications, based on the progress of the technical work packages, as well as during the validation and evaluation phases of the project.

1.3 Methodology and Structure of the Deliverable

This document reports on the activities and effort placed in the integration of the various technologies and tools provided by the WP3: ACROSS Data Governance Framework and WP4: ACROSS Modules Setup towards delivering the first release of a functional ACROSS Integrated prototype. The integration effort is guided by the Agile Software Development methodology, aiming to progress the development work in parallel teams and regularly integrating their output, based on a well-defined design.

The scope of this document is to act as appendix to the current version of the ACROSS integrated prototype and, as such, it is structured as follows:

- Section 2 provides an overview of the main principles of the technical design that have been applied to the development of the ACROSS solution;



- Section 3 presents the components integrated for the initial version of the ACROSS prototype;
- Section 4 concludes this report and presents the next steps for the second release of the ACROSS integrated prototype.

2 Software and System Design principles

This section contains some main principles of technical design which have been applied to the development of the ACROSS solution. Selecting design principles is critical for creating complex software structures and doing it properly at the initial stages of the project leads to better results in the long term in terms of scalability, availability, reliability, and reduced maintenance costs.

2.1 Operational Environment

In order to host and facilitate all the components and layers of the ACROSS ecosystem the operational environment was set upon a dedicated Kubernetes cluster (Version: v1.21.9). The cluster is hosted on Digital Ocean and consists of 3 Linux server nodes that act as cluster workers. Each cluster node consists of 4 VCPUs and 8 GB of RAM.

Every component is deployed, maintained, and scaled on these 3 nodes. Additionally, the Rancher framework was deployed to leverage the provision of a User Interface to administer the cluster.

State	Name	Roles	Version	External/Internal IP	OS	CPU	RAM	Pods	Age
Active	across-k8s-pool-cprhk	All	v1.21.9	178.62.237.78 / 10.110.0.2	Linux	1.8%	19%	8.2%	1.9 days
Active	across-k8s-pool-udto2	All	v1.21.9	134.209.196.252 / 10.110.0.3	Linux	6.9%	76%	26%	114 days
Active	across-k8s-pool-udtos	All	v1.21.9	134.209.204.98 / 10.110.0.4	Linux	6%	65%	24%	114 days

Through the provision of namespaces and projects (as a feature of K8S) the operational testbed is set to host different environments (development-staging-production). Each component is mapped to a dedicated deployment configuration that handles the continuous integration and deployment, scale up and monitoring of the component.



State	Name	Namespace	Image	Endpoints	Ready	Up-to-date	Available	Age	Health
Active	backend-service-deployment	backend-service	atcfogprotect/be:v4	80/HTTP	1/1	1	1	113 days	
Active	citizen-app-be	citizen-application-dev	094360380/wp5-citizen:web-application-be	443/HTTPS	1/1	1	1	76 days	
Active	citizen-app-fe	citizen-application-dev	094360380/wp5-citizen:web-application-fe	443/HTTPS	1/1	1	1	76 days	
Active	dh-be	data-harmonization-dev	094360380/wp4-data-harmonization:be	443/HTTPS	1/1	1	1	77 days	
Active	dh-fe	data-harmonization-dev	094360380/wp4-data-harmonization:fe	443/HTTPS	1/1	1	1	77 days	
Active	dh-sql	data-harmonization-dev	094360380/wp4-data-harmonization:db	31437/TCP	1/1	1	1	77 days	
Active	drawio	default	jgraph/drawio	31408/TCP 443/HTTPS	1/1	1	1	114 days	
Active	hochschulstart	hochschulstart-dev	094360380/wp4-mock-api-hochschulstart	443/HTTPS	1/1	1	1	42 days	
Active	keycloak	security-dev	atacross/security-dev:keycloak-17-postgres	443/HTTPS	1/1	1	1	91 days	
Active	keycloak-postgresql	security-dev	postgres:13		1/1	1	1	91 days	
Active	keycloak-server	security	jboss/keycloak	80/HTTP	1/1	1	1	107 days	
Active	keycloak-v2	security-dev	094360380/security-dev:keycloak18-eidas	443/HTTPS	1/1	1	1	1.9 days	

On top of that, Rancher also handles provision of external access to all these services deployed by invoking nginx ingress instances. This way all the components of ACROSS are accessible to citizens.

State	Name	Namespace	Target	Default	Age
Active	across-keycloak	security	http://across-keycloak.across.com > keycloak-server	—	107 days
Active	across-web-app-be	backend-service	http://across-web-app-be.across.com > backend-service-deployment	—	113 days
Active	across-web-app-fe	user-interface	http://across-web-app-fe.across.com > user-interface-deployment	—	113 days
Active	citizen-app-be-ingress	citizen-application-dev	https://citizen-webapp-be-citizen-application-dev.k8s.across-h2020.eu > citizen-app-be	—	76 days
Active	citizen-app-fe-ingress	citizen-application-dev	https://citizen-webapp-citizen-application-dev.k8s.across-h2020.eu > citizen-app-fe	—	76 days
Active	dh-be-ing	data-harmonization-dev	https://data-harmonization-dh-dev.k8s.across-h2020.eu > dh-be	—	77 days
Active	dh-fe-ing	data-harmonization-dev	https://data-harmonization-dh-dev.k8s.across-h2020.eu > dh-fe	—	77 days

Additionally, a Grafana and Prometheus instance is deployed to effectively monitor the performance of each component in the environment and report incidents.

Finally, combining the Gitlab CI/CD features along with the docker registry, an end-to-end Continuous Integration and Continuous Deployment mechanism is set up leading to instant automated deployments when code is pushed for each dedicated component.

2.2 System Architecture Overview

Based on the D5.1 System Architecture-initial, the ACROSS system architecture can be depicted in detail below:

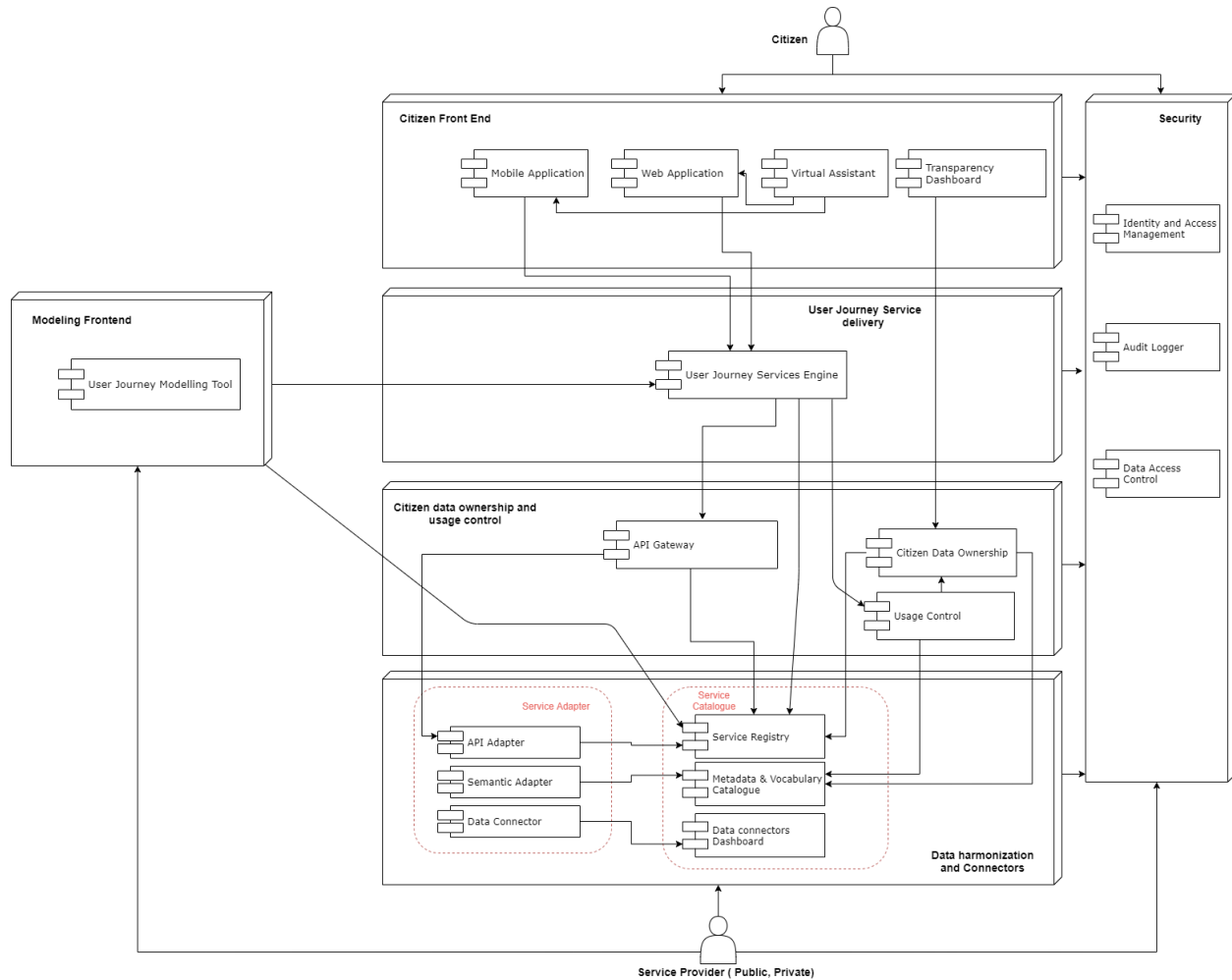


Figure 1: System architecture

From the ACROSS platform perspective, each component of these 6-layer architecture is mapped to a project containing the component deployment configuration. All the component interfaces have been set up either by the definition of internal cluster services or by the definition of external NGINX ingresses.

Regarding the Security of the ACROSS platform (vertical layer in architecture), a single common Identity management system is implemented authenticating users across all applications and leveraging functionalities such as Single Sign On principle.

2.3 Component Interaction

The ACROSS components interaction can be depicted in detail in the following interaction diagram:

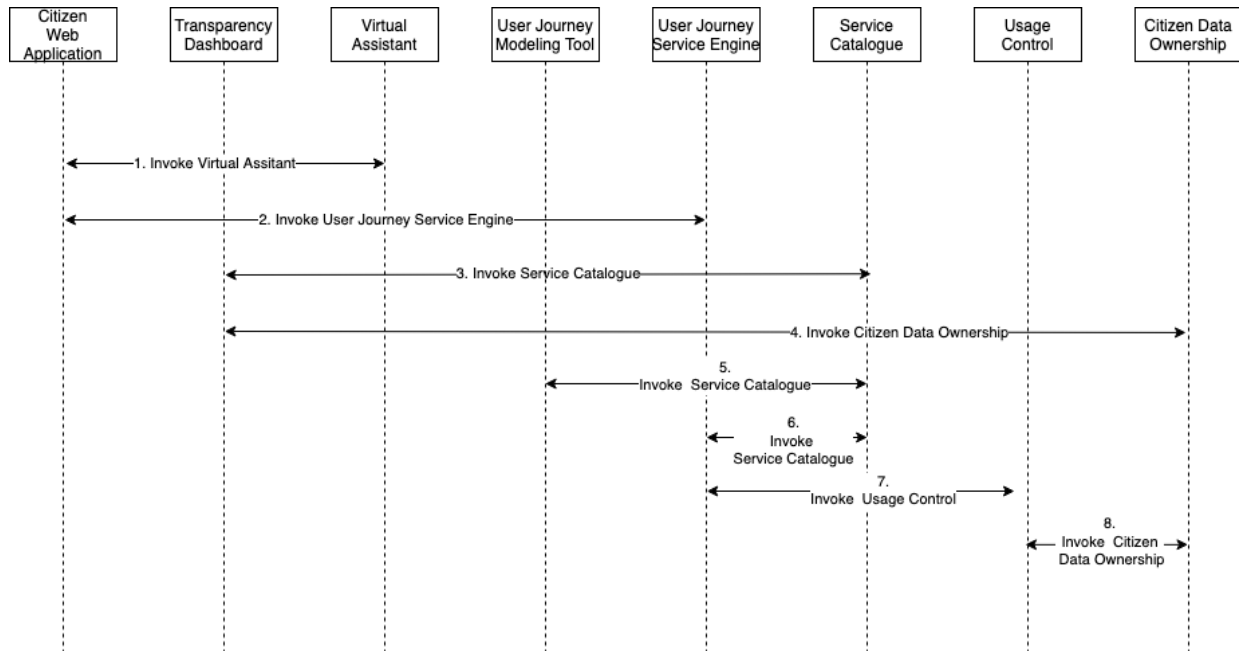


Figure 2: Components Interaction

The following table presents the interactions between the components in detail

Interface	Description
Invoke Virtual Assistant	The Citizen web/mobile application invokes the integrated Virtual Assistant service to provide conversational interfaces to features of the web/mobile application towards the citizens. Through an automated process, the citizen then can navigate and interact with the application (as well as with the virtual assistant itself) through conversational interaction i.e. (voice/chat utterances).
Invoke User Journey Service Engine	The Citizen web/mobile application invokes the User Journey Service Engine to create, edit and delete user journeys. Additionally, the citizen can directly access services, be redirected towards the Transparency dashboard to provide consent policies, and finally execute services from the citizen



	web/mobile application. Through this interface the services are provided towards the user as a one-stop-shop catalog.
Invoke Service Catalogue	The Transparency Dashboard (TD) invokes the Service Catalogue to get the list of available services which make use of personal data, so that the citizen can edit via the TD the consent he/she gives for the use of his/her personal data.
Invoke Citizen Data Ownership	The Transparency Dashboard invokes the Citizen Data Ownership to get the list of given consents by each citizen and the status of them, to grant or withdraw the citizens' consents and to receive notifications about how their data is being used.
Invoke Service Catalogue	The user journey modelling tool invokes the service catalogue to obtain the list of registered services to be used to model the workflow. The invocation can filter/search for specific services and get the needed information in accordance to the service model provided by the Service Catalogue.
Invoke Service Catalogue	The User Journey Services Engine (UJSE) invokes the Service Catalogue to get the description (required inputs, inputs considered personal data, info to invoke the service, etc.) of the services included in the corresponding workflow.
Invoke Usage Control	The UJSE invokes the Usage Control: <ul style="list-style-type: none">• to inform about the services included in the new instanced workflow and that require data consent to be executed.• to verify if the citizen has already given consent for the use of the personal data involved in a specific service execution. If consent is not given, the UJSE won't invoke the service execution• to inform that a specific personal data has been used in a service invocation.
Invoke Citizen Data Ownership	The Usage Control invokes the Citizen Data Ownership to get the consents established between the citizens and the services.



All the participating components are authenticated using Keycloak Identity management server. Through the definition of a common realm, users can be authenticated using the Single Sign On principle across all applications. Additionally, Keycloak is extended to include an eIDAS plugin to provide the option of eIDAS invocation using Keycloak as the central IMS authority.

3 Components integrated for the Initial version

3.1 Scenario implementation

Main Scenario

In the first ACROSS periodic review the proof-of-concept scenario was demonstrated. In this scenario a citizen from Latvia interacted with the ACROSS One-Stop-Shop citizen application to create a journey to Germany for the purposes of studying. The citizen is presented with a straightforward way to invoke all the different services to submit and acquire all the data needed for this journey.

User Journeys will provide to citizens all information needed for their move between countries and moreover will guide them to use relevant online services either by redirecting them to the proper websites or by offering some services directly within the ACROSS platform.

In the above-described scenario, two initial indicative services were integrated, deployed, and invoked:

- Search and Select University
- Apply at University

In order to be able to present journeys towards the citizens, the following preprocessing must take place in ACROSS.

IT Expert Interaction

Initially, by using the user interface of the User Journey Modeling Tool (UJMT), the initial workflow of these two services was designed by an IT Expert. The extensive details of the service descriptions (relevant inputs, invocation details, consent policies needed etc.) were also registered in the service catalogue (Fig 3).

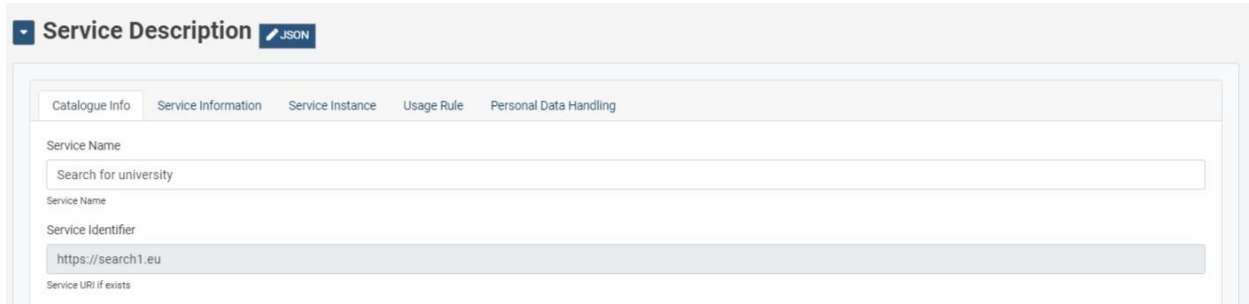


Figure 3: Service information sections included in the service catalogue

Figure 4 shows a draft of the user journey for the proof-of-concept scenario in the UJMT. The UJMT provides the IT Expert with a palette of graphical elements for the services from the Service Catalogue in the left sidebar. The services from the Service Catalogue can also be assigned to graphical elements via right-click (see Figure 5).

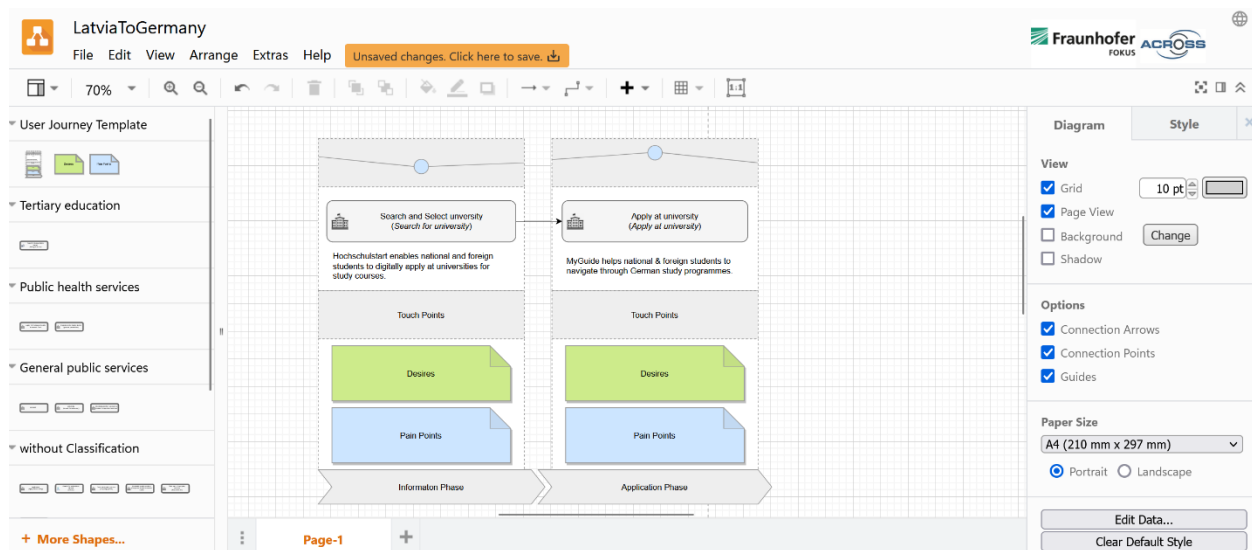


Figure 4: User Interface of the UJMT

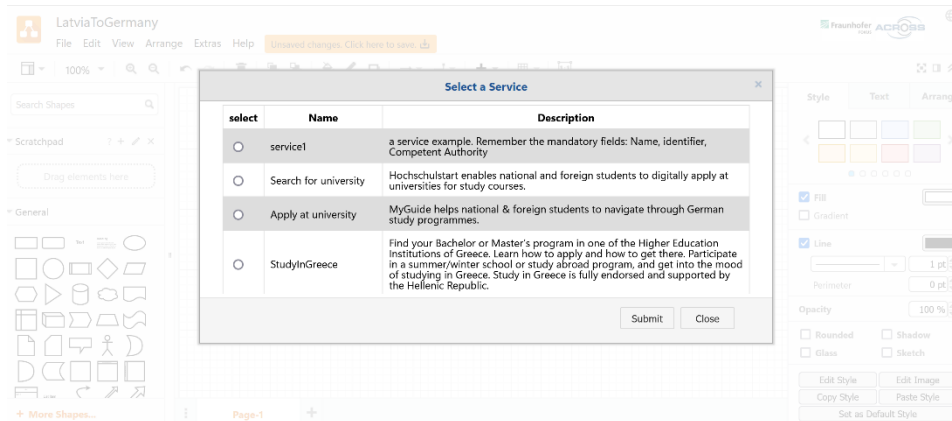


Figure 5: Service Selection for an Action in the UJMT

The BPMN model produced by the User Journey Modeling Tool (see Figure 6) was subsequently also ingested by the User Journey Service Engine. Hence, a new workflow representing a journey in ACROSS was created.

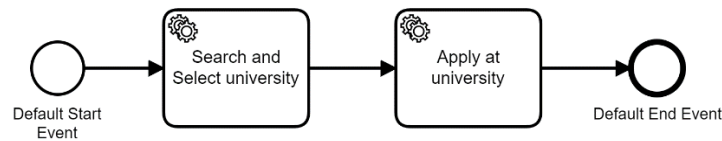


Figure 6: Generated BPMN diagram

From that point and forward, all applications of the Citizen Front End layer (Citizen web/mobile app and Transparency dashboard) have all the relevant information needed to deliver this journey towards the citizens.

Citizen Interaction

As mentioned before, in the proof-of-concept scenario an initial journey from Latvia to Germany for studying purposes was designed. The citizen can access the citizen web application and create the relevant journey.



Start planning your Journey Abroad

Origin: Country: Latvia

Purpose: study

Destination: Country: Germany

Start >

Figure 7: Latvia-to-Germany-to-study Journey creation

After the journey creation the user can start executing steps of these journey with the final goal of completing all of them. In cases where services that need to be invoked require consent policies, the citizen is redirected to the Transparency dashboard where she can manage all consent related to the journey. It is imperative for these consent policies to be granted by the user for the services to be invoked.

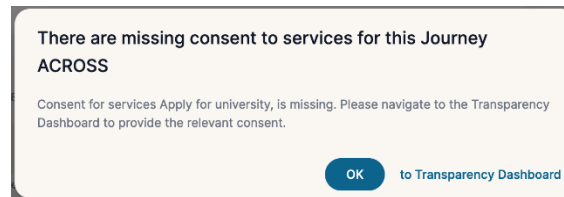


Figure 8: Transparency dashboard redirection

ACROSS

English en Log out ↗

DASHBOARDS

- Dashboard

CONSENT & LOGS

- Data & Event Logs
- Consents management

SERVICES

- Available services 1

Available services

Filters

Service	Email	Updated selected service	Details	Consent	Selected ↓
Create an application	johndoe2@example.com	July 13, 2022	i	ACTIVATED	<input checked="" type="checkbox"/>
Apply at university	johndoe2@example.com	July 15, 2022	i	ACTIVATED	<input checked="" type="checkbox"/>

Figure 9: Transparency dashboard



In the figure above, the citizen through the Transparency dashboard can grant or/and revoke the consent towards services of ACROSS. For this scenario, the “Apply at university” service requires certain consent provision from the citizen. The citizen agrees to these consent policies and subsequently proceeds to the Citizens web application to start executing the steps of the journey.

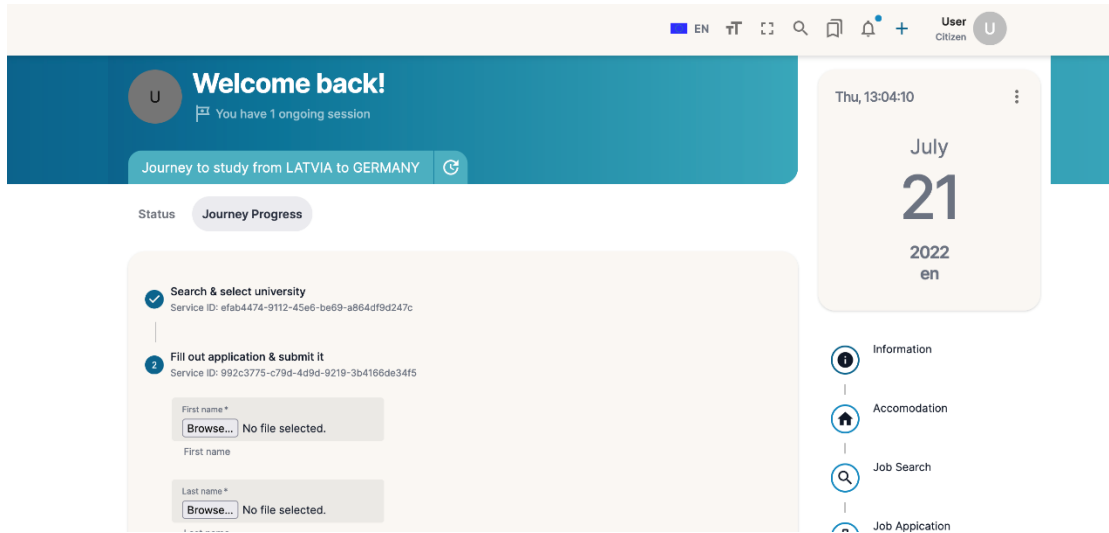


Figure 10: ACROSS Journey Step Progress

In the figure above, the citizen provides all the relevant information to the Citizens web application. After that, the citizen can invoke this service through the ACROSS application in a seamless way.

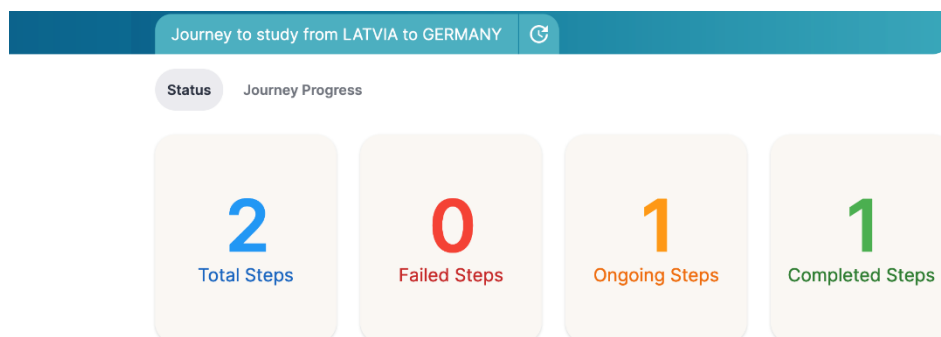


Figure 11: ACROSS Journey Status



At any moment, the citizen can overview the progress of the journey and the step status. Upon successful completion of the journey the citizen has acquired all relevant information and the journey is marked as completed.

A complete live demonstration of the citizen interaction can also be found [here](#). In the first part of the demo the citizen creates the journey and executes the steps. In the last part of the demo, the citizen's consent for these services is withdrawn so the user must access the Transparency dashboard to agree and provide the relevant consent.

Virtual Assistant

Complementary to the above scenario, an initial integration and invocation of the Virtual Assistant was implemented. The citizen was able to create the journey from Latvia to Germany for the purpose of studying. The Virtual Assistant was able to accept spoken user utterances in both English and German language and create the citizen journey through these voice commands.

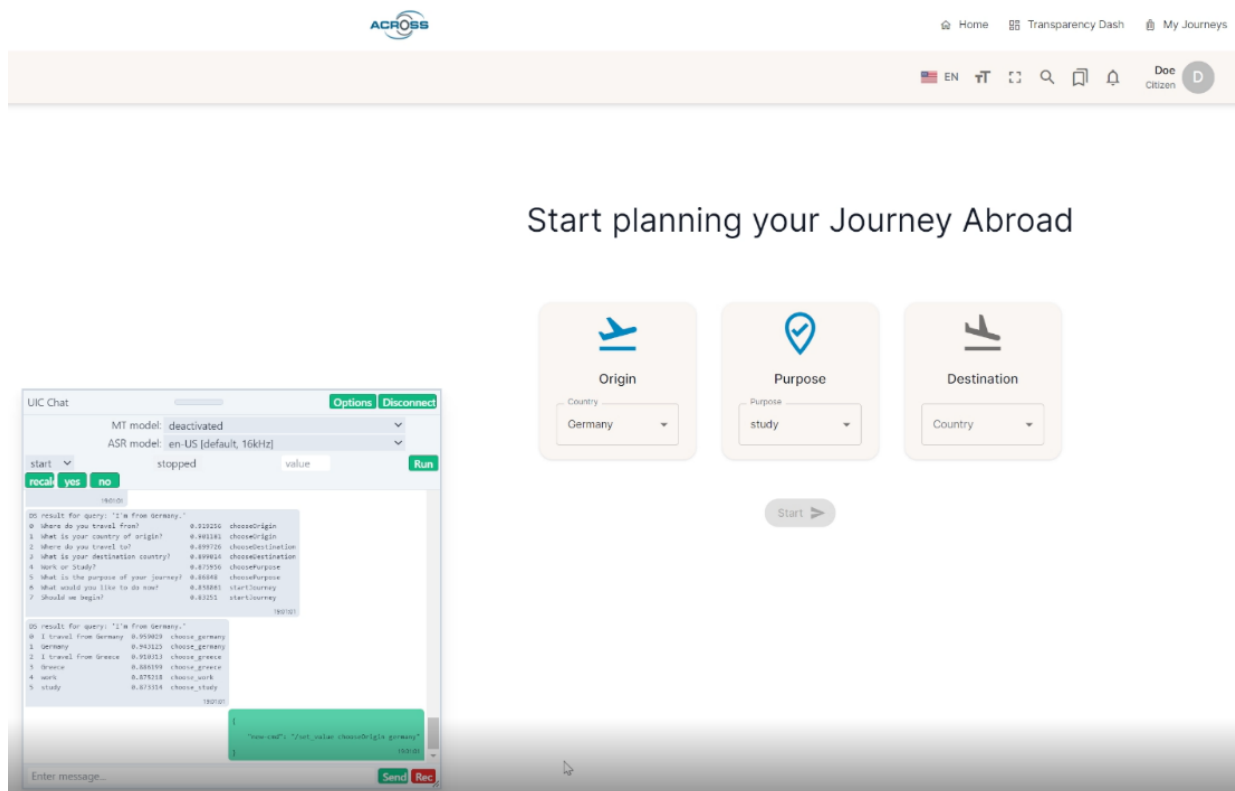


Figure 12: Virtual Assistant User Interface



A detailed demo of the Virtual assistant functionality on top of the Citizen Web User Interface can be found [here](#).

3.2 Components Integration

Component	Git Registry
Service Catalogue	Main release: https://github.com/OPSILab/Service-Catalogue/releases/tag/v1.0.0 Extensions of Service Model vocabulary for PoC scenario: https://github.com/OPSILab/Service-Catalogue/tree/v1.0.1-alpha
User Journey Modelling Tool Frontend (UJMT.F)	https://gitlab.fokus.fraunhofer.de/across/drawio
User Journey Modelling Tool Backend (UJMT.B)	https://gitlab.fokus.fraunhofer.de/across/ujmtbackend
Virtual Assistant (VA)	https://gitlab.fokus.fraunhofer.de/across/virtualassistant
Virtual Assistant UI Connector	https://git.code.tecnalia.com/across/private/citizen-front-end/web-application/web-application-UI/-/tree/virtual assistant
User Journey Service Engine (UJSE)	https://git.code.tecnalia.com/across/private/user-journey-service-delivery/userjourneyserviceengine
Transparency Dashboard	https://git.code.tecnalia.com/across/private/citizen-front-end/transparency-dashboard/transparency-dashboard-ui
Citizen Web Application Frontend	https://git.code.tecnalia.com/across/private/citizen-front-end/web-application/web-application-UI/-/tree/demo-material
Citizen Web Application Backend	https://git.code.tecnalia.com/across/private/citizen-front-end/web-application/web-application-BE



3.2.1 Service Catalogue

The following figure provides the available API used in component interactions:

Service Model		Service Model Description APIs to get and manage service model descriptions.		^
GET	/api/v2/services	Get all the Service Model descriptions.	↓	
PUT	/api/v2/services	Update Service Model description, by replacing the existing one	↓	
POST	/api/v2/services	Create a new Service Model description.	↓	
DELETE	/api/v2/services	Delete Service Model description by Service Id.	↓	
GET	/api/v2/services/specified/**	Get the Service Model descriptions by specified Service Ids.	↓	
GET	/api/v2/services/json/**	Get the Service Model description by Service Id.	↓	
GET	/api/v2/services/isPersonalDataHandling	Get the Service Model descriptions is handling personal data	↓	
GET	/api/v2/services/isPersonalDataHandling/count	Get the count of the Service Model descriptions is personal data handling.	↓	
GET	/api/v2/services/count	Get the count of the registered Service Model descriptions (total, public and private services).	↓	
GET	/api/v2/services/count/thematicArea	Get the Service Models count grouped by Thematic Area.	↓	
GET	/api/v2/services/count/sector	Get the Service Models count grouped by Sector.	↓	
GET	/api/v2/services/count/location	Get the Service Models count grouped by Spatial.	↓	
GET	/api/v2/services/count/groupedBy	Get the Service Models count grouped by GroupedBy.	↓	

Figure 13: Service Catalogue APIs

3.2.2 User Journey Service Engine

The following figure provides the available API used in component interactions. This API is divided into 2 sections:

- citizen-frontend-api-controller: it contains the API for the citizen frontend.
- modelling-tool-api-controller: it contains the API for the modelling tool.



citizen-frontend-api-controller		^
PUT	/workflowExecutionManagement/stepFinishedOffline inform a step has been executed off-line	▼ 🔒
PUT	/workflowExecutionManagement/killWorkflow kill a workflow	▼ 🔒
POST	/workflowExecutionManagement/executeStep execute step	▼ 🔒
POST	/workflowExecutionManagement/createUserWorkflowInstance create a new workflow instance	▼ 🔒
GET	/workflowExecutionManagement/getWorkflowStatus get workflow status	▼ 🔒
GET	/workflowExecutionManagement/getUserWorkflowInstances get list of workflow instances of a user	▼ 🔒
GET	/workflowExecutionManagement/getUserWorkflowInstanceById get an existing workflow instance	▼ 🔒
GET	/workflowExecutionManagement/getStepInfo get step info	▼ 🔒
modelling-tool-api-controller		^
GET	/workflowManagement getAllWorkflows	▼ 🔒
POST	/workflowManagement add/update workflow	▼ 🔒
GET	/workflowManagement/getWorkflowById/{workflowId} getWorkflowById	▼ 🔒
DELETE	/workflowManagement/{workflowId} delete a Workflow giving workflowId	▼ 🔒

Figure 14: UJSE APIs



4 Conclusions

This document is the report following the prototype deliverable D5.3 about the first release of the ACROSS platform. This is a first working version of the prototype that offers an initial set of the expected functionalities, and will act as the test-bed for the ACROSS stakeholders to experience with this initial set of ACROSS provisions and assess the concepts and knowledge conveyed by the project.

The implementation and integration are not limited to the current level. Most ACROSS modules, as well as the platform itself are designed to be flexible and easily customisable. The overall architecture of the system facilitates modifications and makes the platform improvement to be easy, at any point. This will be the aim of the following period: to continue the enhancement of this platform capabilities, make it more stable, more sophisticated and more efficient for the target end users, by plugging additional functionalities and calibrating existing ones. Specifically, the next steps of the implementation consists of the end-to-end integration of participating components through the ACROSS CI/CD pipelines and the application of the ACROSS components to further develop the use cases under scope. Namely, the current work in progress focuses on the integration of Virtual Assistant and the eIDAS integration as an SSO principle.

This version of the prototype will be enriched with more use cases to support the three pilots so that the planned first evaluation iteration will be performed in the next period. The results of this evaluation cycle, along with the already scheduled work in this and the other technical WPs, will lead to another prototype release of the ACROSS integrated platform by M27 (April 2023).

To sum up, this version of the ACROSS platform prototype is going to be updated, according to the feedback that will be received during the evaluation period, as well as the ongoing work in the technical WPs. An updated official release is expected by M27 of the project (April 2023).



5 References

- [1] ACROSS Consortium, “D5.1: System Architecture and Implementation Plan – Initial”, November 2021
- [2] ACROSS Consortium, “D4.1: Components adaptation for SDG, OOP, eIDAS for National public services – Initial”, January 2022
- [3] ACROSS Consortium, “D4.4: Micro Proxies and services catalogue – Initial”, January 2022
- [4] ACROSS Consortium, “D4.7: User Support Tools– Initial”, January 2022