

## H2020-SC6-GOVERNANCE-2018-2019-2020

### DT-GOVERNANCE-05-2018-2019-2020



## D5.1 System Architecture & Implementation Plan - initial

<b>Project Reference No</b>	959157 — ACROSS — H2020-SC6-GOVERNANCE-2018-2019-2020
<b>Deliverable</b>	D5.1 System Architecture & Implementation Plan – Initial
<b>Work package</b>	WP5: ACROSS Applications & Platform Integration
<b>Nature</b>	Report
<b>Dissemination Level</b>	Public
<b>Date</b>	31/10/2021
<b>Status</b>	Final
<b>Editor(s)</b>	Vincenzo Savarino (ENG), Anestis Sidiropoulos (ATC)
<b>Contributor(s)</b>	Iosif Kanakaris (GRNET), Thilo Ernst, Anna Opaska, Ekkart Kleinod (FRAUNHOFER), Urtza Iturraspe, Idoia Murua, Valentin Sanchez (TECNALIA)
<b>Reviewer(s)</b>	Enrique Areizaga, TECNALIA
<b>Document description</b>	This deliverable is the outcome of the first task in WP5 and it presents the overall architecture of the ACROSS platform in terms of the supported functionalities, the respective processes and the components that realise them.



## 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.

© 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	26/04/2021	Table of Content (ToC)	ENG
V0.2	18/05/2021	Definition of Requirements mapping, first description of architecture and related components	ENG, ATC, TECNALIA
V0.3	11/ 06/2021	Update of architecture and refinements of requirements and component cards	ENG, ATC, TECNALIA, FRAUNHOFER, GRNET
V0.4	07/07/2021	Refinements of requirements and component cards. First version of interaction flows	ENG, ATC, TECNALIA, FRAUNHOFER, GRNET
V0.5	22/07/2021	First version of deployment section	ATC
V0.6	15/09/2021	Second refinement of architecture and component cards	ENG, ATC, TECNALIA, FRAUNHOFER, GRNET
V0.7	08/10/2021	Refinement of architecture and interaction flows	ENG, ATC, TECNALIA, FRAUNHOFER, GRNET
V0.8	13/10/2021	First internal review	ENG, ATC
V0.9	18/10/2021	Pre-final for internal review	ENG, ATC
V1.0	31/10/2021	Submitted	ATC



## Executive Summary

ACROSS is a diverse project with several work packages working in parallel in order to achieve common objectives. In order to start this work with a mutual sense of direction, we needed a high-level blueprint of the overall technology development within the project. The role of this document is to be a common point of reference throughout the project.

This deliverable is the outcome of the first task in WP5 and it presents the overall architecture of the ACROSS platform in terms of the supported functionalities, the respective processes and the components that realise them. The document presents a coarse-grained collection of technical components that provide the necessary functionality and defines the principles for their integration in a manner that ensures completeness, safety and efficiency. To this end, we propose a Service-oriented approach, where each functional unit is implemented as a stand-alone service, communicating in a standardized manner with other components.

The document sets the boundaries for the integration of the ACROSS modules by presenting the context and use for the ACROSS platform. As such, it builds on top of the technical project requirements and presents the way that the ACROSS capabilities will be able to integrate into the operational environments of the pilots which are the early adopters of the proposed solution.

We consider the architecture to be a live document that will evolve and be aligned with the ongoing work in the project. An updated version of this deliverable will be delivered in M22.



## Table of Contents

<b>DOCUMENT REVISION HISTORY</b> .....	<b>3</b>
<b>EXECUTIVE SUMMARY</b> .....	<b>4</b>
<b>TABLE OF CONTENTS</b> .....	<b>5</b>
<b>LIST OF FIGURES</b> .....	<b>7</b>
<b>LIST OF TABLES</b> .....	<b>7</b>
<b>LIST OF TERMS AND ABBREVIATIONS</b> .....	<b>8</b>
<b>1 INTRODUCTION</b> .....	<b>1</b>
1.1 PURPOSE AND SCOPE .....	1
1.2 APPROACH FOR WORK PACKAGE AND RELATION TO OTHER WORK PACKAGES AND DELIVERABLES .....	1
1.3 STRUCTURE OF THE DELIVERABLE .....	1
<b>2 ACROSS REQUIREMENTS</b> .....	<b>3</b>
2.1 PLATFORM ARCHITECTURE AND INTEROPERABILITY .....	3
2.2 SECURITY AND PRIVACY .....	3
2.3 CONNECTORS TO INTEGRATE THE PRIVATE AND PUBLIC SECTOR OFFERING .....	3
2.4 VIRTUAL ASSISTANT .....	3
2.5 DATA GOVERNANCE AND SOVEREIGNTY .....	4
2.6 WEB & MOBILE APPLICATIONS .....	4
<b>3 ACROSS ARCHITECTURE HIGH LEVEL OVERVIEW</b> .....	<b>5</b>
<b>4 ARCHITECTURE COMPONENTS DESCRIPTION</b> .....	<b>7</b>
4.1 DATA HARMONIZATION AND CONNECTORS .....	8
4.1.1 <i>API adapter</i> .....	8
4.1.2 <i>Semantic adapter</i> .....	9
4.1.3 <i>Data connector</i> .....	11
4.1.4 <i>Service Registry</i> .....	12
4.1.5 <i>Metadata and Vocabulary Catalogue</i> .....	13
4.1.6 <i>Data connectors dashboards</i> .....	14
4.2 CITIZEN DATA OWNERSHIP AND USAGE CONTROL .....	15
4.2.1 <i>Citizen Data Ownership</i> .....	15
4.2.2 <i>Usage Control</i> .....	17



4.2.3	API Gateway .....	18
4.3	USER JOURNEY SERVICE MODELING AND DELIVERY .....	19
4.3.1	User Journey Modelling Tool.....	19
4.3.2	User Journey Services Engine .....	21
4.4	CITIZEN FRONT-END .....	23
4.4.1	Mobile Application .....	24
4.4.2	Web Application.....	25
4.4.3	Virtual Assistant.....	27
4.4.4	Transparency Dashboard.....	29
4.5	SECURITY.....	31
4.5.1	Identity and access management .....	31
4.5.2	Audit Logger.....	32
4.5.3	Data Access Control .....	33
<b>5</b>	<b>INTERACTION SCENARIOS.....</b>	<b>35</b>
5.1	CITIZEN FRONT-END: WEB/MOBILE APPLICATIONS AND VIRTUAL ASSISTANT .....	35
5.2	USER JOURNEY SERVICE MODELLING AND DELIVERY .....	35
5.3	CITIZEN OWNERSHIP AND DATA USAGE CONTROL .....	37
5.4	DATA HARMONIZATION AND CONNECTORS .....	38
<b>6</b>	<b>DEPLOYMENT SCENARIOS AND IMPLEMENTATION GUIDELINES .....</b>	<b>41</b>
6.1	DEPLOYMENT AND IMPLEMENTATION APPROACH .....	41
6.2	DEPLOYMENT SCHEMAS.....	42
<b>7</b>	<b>CONCLUSIONS .....</b>	<b>44</b>
<b>8</b>	<b>ANNEX 1-REQUIREMENTS TABLES .....</b>	<b>45</b>
<b>9</b>	<b>REFERENCES .....</b>	<b>52</b>



## List of Figures

FIGURE 1: ACROSS CONCEPTUAL ARCHITECTURE .....	5
FIGURE 2: COMPONENT VIEW OF ACROSS PLATFORM .....	7
FIGURE 3: WEB/MOBILE APPLICATIONS AND VIRTUAL ASSISTANT INTERACTION FLOW.....	35
FIGURE 4: USER JOURNEY MODELLING AND DELIVERY INTERACTION FLOW .....	36
FIGURE 5: CITIZEN OWNERSHIP AND DATA USAGE CONTROL INTERACTION FLOW .....	37
FIGURE 6: SERVICE DESCRIPTION INTERACTION FLOW.....	39
FIGURE 7: SERVICE REGISTRATION AND CONNECTION INTERACTION FLOW .....	40

## List of Tables

TABLE 1: API ADAPTER COMPONENT CARD .....	9
TABLE 2: SEMANTIC ADAPTER COMPONENT CARD.....	10
TABLE 3: DATA CONNECTOR COMPONENT CARD.....	12
TABLE 4: SERVICE REGISTRY COMPONENT CARD.....	13
TABLE 5: METADATA AND VOCABULARY COMPONENT CARD .....	14
TABLE 6: DATA CONNECTORS DASHBOARDS COMPONENT CARD .....	15
TABLE 7: CITIZEN DATA OWNERSHIP COMPONENT CARD .....	17
TABLE 8: USAGE CONTROL COMPONENT CARD .....	18
TABLE 9: API GATEWAY COMPONENT CARD .....	19
TABLE 10: USER JOURNEY MODELLING TOOL COMPONENT CARD .....	21
TABLE 11: USER JOURNEY SERVICES ENGINE COMPONENT CARD .....	23
TABLE 12: MOBILE APPLICATION COMPONENT CARD.....	25
TABLE 13: WEB APPLICATION COMPONENT CARD.....	27
TABLE 14: VIRTUAL ASSISTANT COMPONENT CARD .....	29
TABLE 15: TRANSPARENCY DASHBOARD COMPONENT CARD .....	31
TABLE 16: IDENTITY AND ACCESS MANAGEMENT COMPONENT CARD .....	32
TABLE 17: AUDIT LOGGER COMPONENT CARD.....	33
TABLE 18: DATA ACCESS CONTROL COMPONENT CARD.....	34



## List of Terms and Abbreviations

Abbreviation	Definition
WP	Work Package
APIs	Application Programming Interface
PA	Public Administration
eIDAS	electronic IDentification, Authentication and trust Services
VA	Virtual Assistant
Req	Requirement
UI	User Interface





# 1 Introduction

## 1.1 Purpose and Scope

This document is the initial report detailing the design of the complete ACROSS system architecture. The main scope is to identify the main building blocks, capabilities, interfaces and interactions, which will satisfy the identified functional and non-functional requirements. Indeed, the process for the definition of the architecture started from the analysis of the requirements elicited from the pilot use cases, but also related to the overall objectives of the ACROSS project. The document, starting from the conceptual model of ACROSS platform, includes the specification of the overall logical structure of the system, from a static to dynamic perspective, giving special focus on the integration framework by defining the programming interfaces that will enable the interaction and communication among the individual components identified and described in the document. This document does not contain details about specific technologies or products to be used to implement each component. Despite the fact that this document is an input for the rest of technical developments of ACROSS components, the architectural design phase itself was also influenced by the on-going work related to the other tasks of the technical work packages (mainly WP3 and WP4 and WP5 itself), that will be finally in charge of the detailed design and concrete implementation of the specific architectural components. The ACROSS implementation phase involves a series of development sprints to develop, extend and customize existing technological assets in accordance with an agile development philosophy to constantly build and update the ACROSS platform based on feedback received from the users. To this end, this document provides a preliminary description of deployment and implementation guidelines the project will follow.

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

WP5 aims at providing the architectural and implementation aspects for the delivery of the ACROSS tools platform taking into account the full range of requirements for such service. The design of the ACROSS will drive the design and implementation of the various components produced in the context of WPs: WP3, WP4 & WP5. The decisions presented in this deliverable are a subject to refinements and modifications, based on the progress of the technical work packages, as well as the validation and evaluation phases.

## 1.3 Structure of the Deliverable

The deliverable is organized as follows:

- Section 2 presents an overview of the list of requirements.



- Section 3 provides the description of the high-level architecture of the ACROSS platform.
- Section 4 provides the description of the ACROSS components.
- Section 5 presents the interaction scenarios.
- Section 6 presents the deployment scenarios and implementation guidelines
- Sections 7, 8 and 9 provides the conclusion of the deliverable, the Annex and the References



## 2 ACROSS Requirements

To organize in several subsections (requirements group) from the list of requirements. The following section provides an overview of each group while the complete list is provided in Annex 1-Requirements tables.

### 2.1 Platform architecture and interoperability

ACROSS Platform must pay particular attention to the core interoperability principles (openness, transparency, reusability, and technological neutrality and data portability) and the principles related to generic user needs and expectations (user-centricity, inclusion and accessibility, security and privacy, multilingualism). The platform should rely on Open Specifications (APIs and shared data models) that support and increase replicability opportunity and the development of an ecosystem of applications and services and its interaction with existent PA legacy systems. ACROSS platform should support scalability and modularity while assuring integrity of its components.

### 2.2 Security and Privacy

ACROSS must ensure that the citizen is the sole owner of their personal data, and that any components or services that need to access this data may do so after having the consent from the user.

To ensure that this consent is valid, the user needs to be authenticated (and subsequently authorized) against an official authority of authentication. eIDAS will be the preferred option as it provides a Europe-wide accepted authentication scheme. ACROSS platform must also ensure authentication and authorization features to the access and use platform tools and services.

### 2.3 Connectors to integrate the private and public sector offering

ACROSS must provide functionalities to connect and access public/private services offered by both the Public Administration and third parties and to get data from heterogeneous sources such as repositories, existing systems (e.g. owned by PA), etc. that could expose different interfaces. The ACROSS system should support the harmonization of data coming from connected data sources and third-party services enabling semantic interoperability according to data models and to provide a standard and common model of service description.

### 2.4 Virtual Assistant

In ACROSS, a Multi-lingual Virtual Assistant (VA) API should be provided as a service. All User-interfacing ACROSS applications should be able to connect to the VA and benefit from their User Experience (in



particular, regarding Usability and Accessibility), by the VA enabling a conversational interaction mode through natural language and speech. When interacting conversationally through the VA, navigation through the steps of the process defined by the User Journey shall be supported by the VA in a sufficiently functionally similar manner as when traditional interaction is used. The VA software components should be designed for appropriate privacy and data protection, should support scalability, and should be easy to integrate with the other software and infrastructure components of ACROSS during development and operation. The VA's front-end part should provide browser compatibility with the main ACROSS Web Application.

## 2.5 Data governance and sovereignty

ACROSS must offer secure storage for user data and documents, in the case it stores these data. This ensures the Cross borders public services adaptation. It must also provide a Transparency dashboard to the users to control how their (personal) data are used by public administrations, businesses, or data brokers to easily manage and handle sensitive information. The user must be able to define the rules and conditions on how data, used in cross border services, must be used (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.

## 2.6 Web & Mobile applications

ACROSS must provide a complete Web and Mobile User Application, which act as the visual interface for the one-stop-shop service. Based on the models received by the User Journey Service Engine, the applications should enable users to clearly understand the sequence of individual steps that need to be taken for the moving abroad journey. Additionally, by invoking the Virtual Assistant module, the user experience should be optimized since the users are guided and assisted during the whole process to complete the steps through a comprehensive and straight-forward User Journey Experience.

### 3 ACROSS architecture high level overview

The following Figure 1 provides a conceptual view of ACROSS architecture by identifying the related layers and actors.

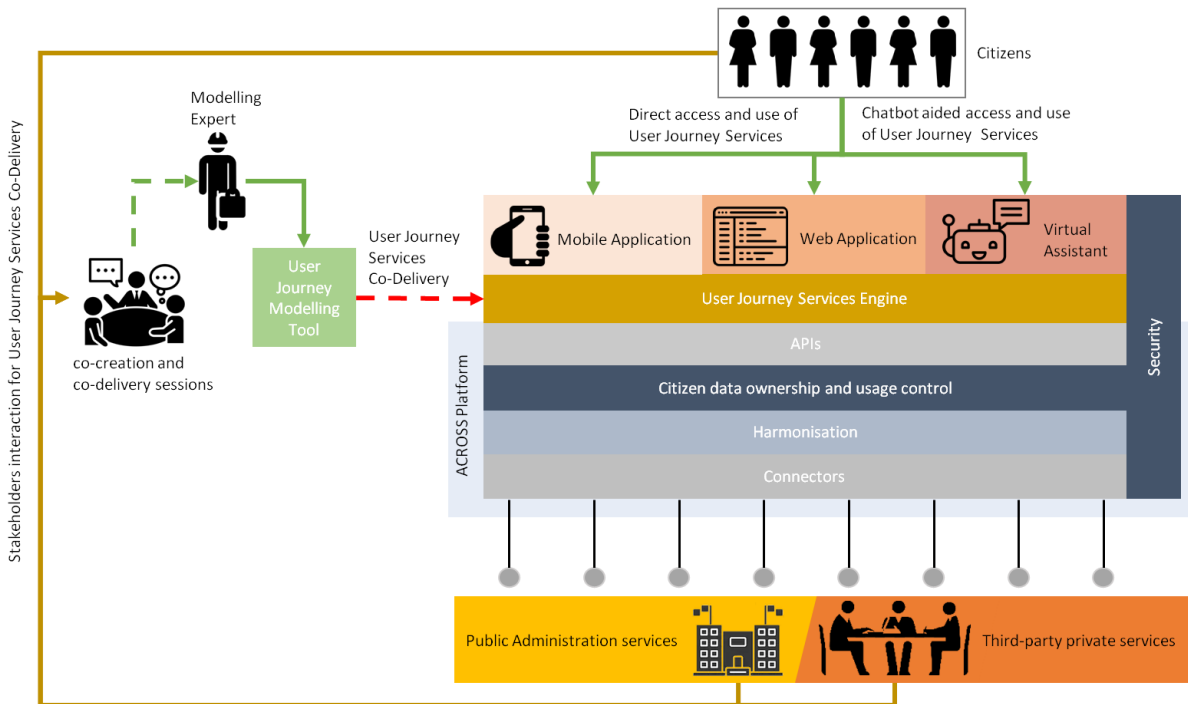


Figure 1: ACROSS conceptual architecture

The Modelling Expert is responsible for modelling the identified cross-border services use as User Journey Models with the User Journey Modelling Tool. From each User Journey Model, the User Journey Modelling Tool creates a machine-readable workflow description for the service orchestration which is provided to the User Journey Services Engine component.

The User Journey Services Engine is responsible for providing and initiating these User Journey Service workflows to the Citizens as well as invoking the specific services from which they are composed. The Citizens can access those workflows and specific services through the Mobile Application, the Web Application, or the Virtual Assistant. Thus, the User Journey Services Engine is responsible for performing the service orchestration by interpreting the provided workflow description and for intermediating between user interface requests and specific services.

The Web & Mobile Application and Virtual Assistant layer provides the assisted visual access towards citizens. Through the applications' User Interface, the citizen can invoke the User Journey Services while



being guided by the Virtual Assistant chatbot in order to complete the necessary steps for the journey abroad process.

The citizen data ownership and usage control will ensure that the data provided by the citizens to the services, is used by the service providers taking into account the consents approved by them.

Harmonization and Connectors layers provide the south bound connection with external systems and services from public and private sectors to provide a uniform description of the services and the related invocation. To this end these layers will provide for each identified service a service adapter, that is a single instance of adaptation of that service (public & private) for its use in ACROSS Platform from several points of view for its use in the upper layers: informational, data governance and service invocation. To support that adaptation, Harmonization and Connectors layers will include a service catalogue with the aim to provides all functionality to register, model, map and publish and manage all the service information, according to the three above points of view, needed to support the uses of each service by the upper layers of ACROSS Platform.

The Security Layer ensures that all information is accessed by authenticated and authorized players, be it the citizens requesting the services or machine components that interconnect data. Moreover, Citizen Data Ownership is ensured through mechanisms of Access Control and possible Data breaches are swiftly addressed by auditing and logging of security events.

## 4 Architecture components description

The following sections will detail the different layers of the ACROSS platform.

The following Figure 2 provides an overall view of the main components of ACROSS platform:

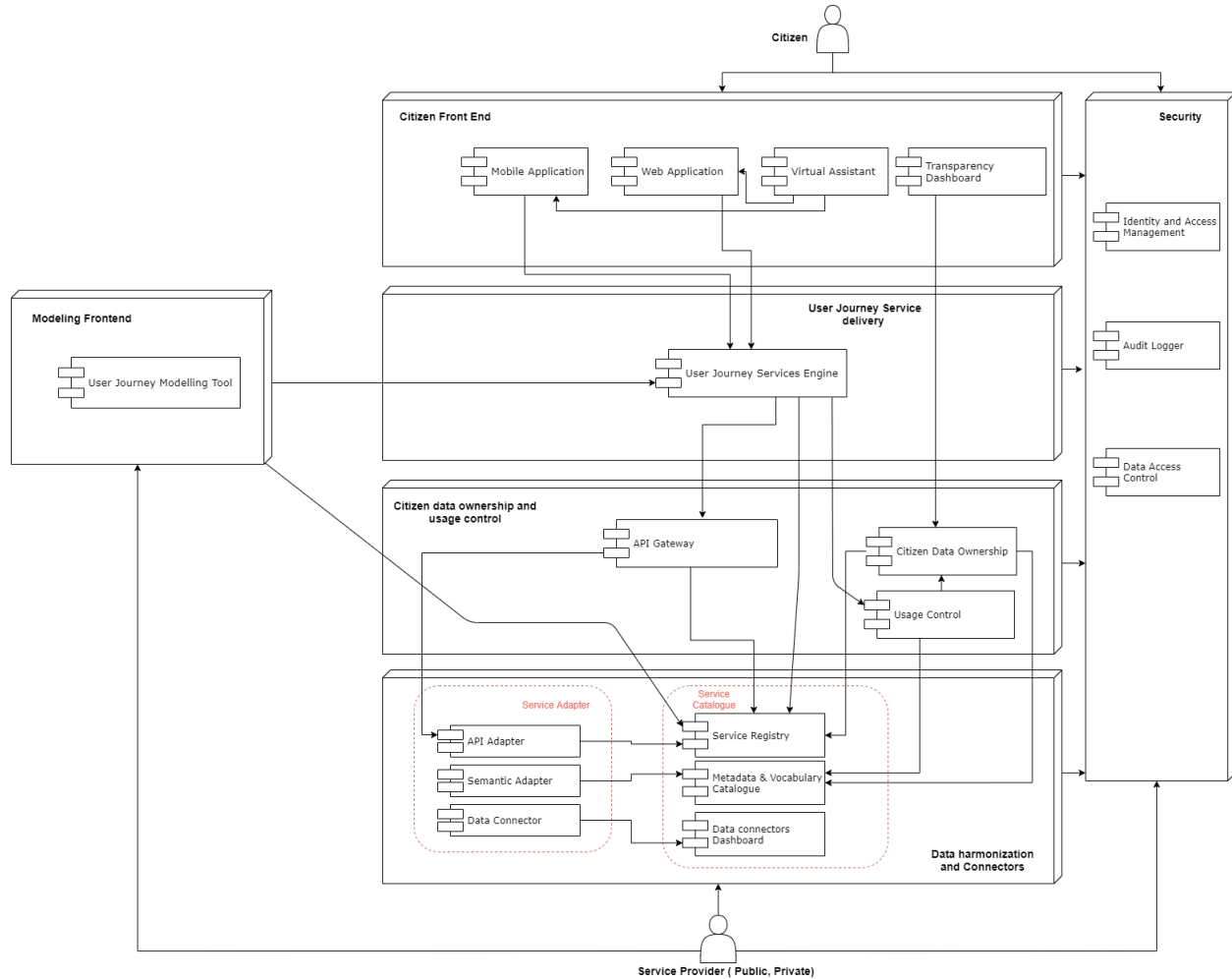


Figure 2: Component View of ACROSS Platform



## 4.1 Data harmonization and connectors

This layer is the connection between ACROSS Platform and external services provided by PAs and private entities to support the composition of user journey services. The main features of this layer are the following:

- Secure connection to internal system of the PAs and services.
- Secure connection to private services to support cross border services.
- Uniform API for data access.
- Exposure of sub-set of existing functionalities/services of PAs and private
- Data access management based on authentication, authorization and privacy management for requested information.
- Semantic adaption to support interoperability according to common vocabularies
- Catalogue of services instances, models and metadata registry

### 4.1.1 API adapter

<b>Component Name</b>	<b>API adapter</b>	
<b>Module Description</b>	The responsibility of this module is to expose a standard set of APIs for data access. It is also responsible for exposing possible functionalities/services supplied by internal or legacy systems of PAs and private service providers.	
<b>Main functionalities</b>	The component provides the following functionalities: <ul style="list-style-type: none"> <li>• Uniform APIs as set of methods in order to provide a shared and common way to access most common data.</li> <li>• Common interfaces to custom methods to access specific data o services.</li> </ul>	
<b>Main logical Interfaces</b>	<b>Interface name</b>	<b>Description</b>
	proxy	Interface to submit request from/to ACROSS platform, mediated by the API Gateway, to the connected service by means of Data connector





<b>Requirements Mapping</b>	List of requirements covered by this component Req_09,Req_13, Req_15, Req_16, Req-17, Req_18, Req_19, Req_29, Req_30													
<b>Interaction with other components</b>	<table border="1"> <thead> <tr> <th style="background-color: #4F81BD; color: white;">Interfacing Component</th> <th style="background-color: #4F81BD; color: white;">Interface Description</th> </tr> </thead> <tbody> <tr> <td>Audit Logger</td> <td>keeps track of operation executed by API adapter</td> </tr> <tr> <td>Semantic Adapter</td> <td>Processes information retrieved by the Data Connector in order to provide requested data. API messages and format are harmonized with common vocabularies.</td> </tr> <tr> <td>Service Registry</td> <td>API instances and descriptions are registered for query and search</td> </tr> <tr> <td>API Gateway</td> <td>Uniform API are mediated by API Gateway</td> </tr> <tr> <td>Identity and Access Management</td> <td>Grants or denies other applications to access an exposed method (authentication and authorization)</td> </tr> </tbody> </table>		Interfacing Component	Interface Description	Audit Logger	keeps track of operation executed by API adapter	Semantic Adapter	Processes information retrieved by the Data Connector in order to provide requested data. API messages and format are harmonized with common vocabularies.	Service Registry	API instances and descriptions are registered for query and search	API Gateway	Uniform API are mediated by API Gateway	Identity and Access Management	Grants or denies other applications to access an exposed method (authentication and authorization)
Interfacing Component	Interface Description													
Audit Logger	keeps track of operation executed by API adapter													
Semantic Adapter	Processes information retrieved by the Data Connector in order to provide requested data. API messages and format are harmonized with common vocabularies.													
Service Registry	API instances and descriptions are registered for query and search													
API Gateway	Uniform API are mediated by API Gateway													
Identity and Access Management	Grants or denies other applications to access an exposed method (authentication and authorization)													

**Table 1: API adapter component card**

#### 4.1.2 Semantic adapter

<b>Component Name</b>	<b>Semantic Adapter</b>
<b>Module Description</b>	<p>Responsible for processing data gathered from Data Connector in order to supply them in the expected format.</p> <p>Semantic adapter component performs data transformation to semantic data from non-semantic data. The component is connected to data connectors to receive non semantic data, according to the models adopted, transform to semantic data and pass on transformed semantic data to upstream components. This component is also responsible of semantic adaptation of service descriptions according to the standard and shared model adopted by the platform and managed in the service catalogue.</p>



<p><b>Main functionalities</b></p>	<p>The component provides the following functionalities:</p> <ul style="list-style-type: none"> <li>• Integration Logic</li> <li>• Normalize and/or process gathered data in order to produce requested information</li> <li>• Data mapping with common data models and vocabularies</li> <li>• Data ingestion with provenance for each data connector</li> <li>• Data transformation rules definition for each source</li> <li>• Data transformation from non-semantic to semantic data</li> <li>• Service description ingestion and mapping</li> </ul>									
<p><b>Main logical Interfaces</b></p>	<table border="1"> <thead> <tr> <th data-bbox="431 783 699 825">Interface name</th> <th data-bbox="708 783 1412 825">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="431 835 699 1024">semantic Adaptation</td> <td data-bbox="708 835 1412 1024">This is the interfaces where other components ask the semantic adapter to transform service descriptions, raw or proprietary data in semantic data according to shared or standard models.</td> </tr> </tbody> </table>		Interface name	Description	semantic Adaptation	This is the interfaces where other components ask the semantic adapter to transform service descriptions, raw or proprietary data in semantic data according to shared or standard models.				
Interface name	Description									
semantic Adaptation	This is the interfaces where other components ask the semantic adapter to transform service descriptions, raw or proprietary data in semantic data according to shared or standard models.									
<p><b>Requirements Mapping</b></p>	<p>List of requirements covered by this component:</p> <p>Req_01, Req_09, Req_13, Req_15, Req_16, Req_17, Req_27, Req_28, Req_29, Req_30</p>									
<p><b>Interaction with other components</b></p>	<table border="1"> <thead> <tr> <th data-bbox="431 1276 833 1318">Interfacing Component</th> <th data-bbox="841 1276 1412 1318">Interface Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="431 1329 833 1413">Data Connector</td> <td data-bbox="841 1329 1412 1413">It receives no structured or semantic data from external services</td> </tr> <tr> <td data-bbox="431 1423 833 1507">Metadata &amp; Vocabulary Catalogue</td> <td data-bbox="841 1423 1412 1507">It retrieves metadata and data mapping for semantic adaptation</td> </tr> <tr> <td data-bbox="431 1518 833 1654">Identity and Access Management</td> <td data-bbox="841 1518 1412 1654">Grants or denies other applications to access an exposed method (authentication and authorization)</td> </tr> </tbody> </table>		Interfacing Component	Interface Description	Data Connector	It receives no structured or semantic data from external services	Metadata & Vocabulary Catalogue	It retrieves metadata and data mapping for semantic adaptation	Identity and Access Management	Grants or denies other applications to access an exposed method (authentication and authorization)
Interfacing Component	Interface Description									
Data Connector	It receives no structured or semantic data from external services									
Metadata & Vocabulary Catalogue	It retrieves metadata and data mapping for semantic adaptation									
Identity and Access Management	Grants or denies other applications to access an exposed method (authentication and authorization)									

**Table 2: Semantic Adapter component card**



### 4.1.3 Data connector

<b>Component Name</b>	<b>Data connector</b>	
<b>Module Description</b>	This component allows the integration of Legacy/Proprietary Systems. They are components, often ad hoc developed, which have the responsibility to correctly query the legacy systems and/or to expose interfaces suitable for receiving information from them (pull/push). Data Connector should also be able to deal with security aspects (authentication and authorization of the channel). This component should be able to access to legacy/proprietary data using several approaches (e.g. read from API, read csv or json file or other files, read from SQL or NoSQL databases, etc.)	
<b>Main functionalities</b>	<p>The component provides the following functionalities:</p> <ul style="list-style-type: none"> <li>• Read data from the legacy/proprietary systems (Public and Private)</li> <li>• Expose an interface to allow legacy/proprietary systems to send data to the Data Connector</li> </ul>	
<b>Main logical Interfaces</b>	<b>Interface name</b>	<b>Description</b>
	receiveData	to receive data from Legacy/Proprietary systems
	sendData	To send data to Legacy/Proprietary systems
<b>Requirements Mapping</b>	<p>List of requirements covered by this component:</p> <p>Req_09, Req_13, Req_15, Req_16, Req-17, Req_18, Req_19, Req_29, Req_30</p>	
<b>Interaction with other components</b>	<b>Interfacing Component</b>	<b>Interface Description</b>
	Data connectors Dashboard	It notifies activity and status log of the instantiated data connector for the specific external service
	Audit Logger	It traces all security logs



<b>Identity and Access Management</b>	<b>Access</b>	Grants or denies other applications to access an exposed method (authentication and authorization)
---------------------------------------	---------------	--

**Table 3: Data Connector component card**

#### 4.1.4 Service Registry

<b>Component Name</b>	<b>Service Registry</b>	
<b>Module Description</b>	This component is one of the modules of Service Catalogue to provide human and machine readable description of services that will be available in ACROSS platform for user journey services provisioning. The registry enables the storage and publishing of service by providing general, technical and data processing information based on standard models (e.g. ISA2).	
<b>Main functionalities</b>	<p>The component provides the following functionalities:</p> <ul style="list-style-type: none"> <li>publishing, searching, and retrieving of an already available service in the platform</li> <li>Service Description versioning</li> <li>API for programmatically interaction with the registry</li> <li>User Dashboard and service editor</li> </ul>	
<b>Main logical Interfaces</b>	<b>Interface name</b>	<b>Description</b>
	Store/delete	Store or delete a service description
	Search	Search a service description according to several metadata in accordance to the adopted service model
	Publish	The service description is active and available and searchable
<b>Requirements Mapping</b>	<p>List of requirements covered by this component</p> <p>Req_01, Req_13, Req_15, Req_16, Req_17, Req_27, Req_28, Req_29, Req_30, Req_35, Req_36.</p>	



Interaction with other components	<b>Interfacing Component</b>	<b>Interface Description</b>
	API Gateway	API instances and descriptions are registered for query and search
	Citizen data ownership	List of available services and related information and data processing information (type of data, legal basis, processing type, purpose)
	Metadata and Vocabulary Catalogue	It gets references for the mapping with the adopted models and vocabulary concepts and terms

Table 4: Service Registry component card

#### 4.1.5 Metadata and Vocabulary Catalogue

<b>Component Name</b>	Metadata and Vocabulary Catalogue	
<b>Module Description</b>	This component is one of the modules of Service Catalogue to support the storage of shared models and vocabularies used to describe public and private service available by means of the platform. The module also allows users to create service and data mappings by means of web interfaces online between the stored metadata and vocabulary and any other data model for describing public and private services and related data processing.	
<b>Main functionalities</b>	<p>The component provides the following functionalities:</p> <ul style="list-style-type: none"> <li>• Basic functions to create, read, update, and delete metadata</li> <li>• Data Model mapping</li> <li>• Service Model mapping</li> </ul>	
<b>Main logical Interfaces</b>	<b>Interface name</b>	<b>Description</b>
	Edit metadata	Edit a new or already existing metadata
	Store vocabulary	Store a new vocabulary
	Edit data model mapping	Create a data map between an external attribute and a referenced metadata



	<table border="1"> <tr> <td>Edit service model mapping</td> <td>Create a service model map between external service description attributes and referenced metadata</td> </tr> </table>	Edit service model mapping	Create a service model map between external service description attributes and referenced metadata				
Edit service model mapping	Create a service model map between external service description attributes and referenced metadata						
<b>Requirements Mapping</b>	List of requirements covered by this component Req_17, Req_28						
<b>Interaction with other components</b>	<table border="1"> <thead> <tr> <th>Interfacing Component</th> <th>Interface Description</th> </tr> </thead> <tbody> <tr> <td>Service Registry</td> <td>It provides metadata references for service data mapping</td> </tr> <tr> <td>Citizen Data Ownership</td> <td>It provides metadata references for service personal data mapping</td> </tr> </tbody> </table>	Interfacing Component	Interface Description	Service Registry	It provides metadata references for service data mapping	Citizen Data Ownership	It provides metadata references for service personal data mapping
Interfacing Component	Interface Description						
Service Registry	It provides metadata references for service data mapping						
Citizen Data Ownership	It provides metadata references for service personal data mapping						

Table 5: Metadata and Vocabulary component card

#### 4.1.6 Data connectors dashboards

<b>Component Name</b>	Data connectors dashboards						
<b>Module Description</b>	This component is one of the modules of Service Catalogue to provide a visual dashboard to manage the status of services connected to the platform by means of a specific data connector instance.						
<b>Main functionalities</b>	<p>The component provides the following functionalities:</p> <ul style="list-style-type: none"> <li>The dashboard gives insights on many aspects of the data connector and how it is performing, including logs, performance metrics, and the status of alerting policies.</li> </ul>						
<b>Main logical Interfaces</b>	<table border="1"> <thead> <tr> <th>Interface name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Logs</td> <td>It provides different view of data connector logs</td> </tr> <tr> <td>Status</td> <td>It provides information about data connector status</td> </tr> </tbody> </table>	Interface name	Description	Logs	It provides different view of data connector logs	Status	It provides information about data connector status
Interface name	Description						
Logs	It provides different view of data connector logs						
Status	It provides information about data connector status						
<b>Requirements Mapping</b>	List of requirements covered by this component						



	Req_11 ,Req_30, Req_36	
Interaction with other components	<b>Interfacing Component</b>	<b>Interface Description</b>
	Data Connector	It provides logs and status of the connector

**Table 6: Data connectors dashboards component card**

## 4.2 Citizen data ownership and usage control

This layer is part of the personal data framework architecture that will be designed in WP3, which will allow citizens / users to register in a series of services (Service Registry) and allow the use of their data based on consents that should be approved by them. To carry out this transfer of information in a secure way, the Usage Control module will be used, which will allow the usage of data based on previously defined usage policies.

### 4.2.1 Citizen Data Ownership

<b>Component Name</b>	<b>Citizen Data Ownership</b>
<b>Module Description</b>	This component allows the citizens to manage their personal data and allows the organizations/services to fulfil the requirements in line with the GDPR. It will expose several interfaces for the Transparency Dashboard, so that the individuals can grant and withdraw their consents and receive notifications about how their data is being used. On the other hand, it will expose several interfaces for the services, so that they can be informed about the consents of the citizens, and they can send notifications about the data that is being used.
<b>Main functionalities</b>	The component provides the following functionalities: <ul style="list-style-type: none"> <li>• Enable the citizens to grant and withdraw their consents for data processing.</li> <li>• Provide to the citizens notifications about their data processing.</li> <li>• Provide to the services information about the consents of the citizens.</li> </ul>



	<ul style="list-style-type: none"> <li>• Enable the services to send the notifications about the usage of the data.</li> </ul>	
<b>Main logical Interfaces</b>	<b>Interface name</b>	<b>Description</b>
	SearchConsent	Search consents by different criteria.
	ModifyConsent	Modify consent status (e.g.: withdraw), enable or disable specific data to which consent applies, change organizations to which data is shared.
	ViewLogs	Show information about the events that have happened related to the linked services and the consents given/withdrawn.
	SearchServices	Search services by different criteria.
	LinkUserToService	Link a user to a service, so that he can manage the consents given to that service.
<b>Requirements Mapping</b>	<p>List of requirements covered by this component</p> <p>Req_02, Req_04, Req_16, Req_17, Req_18, Req_20, Req_21, Req_22, Req_30, Req_31, Req_34</p>	
<b>Interaction with other components</b>	<b>Interfacing Component</b>	<b>Interface Description</b>
	Transparency Dashboard	Read, insert, modify or remove consents of the citizens.  Access notifications of usage of the citizens data.
	Service Registry	Access services related information saved in the registry and link services to citizens.
	Metadata and Vocabulary Catalogue	To map the different data models used by the different services involved in ACROSS.
	Audit Logger	It traces all security logs.





	Identity and Access Management	Grants or denies other applications to access an exposed method (authentication and authorization).
	Usage Control	List of consents established between the citizens and the services.

**Table 7: Citizen Data Ownership component card**

#### 4.2.2 Usage Control

<b>Component Name</b>	<b>Usage Control</b>							
<b>Module Description</b>	The component provides the enforcement mechanism to apply usage policies according to previously defined consents.							
<b>Main functionalities</b>	The main objective is to provide the security and privacy aspects of the data shared among the different systems connected. Existing data usage control mechanisms will be studied and adapted to the platform.							
<b>Main logical Interfaces</b>	<table border="1"> <thead> <tr> <th>Interface name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>UsageControlEnforcement</td> <td>Apply usage policies so that data is used accordingly</td> </tr> </tbody> </table>	Interface name	Description	UsageControlEnforcement	Apply usage policies so that data is used accordingly			
Interface name	Description							
UsageControlEnforcement	Apply usage policies so that data is used accordingly							
<b>Requirements Mapping</b>	List of requirements covered by this component: Req_02, Req_04, Req_16, Req_20, Req_21, Req_22, Req_30, Req_31							
<b>Interaction with other components</b>	<table border="1"> <thead> <tr> <th>Interfacing Component</th> <th>Interface Description</th> </tr> </thead> <tbody> <tr> <td>Audit Logger</td> <td>It traces all security logs</td> </tr> <tr> <td>Identity and Access Management</td> <td>Grants or denies other applications to access an exposed method (authentication and authorization).</td> </tr> </tbody> </table>	Interfacing Component	Interface Description	Audit Logger	It traces all security logs	Identity and Access Management	Grants or denies other applications to access an exposed method (authentication and authorization).	
Interfacing Component	Interface Description							
Audit Logger	It traces all security logs							
Identity and Access Management	Grants or denies other applications to access an exposed method (authentication and authorization).							



	Metadata & Vocabulary Catalogue	It retrieves metadata and data mapping for semantic adaptation
	Citizen Data Ownership	Access the consents established between the citizens and the services.

**Table 8: Usage Control component card**

### 4.2.3 API Gateway

<b>Component Name</b>	<b>API Gateway</b>	
<b>Module Description</b>	This components provides infrastructure commodities as bridge between the internal service invocation and the actual API invocation exposed by the API adapter of each service adapter instance. This components also will support the micro-service approach on deployment of ACROSS component supporting not only the north-south interactions but also providing API management, interacting with the Service Catalogue, and authentication and authorization filtering features by interacting with the Security layer.	
<b>Main functionalities</b>	API Proxy and management	
<b>Main logical Interfaces</b>	<b>Interface name</b>	<b>Description</b>
	proxy	It proxies service API invocation
<b>Requirements Mapping</b>	List of requirements covered by this component: Req_07, Req_11,Req_12, Req_13, Req_14, Req_15, Req_16, Req_17, Req_19, Req__20, Req_30	
<b>Interaction with other components</b>	<b>Interfacing Component</b>	<b>Interface Description</b>
	Audit Logger	It traces all security logs



	Identity and Access Management	Filter any authentication and authorization constraints
	API Adapter	It proxies service invocation to the actual service instance API.
	Service Registry	It gets information about API management

Table 9: API Gateway component card

### 4.3 User Journey Service modeling and delivery

This layer is the connection between the Citizen Front End and the public/private services to be used by the citizens to complete the user journey. It is composed of the User Journey Modelling tool, to be used by the Modelling Expert to model graphically the abstract workflow of the journey (work study, etc.), and the User Journey Services Engine, which will instantiate a selected abstract workflow into a concrete one, according to the Citizen’s destination country, and using the services description provided by the Service Registry.

#### 4.3.1 User Journey Modelling Tool

<b>Component Name</b>	<b>User Journey Modelling Tool (UJMT)</b>
<b>Module Description</b>	This component offers support for User Journey modelling and for implementation of service orchestrations based on User Journey Models (UJM).
<b>Main functionalities</b>	<p>The component comprises:</p> <ol style="list-style-type: none"> <li>1. a graphical <i>front-end tool</i> (UJMT.F) for interactively, in a diagram-like style, <i>modelling</i> central aspects of a User Journey that will directly affect the intended service orchestration. This tool will either be implemented within the project, or a suitable external SaaS offering may be employed. In the latter case, the SaaS UJMT.F tool 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).</li> </ol>



	<p>2. a <i>back-end tool</i> (UJMT.B) that provides support for implementing the intended service orchestration, based on the objects and relationships of the model built in the first step. From a User Journey Model, this tool will create a User Journey Workflow Template (JWT). The JWT is a machine-readable abstract orchestration description which is based on the modelled workflow. It will be provided to the User Journey Service Engine, where it is used to instantiate the concrete User Journey Service Workflows.</p>									
<p><b>Main logical Interfaces</b></p>	<table border="1"> <thead> <tr> <th data-bbox="430 663 721 709">Interface name</th> <th data-bbox="727 663 1414 709">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="430 718 721 806">UJMT.F.WebUI</td> <td data-bbox="727 718 1414 806">Web user interface enabling users to interactively construct and modify User Journey Models</td> </tr> <tr> <td data-bbox="430 814 721 999">UJMT.F.ConfigurationI</td> <td data-bbox="727 814 1414 999">Interface for configuring the User Journey Modelling Tool, e.g. by selection of suitable templates for a particular use-case domain (may be file-based or service-based)</td> </tr> <tr> <td data-bbox="430 1008 721 1096">UJMT.B.ModelinputI</td> <td data-bbox="727 1008 1414 1096">Service Interface for accepting an UJM model representation saved or exported from UJMT.F</td> </tr> </tbody> </table>		Interface name	Description	UJMT.F.WebUI	Web user interface enabling users to interactively construct and modify User Journey Models	UJMT.F.ConfigurationI	Interface for configuring the User Journey Modelling Tool, e.g. by selection of suitable templates for a particular use-case domain (may be file-based or service-based)	UJMT.B.ModelinputI	Service Interface for accepting an UJM model representation saved or exported from UJMT.F
Interface name	Description									
UJMT.F.WebUI	Web user interface enabling users to interactively construct and modify User Journey Models									
UJMT.F.ConfigurationI	Interface for configuring the User Journey Modelling Tool, e.g. by selection of suitable templates for a particular use-case domain (may be file-based or service-based)									
UJMT.B.ModelinputI	Service Interface for accepting an UJM model representation saved or exported from UJMT.F									
<p><b>Requirements Mapping</b></p>	<p>List of requirements covered by this component</p> <p>Req_09, Req_10, Req_15, Req_33, Req_35, Req_36</p>									
<p><b>Interaction with other components</b></p>	<table border="1"> <thead> <tr> <th data-bbox="430 1289 834 1377">Interfacing Component Used Interface</th> <th data-bbox="841 1289 1414 1377">Interface Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="430 1386 834 1717">User Journey Service Engine</td> <td data-bbox="841 1386 1414 1717">The component is able to provide the created JWTs to the User Journey Service Engine. It is also able to update or delete previously provided JWTs. In the User Journey Service Engine, the JWTs are used for the instantiation of the concrete User Journey Service Workflows.</td> </tr> <tr> <td data-bbox="430 1726 834 1860">Service Registry</td> <td data-bbox="841 1726 1414 1860">The component will provide the information for each service, already registered, to be included in the workflow.</td> </tr> </tbody> </table>		Interfacing Component Used Interface	Interface Description	User Journey Service Engine	The component is able to provide the created JWTs to the User Journey Service Engine. It is also able to update or delete previously provided JWTs. In the User Journey Service Engine, the JWTs are used for the instantiation of the concrete User Journey Service Workflows.	Service Registry	The component will provide the information for each service, already registered, to be included in the workflow.		
Interfacing Component Used Interface	Interface Description									
User Journey Service Engine	The component is able to provide the created JWTs to the User Journey Service Engine. It is also able to update or delete previously provided JWTs. In the User Journey Service Engine, the JWTs are used for the instantiation of the concrete User Journey Service Workflows.									
Service Registry	The component will provide the information for each service, already registered, to be included in the workflow.									



**Table 10: User Journey Modelling Tool component card**

#### 4.3.2 User Journey Services Engine

<b>Component Name</b>	<b>User Journey Services Engine</b>
<b>Module Description</b>	<p>This component has two main responsibilities:</p> <ol style="list-style-type: none"><li>1. It manages machine-readable orchestration descriptions from the User Journey Modelling Tool and instantiates them into concrete User Journey Service Workflows (which include the concrete executable services) to the frontend components and thus the citizens. This instantiation will be carried out when the user specifies the abstract workflow (e.g.: work abroad, study abroad) he wants to carry out and the city/country from/to where to travel.</li><li>2. It performs the service orchestration and thus coordinates the communication with the API-Gateway that is necessary for the User Journey Service execution. The orchestration might include processing of external events.</li></ol>
<b>Main functionalities</b>	<p>The component provides the following functionalities:</p> <ol style="list-style-type: none"><li>1. It is able to receive and manage orchestration descriptions (abstract workflow templates) from the User Journey Modelling Tool.</li><li>2. It is able to provide an overview of the available User Journey Services.</li><li>3. Following a request from a citizen, it orchestrates the necessary service calls by interpreting the according service orchestration description and coordinating the communication with the API-Gateway.</li><li>4. It is responsible for intermediating user interface requests bidirectionally between the user interface components and the individual services executed within the concrete workflow's orchestration.</li><li>5. This intermediation is based on information received from the service registry.</li></ol>



<b>Main logical Interfaces</b>	<b>Interface name</b>	<b>Description</b>
	Manage workflow templates	Store, modify, delete workflow templates
	Get available workflow templates	Get the available workflow templates to instantiate.
	Instantiate workflow template	According to a workflow template and a destination country, provided as input parameters to the interface, translate workflow steps into concrete services.
	Steer workflow	Interface for interfacing from the UI to the User Journey Services Engine: start a workflow, kill a workflow, inform that a concrete workflow step has been completed offline (e.g.: by phone, etc.), etc..
	Monitor workflow status	Get status of workflow: before start, running at step xyz, completed, exited successfully, exited unsuccessfully at step xyz, etc.
<b>Requirements Mapping</b>	List of requirements covered by this component Req_09, Req_10, Req_15, Req_19, Req_29, Req_30	
<b>Interaction with other components</b>	<b>Interfacing Component</b>	<b>Interface Description</b>
	User Journey Modelling Tool	The component is able to receive and manage orchestration descriptions from the UJM backend tool.
	API Gateway	The component intermediates between user interface and individual services via the API Gateway.
	Web Application	The component is able to provide to the Web Application the available abstract workflows and the available User Journey Services. It is also able to send and receive user requests to and from the Web Application.
	Mobile Application	The component is able to provide to the Mobile Application the available abstract



		workflows and the available User Journey Services. It is also able to send and receive user requests to and from the Mobile Application.
	Service Registry	The component receives from the Service Catalogue (via service registry component) the necessary information for the invocation of service instance.
	Usage Control	Whenever the citizen inputs data for a service, before transferring it to that service, this data must be processed by the Usage Control component to enforce previously defined usage policies.

**Table 11: User Journey Services Engine component card**

#### 4.4 Citizen Front-End

This layer provides all the ACROSS User Interfaces towards citizens. The main interactions of the Citizen Front End layer are with the User Journey Service delivery layer and the Citizen data ownership and usage control layer.

Citizen Front-End consists of 4 distinct components:

- Mobile Application
- Web Application
- Transparency Dashboard
- Virtual Assistant

The key features of this layer are:

- Secure connection to Citizen data ownership layer so the citizens can handle their personal data and the way they are used
- Secure connection to the Web/Native application to access and invoke the delivered services of the User Journey
- Secure access to the Virtual assistant to improve the overall user experience.



#### 4.4.1 Mobile Application

<b>Component Name</b>	<b>Mobile Application</b>											
<b>Module Description</b>	<p>A native application for one-stop-shop service, enabling users to better understand, complete and seamless information exchange within the whole cross-border digital service and the involved stakeholders. The application enhances the process and tackles the difficulty of finding the correct digital public service and the sequence of the steps that need to be taken.</p> <p>The Mobile Native Application will be developed with the official SDKs and Tools provided by Apple and Google for iOS and Android respectively. Standard libraries and frameworks, like Android Jetpack, will be employed to deliver high-quality applications.</p>											
<b>Main functionalities</b>	<p>The component provides the following functionalities:</p> <ul style="list-style-type: none"> <li>• Access to the User Journey Services.</li> <li>• Visualization of the services orchestration of the User Journey for moving.</li> <li>• Virtual assistance in the User Journey Experience.</li> </ul>											
<b>Main logical Interfaces</b>	<table border="1"> <thead> <tr> <th data-bbox="422 1218 693 1276">Interface name</th> <th data-bbox="699 1218 1414 1276">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="422 1276 693 1415">User Authentication</td> <td data-bbox="699 1276 1414 1415">This is a REST based interface that authenticates the user through the Identity and Access Management component of the Security layer</td> </tr> <tr> <td data-bbox="422 1415 693 1514">Virtual Assistant Invocation</td> <td data-bbox="699 1415 1414 1514">This is a REST based interface that invokes the Virtual Assistant Service.</td> </tr> <tr> <td data-bbox="422 1514 693 1652">Citizen Data Management</td> <td data-bbox="699 1514 1414 1652">This is a REST based interface that allows citizens to define and store their data to the Citizen Data Ownership component through the Transparency dashboard.</td> </tr> <tr> <td data-bbox="422 1652 693 1749">User Journey Service Invocation</td> <td data-bbox="699 1652 1414 1749">This is a REST based interface to expose access the User Journey Services.</td> </tr> </tbody> </table>	Interface name	Description	User Authentication	This is a REST based interface that authenticates the user through the Identity and Access Management component of the Security layer	Virtual Assistant Invocation	This is a REST based interface that invokes the Virtual Assistant Service.	Citizen Data Management	This is a REST based interface that allows citizens to define and store their data to the Citizen Data Ownership component through the Transparency dashboard.	User Journey Service Invocation	This is a REST based interface to expose access the User Journey Services.	
	Interface name	Description										
	User Authentication	This is a REST based interface that authenticates the user through the Identity and Access Management component of the Security layer										
	Virtual Assistant Invocation	This is a REST based interface that invokes the Virtual Assistant Service.										
	Citizen Data Management	This is a REST based interface that allows citizens to define and store their data to the Citizen Data Ownership component through the Transparency dashboard.										
User Journey Service Invocation	This is a REST based interface to expose access the User Journey Services.											





<b>Requirements Mapping</b>	List of requirements covered by this component Req_06, Req_09, Req_10, Req_35, Req_36.	
<b>Interaction with other components</b>	<b>Interfacing Component</b>	<b>Interface Description</b>
	User Journey Service Engine	This interface enables the interaction of the users with the cross-border journey services provided by ACROSS.
	Identity and Access Management	This interface authenticates and authorizes the user to access the web application.
	Virtual Assistant	This interface enables the user to access speech and textual chat interfaces improving the overall accessibility of the existing integrated into ACROSS public services through the web application.
	Citizen Data Ownership	This interface enables the interactions with the citizen data ownership component through which the user can define her personal data.

**Table 12: Mobile Application component card**

#### 4.4.2 Web Application

<b>Component Name</b>	<b>Web Application</b>
<b>Module Description</b>	<p>A web application (Android and iOS) for one-stop-shop service, enabling users to better understand, complete and seamless information exchange within the whole cross-border digital service and the involved stakeholders. The application enhances the process and tackles the difficulty of finding the correct digital public service and the sequence of the steps that need to be taken.</p> <p>The Web Application will make use of the overall JavaScript framework. JavaScript is used as a basis for many different solution frameworks popular to implement Single Page Applications and build beautiful user interfaces. The web application will</p>



	<p>be implemented with a mobile-first approach and will follow the guidelines of material design.</p>											
<p><b>Main functionalities</b></p>	<p>The component provides the following functionalities:</p> <ul style="list-style-type: none"> <li>• Access to the User Journey Services.</li> <li>• Orchestration and visualization of the User Journey for moving.</li> <li>• Virtual assistance in the User Journey Experience.</li> </ul>											
<p><b>Main logical Interfaces</b></p>	<table border="1"> <thead> <tr> <th data-bbox="423 865 699 919">Interface name</th> <th data-bbox="699 865 1421 919">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="423 919 699 1062">User Authentication</td> <td data-bbox="699 919 1421 1062">This is a REST based interface that authenticates the user through the Identity and Access Management component of the Security layer</td> </tr> <tr> <td data-bbox="423 1062 699 1157">Virtual Assistant Invocation</td> <td data-bbox="699 1062 1421 1157">This is a REST based interface that invokes the Virtual Assistant Service.</td> </tr> <tr> <td data-bbox="423 1157 699 1304">Citizen Data Management</td> <td data-bbox="699 1157 1421 1304">This is a REST based interface that allows citizens to define and store their data to the Citizen Data Ownership component through the Transparency dashboard.</td> </tr> <tr> <td data-bbox="423 1304 699 1398">User Journey Service Invocation</td> <td data-bbox="699 1304 1421 1398">This is a REST based interface to expose access the User Journey Services.</td> </tr> </tbody> </table>		Interface name	Description	User Authentication	This is a REST based interface that authenticates the user through the Identity and Access Management component of the Security layer	Virtual Assistant Invocation	This is a REST based interface that invokes the Virtual Assistant Service.	Citizen Data Management	This is a REST based interface that allows citizens to define and store their data to the Citizen Data Ownership component through the Transparency dashboard.	User Journey Service Invocation	This is a REST based interface to expose access the User Journey Services.
Interface name	Description											
User Authentication	This is a REST based interface that authenticates the user through the Identity and Access Management component of the Security layer											
Virtual Assistant Invocation	This is a REST based interface that invokes the Virtual Assistant Service.											
Citizen Data Management	This is a REST based interface that allows citizens to define and store their data to the Citizen Data Ownership component through the Transparency dashboard.											
User Journey Service Invocation	This is a REST based interface to expose access the User Journey Services.											
<p><b>Requirements Mapping</b></p>	<p>List of requirements covered by this component</p> <p>Req_05, Req_09, Req_10, Req_35 ,36.</p>											
<p><b>Interaction with other components</b></p>	<table border="1"> <thead> <tr> <th data-bbox="423 1585 834 1640">Interfacing Component</th> <th data-bbox="834 1585 1421 1640">Interface Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="423 1640 834 1787">User Journey Service Engine</td> <td data-bbox="834 1640 1421 1787">This interface enables the interaction of the users with the cross-border journey services provided by ACROSS.</td> </tr> <tr> <td data-bbox="423 1787 834 1881">Identity and Access Management</td> <td data-bbox="834 1787 1421 1881">This interface authenticates and authorizes the user to access the web application.</td> </tr> </tbody> </table>		Interfacing Component	Interface Description	User Journey Service Engine	This interface enables the interaction of the users with the cross-border journey services provided by ACROSS.	Identity and Access Management	This interface authenticates and authorizes the user to access the web application.				
Interfacing Component	Interface Description											
User Journey Service Engine	This interface enables the interaction of the users with the cross-border journey services provided by ACROSS.											
Identity and Access Management	This interface authenticates and authorizes the user to access the web application.											



Component Card	Virtual Assistant	This interface enables the user to access speech and textual chat interfaces improving the overall accessibility of the existing integrated into ACROSS public services through the web application.
	Citizen Data Ownership	This interface enables the interactions with the citizen data ownership component through which the user can define her personal data.

**Table 13: Web Application component card**

#### 4.4.3 Virtual Assistant

Component Name	<b>Virtual Assistant</b>
Module Description	This component offers conversational (chat and voice) user interfaces to the Mobile Application ( <b>Error! Unknown switch argument.</b> ) and Web Application ( <b>Error! Unknown switch argument.</b> ).
Main functionalities	<p>The component provides two additional <i>conversational</i> interaction modes (chat and voice) to the Mobile &amp; Web Applications’ UIs, in addition to the traditional interaction mode of using screen, keyboard and pointing device for navigating through a (Web) UI and filling out its input elements. Towards this end, it supports the following functionalities:</p> <ul style="list-style-type: none"> <li>- Textual chat-bot user interface</li> <li>- Maintaining an interaction model of the Application’s UI</li> <li>- Conversational UI navigation support based on the interaction model</li> <li>- Conversational support for executing individual interactions (maintaining and perusing a database for obtaining textual “system questions” corresponding to the individual interaction elements; intent analysis and recognition of input values within the textual user responses)</li> <li>- Support for multi-lingual operation, for selected languages corresponding to the use-cases.</li> </ul>



	<ul style="list-style-type: none"> <li>- For selected languages: Acoustic speech user interfaces, converting speech to text (STT) and text to speech (TTS), so the above textual conversation support can be applied to acoustic speech as well.</li> </ul>	
Main logical Interfaces	<b>Interface name</b>	<b>Description</b>
	TextIO.Stream	Bi-directional text streaming of chat messages
	VoiceIO.Stream	Bi-directional audio streaming of voice utterances
	ControlIO.Stream	Bi-directional streaming of control commands
		<b>NOTE:</b> The above three logical interfaces can be collapsed into a single physical streaming connection between the Application (Web)UI and the Virtual Assistant, over which the different message packets (associated to the logical interfaces) are then multiplexed.
Requirements Mapping	List of requirements covered by this component  Req_03, Req_07, Req_10, Req_12, Req_22, Req_33, Req_35, Req_36	
Interaction with other components	<b>Interfacing Component Used Interface</b>	<b>Interface Description</b>
	Web Application, Mobile Application TextIO.Stream	Virtual Assistant <ul style="list-style-type: none"> <li>- receives user's chat input from the Web/Mobile Applications' (Web)UIs</li> <li>- sends its chat responses to the Web/Mobile Applications' (Web)UIs, and thus, to the user</li> </ul>
	Web Application, Mobile Application VoiceIO.Stream	Virtual Assistant <ul style="list-style-type: none"> <li>- receives user's voice input from the microphone...</li> <li>- sends its synthetic voice output to the speaker of the mobile device/browser on/in which the Web/Mobile Application is running</li> </ul>



	<p>Web Application, Mobile Application ControlIO.Stream</p>	<p>Web/Mobile Application UI sends control commands to the Virtual Assistant for:</p> <ul style="list-style-type: none"> <li>- starting an elementary conversational interaction</li> <li>- aborting an ongoing conversational interaction</li> </ul> <p>Virtual Assistant sends control commands to Web/Mobile Application UI for:</p> <ul style="list-style-type: none"> <li>- activating and deactivating voice recording and streaming (through VoiceIO.Stream) within the Web/Mobile Application</li> <li>- having the Web/Mobile Application UI navigate to a particular interaction element</li> <li>- inserting/rendering a value received conversationally for an interaction element in the Web/Mobile Application UI</li> </ul>
--	---	--

**Table 14: Virtual Assistant component card**

#### 4.4.4 Transparency Dashboard

<b>Component Name</b>	<b>Transparency Dashboard</b>
<b>Module Description</b>	A web application that uses a human centric approach to liberate the potential of personal data and to facilitate its controlled flow from multiple data sources to applications and services. Citizens must be able to opt-in and out from the use of their personal data, in line with the requirements of the GDPR.



<p>Main functionalities</p>	<p>The main objective is to give the individual control of their own data.</p> <p>The component provides the following functionalities:</p> <ul style="list-style-type: none"> <li>• Monitor which data are available and how they are used or how it can be accessed. It contains individual’s linked services, and data use related policies and consents.</li> <li>• Users receive notifications about realised data processing at services.</li> <li>• Give users control over their data allowing them to add as well as delete or modify information.</li> </ul>													
<p>Main logical Interfaces</p>	<table border="1"> <thead> <tr> <th data-bbox="428 747 704 800">Interface name</th> <th data-bbox="704 747 1417 800">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="428 800 704 852">SearchConsent</td> <td data-bbox="704 800 1417 852">Search consents by different criteria.</td> </tr> <tr> <td data-bbox="428 852 704 999">ModifyConsent</td> <td data-bbox="704 852 1417 999">Modify consent status (e.g.: withdraw), enable or disable specific data to which consent applies, change organizations to which data is shared.</td> </tr> <tr> <td data-bbox="428 999 704 1146">ViewLogs</td> <td data-bbox="704 999 1417 1146">Show information about the events that have happened related to the linked services and the consents given/withdrawn.</td> </tr> <tr> <td data-bbox="428 1146 704 1199">SearchServices</td> <td data-bbox="704 1146 1417 1199">Search services by different criteria.</td> </tr> <tr> <td data-bbox="428 1199 704 1293">LinkUserToService</td> <td data-bbox="704 1199 1417 1293">Link a user to a service, so that he can manage the consents given to that service.</td> </tr> </tbody> </table>		Interface name	Description	SearchConsent	Search consents by different criteria.	ModifyConsent	Modify consent status (e.g.: withdraw), enable or disable specific data to which consent applies, change organizations to which data is shared.	ViewLogs	Show information about the events that have happened related to the linked services and the consents given/withdrawn.	SearchServices	Search services by different criteria.	LinkUserToService	Link a user to a service, so that he can manage the consents given to that service.
Interface name	Description													
SearchConsent	Search consents by different criteria.													
ModifyConsent	Modify consent status (e.g.: withdraw), enable or disable specific data to which consent applies, change organizations to which data is shared.													
ViewLogs	Show information about the events that have happened related to the linked services and the consents given/withdrawn.													
SearchServices	Search services by different criteria.													
LinkUserToService	Link a user to a service, so that he can manage the consents given to that service.													
<p>Requirements Mapping</p>	<p>List of requirements covered by this component:</p> <p>Req_4, Req_31, Req_35, Req_36</p>													
<p>Interaction with other components</p>	<table border="1"> <thead> <tr> <th data-bbox="428 1482 834 1545">Interfacing Component</th> <th data-bbox="834 1482 1417 1545">Interface Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="428 1545 834 1619">Audit Logger</td> <td data-bbox="834 1545 1417 1619">It traces all security logs</td> </tr> <tr> <td data-bbox="428 1619 834 1787">Identity and Access Management</td> <td data-bbox="834 1619 1417 1787">Grants or denies other applications to access an exposed method (authentication and authorization)</td> </tr> </tbody> </table>		Interfacing Component	Interface Description	Audit Logger	It traces all security logs	Identity and Access Management	Grants or denies other applications to access an exposed method (authentication and authorization)						
Interfacing Component	Interface Description													
Audit Logger	It traces all security logs													
Identity and Access Management	Grants or denies other applications to access an exposed method (authentication and authorization)													



	Citizen Data Ownership	Manages control over the data, being able to establish consents on them as well as modify or revoke them
--	------------------------	--

**Table 15: Transparency Dashboard component card**

## 4.5 Security

This layer provides all the security features needed for a citizen to be authenticated and authorized for using a national service, against their country’s eIDAS node. Besides, this layer provides all security features (authentication, authorization, logging) during the interactions among all components of ACROSS platform.

Main component will be the Identity and access management component, which will act as a middleware or proxy, to allow said services, be it public or private, to receive identification information of the individual and authorize them to use the services.

The component will interface to the services on one side and on the other side to the eIDAS network of nodes.

### 4.5.1 Identity and access management

<b>Component Name</b>	<b>Identity and access management (IAM)</b>
<b>Module Description</b>	The module will provide authentication against the eIDAS node for the citizen and authorization to use the services provided by PAs and Private Providers to the ACROSS platform. Moreover, authorization to use the backend components will be granted. It provides REST based authentication and Identity Management and role based authorization.
<b>Main functionalities</b>	<p>The component provides the following functionalities:</p> <ul style="list-style-type: none"> <li>• Secure and private authentication</li> <li>• Authorization and Trust management</li> <li>• Personal data ownership</li> <li>• SSO</li> <li>• Role based authorization</li> </ul>



<b>Main logical Interfaces</b>	<b>Interface name</b>	<b>Description</b>
	REST interface	All interactions with other modules and/or applications are implemented through REST mechanisms
<b>Requirements Mapping</b>	List of requirements covered by this component Req_02, Req_16, Req_18, Req_20, Req_22, Req_23-27, Req_30, Req_32	
<b>Interaction with other components</b>	<b>Interfacing Component</b>	<b>Interface Description</b>
	[External]EIDAS or other SSO connector	Performs authorization against eIDAS or other eID provider
	WEB Application	Provides the WEB application with the authorization confirmation of the user
	Mobile Application	Same
	Transparency dashboard	Confirms authorization
	Audit Logger	Logs authentication events
	Citizen Data Ownership	Provides authorization

**Table 16: Identity and access management component card**

#### 4.5.2 Audit Logger

<b>Component Name</b>	<b>Audit Logger</b>
<b>Module Description</b>	<p>This component is in charge of tracking several events occurred among the components of all the layers of the platform, such as the usage of data during data exchange transactions and the enforcement of usage restrictions, providing evidence of compliant data usage: how and by whom data was accessed or processed.</p> <p>This component can work either with event-based (centralized) or flow-interception (distributed) way of logging data exchange transactions.</p>
<b>Main functionalities</b>	Centralized or distributed auditing of data exchange transactions by either receiving events or actively intercept data flows.





	Query and Filtering services of audited data.	
Main Logical Interfaces	<b>Interface name</b>	<b>Description</b>
	Get Event Log	Query and Filtering event logs
	Post Event Log	Store specific event/s
Requirements Mapping	List of requirements covered by this component: Req_02, Req_16, Req_18, Req_20, Req_22, Req_23-27, Req_30, Req_32	
Interaction with other components	<b>Interfacing Component</b>	<b>Interface Description</b>
	Data Connectors Dashboards	Get/register events
	Transparency Dashboard	Get/register events
	User Journey Service Engine	Get/register events
	Citizen data ownership and usage control layer	Register events
	Service Adapter	Register events
	IAM	Register events

Table 17: Audit Logger component card

#### 4.5.3 Data Access Control

<b>Component Name</b>	<b>Data Access Control</b>
<b>Module Description</b>	This component ensures further authorization enforcement restricting unauthorized access to data resources, by relying on Role Based and/or Attribute Based Access Control models (RBAC, ABAC). Once the access token has been validated by the Authorization Server (IAM), this component then can perform Access Control decisions on requested data, also relying on policy languages such as XACML (Extensible Access Control Markup Language) or ODRL (Open Digital



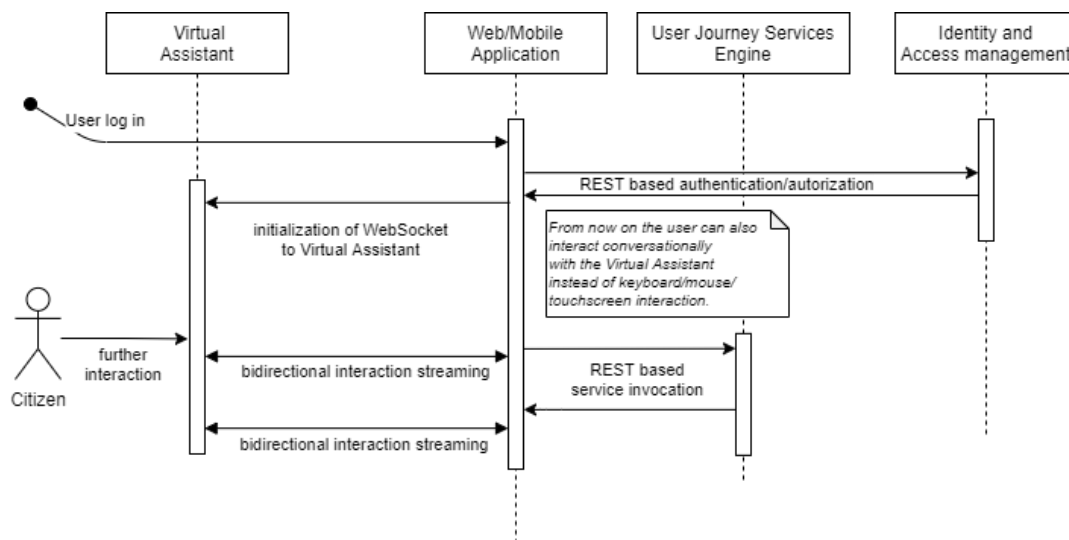
	Rights Language). This component will interact with the API Gateway intercepting incoming data requests and it will grant access on the decisions made by the Usage Control component.											
Main functionalities	Role-based access control (RBAC) Attribute-based access control (ABAC) Policy Enforcement Point. Policy Decision Point											
Main logical Interfaces	<table border="1"> <thead> <tr> <th>Interface name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Grant data access</td> <td>Grant access to a specific resource against specific access and usage policy and context</td> </tr> <tr> <td></td> <td></td> </tr> </tbody> </table>		Interface name	Description	Grant data access	Grant access to a specific resource against specific access and usage policy and context						
Interface name	Description											
Grant data access	Grant access to a specific resource against specific access and usage policy and context											
Requirements Mapping	List of requirements covered by this component: Req_02, Req_16, Req_18, Req_20, Req_22, Req_23-27, Req_30, Req_32											
Interaction with other components	<table border="1"> <thead> <tr> <th>Interfacing Component</th> <th>Interface Description</th> </tr> </thead> <tbody> <tr> <td>IAM</td> <td>Validate Access Token</td> </tr> <tr> <td>Data Usage Control</td> <td>Verify usage rules</td> </tr> <tr> <td>Audit Logger</td> <td>Store access events</td> </tr> <tr> <td>API Gateway</td> <td>Perform access enforcement</td> </tr> </tbody> </table>		Interfacing Component	Interface Description	IAM	Validate Access Token	Data Usage Control	Verify usage rules	Audit Logger	Store access events	API Gateway	Perform access enforcement
Interfacing Component	Interface Description											
IAM	Validate Access Token											
Data Usage Control	Verify usage rules											
Audit Logger	Store access events											
API Gateway	Perform access enforcement											

Table 18: Data Access Control component card

## 5 Interaction Scenarios

The main high-level interaction diagram of Citizen Front End layer can be summarized in the flow diagrams below:

### 5.1 Citizen Front-End: Web/Mobile Applications and Virtual Assistant



**Figure 3: Web/Mobile Applications and Virtual Assistant interaction flow**

The main interaction scenarios are:

- Authentication/ Authorization of the user. This interaction scenario invokes the User authentication interface towards Identity and Access Management.
- User Journey Service invocation. This scenario includes the access of the journey services through the provided open APIs to implement the User Journey move scenario.
- Virtual Assistant Service invocation. This interaction invokes the Virtual Assistance Invocation interface towards the Virtual Assistant Component.

### 5.2 User Journey Service modelling and delivery

The main high level interaction diagram of the User Journey Service modelling and delivery layer can be summarized in the image below:

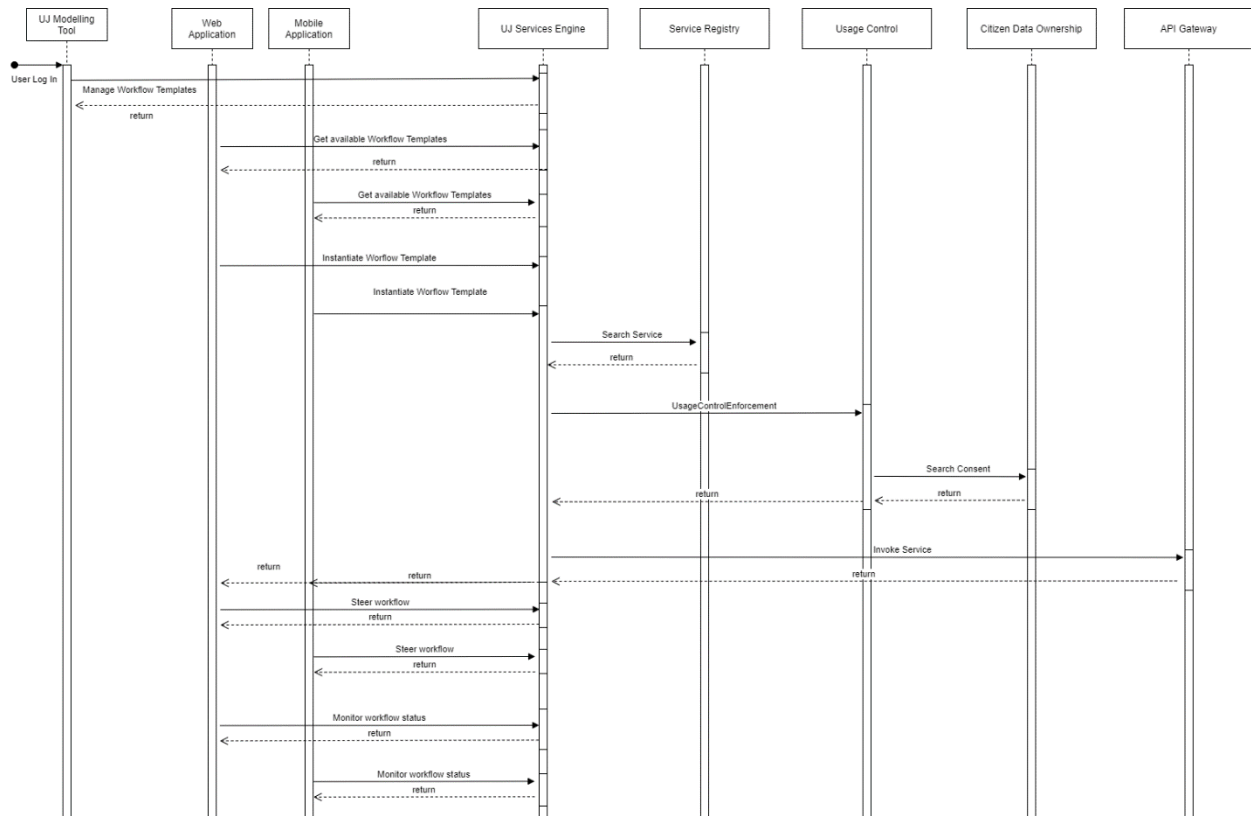


Figure 4: User Journey modelling and delivery interaction flow

The main interaction scenarios are:

- User Journey Service Invocation from the following modules:
  - o User Journey Modelling Tool: This module will invoke the User Journey Services Engine to manage the workflow templates.
  - o Web Application and Mobile Application: these modules will invoke the User Journey Services Engine:
    - To get the available workflows templates to instantiate
    - to instantiate every workflow template
    - To steer the workflow, that is, to start it, kill it, inform that a workflow step has been completed offline (e.g.: by phone, etc.), etc.
    - To monitor the workflow status (running, finished, etc.)
- Service Registry invocation from the User Journey Services Engine to translate the abstract services in the workflow template into concrete services.

- Usage Control invocation from the User Journey Services Engine to apply usage policies so that data is used accordingly. The Usage Control component will invoke the Citizen Data Ownership component to get the consents established between the citizens and the services.
- API Gateway invocation from the User Journey Services Engine to reach and invoke the individual services.

### 5.3 Citizen Ownership and Data Usage Control

The main high-level interaction diagram of Citizen Ownership and Data Usage Control layer can be summarized in the image below:

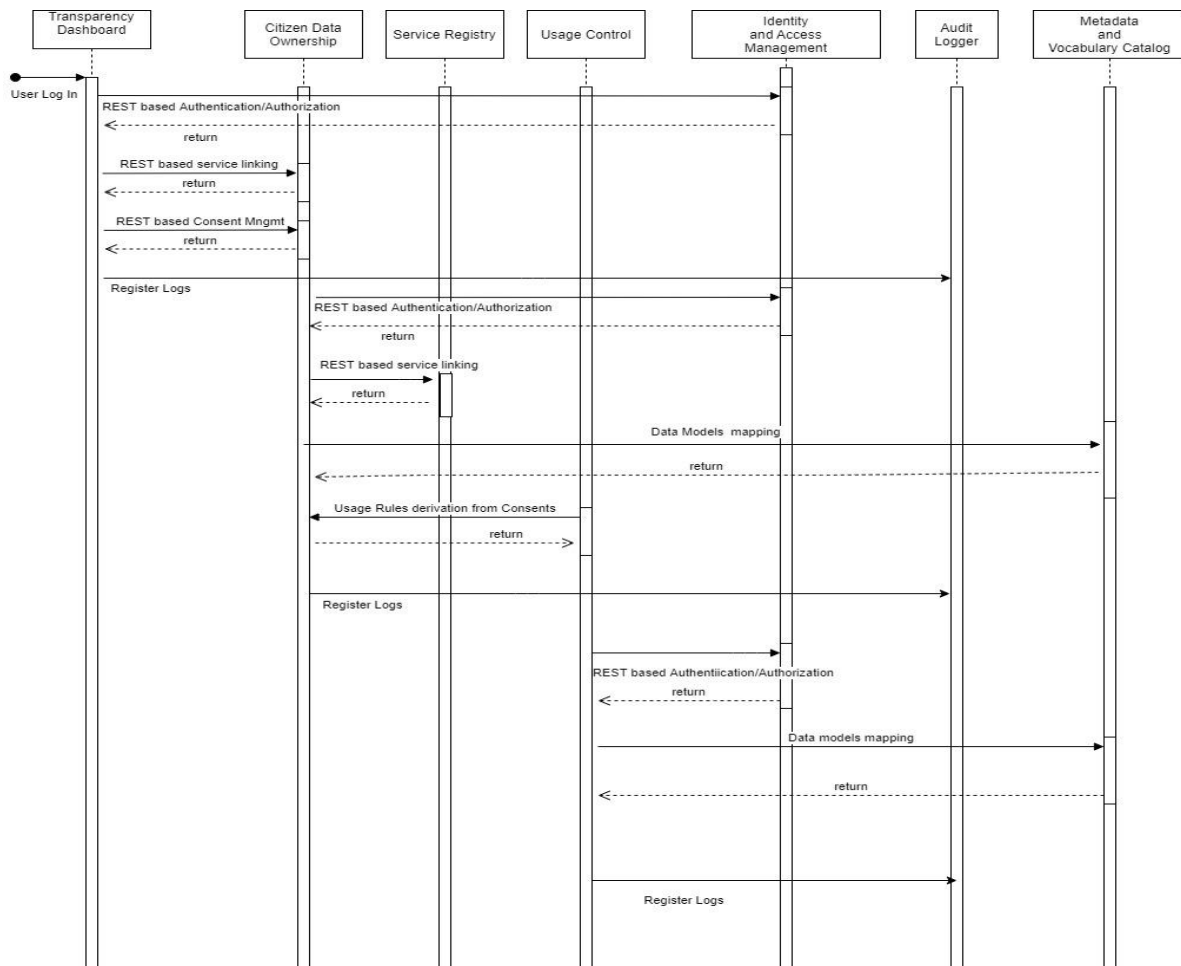


Figure 5: Citizen Ownership and Data usage control interaction flow



The main interaction scenarios are:

- Transparency Dashboard invocation. This scenario includes the functionalities that gives to individual control of their own data.
- Citizen Data Ownership invocation. Expose several APIs for the Transparency Dashboard so that the individuals can grant and withdraw their consents and receive notifications about how their data is being used
- Service Registry invocation provide human and machine-readable description of services that will be available in ACROSS platform for user journey services provisioning.
- Usage Control invocation. This scenario provides the enforcement mechanism to apply usage policies according to previously defined consents.
- Identity and Access Management. This interaction scenario includes the interface with the Authentication node (eIDAS, SSO)
- Audit Logger invocation. It traces all security logs.
- Metadata and Vocabulary Catalogue invocation. Expose several APIs to support the storage of shared models and vocabularies used to describe public and private service available by means of the platform and allows users to create data mappings.

## 5.4 Data harmonization and Connectors

The main high-level interaction diagram of Data harmonization and Connectors can be summarized in the two flow diagrams below:

### 1. Service description adaptation

This first flow describes the scenarios of interaction among the components in order to provide functionalities for the provisioning of a common information model of the service registered to the platform.

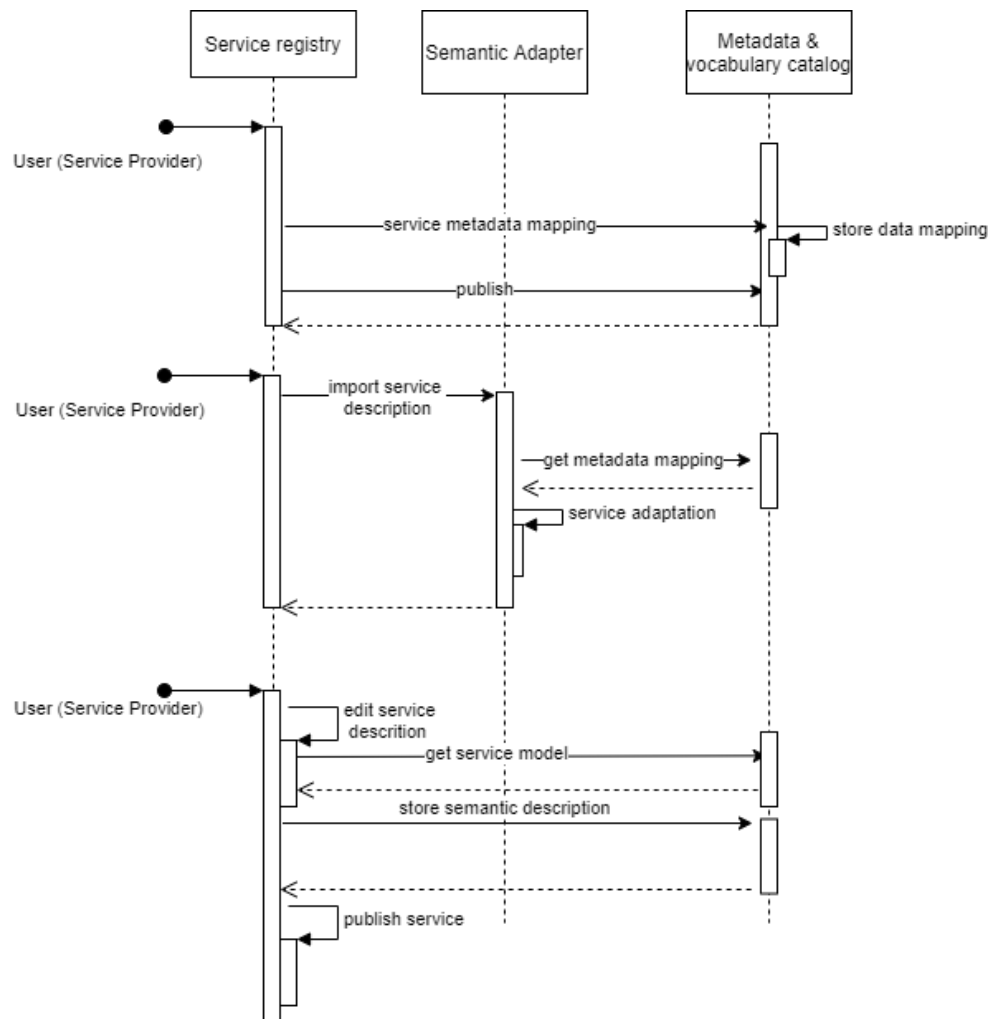


Figure 6: Service description interaction flow

The main interaction scenarios are:

- Service Provider user, responsible to register a new service in ACROSS platform access to Service Registry to model a new service or import & adapt and external description.
- Service Registry interacts with Service Adapter to define the metadata mapping between the service model adopted in ACROSS platform and the external one.
- Metadata & Vocabulary Catalogue is invoked to store and retrieve the semantic data mapping and the produced informational models

## 2. Service/ Data source registration and connection

This second flow describe the interaction of ACROSS platform with an external service supported by the definition of a data connector and related data model adaptation.

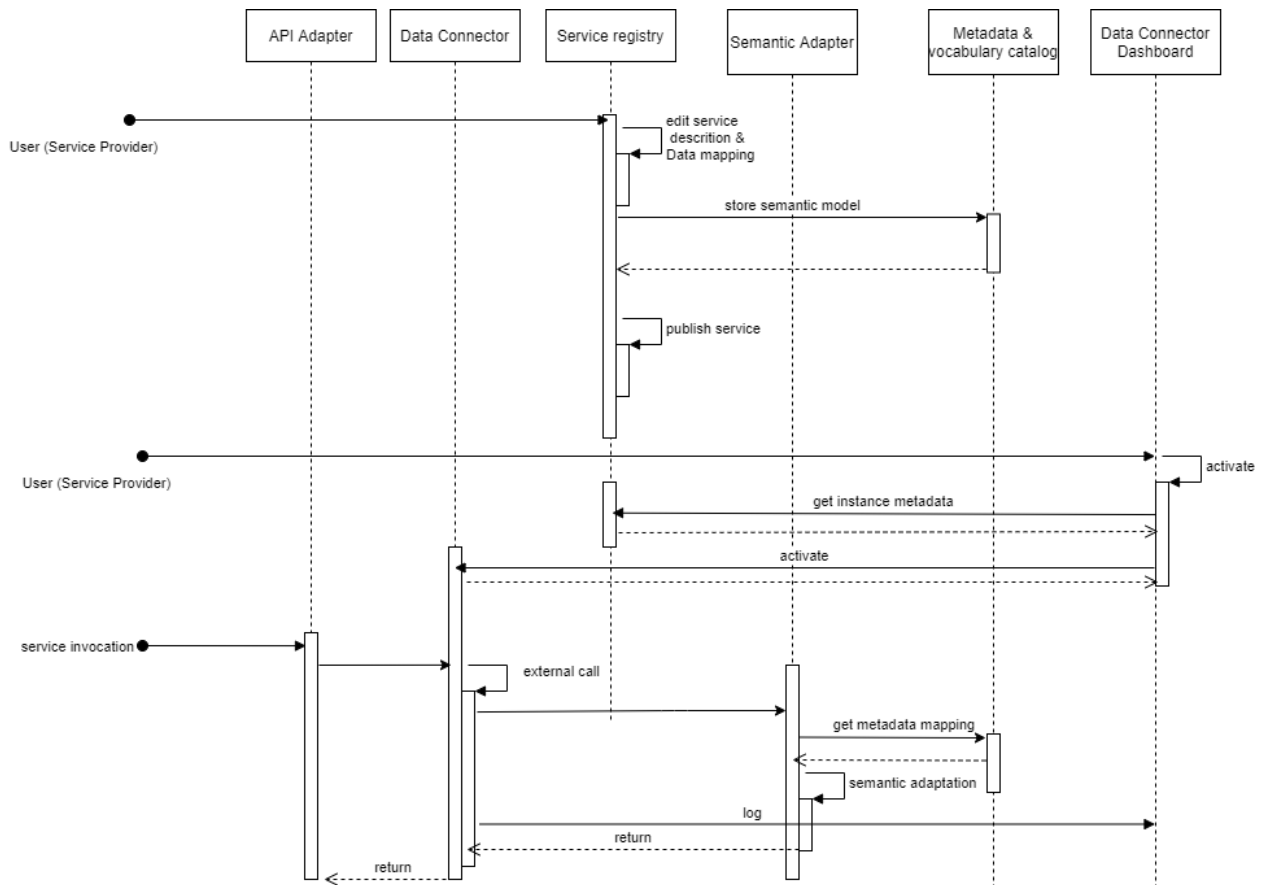


Figure 7: Service registration and connection interaction flow

The main interaction scenarios are:

- Service Provider register the new service to invoke describing it and defining a data mapping. All this information is stored in metadata and vocabulary catalogue
- Once service is described and registered, the service provider activates the instance of data connector that will actually interact with external service.
- Each service invocation from the upper layers is mediated and adapted by means of API adapter that will interact with data connector
- Data Connector will perform external service invocation and it will interact with the Semantic Adapter to perform the semantic adaptation in accordance to the define data mapping.



## 6 Deployment scenarios and implementation guidelines

### 6.1 Deployment and implementation approach

As described in the requirements, a pipeline of processes will be set up to ensure the successful integration, deployment, testing and delivery of the services. The DevOps processes, development and operations, will be integrated into a single entity with common goals: high-quality software, faster releases and improved users' satisfaction. Hence, we can leverage the value of DevOps tools to ensure Continuous Integration and Continuous Deployment.

Initially, Gitlab will be used as the code versioning tool. The relevant repositories will be set up to facilitate the components. In addition, Gitlab offers the CI feature that will trigger the automatic build and deployment of components.

All components and applications will be deployed on a testbed server under 2 environments: staging and production. After the successful deployment and testing activities on staging the code can be deployed live, thus assuring the quality of the services and the platform.

The above-mentioned methodology can be summarized in the image below:

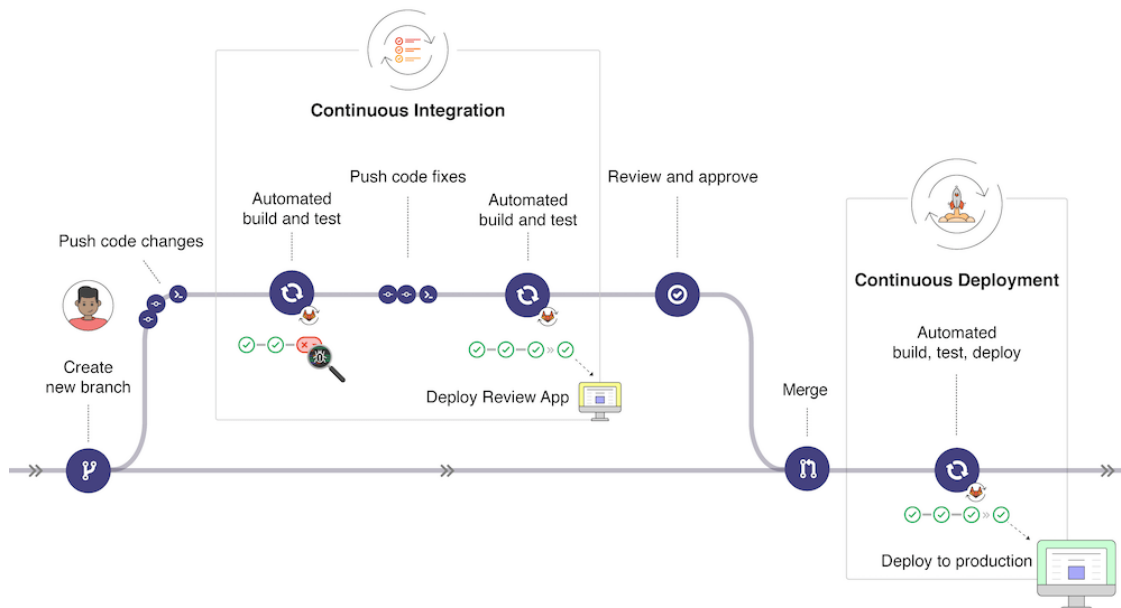


Figure 8: Continuous Integration / Continuous Delivery Overview

## 6.2 Deployment schemas

For the purposes of defining the overall deployment schema, Docker Registry along with the Kubernetes orchestration framework shall be used. During the initial stages of the implementation and deployment Kubernetes kind could also be used to minimize the required resources and achieve time efficiency. Kind is a tool for running local Kubernetes clusters using Docker container “nodes”. Subsequently, since the deployment schema is customized to fit the project purposes, it can be seamlessly migrated to an actual Kubernetes cluster, once the requirements of the component stack are clearly defined and met, leveraging all the benefits of Kubernetes.

In the image below an initial blueprint of the deployment schema is provided:

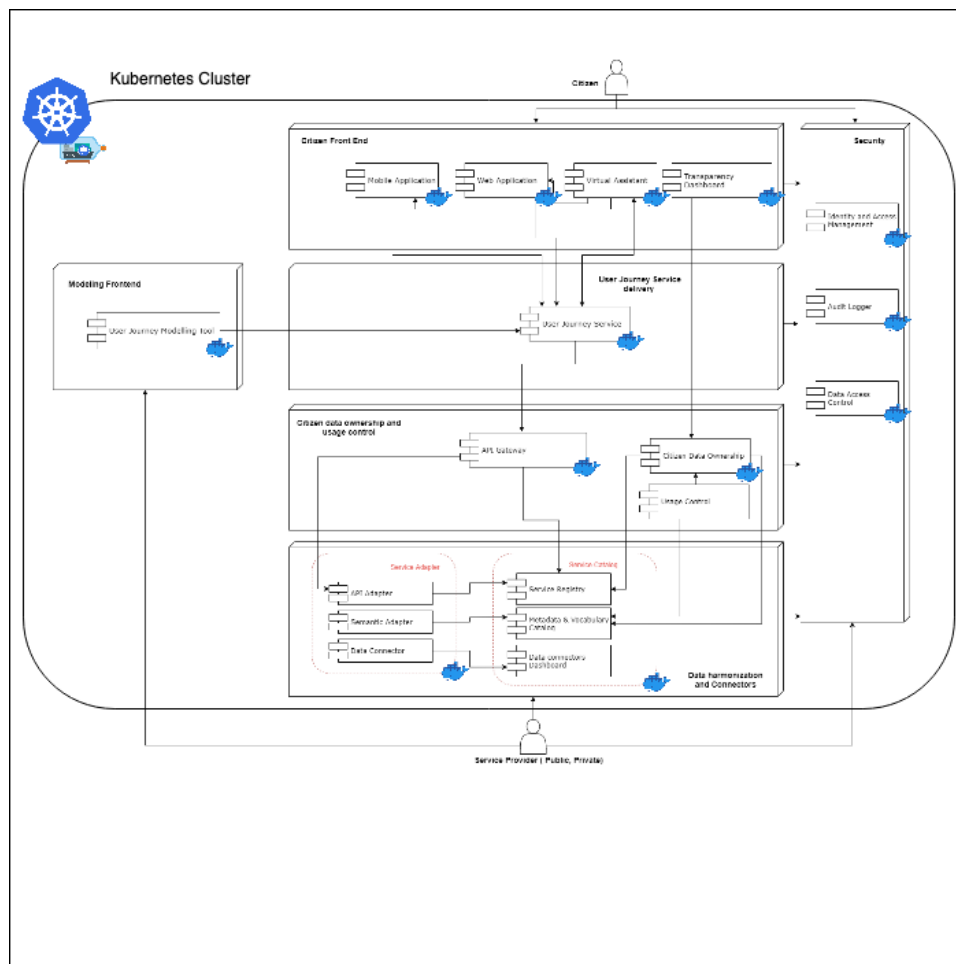
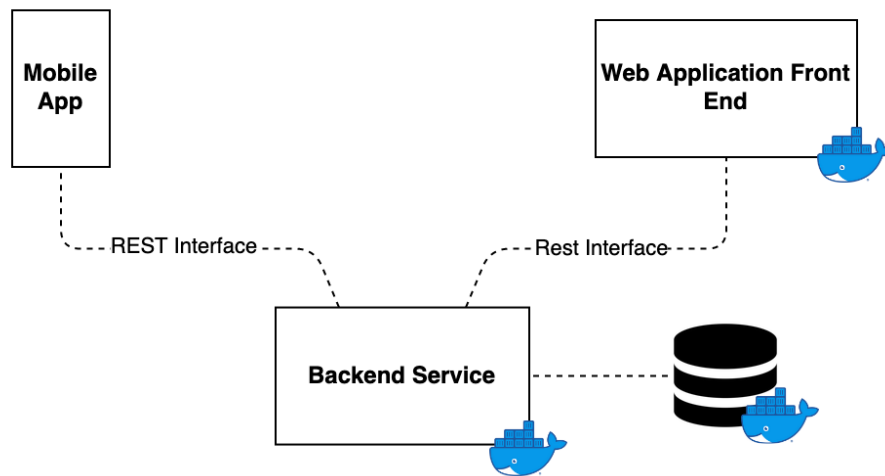


Figure 9: Deployment Blueprint Overview

As shown in the diagram, the individual work package components are indicated as docker containers orchestrated by Kubernetes. The interfacing among the components is implemented via REST interfaces.

Finally, as the components achieve their maturity levels, possible refactors on the docker containers might occur. For example, in the case of the Web application the initial docker container might be split into 3 containers including the frontend application, the backend application and the relevant database.



**Figure 10: Detailed Module Deployment blueprint**



## 7 Conclusions

The current document includes the architecture specifications and design of the integrated ACROSS platform and serves as the basis for the development tasks of the project. Information about the functionalities from the system point of view, the characteristics of the components of the system, and the data flow between them are presented in detail.

This architecture description document will be very useful to define and communicate the initial blueprint of the ACROSS platform. The architecture will continue to evolve throughout the project and the most important point is to make sure that it is consistent and in line with the design and implementation work being described in the other technical work packages, as well as with the early pilot activities of the project. This deliverable acts as the reference point for the actual development of this platform and offers a shared and common background for the Consortium participants on the envisaged technologies that are necessary to build such a platform.



## 8 Annex 1-Requirements tables

<b>Id</b>	<b>Title</b>	<b>Description</b>	<b>Type</b>	<b>Category</b>
<b>Req_01</b>	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
<b>Req_02</b>	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
<b>Req_03</b>	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
<b>Req_04</b>	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
<b>Req_05</b>	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



<b>Req_06</b>	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
<b>Req_07</b>	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: high-quality software, faster releases, and improved users' satisfaction.	non functional	Platform architecture and interoperability
<b>Req_08</b>	Continuous Platform Integration	The DevOps approach should be followed in ACROSS Platform to include the continuous integration tools.	non functional	Platform architecture and interoperability
<b>Req_09</b>	Free access to other countries' e-services	Citizens should be able to access other countries' services	non functional	Platform architecture and interoperability
<b>Req_10</b>	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
<b>Req_11</b>	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



<b>Req_12</b>	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
<b>Req_13</b>	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
<b>Req_14</b>	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
<b>Req_15</b>	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
<b>Req_16</b>	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



<b>Req_17</b>	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
<b>Req_18</b>	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
<b>Req_19</b>	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
<b>Req_20</b>	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
<b>Req_21</b>	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





<b>Req_22</b>	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
<b>Req_23</b>	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
<b>Req_24</b>	OpenID Connect - Role-Management	For securely exchanging roles between two parties it is needed to define a data format	functional	Security and Privacy
<b>Req_25</b>	OpenID Connect - Client-Registration	It is needed to register the clients that want to use the ACROSS-Platform	non functional	Security and Privacy
<b>Req_26</b>	OpenID Connect - Client-Registration	Definition of the client registration workflow	business	Security and Privacy
<b>Req_27</b>	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
<b>Req_28</b>	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
<b>Req_29</b>	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



<b>Req_30</b>	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
<b>Req_31</b>	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 (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
<b>Req_32</b>	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
<b>Req_33</b>	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
<b>Req_34</b>	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



<b>Req_35</b>	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
<b>Req_36</b>	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



## 9 References

- [1] Sample bibliography
- [2] IEEE Standard 802.11-1997, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 26 June 1997.
- [3] P. Brenner, A Technical Tutorial on the IEEE 802.11 Protocol, Breezecom Wireless Communications edition, 1997.