**H2020-SC6-GOVERNANCE-2018-2019-2020**

**DT-GOVERNANCE-05-2018-2019-2020**

Towards user journeys
for the delivery of cross-border services
ensuring data sovereignty

# D4.7 User Support Tools – Initial

| | |
|---|---|
| **Project Reference No** | 959157 — ACROSS — H2020-SC6-GOVERNANCE-2018-2019-2020 |
| **Deliverable** | D4.7 User Support Tools – Initial |
| **Work package** | WP4 - ACROSS Modules Set-Up |
| **Nature** | Other |
| **Dissemination Level** | Public |
| **Date** | 10.02.2022 |
| **Status** | Final V1.0 |
| **Editor(s)** | Thilo Ernst (FRH) |
| **Contributor(s)** | Anna Opaska (FRH), Mustafa Can (FRH), Ekkart Kleinod (FRH),<br>Anestis Sidiropoulos (ATC), Enrique Areizaga (TEC), Vincenzo Savarino (ENG) |
| **Reviewer(s)** | Vincenzo Savarino (ENG), Enrique Areizaga (TEC) |
| **Document description** | This report focuses on the ACROSS User Support Tools *User Journey Modelling Tool (UJMT)* and *Virtual Assistant (VA)*. It documents the requirements, design, description of modules, and description of APIs applying to these components. It also describes the respective initial PoCs. Finally, it contains a description of the relevant baseline technologies being used for the implementation of the user support tools. |

## 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. The project kicked off its activities in February 2021, with an energising online meeting, where all partners took the floor to present their plans to make the project a great success.

**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.

## Document Revision History

| Version | Date | Modifications Introduced | |
|---|---|---|---|
| | | Modification Reason | Modified by |
| V0.1 | 12/11/2021 | Table of contents | FRH |
| V0.2 | 10/01/2022 | First Draft | FRH |
| V0.3 | 21/01/2022 | Second Draft | FRH |
| V0.9 | 08/02/2022 | Pre-Final Draft | FRH |
| V1.0 | 10/02/2022 | Final Version | FRH |

## Executive Summary

The main objective of the ACROSS project is to provide the means (tools, methods and techniques) to enable user-centric design and implementation of interoperable cross-border (digital) public services compliant with the current European regulations where the private sector can also interconnect their services while ensuring the data sovereignty of the citizens, who can set the privacy level that will allow the public and private sector to access to their data based on their requirements.

This deliverable “D4.7 User Support Tools – Initial” documents the design and essential implementation aspects of two software components developed by Fraunhofer FOKUS(FRH) within Work Package 4 “ACROSS Modules Set-up” and concretely in Task T4.3 “User support tools implementation” of the ACROSS project: The User Journey Modelling Tool, and the Virtual Assistant, including the corresponding first Proof-of-Concept Prototypes (*PoC’*s).

The focus of the User Journey Modelling Tool (UJMT) is on *modellers* - expert users who think about and design *user journeys*, i.e., among other aspects, plan in which combination and ordering end-users (citizens) will need to access the services offered through the ACROSS platform in a specific situation (or use-case). Modellers have expert knowledge about the available services but they are not expected to have in-depth IT expertise. Using the UJMT, a modeller will be able to interactively create graphical models of the “service workflows” the end-users will need to go through in each user journey. Each modelled user journey-specific service workflow is converted into a suitable machine-readable presentation and is then handed over to the User Journey Service Engine which, in collaboration with other ACROSS subsystems, enables the orchestration and actual execution of the specified service workflow. Eventually, the workflow can be selected, started and used through the ACROSS Web/Mobile App (and the Virtual Assistant, see below) by the end-user citizen, exactly according to the User Journey model created at the beginning. The UJMT is created based on a carefully selected and powerful open-source package which is substantially modified and extended, and then integrated into the service-oriented ACROSS architecture.

The Virtual Assistant’s focus, on the other hand, is on the *end user/citizen* - its purpose is to enable superior ease of use and accessibility (in particular, for users with disabilities) for the user-exposed parts of the ACROSS system - concretely, the ACROSS Web and Mobile App. This goal is pursued by offering dedicated conversational interfaces (i.e. multi-lingual textual chat-bot and speech interfaces) for controlling user interface features of the ACROSS Web and Mobile App by natural language. The Virtual Assistant is implemented by building upon modern web standards and open-source software and integrating it both with the other ACROSS subsystems and with background technology from Fraunhofer FOKUS using an innovative, minimally invasive integration approach, utilizing a state-of the art service-based architecture.

## Table of Contents

## List of Figures

## List of Tables

## List of Terms and Abbreviations

| Abbreviation | Definition |
|---|---|
| BPMN | Business Process Modelling and Notation |
| CPSV | Core Public Service Vocabulary |
| CPSV-AP | Core Public Service Vocabulary Application Profile |
| DoA | Description of the Action |
| EIF | European Interoperability Framework |
| FDM | FISA Dialog Manager |
| FISA | Fraunhofer FOKUS Intelligent Speech Assistant |
| ICT | Information and Communications Technologies |
| KPI | Key Performance Indicator |
| MO | Measurable Outcome |
| OOP | Once-only principle |
| PA | Public Administration |
| PoC | Proof of Concept |
| SDG | Single Digital Gateway |
| UIA | User Interface Adapter |
| UIC | User Interface Connector |
| UJM | User Journey Model |
| UJMT | User Journey Modelling Tool |
| UJSE | User Journey Service Engine |
| UJWD | User Journey Workflow Description |

# 1 User Support Tools - Introduction and Overview

## 1.1 Common ACROSS Context

The ACROSS project envisions two User Support Tools, the User Journey Modelling Tool (UJMT) and the Virtual Assistant (VA). Both are motivated from main objectives / Measurable Outcomes (MOs) as defined within the Description of the Action (DoA) of the ACROSS project, and their detailed success criteria (Key Performance Indicators (KPIs)):

- MO1.1: A User journey methodology, approach **and supporting tool** to define and model user-centric digital public services, complemented with an example, namely, the application to the cross-border mobility life event. […] KPI1.1.2: An **online tool for modelling user journeys** to define the interaction of the user with the digital public services and their orchestration.

- MO1.3: **Multi-lingual Virtual Assistant providing speech and textual chat interfaces** guiding the user in the user journey of the service. The assistant will provide superior accessibility (in particular, for users with disabilities), by offering multi-lingual textual chat-bot and speech interfaces to be integrated with the ACROSS Platform. Success criteria: KPI1.3.1: 100% of the requirements and functionalities specified are implemented. KPI1.3.2: It is able to support in all the steps of the user journey.

These and the other MOs have their roots in European legislations and initiatives such as Single Digital Gateway (SDG), Once-Only principle (OOP) and European Interoperability Framework (EIF). MO1.3 also is strongly motivated by the European Accessibility Act, which has the primary goal of benefitting persons with disabilities and elderly people by providing more accessible products and services in the European market.

## 1.2 Methodology and Structure of the Deliverable

As the two User Support Tools have very different purposes and roles within ACROSS, and collaborations to other ACROSS components, they will be described independently, each within its own main chapter. However, a common description format will be used, covering

- the Objectives pursued by each tool together with the scope in which the tool is relevant,
- the context motivating the creation and influencing the design and implementation of each tool (both on European level and specifically within ACROSS),
- the specific requirements that have been gathered for the tool and that steer its design and implementation

- the design of the tool (including architecture, sub-modules, interfaces, and collaborations)
- a description of the Proof of Concept (PoC) delivered at the first milestone
- and a description of the baseline technologies used in the implementation of the tool.

# 2 User Journey Modelling Tool (UJMT)

## 2.1 UJMT - Objectives and Scope

The User Journey Modelling Tool (UJMT) is an online supporting tool for defining and modelling user-centric digital public services. The main objectives of the UJMT are:

- Provision of the functionalities that are necessary for modelling user interactions with digital public services as user journeys
- Provision of corresponding machine-readable descriptions for the service orchestration to the User Journey Service Engine (UJSE)
- Integration of available public administration (PA) and third-party private service information from the Service Catalogue

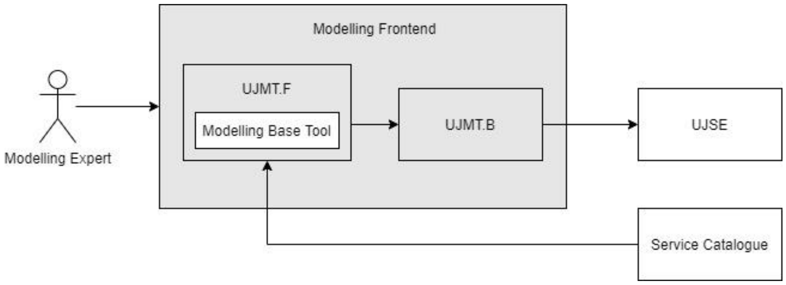


**Figure 1: UJMT Overview**

Figure 1: UJMT Overview shows that the UJMT consists of two main components: The frontend UJMT.F is based on an open-source modelling base tool and allows the Modelling Expert to interactively create User Journey Models (UJMs) for ACROSS. The backend UJMT.B creates from the UJM a User Journey Workflow Description (UJWD)[1], a machine-readable concrete orchestration description based on the modelled workflow. The two UJMT Modules are described in more detail in section 2.4.3.

[1] In the component card for the UJMT [12] we referred to the orchestration description as User Journey Workflow Template (UJWT). This term had been introduced to describe a more abstract orchestration description that meanwhile (i.e. after continued design discussions between the ACROSS technical partners) has been discarded.

For the PoC, a part of the scenario “Study Abroad in Germany” will be modelled. Figure 2 shows an abstract workflow for that scenario provided by Dataport. The colouring of the workflow steps represents the expected realization of the involved services from the citizens’ perspective (red = only information provided; yellow = link to external resources; green = planned to integrate).

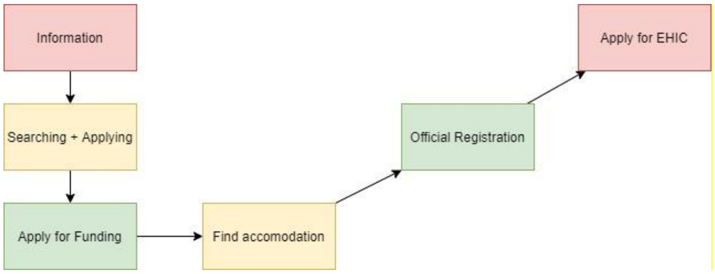
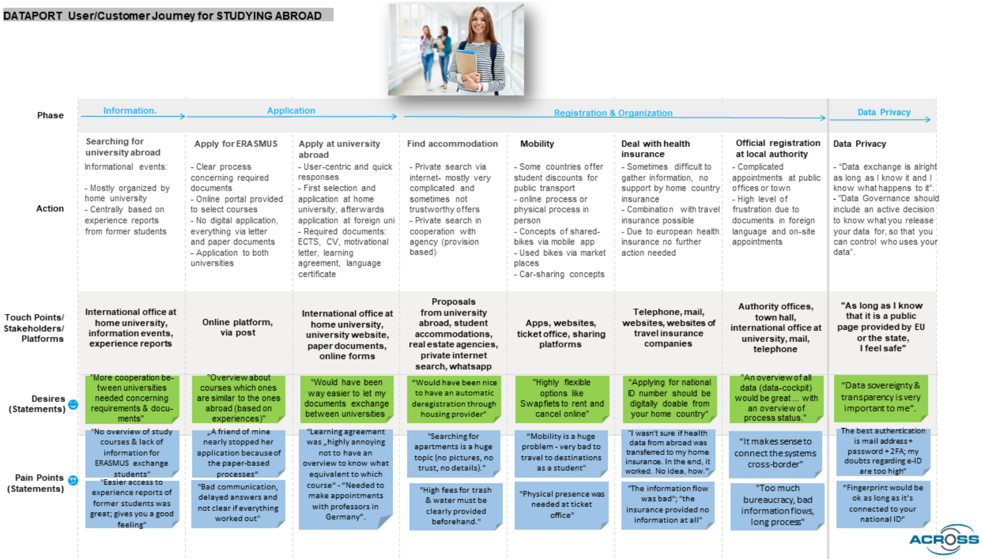


Figure 2: Abstract Workflow for the scenario “Study Abroad” provided by Dataport

The UJMT aims to support mapping and specification of abstract workflows. Based on the current state of considerations within the use cases work packages, we have decided to use, for this initial iteration of the UJMT implementation, a simple, sequential representation, covering the individual workflow steps as successive user journey phases. A similar mapping is visible in the associated example user journey by Dataport (s. Figure 3) where the workflow steps are partially mapped to phases and/or actions.



DATAPORT User/Customer Journey for STUDYING ABROAD

| Phase | Information. | Application | | Registration & Organization | | | | Data Privacy |
|---|---|---|---|---|---|---|---|---|
| Action | **Searching for university abroad**<br>Informational events:<br>- Mostly organized by home university<br>- Centrally based on experience reports from former students | **Apply for ERASMUS**<br>- Clear process concerning required documents<br>- Online portal provided to select courses<br>- No digital application, everything via letter and paper documents<br>- Application to both universities | **Apply at university abroad**<br>- User-centric and quick responses<br>- First selection and application at home university, afterwards application at foreign uni<br>- Required documents: ECTS, CV, motivational letter, learning agreement, language certificate | **Find accommodation**<br>- Private search via internet- mostly very complicated and sometimes not trustworthy offers<br>- Private search in cooperation with agency (provision based) | **Mobility**<br>- Some countries offer student discounts for public transport<br>- online process or physical process in person<br>- Concepts of shared-bikes via mobile app<br>- Used bikes via market places<br>- Car-sharing concepts | **Deal with health insurance**<br>- Sometimes difficult to gather information, no support by home country insurance<br>- Combination with travel insurance possible<br>- Due to european health insurance no further action needed | **Official registration at local authority**<br>- Complicated appointments at public offices or town<br>- High level of frustration due to documents in foreign language and on-site appointments | **Data Privacy**<br>- “Data exchange is alright as long as I know it and I know what happens to it”.<br>- "Data Governance should include an active decision to know what you release your data for, so that you can control who uses your data”. |
| Touch Points/ Stakeholders/ Platforms | International office at home university, information events, experience reports | Online platform, via post | International office at home university, university website, paper documents, online forms | Proposals from university abroad, student accommodations, real estate agencies, private internet search, whatsapp | Apps, websites, ticket office, sharing platforms | Telephone, mail, websites, websites of travel insurance companies | Authority offices, town hall, international office at university, mail, telephone | "As long as I know that it is a public page provided by EU or the state, I feel safe" |
| Desires (Statements) 😊 | "More cooperation between universities needed concerning requirements & documents" | "Overview about courses which ones are similar to the ones abroad (based on experiences)" | “Would have been way easier to let my documents exchange between universities | “Would have been nice to have an automatic deregistration through housing provider” | "Highly flexible options like Swapfiets to rent and cancel online" | "Applying for national ID number should be digitally doable from your home country" | "An overview of all data (data-cockpit) would be great … with an overview of process status." | “Data sovereignty & transparency is very important to me”. |
| Pain Points (Statements) 😟 | "No overview of study courses & lack of information for ERASMUS exchange students"<br>"Easier access to experience reports of former students was great; gives you a good feeling" | „A friend of mine nearly stopped her application because of the paper-based processes"<br>"Bad communication, delayed answers and not clear if everything worked out" | "Learning agreement was „highly annoying not to have an overview to know what equivalent to which course" - “Needed to make appointments with professors in Germany”. | "Searching for apartments is a huge topic (no pictures, no trust, no details)."<br>"High fees for trash & water must be clearly provided beforehand." | “Mobility is a huge problem - very bad to travel to destinations as a student"<br>"Physical presence was needed at ticket office" | "I wasn't sure if health data from abroad was transferred to my home insurance. In the end, it worked. No idea, how.”<br>"The information flow was bad"; "the insurance provided no information at all" | “It makes sense to connect the systems cross-border”<br>"Too much bureaucracy, bad information flows, long process" | The best authentication is mail address + password + 2FA; my doubts regarding e-ID are too high”<br>“Fingerprint would be ok as long as it’s connected to your national ID” |

Figure 3: Example User Journey for the scenario "Study Abroad" provided by Dataport

Following the example user journey, the UJMT allows the definition of phases and actions, and the assignment of several actions to each phase. We transferred the proposed colouring concept to the individual actions in order to allow an initial classification at this level. The UJMT furthermore supports an initial form of the concretization of each action, by allowing a workflow at operational level to be defined on a linked page.

## 2.2 UJMT - ACROSS Context

This section describes the context that has influenced the definition of the UJMT, both on the European level and within the ACROSS project.

### 2.2.1 Relevant European initiatives and legislation

A considerable number of European projects have been bringing forward innovation in the field of business process modelling and workflow management within different application domains over the last decade. Examples include PBM4PEOPLE [1] [2], Productive4.0 [3], HORSE [4], and COMPOSITION [5]. The intermediate level of abstraction exemplified in business process/workflow formalisms such as Business Process Modelling and Notation (BPMN) has proven very successful both as regards the modelling side, where process experts can express their designs without being distracted by overwhelming details of the individual steps, and as sound foundation for process/workflow execution and orchestration.

In recent years, new agile, user-centric design methodologies such as User Journey Mapping or Customer Journey Mapping have emerged and are becoming increasingly popular in product and service design on an international scale and across multiple industries. On the European level, this trend is exemplified by projects like PASSME [6] [7], which pushed for optimization of the air travel experience based upon the notion of the "passenger journey", or GRAVITATE-HEALTH [8], which focuses on optimizing "patients' journeys" in the digital health information domain. The ACROSS project, too, is built upon the premise that User Journeys provide a superior framework for collaboratively thinking about and designing the interaction of users (in the chosen application domain of ACROSS: citizens) with a complex Information and Communications Technologies (ICT) system.

User Journeys, however, are inherently more abstract and loosely-defined when compared to business process or workflow models, even though they clearly contain workflow aspects. The development of the ACROSS User Journey Modelling Tool strives to bridge this gap. The UJMT's goal is to empower User Journey Modelling experts to lay down and refine the workflow aspects of the User Journey Maps they create in the form of a reusable, digital *abstract workflow* model.  After a subsequent concretization step, this model can directly be used to orchestrate and execute workflows within the ACROSS platform. The individual steps of these *concrete workflows* will be the elementary services that the ACROSS platform is built to make available to its users - the citizens.

These elementary Public Services are another aspect where existing European initiatives are very relevant for the development of the UJMT. In particular, several present and future results of the ISA² programme [9] of the EU will serve as the basis for empowering the modelling experts to discover, select and integrate concrete Public Services into their User Journey Models, both on abstract and concrete workflow level. Even though details remain to be worked out, several parts of the ISA² service model are clearly relevant her, most importantly, the Core Public Service Vocabulary (CPSV), the Core Public Service Vocabulary Application Profile (CPSV-AP) [10], and the European taxonomy for public services [11].

### 2.2.2 Approach and Relation to other Work Packages and Deliverables

The R&D work being done on the UJMT is closely related to several other Work Packages of ACROSS and their respective Deliverables. First and foremost, the primary motivator and inspiration of the R&D activities is the work being done in WP2 "ACROSS new Governance Model" T2.1 "User Journey Methodology definition" and WP6 "Use cases deployment, evaluation & impact assessment" T6.1 "Use cases definition and planning". From both these Work Packages and Tasks, important requirements to the UJMT are gathered, while the latter also provides insights about the nature and details of the concrete services (to be accessed within the use cases), which the UJMT will help orchestrate and make available through the ACROSS platform. Of course, with the UJMT to work in close coupling to several other parts of the ACROSS framework, there is also a close relationship with WP5 "Platform Integration & Mobility Applications" and all its tasks. For the same reason, there are close ties with the other Tasks within WP4 "ACROSS Modules Set-Up").

With respect to intra-project collaboration and synergy, the approach taken for implementing the UJMT R&D activities thus relies on two pillars:

a) close collaboration and communication with the responsible WP2 and WP6 partners, for gathering requirements to the UJMT
b) close collaboration with the WP5 partners, especially regarding T5.1 System Architecture, for achieving well-founded and strong integration with the relevant other parts of the ACROSS platform (most importantly, the User Journey Service Engine and the Service Catalogue)

Apart from that, the methodology employed in the R&D work on the UJMT emphasizes continuous orientation on the current SOTA both in industry and academic research, a strong reliance on mature open-source software, and consistent use of agile development practices in a form tailored to the environment of an applied research organization.

## 2.3 UJMT – Requirements

The User Journey Modelling Tool covers some general ACROSS requirements and fulfils special requirements for the UJMT component. Those requirements are listed in this section.

### 2.3.1 Requirements from WP5 and General ACROSS requirements

WP5 has gathered a first set of requirements for the whole ACROSS platform and ICT modules [12]. Some of these requirements are directly related to the UJMT while others are generic IT requirements or non-functional requirements. The full list of requirements gathered in WP5 is included in Annex I – WP5 Requirements. Those relevant to the UJMT are also shown in this subsection. Req_10 is the foundational requirement from WP5 applying to the UJMT, whereas the remaining requirements shown below are general requirements to the components implemented in ACROSS, which also apply to the VA (to an appropriate extent).

**Table 1 Requirements to the UJMT from WP5**

| Id | Title | Description | Type | Category |
|---|---|---|---|---|
| **Req_09** | Free access to other countries' e-services | Citizens should be able to access other countries' services | non functional | Platform architecture and interoperability |
| **Req_10** | User Journey Experience | Citizen should be able to navigate in a straightforward clearly defined way through the whole process of the User Journey provided user experience. I would also like to have support during the steps of the moving abroad process. | non functional | Platform architecture and interoperability |
| **Req_15** | Easy to use service integration and orchestration tools | In order to create cross border services, the platform has to support Public and Private providing a set of tools and applications that will help them to easily implement service integration. | functional | Connectors to integrate the private and public sector offering |

| Id | Title | Description | Type | Category |
|---|---|---|---|---|
| **Req_19** | Reliability and Integrity | The implementation of ACROSS should follow open standards and use well-known and widely accepted technologies in order to ensure integrity. The ACROSS platform has to be reliable assuring integrity of the components/tools that are part of it. | non functional | Platform architecture and interoperability |
| **Req_29** | No vendor lock-in | I want the ACROSS reference architecture to be technologically agnostic to avoid vendor lock-in. | non functional | Platform architecture and interoperability |
| **Req_30** | Open source | I want the ACROSS reference architecture to reuse already available open source solutions and only create or improve those aspects that are not covered by the existing solutions | non functional | Platform architecture and interoperability |

### 2.3.2 UJMT – Requirements from WP2 (ACROSS New Governance Model) and WP6 (Use Cases)

In addition to the general requirements, the following requirements have been agreed upon in discussion with the WP2 partners. Due to the ongoing work, this list is incomplete and will be supplemented:

REQ-UJMT-1 As a Modelling Expert in ACROSS I want to be able to work on models in a graphical, interactive manner, on an appropriate level of abstraction.

REQ-UJMT-2 As a Modelling Expert in ACROSS I want to be able to add comments to modelling elements.

### 2.3.3 UJMT – Specific Requirements

#### *2.3.3.1 User Journey Modelling Requirements*

REQ-UJMT-3 As a Modelling Expert in ACROSS I want to be able to easily create UJMs for different cross-border scenarios via graphical interactions such as drag and drop.

REQ-UJMT-4 As a Modelling Expert in ACROSS I want to be able to store the created UJMs.

REQ-UJMT-5 As a Modelling Expert in ACROSS I want to be able to load and edit previously stored UJMs.

REQ-UJMT-6 As a Modelling Expert in ACROSS I want to be able to export the created UJMs as PDF.

#### 2.3.3.2 Modelling of Abstract Workflows

REQ-UJMT-7 As a Modelling Expert in ACROSS I want to be able to create abstract workflow steps as part of the UJM and combine them into abstract workflows.

#### 2.3.3.3 Transition to Concrete Workflows for Orchestration

REQ-UJMT-8 As a Modelling Expert in ACROSS I want to be able to technically refine abstract workflows into concrete workflows in order to prescribe the service orchestration for the UJME.

REQ-UJMT-9 As a Modelling Expert in ACROSS I want to be able to map abstract workflow steps to concrete services from the Service Catalogue.

REQ-UJMT-10 As a Modelling Expert in ACROSS I want to be able to map abstract workflow steps to concrete sub-workflows that contain several concrete services from the Service Catalogue.

REQ-UJMT-11 As a Modelling Expert in ACROSS I want to be able to provide all the necessary data for the service orchestration in the UJM.

#### 2.3.3.4 Providing Building Blocks / Templates

REQ-UJMT-12 As a Modelling Expert in ACROSS I want to be able to use pre-made templates that support the user journey modelling.

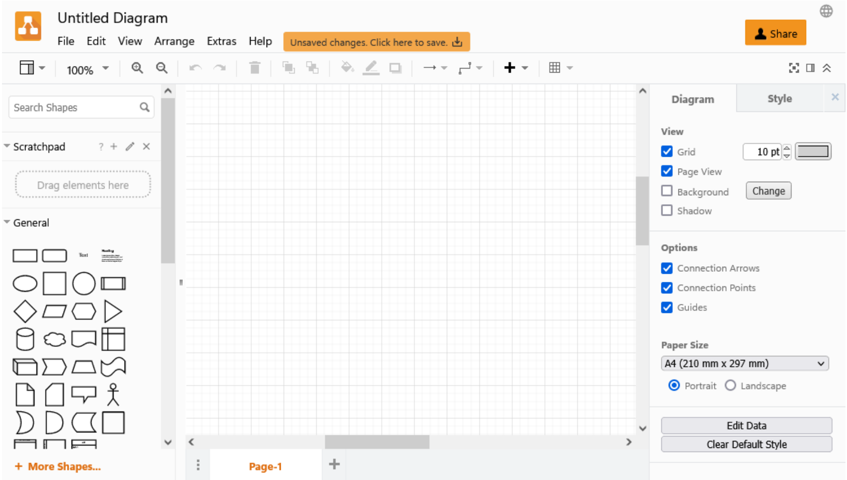
REQ-UJMT-13 As a Modelling Expert in ACROSS I want to be able to add my own templates for further use.

## 2.4 UJMT - Initial design

This section describes the design, architecture and interfaces for the initial implementation of the User Journey Modelling Tool.

### 2.4.1 UJMT - Base Tool Selection

A comparison of several open-source modelling tools was carried out for the selection of the UJMT.F Base Tool. The evaluation showed that Draw.io (see Figure 4: Draw.io User Interface) – despite its complex code base - is the most suitable among the candidates.



View

☑ Grid 10 pt
☑ Page View
☐ Background Change
☐ Shadow

Options

☑ Connection Arrows
☑ Connection Points
☑ Guides

Paper Size

A4 (210 mm x 297 mm)
● Portrait ○ Landscape

Figure 4: Draw.io User Interface

### 2.4.1.1 Candidates

We conducted research and identified the following open-source modelling tools as potential base tools for the UJMT:

- **Draw.io** [12] is a very comprehensive application for creating various graphic models which is owned and developed by JGraph Ltd. The code is licensed under the Apache v2. Draw.io is available as a desktop and online app.
- **Excalidraw** [13] is a sketching application that allows collaborative modelling and provides end-to-end encryption.
- **Camunda Modeler** [14] is a modelling tool for Business Process Model and Notation (BPMN) and Decision Model and Notation (DMN), and a feature of the Camunda Business Process Management platform.
- **Bonita Studio** [15] is a modelling application based on Eclipse. It is part of the Bonita Business Process Management platform.

### 2.4.1.2 Evaluation Criteria

In the following we describe the evaluation criteria of our selection process.

- **Open-source**: This criterion is assumed by the WP5 requirement Req_30 (open-source) which states that the reference architecture must reuse available open-source solutions.
- **Aesthetic presentation quality (for export)**: This criterion is based on the experience that user journey models are often used as a “sales tool” in presentations. We want to support this general use case.

- **Appropriate usability**: This criterion is partially assumed by the WP5 requirement Req_15 (easy to use service integration and orchestration tools) which states that the supporting tools must help the user to easily implement service integration.
- **Modelling Features**:
    - **Extendable templates**: This criterion is assumed by the UJM requirements REQ-UJMT-12 and REQ-UJMT-13 which state that the Modelling Expert should be able to add and use pre-mad user journey templates.
    - **Ideally, template / building blocks for UJM already available**: This criterion expands on the previous one.
    - **Annotations / links between model entities possible**: This criterion describes a necessity resulting from UJM requirement … which states that workflows must be extracted from the user journey models.
    - **Hierarchical modelling**: This criterion describes a necessity resulting from UJM requirement REQ-UJMT-8 which states that the user must be able to specify abstract workflows.
    - **Multi-document support**: This criterion describes a possible implementation of the previous criterion.
    - **Collaborative editing**: This criterion is assumed by the requirement REQ-UJMT-1 which states that Modelling Experts should be able to use the tool collaboratively.
- **Existing Code Base**:
    - **Size**: If the required functionalities are implemented, we prefer a code base with a smaller scope.
    - **Documentation**: We prefer a well-documented code base.
    - **API available/documented**: We prefer a base tool that can be operated via documented interfaces.
    - **Programming Language**: We prefer a code base with widely used programming languages.
- **Project Community**:
    - **Currently ongoing development**: We prefer a code base that is regularly maintained and has an active community.
    - **Maturity level (development time)**: We prefer a code base with a higher level of maturity to ensure the highest possible reliability and error tolerance.
- **Output Data formats**: In order to be able to implement the translation of the user journey model into the according orchestration description, we prefer a base tool that supports processable

output data formats like YAML, JSON and/or XML. The requirement REQ-UJMT-6 also specifies PDF as output format.

#### 2.4.1.3 Tool Selection

Since Draw.io is the only tool that contains all the necessary functionalities - in particular multi-document support for hierarchical modelling and sufficient modelling elements to fully support the modelling of user journeys, it best meets the evaluation criteria. We therefore plan on using Draw.io as the base tool for the UJMT, despite its extensive code base, and pursue the goal of implementing as much as possible of the UJMT as a plugin.

### 2.4.2 UJMT - Architecture

The following section provides an overview of the main modules of the UJMT.

### 2.4.3 UJMT - Modules

#### 2.4.3.1 UJMT Front-End

The UJMT.F is the graphical front-end tool for interactively, in a diagram-like style, modelling User Journey Models for ACROSS. The modelling of central workflow related aspects will directly affect the intended service orchestration. The UJMT.F will be implemented as an extension of the open-source project Draw.io. It must be able to export a file-based model representation in an open and documented structured data format (e.g., based on XML or JSON), and it needs to be reliably deployed externally (i.e., out of the scope of the ACROSS platform). A web user interface must enable the Modelling Experts in ACROSS to interactively construct and modify User Journey Models. The UJMT should be configurable and allow e.g., the selection of suitable templates for a particular use-case domain (may be file-based or service-based). It should be able to incorporate service information from the Service Catalogue.

#### 2.4.3.2 UJMT Back-end

The UJMT.B is the back-end tool that provides support for implementing the intended service orchestration, based on the objects and relationships in the User Journey Model built in the first step. From the model, the UJMT.B will create a machine-readable concrete orchestration description based on the modelled workflow (referred to as a User Journey Workflow Description, or UJWD). This orchestration description will be provided to the UJSE, where it is used to orchestrate and execute the User Journey Service Workflows.

### 2.4.4 UJMT - Interfaces and Collaborations

#### *2.4.4.1 User Journey Service Engine*

The UJMT is able to provide the created UJWDs to the UJSE, as well as update or delete previously provided UJWDs. In the UJSE, the UJWDs are used for the instantiation of the concrete User Journey Service Workflows. In order to communicate with the UJSE, an interface (e.g. REST) must be implemented.

#### *2.4.4.2 Service Catalogue*

The UJMT has an interface to the Service Catalogue in order to receive information about already registered services. In the UJMT, the Modelling Expert should be able to use the information for the creation of concrete workflows.

## 2.5 UJMT - Description of Initial PoC

As part of the PoC for the first milestone we will implement a dialog that simplifies the start of modelling. The user can first define phases and actions in the dialog and then generate according pre-defined graphical elements from the input. This allows the user to start modelling with a prepared set of modelling elements which can be arranged and completed with additional elements from the toolbar. The generated elements also contain necessary information for the orchestration.

The user can call the 'Create new User-Journey' dialog from the menu. Figure 5 shows an initial wireframe for the proposed dialog. In the dialog the user can submit phases and related actions of the UJ and provide additional information. From the input data, the modelling tool creates a structured set of modelling elements (s. Figure **6**) as a first draft of the UJ. From here, the user can manually assemble the modelling elements, add further elements from the toolbar and technically refine individual actions in linked pages.

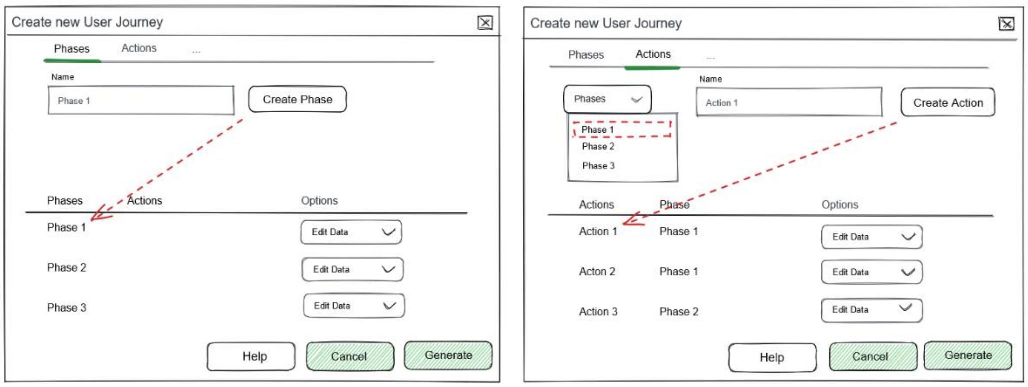


**Figure 5: Initial UJMT Dialog wireframe**

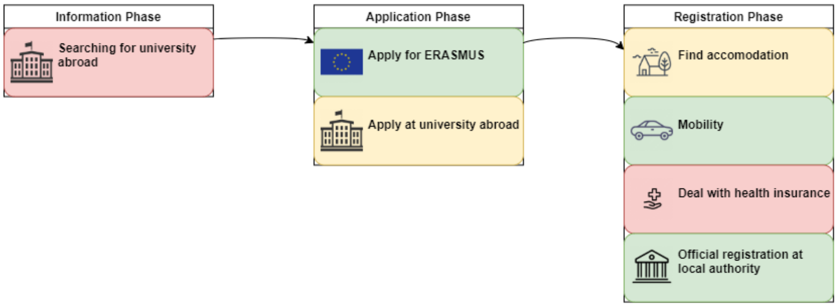

Information Phase
Searching for university abroad
Application Phase
Apply for ERASMUS
Apply at university abroad
Registration Phase
Find accomodation
Mobility
Deal with health insurance
Official registration at local authority

**Figure 6: Example for generated modelling elements**

## 2.6 UJMT - Baseline technologies

The UJMT uses different standard baseline technologies. The most relevant of these are listed below:

**Representational state transfer (REST):** architectural style for implementing hypermedia systems in the Web. In a narrower meaning that is used here, it provides a definition (rather guidelines) of stateless web APIs. Web APIs that conform to REST guidelines are called RESTful APIs. REST is not standardized per se, but it is well defined by papers and convention.

**JavaScript**: is a programming language, it is a core technology of the Web, just as HTML or CSS. It conforms to the ECMAScript standard but comes in different flavours by different vendors. Nonetheless, a core JavaScript language definition is supported in all flavours, thus making the use of the language safe for our needs.

**mxGraph**: is a JavaScript library for browser-based interactive graph applications. The library is a fundamental part of Draw.io.

**Business Process Modelling and Notation (BPMN)**: is a machine-readable graphical standard notation for describing business processes in a model. BPMN is maintained by the Object Management Group (OMG). The current version is BPMN 2.0 and the usual exchange format is XML.

## 2.7 UJMT - Conclusions and next steps

The next steps in the further development of the tool include:

- the conception and implementation of the technical connection to the UJSE via an available REST API

- the coordination, conception and implementation of the integration of service related information from the Service Catalogue
- the further adaptation of the UJMT to the stated requirements as well as requirements that are continuously developed in discussions with the partners

For the implementation of the interface between the UJMT and the UJSE, it must be clarified which engine is used and which information is required from the model for orchestration. In the first step, a reduced section of a scenario will be selected and implemented with the partners.

# 3 Virtual Assistant (VA)

The Virtual Assistant provides conversational interfaces (i.e. chat or speech interfaces using natural language) to the user-facing components of ACROSS, i.e. the Web and Mobile App. Thus, the ACROSS applications can be used traditionally by keyboard and mouse or touchscreen, but the VA empowers the user to control them through natural language as well.

## 3.1 VA - Objectives and Scope

The main technical objective of the Virtual Assistant is to make features of the ACROSS web and mobile apps controllable through conversational interfaces, i.e. natural language. This objective is motivated by two more general, non-technical objectives.

First**, ease of use and convenience**: Conversational control to many users seems more natural and convenient than using a classical “point and click” graphical user interface. Many users prefer to ask a chat-bot for information over having to click through a complex website for finding it. With a voice interface, there is the additional advantage of not having to use one’s hands, which, apart from mere convenience, may considerably ease uses e.g. in non-standard environments or situations.

Secondly, **accessibility**: The advantages mentioned above are even more relevant for disabled citizens, who by utilizing this additional interaction channel can work around their impairments, so the barrier to use the ACROSS apps is removed.  For instance, people with reduced vision can access the application through voice interaction.  Similarly, there are analphabetic or dyslexic people or who are unable to read or write or uncomfortable with reading and writing, to whom a speech interface would also be strongly preferable.  Or a person with a slight motor impairment that affects fine control of a pointing device while still allowing keyboard use might prefer to communicate with the app and the underlying services through a chat-bot style textual interface.

Even though this is not an objective in a technical sense, from which specific requirements could be derived, a third aspect should also be mentioned:  Conversational interfaces enjoy broad popularity nowadays, both in their textual form (e.g. as chat-bots for customer contact purposes in online shops) and as voice assistant devices (in the consumer electronics space). Given that, making existing services accessible through natural or even spoken language can be expected to render these services more innovative and attractive in general.

## 3.2 VA - ACROSS Context

This section describes the context that has influenced the definition of the VA, both on the European level and within the ACROSS project.

### 3.2.1 Relevant European initiatives and legislation

**Connecting Europe Facility – CEFDigital eTranslation service**

Within the Digital Europe programme, the Connecting Europe Facility [17]has developed a set of Digital Service Infrastructures Building Blocks that can be reused in any European project to facilitate the delivery of digital public services across borders and sectors. Amongst these is the CEF eTranslation service [16] which is able to automatically translate formatted documents and plain text between any pair of EU official languages, as well as Icelandic and Norwegian. This mature and operational machine translation service can be directly invoked through an API and thus is an ideal foundation for the multilingual aspects of projects like ACROSS. Concretely, even though its integration is not yet covered within the initial design, the CEF Digital eTranslation service will serve as an important basis for the multilingual operation of the ACROSS Virtual Assistant in the future.

One of the general objectives pursued with the development of the Virtual Assistant is accessibility. Apart from being a laudable goal in general, this objective is also clearly rooted in European legislation. Concretely, the following EU regulations are strongly relevant - their content, but also their rationales (documented in the preambles) attest the considerable esteem the European Union and its member states devote to issues of accessibility, in particular with respect to IT products and services:

**European Accessibility Act**

This regulation [17], formally known as “DIRECTIVE (EU) 2019/882 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 17 April 2019 on the accessibility requirements for products and services” aims to “increase the availability of accessible products and services in the internal market and improve the accessibility of relevant information.” and specifies a list of product and services categories to which additional accessibility requirements and constraints will apply after a certain deadline (June 28, 2025). Apart from a number of hardware product categories, this list also includes software product categories, amongst other:

- consumer banking services

- e-commerce services

- "websites” and “mobile device-based services including mobile applications” if they are “elements of air, bus, rail and waterborne passenger transport services, except for urban, suburban and regional transport services”

The services mentioned above are very relevant to ACROSS and the Virtual Assistant, in that they clearly represent examples for private-sector services that can be needed within service workflows according to the ACROSS use cases.

**Web Accessibility Directive**

This directive [18], formally known as “Directive (EU) 2016/2102” aims to provide people with disabilities with better access to websites and mobile apps of public services. It obliges websites and mobile apps of public sector bodies to meet specific technical accessibility standards and includes provisions for regular monitoring of public sector websites and apps by Member States.

This regulation, too, is highly relevant to the ACROSS project and the Virtual Assistant, because it explicitly focuses on the web-based and mobile app offerings of the public sector, which clearly includes many of the services that can be needed within service workflows according to the ACROSS use cases. But notably, this category would also include a future, operational version of the ACROSS web and mobile app itself.

### 3.2.2 Approach and Relation to other Work Packages and Deliverables

The Virtual Assistant is part of the citizen front end part of ACROSS. It is closely coupled to the ACROSS web application and mobile application in order to provide the aforementioned additional conversational control of the applications. The user journey service developed influences the Web UI, and thus, indirectly, the Virtual Assistant.

Because of the minimally-invasive integration approach used, which does not require in-depth modification of the UI it is used to control, the Virtual Assistant can be regarded mainly as an add-on to the ACROSS apps, instead of an interwoven ingredient.

As is requires no direct couplings with other ACROSS components, the VA builds on the output of the other work packages (notably requirements arising from the use cases work in WP2 and WP6), but has minimal impact in the reverse direction.

## 3.3 VA – Requirements

The Virtual Assistant covers some general ACROSS requirements and fulfils special requirements for the VA component. Those requirements are listed in this section.

### 3.3.1 Requirements from WP5 and general ACROSS requirements

WP5 has gathered a first set of requirements for the whole ACROSS platform and ICT modules [12]. Some of these requirements are directly related to the VA while others are generic IT requirements or non-functional requirements. The full list of requirements gathered in WP5 is included in Annex I – WP5 Requirements. Those relevant to the VA are also shown in this subsection. Req_03 is the foundational requirement from WP5 applying to the VA, whereas the remaining requirements shown below are general requirements to the components implemented in ACROSS, which also apply to the VA (to an appropriate extent).

**Table 2 Requirements to the VA from WP 5**

| Id | Title | Description | Type | Category |
|---|---|---|---|---|
| **Req_03** | User Journey Chatbot implementation | A Multi-lingual Virtual Assistant API should be provided as a service. All the User applications should be able to connect to that API and benefit in their User Experience. | functional | Virtual assistant to guide the citizen |
| **Req_07** | DevOps Processes Setup | A full end-to-end pipeline of processes should be set up to ensure the successful integration, deployment, testing and delivery of the services. The DevOps processes, development and operations should be integrated into a single-minded entity with common goals:<br>high-quality software, faster releases, and improved users' satisfaction. | non functional | Platform architecture and interoperability |
| **Req_10** | User Journey Experience | Citizen should be able to navigate in a straightforward clearly defined way through the whole process of the User Journey provided user experience. I would also like to have support during the steps of the moving abroad process. | non functional | Platform architecture and interoperability |

| | | | | |
|---|---|---|---|---|
| **Req_12** | Scalability | The ACROSS platform should be designed to be scalable in terms of computational load, number of users accessing applications and amount of data storage. In particular, the platform should be able to scale horizontally (e.g. add more nodes to a computational network)and vertically (e.g. add resources such as Memory, CPU to a single node in a system). | non functional | Platform architecture and interoperability |
| **Req_19** | Reliability and Integrity | The implementation of ACROSS should follow open standards and use well-known and widely accepted technologies in order to ensure integrity. The ACROSS platform has to be reliable assuring integrity of the components/tools that are part of it. | non functional | Platform architecture and interoperability |
| **Req_22** | Privacy and Data Protection | The ACROSS platform has to be compliant with the EU legislation regarding privacy and data protection. It should adopt all the necessary technologies, standards and methods to protect privacy of the users of the platform services and to secure stored information that could be considered private. | non functional | Security and Privacy |
| **Req_29** | No vendor lock-in | I want the ACROSS reference architecture to be technologically agnostic to avoid vendor lock-in. | non functional | Platform architecture and interoperability |
| **Req_30** | Open source | I want the ACROSS reference architecture to reuse already available open source solutions and only create or improve those aspects that are not covered by the existing solutions | non functional | Platform architecture and interoperability |
| **Req_33** | Accessibility | The front-ends of the system should comply with the current Web Accessibility Directives and in particular with EN301549 (included in WCAG-2.1) | non functional | Web&Mobile applications |

| | | | | |
|---|---|---|---|---|
| **Req_35** | Usability and adaptability | The provided solutions in the platform should be user-friendly and easy to use and should be multilingual. No piece of text that might be displayed to a user shall reside in source code and solution and user should be able to select the preferred language. The implementation of the system should follow open standards and use well-known and widely accepted technologies in order to ensure ease of use. | non functional | Platform architecture and interoperability |
| **Req_36** | Minimal browser support. | The component user interface (where available e.g. dashboards, forms, etc.) should provide support for the wide range of widely used browsers. | non functional | Web&Mobile applications |

In addition, several general ACROSS requirements were identified from project documents or in discussions with project partners, which the Virtual Assistant shall abide to.

REQ-GEN-1. As a developer in ACROSS I want the Virtual Assistant to be easy to integrate with the other software and infrastructure components of ACROSS during development and operation.

REQ-GEN-2. As a developer in ACROSS I want integration with the Virtual Assistant to not hamper the functionality of other ACROSS components, in particular the Web or Mobile Application.

REQ-GEN-3. As a developer in ACROSS I want the Virtual Assistant to use the Apps, and, indirectly, the User journey Services Engine and the results of the User Journey Modelling Tool in order to access the elementary Public Services.

### 3.3.2 VA – Requirements from WP2 (ACROSS New Governance Model) and WP6 (Use cases)

REQ-VA-1. As a user of the Virtual Assistant I want the Virtual Assistant to provide access to the steps of the process defined by the User Journey in a sufficiently functionally similar manner as when traditional interaction would be used.

REQ-VA-2. As a user of the Virtual Assistant I want the Virtual Assistant to communicate and to steer the sequence of individual steps that need to be executed for the individual user journeys.

### 3.3.3 VA-specific requirements

#### *3.3.3.1 Conversational UI control*

REQ-VA-3. As a user of the Virtual Assistant I want to have additional *conversational* control over the application via natural language and speech for a convenient and accessible User Journey Experience.

REQ-VA-4. As a user of the Virtual Assistant I want the Virtual Assistant to be closely integrated with the ACROSS application UI.

REQ-VA-5. As a user of the Virtual Assistant I want the Virtual Assistant to provide access to an appropriate part of the features the UI is providing.

REQ-VA-6. As a user of the Virtual Assistant I want interaction through the Virtual Assistant to be as equivalent to traditional UI interaction as possible (with a conversational interaction mode) regarding input, navigation, and output.

#### *3.3.3.2 Multi-language support*

REQ-VA-7. As a user of the Virtual Assistant I want the Virtual Assistant to have multilingual capability.

#### *3.3.3.3 Customization for Services and Workflows*

REQ-VA-8. As a user of the Virtual Assistant I want the Virtual Assistant to be customizable to the specific services and workflows.

#### *3.3.3.4 Other VA-specific requirements*

REQ-VA-9. As a developer and user in ACROSS I want the Virtual Assistant to work without slowing down the browser, server, and general user experience (beyond the extent that is inevitable due to the limited interaction bandwidth of natural language and speech)

## 3.4 VA - Design

### 3.4.1 VA - Architecture

The following figure provides an overall view of the main subsystems and modules of the VA and how they collaborate with the different parts of the ACROSS platform.

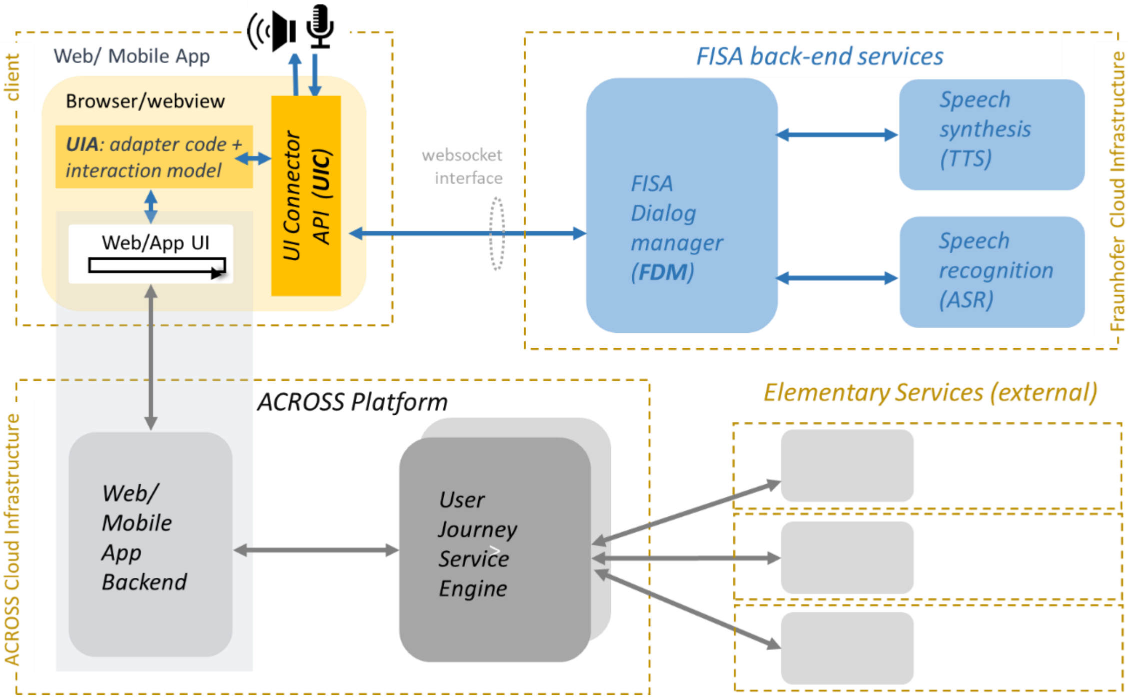


Figure 7: Subsystems and modules of the VA and their collaboration with the ACROSS platform (orange: front-end components, blue: back-end services)

## 3.4.2 VA - Modules

The ACROSS Virtual Assistant consists of two modules: The Web User Interface(UI) Connector (UIC) and the WebUI Adapter (UIA). It also collaborates with the FISA Dialog Manager (FDM) - this component is not developed within ACROSS but is background material[2]. Nevertheless, it will be summarily described below for completeness.

### 3.4.2.1 VA UI Connector (UIC)

The UI Connector (UIC) connects the UI and the backend bi-directionally. It provides a generic, application independent library for connecting a WebUI with the FISA Dialog Manager (FDM) in order to add a conversational (chat and/or speech) interface to the WebUI. This way, every WebUI can be connected to the Dialog Manager, and the same holds for the ACROSS mobile App (as it its development is based on the same web technology as the ACROSS Web App, essentially mirroring it).

[2] Relying on background material that currently is not open-source inevitably induces risks, concretely regarding ACROSS WP5 requirement Req_29 [12]. On the plus side, reliance on background material makes sense economically though, in that more functionality can be achieved with the given project resources. A future version of this deliverable will document the interface used by the FDM. That way, switching to another dialog manager implementation will be possible if need be, and any vendor lock-in risk is mitigated.

A WebUI to be controlled by the Virtual Assistant is extended with a WebUI Adapter, which through the UIC communicates with the Dialog Manager (FDM). By bridging the gap between the WebUI and the UIC, the WebUI Adapter effectively makes the WebUI controllable from the FDM.

Technically, the UIC creates and maintains a websocket connection to the FDM. For each conversational interaction to execute, the UIC receives a request containing an interaction state (also called interaction point) ID from the WebUI Adapter, and delivers it to FDM through the websocket connection using a dedicated wire protocol. (For maintaining the websocket connection, the VA implementation employs a Socket.IO open-source component, which reduces the implementation effort and increases resilience against important failure modes.)

For the other direction, the UIC decodes messages from the FDM received through the websocket connection, and forwards them to the WebUI Adapter. This includes input values or navigation requests recognized from user utterances within the ongoing conversation.

The implemented FISA wire protocol is a bi-directional streaming protocol and uses, amongst others, command, text, and audio packets based on a protocol schema (specified using features of the Socket.IO implementation).

The UIC is also responsible for communicating with the client system audio resources (microphone and speaker) available through the browser or mobile webview. The UIC thus provides all audio input and output handling (as far as needed within the client for voice interfaces). In this context, it also addresses browser autoplay restrictions (to the maximum extent allowed by the browser vendors).

#### *3.4.2.2 VA WebUI Adapter (UIA)*

The WebUI Adapter (UIA) is the “remote control” of the WebUI. It adapts and couples the WebUI to the UI Connector, which in turn enables communication with the FDM backend.

The UIA implements detailed, application specific adapter code for a given WebUI. This way it enables ways for remotely controlling the WebUI in both directions: sending user actions and receiving according signals; as well as sending interaction states and receiving spoken output for the user.

In addition to implementing application-specific details of how to control the *individual WebUI elements* (e.g. code for filling specific input fields), the UIA is also responsible for letting the UIC *control the WebUI* *on the macroscopic (interaction workflow[3]) level*, namely by supporting *navigation* within the WebUI in a conversational manner.

In traditional (screen/keyboard/pointing device) interaction, the user's navigation needs are covered by web browser functionality: The user may at any point select (click into) a specific input field in order to use it next, thus making this field obtain keyboard input focus. In form-based UIs, the browser also often offers the option of just pressing the TAB key to go to the next input field, or SHIFT-TAB to go back to the previous one. However, this simplified navigation only works for well-structured web pages. In a conversational UI interaction mode, it is essential to always have a reliable basis for interaction workflow-level navigation, especially so since the fall-back option of just explicitly selecting the field the user has decided to use next is not practical (because there is no pointing device for instantaneously expressing this intent in two dimensions).

Hence, the WebUI adapter also has to provide a basic navigation model of the WebUI to the UIC. This model (also called the Navigator) must be able, at any current interaction point within the interaction workflow, to determine all "target" interaction points the user might want to navigate to from here. I.e., the *next and the previous interaction point*, the *next and the previous page or screen*, the *first and last interaction point* (on a page/screen), and all "*jump targets*" reachable from the current page/screen, such as help pages/screens or pop-ups.

Whenever the UIA has determined the user's intent to navigate to a new interaction point, this interaction point is passed on to the FDM through the UIC. The backend then starts a conversational interaction for that interaction point, e.g. by generating a system utterance to be presented to the user. For example, the system utterance might request the user to utter input data for the input field associated to this interaction point. This system utterance is then passed through the websocket connection to the UIC where it is shown/played to the user.

The following user utterance is streamed back by the UIC to the FDM, where its content is analyzed.

When the FDM has recognized a valid input value in the user's utterance, an appropriate command is streamed to the UIC, which triggers execution of application-specific code within the UIA for value assignment to the input field. Or, if the user responded with a navigation request, a specific command

[3] Within Chapter 3, the term "*workflow*" is used in a somewhat different and more fine-granular sense than it has been in Chapter 2. In the context of the UJMT, the elements of a (user journey) workflow are (abstract or concrete) *services*. In the VA, however, the focus is on *individual interactions* (such as data inputs or button clicks) occurring during a user session - and using a single service might already require a complex sequence of individual interactions. Therefore, to eliminate ambiguity, the more specific term "*interaction workflow*" is used throughout this chapter.

expressing that intent is sent from the FDM to the UIC, which then invokes the navigation model part of the UIA (also called the Navigator) to make the navigation actually happen in the WebUI.

### *3.4.2.3 FISA Dialog Manager (FDM)*

The FISA Dialog Manager (FDM) is part of the aforementioned back-end that receives user input or interaction state, analyses textual user utterances and/or speech and generates system utterances (text and/or speech) or commands accordingly.

In detail, the FDM enables conversational control for WebUIs by handling all dialog management needs for conversationally controlled interactions. It provides NLU / intent recognition by different means, e.g. by pattern matching or constraint rules. The FDM enables data input recognition (also called “slot filling”), i.e. assigning values recognized in user input texts/utterances to “input slots” corresponding to input elements of the WebUI. The FDM also provides conversational response generation, i.e. creation of natural language output (system utterances) in text/speech form.

The FDM is also able to support spoken-language interfaces, by delegating to sub-services for speech recognition or speech synthesis. These components will not require separate, direct connections to the front-end – all data communications for them will be channelled through the FDM.

In order to fulfil its task, the FDM receives messages from the WebUI via the UIA and UIC. The messages must conform to the FDM protocol. Based on these messages, the FDM reacts to conversational user input (i.e. textual or speech utterances) during an ongoing conversational interaction, and to client-triggered interaction point changes (which abort any still-ongoing conversational interaction and start a new one). Interaction point changes may originate from the Navigator part of the adapter code (in reaction to an FDM message originally caused by a user utterance, see below), or from traditional screen interactions of the user (because the user may want to switch back-and-forth between conversational and traditional interaction).

The FDM sends messages, again conforming to the FDM protocol, to the WebUI via UIC and UIA in order to provide answer texts/utterances, provide value assignment requests resulting from user intents, communicate WebUI navigation requests resulting from user intents, and control the audio input (i.e. microphone activation) and output (playback of system utterances) behaviour to occur on client-side.

The FDM has restricted state context, it only “sees” either a single interaction point or a small collection of interaction points, e.g. the one corresponding to all interaction elements on a single page or screen. The FDM does not persist information beyond the end of the conversational interactions being executed for this interaction point (or set of points).

### 3.4.3 VA - Interfaces and Collaborations

#### *3.4.3.1 WebUI - UIA Interface*

The WebUI – UIA interface connects the Web/Mobile App with the Virtual Assistant, namely its UIA.

It is an external interface.

The interface is defined in terms by one or more of the following mechanisms:

- specific JavaScript (JS) API calls of the UI framework in which the WebUI is implemented
- specific URLs for opening specific pages or screens of the WebUI
- selected concrete DOM Elements belonging to the WebUI and uniquely identified by a suitable mechanism (e.g. tags)
- selected specific DOM event bindings, for selected events to which the VA is intended to react

This interface is application/WebUI-specific.

#### *3.4.3.2 UIA - UIC Interface*

The UIA - UIC Interface connects WebUI Adapter (UIA) and UI Connector (UIC).

It is an internal interface.

The interface is an asynchronous JavaScript API provided by the UIC. Method calls for actions originate from the UIA. Callbacks defined within the UIA for actions originate from the UIC (and are triggered by messages arriving from the FDM).

This interface is documented by UIC JSdoc.

#### *3.4.3.3 UIC – FDM Interface*

The UIC – FDM Interface connects UI Connector (UIC) and FISA Dialog Manager (FDM).

It is an internal interface.

The interface provides a websocket connection using a specific FDM wire protocol. This interface and protocol are documented within the FDM.

The UIC is responsible for the connection life cycle, i.e. (re-)creating a connection whenever it is needed and destroying the connection when no longer needed.

There are security considerations for the interface: The FDM needs to accept connections from any browser/mobile device that executes the ACROSS WebUI. In order to protect the websocket server e.g. from DOS attacks, a suitable (fixed-key or time-based) authentication mechanism is employed.

## 3.5 VA – Description of Initial PoC

The initial proof of concept (PoC) for the Virtual Assistant is based on the Web/Mobile App PoC. It will demonstrate the use of the Virtual Assistant in selected areas of the Web/Mobile App PoC.

The PoC will make a set of essential interactions offered by the Web/App prototypes controllable through a conversational (natural language) interface. It will focus on conversational navigation for essential user journey-relevant interactions offered by the WebUI.

In order to demonstrate the abilities for the selected interactions, the Virtual Assistant will be instantiated as follows:

An interaction model will be built for the selected interactions. Adapter code is written for coupling the selected WebUI interactions with the UI Connector. The WebUI components, their dependencies (i.e. UI framework packages) and the VA components UIA and UIC are integrated into a single ACROSS front-end subsystem and run together in the browser/webview.

This integrated front-end subsystem then communicates both with the standard ACROSS back-end services and with the FDM.

On the backend side, the necessary FISA DM customizations are done: recognition of the navigation intents needed, and response generation for state information and other information aspects.

## 3.6 VA - Baseline technologies

The Virtual Assistant uses some standardized baseline technologies. Those technologies are:

**WebSockets:** a communication protocol, standardized by the IETF. It provides full-duplex communication channels using a single TCP connection. It is a mature, well-defined, well-known standard for communication.

**Socket.IO** is a quasi-standard cross-language API based on the IETF WebSocket protocol. It enables real-time, bidirectional and event-based communication between the browser and the server. It is developed in an open and distributed manner under the guidance the Open Source Collective, sponsored by almost 200 companies and organizations and routinely used by an even larger professional audience.

**Document Object Model (DOM)**: Standard interface for manipulating HTML documents, treating the documents as tree structures with identifiable objects. It is standardized by the W3C and the WHATWG. DOM is used to programmatically access, read and modify the tree or the contents of its nodes.

**Representational state transfer (REST)** is an architectural style for implementing hypermedia systems in the Web. In a narrower meaning that is used here, it provides a definition (rather guidelines) of stateless web APIs. Web APIs that conform to REST guidelines are called RESTful APIs. REST is not standardized per se, but it is well defined by papers and convention.

**JavaScript** (also called ECMAScript) is a programming language, it is a core technology of the Web, just as HTML or CSS. It conforms to the ECMAScript standard but comes in different flavors by different vendors. Nonetheless, a core JavaScript language definition is supported in all flavors, thus making the use of the language safe for our needs.

## 3.7 VA - Conclusions and next steps

Despite limited functionality, the demonstrator already fully validates the principal approach of integration of the VA in ACROSS. It also demonstrates the potential the VA offers for extending any desired set of possible Web/Mobile App interactions with a conversational interface, both for written natural language (i.e. in a chatbot setting) and for spoken natural language (i.e. in a voice-assistant setting). For the next version, firstly the scope of the covered interactions will be extended (e.g. data input). Secondly an extension of the interaction modes will be targeted (i.e. spoken voice interaction).

# 4 References

[1] "Final Report Summary - BPM4PEOPLE (Business Process Modelling for Participatory Enterprises, Organizations, and Public Aministration Bodies)," [Online]. Available: https://cordis.europa.eu/project/id/285929/reporting.

[2] "BPM4PEOPLE project page in CORDIS," [Online]. Available: https://cordis.europa.eu/project/id/285929.

[3] "Productive4.0," [Online]. Available: https://productive40.eu.

[4] J. Erasmus, I. T. P. Vanderfeesten, K. Traganos, R. Keulen and P. W. P. J. Grefen, "The HORSE Project: The Application of Business Process Management for Flexibility in Smart Manufacturing," 2020.

[5] "The COMPOSITION project, Ecosystem for Collaborative Manufacturing Processes – Intra- and Interfactory Integration and Automation," [Online]. Available: https://www.composition-project.eu.

[6] "PASSME Website," [Online]. Available: https://passme.eu.

[7] "PASSME project page in CORDIS," [Online]. Available: https://cordis.europa.eu/project/id/636308.

[8] "Gravitate–Health project page in CORDIS," [Online]. Available: https://cordis.europa.eu/project/id/945334.

[9] "About ISA² - European Commission," [Online]. Available: https://ec.europa.eu/isa2/isa2_en.

[10] "Core Public Service Vocabulary Application Profile - Joinup.eu," [Online]. Available: https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/solution/e-government-core-vocabularies/core-public-service-vocabulary-application-profile.

[11] "European taxonomy for public services - Joinup.eu," [Online]. Available: https://joinup.ec.europa.eu/sites/default/files/news/2019-09/ISA2_European%20taxonomy%20for%20public%20services.pdf .

[12] "ACROSS D5.1 System Architecture & Implementation Plan – Initial".

[13] “jgraph/drawio: Source to app.diagrams.net - GitHub,” [Online]. Available: https://github.com/jgraph/drawio.

[14] “Excalidraw - Github,” [Online]. Available: https://github.com/excalidraw/excalidraw.

[15] “Camunda Modeler - GitHub,” [Online]. Available: https://github.com/camunda/camunda-modeler.

[16] “Bonita Studio - GitHub,” [Online]. Available: https://github.com/bonitasoft/bonita-studio.

[17] “Connecting Europe Facility,” [Online]. Available: https://ec.europa.eu/inea/en/connecting-europe-facility.

[18] “CEF eTranslation service,” [Online]. Available: https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Machine+translation.

[19] “European Accessibility Act,” [Online]. Available: https://ec.europa.eu/social/main.jsp?catId=1202.

[20] “Web Accessibility Directive,” [Online]. Available: https://directive2102.eu/.

# 5 Annex I – WP5 Requirements

| Id | Title | Description | Type | Category |
|---|---|---|---|---|
| **Req_01** | Semantic and technical interoperability with SDG | The system should ensure an alignment of semantic and technical interoperability with SDG IT Tools | non functional | Platform architecture and interoperability |
| **Req_02** | Data Governance and Data Sovereignty features should support cross border services | This implementation aims at the secure storage for user data and documents. This requirements ensures the Cross borders public services adaptation | functional | Data governance and sovereignty |
| **Req_03** | User Journey Chatbot implementation | A Multi-lingual Virtual Assistant API should be provided as a service. All the User applications should be able to connect to that API and benefit in their User Experience. | functional | Virtual assistant to guide the citizen |
| **Req_04** | Data Governance framework Implementation for transparency | A Transparency dashboard should be provided to the users to control how their (personal) data are used by public administrations, businesses, or a data brokers to easily manage and handle sensitive information. | functional | Data governance and sovereignty |
| **Req_05** | Web Application for moving abroad Implementation | A web application for one-stop-shop service should be implementend to support the users to better understand, complete and seamless information exchange within the whole cross-border digital service and the involved stakeholders. | functional | Web&Mobile applications |

| | | | | |
|---|---|---|---|---|
| **Req_06** | Native Application for moving abroad Implementation | A native application for one-stop-shop service should be implemented to support the users to better understand, complete and seamless information exchange within the whole cross-border digital service and the involved stakeholders. | functional | Web&Mobile applications |
| **Req_07** | DevOps Processes Setup | A full end-to-end pipeline of processes should be set up to ensure the successful integration, deployment, testing and delivery of the services. The DevOps processes, development and operations should be integrated into a single-minded entity with common goals:<br>high-quality software, faster releases, and improved users' satisfaction. | non functional | Platform architecture and interoperability |
| **Req_08** | Continuous Platform Integration | The DevOps approach should be followed in ACROSS Platform to include the continuous integration tools. | non functional | Platform architecture and interoperability |
| **Req_09** | Free access to other countries' e-services | Citizens should be able to access other countries' services | non functional | Platform architecture and interoperability |
| **Req_10** | User Journey Experience | Citizen should be able to navigate in a straightforward clearly defined way through the whole process of the User Journey provided user experience. I would also like to have support during the steps of the moving abroad process. | non functional | Platform architecture and interoperability |
| **Req_11** | Service/Application execution environment | The ACROSS platform has to provide a virtual environment to deploy and execute applications and services and support communication among them. This environment should support different technologies and standards such as different operative systems and software platforms. It should be possible to remotely administrate this environment. | functional | Platform architecture and interoperability |

| | | | | |
|---|---|---|---|---|
| **Req_12** | Scalability | The ACROSS platform should be designed to be scalable in terms of computational load, number of users accessing applications and amount of data storage. In particular, the platform should be able to scale horizontally (e.g. add more nodes to a computational network)and vertically (e.g. add resources such as Memory, CPU to a single node in a system). | non functional | Platform architecture and interoperability |
| **Req_13** | Interoperability with legacy systems | It has to be possible to connect the ACROSS platform with the existent PA legacy systems (e.g. databases, web services). Secure and reliable communication with the existing public administration information systems have to be provided without requiring changes in these systems. The platform should also provide tools and predefined components to facilitate the interoperability. | non functional | Platform architecture and interoperability |
| **Req_14** | Accounting | The platform has to be able to measure, track and record all the information regarding the usage of services and resources. | functional | Security and Privacy |
| **Req_15** | Easy to use service integration and orchestration tools | In order to create cross border services, the platform has to support Public and Private providing a set of tools and applications that will help them to easily implement service integration. | functional | Connectors to integrate the private and public sector offering |
| **Req_16** | Open API access | Data and services available in the ACROSS platform have to be accessible via a set of APIs using standardized approaches (e.g. RESTful API). | functional | Platform architecture and interoperability |

| | | | | |
|---|---|---|---|---|
| **Req_17** | Service Registries | ACROSS platform has to maintain registries of all available services offered by different PAs, SMEs and by the platform itself. Every service should be well-described using standard metamodels | functional | Connectors to integrate the private and public sector offering |
| **Req_18** | Cross Border Authentication | The services deployed and executed in ACROSS platform should have the possibility to be integrated, if needed, with eIDAS system. The platform can optionally support single-sign-on mechanism to simplify authentication on multiple applications and services internally to the platform. | functional | Security and Privacy |
| **Req_19** | Reliability and Integrity | The implementation of ACROSS should follow open standards and use well-known and widely accepted technologies in order to ensure integrity. The ACROSS platform has to be reliable assuring integrity of the components/tools that are part of it. | non functional | Platform architecture and interoperability |
| **Req_20** | Security access | Access to services and data has to be available to authorized users/applications only. Only audited applications are allowed to be deployed to ensure compliance with the security policies. Every security violation should be reported and the necessary actions to protect information and applications present in the platform has to be performed. | functional | Security and Privacy |
| **Req_21** | Secure storage | ACROSS platform has to provide secure storage functionalities in order to record data needed for the execution of specific services on the platform. | functional | Security and Privacy |

| | | | | |
|---|---|---|---|---|
| **Req_22** | Privacy and Data Protection | The ACROSS platform has to be compliant with the EU legislation regarding privacy and data protection. It should adopt all the necessary technologies, standards and methods to protect privacy of the users of the platform services and to secure stored information that could be considered private. | non functional | Security and Privacy |
| **Req_23** | OpenID Connect - Role-Management | The clients using the ACROSS-Platform have to be authenticated and authorised to get only the permissions that are required. This has to be defined by roles. | functional | Security and Privacy |
| **Req_24** | OpenID Connect - Role-Management | For securely exchanging roles between two parties it is needed to define a data format | functional | Security and Privacy |
| **Req_25** | OpenID Connect - Client-Registration | It is needed to register the clients that want to use the ACROSS-Platform | non functional | Security and Privacy |
| **Req_26** | OpenID Connect - Client-Registration | Definition of the client registration workflow | business | Security and Privacy |
| **Req_27** | Catalogue of services (public/private) data model | The Catalogue of services data model will follow the common public core vocabularies coming from ISA2 and the EIF implementation regulation and will support interoperability with SDG | functional | Platform architecture and interoperability |
| **Req_28** | Catalogue of services (public/private) objective | The catalogue of services will take care of harmonisation of the private and public services and related data enabling semantic interoperability and supporting the selected common vocabularies should be used to express the metadata. | functional | Platform architecture and interoperability |
| **Req_29** | No vendor lock-in | I want the ACROSS reference architecture to be technologically agnostic to avoid vendor lock-in. | non functional | Platform architecture and interoperability |

| | | | | |
|---|---|---|---|---|
| **Req_30** | Open source | I want the ACROSS reference architecture to reuse already available open source solutions and only create or improve those aspects that are not covered by the existing solutions | non functional | Platform architecture and interoperability |
| **Req_31** | Data Usage control | I want to define a Data Usage Policy so that I can retain management rights, define rules and conditions on how data, used in cross border services, must be used<br>(e.g. who can see my data and which parts, prohibit forwarding to 3rd parties and other participants, merging data, the use that can be given to my data, etc.) | functional | Data governance and sovereignty |
| **Req_32** | Identity Certificate | I want to have a unique identity ACROSS ecosystem in the form of a certificate, so that secure and trusted connections to all parties, internally and externally (e.g public and private service) can be established during cross border service provisioning. | functional | Security and Privacy |
| **Req_33** | Accessibility | The front-ends of the system should comply with the current Web Accessibility Directives and in particular with EN301549 (included in WCAG-2.1) | non functional | Web&Mobile applications |
| **Req_34** | Confidentiality | The platform has to follow the 'privacy-by-design' and 'security-by-design' approaches and in particular should comply with the principle that users should provide only the information that is absolutely necessary. | non functional | Security and Privacy |

| | | | | |
|---|---|---|---|---|
| **Req_35** | Usability and adaptability | The provided solutions in the platform should be user-friendly and easy to use and should be multilingual. No piece of text that might be displayed to a user shall reside in source<br>code and solution and user should be able to select the preferred language. The implementation of the system should follow open standards and use well-known and widely accepted technologies in order to ensure ease of use. | non functional | Platform architecture and interoperability |
| **Req_36** | Minimal browser support. | The component user interface (where available e.g. dashboards, forms, etc.) should provide support for the wide range of widely used browsers. | non functional | Web&Mobile applications |