

H2020-SC6-GOVERNANCE-2018-2019-2020

DT-GOVERNANCE-05-2018-2019-2020



D3.5 Implementation of the ACROSS Data Governance framework for data sovereignty - Final

Project Reference No	959157 — ACROSS — H2020-SC6-GOVERNANCE-2018-2019-2020
Deliverable	D3.5 Implementation of the ACROSS Data Governance framework for data sovereignty - Final
Work package	WP3: ACROSS Data Governance framework
Nature	Other
Dissemination Level	Public
Date	31/07/2023
Status	Final version
Editor(s)	Valentín Sánchez (TEC)
Contributor(s)	Idoia Murua (TEC), Urtza Iturraspe (TEC)
Reviewer(s)	Vincenzo Savarino (ENG), Heinrich Hammerstein (DATAPORT), Enrique Areizaga (TEC)
Document description	This report includes the description of the final version of the ACROSS Personal Data Management Framework covering the complete set of functionalities. This version reflects the changes requested by the use cases regarding usability and the solution to the issues arising during the validation phase.



Document Revision History

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
V0.1	14/07/2023	First draft for revision	TECNALIA
V0.2	19/07/2023	Formal revision	ENGINEERING
V0.3	28/07/2023	Formal revision	DATAPORT
V1.0	31/07/2023	Final version	TECNALIA



Executive Summary

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

In order to ensure the protection of personal data (and documents) and its compliance with GDPR and other relevant regulations, especially when shared between organizations, ACROSS will design and implement with a **data governance framework** where data subjects can control the use of their personal data empowering them.

The **data governance framework will** allow users to:

- 1) monitor which data are available and how they are used or how it has been accessed,
- 2) control: add, delete, or change data, provide or block access to public bodies (where permitted or enforced by the law), businesses or data brokers, giving individuals the power to determine how their data can be used.

From a technical point of view the Data governance framework includes:

- 1) A “private/personal data” governance platform including a Personal data management Web-application - the Transparency Dashboard - which provides a user interface to define manage and control the use of personal data.
- 2) A set of APIs/libraries to interact with the ACROSS platform.

A previous deliverable [1] gathered the ACROSS data governance framework requirements, including the data governance, security, and privacy requirements from the use cases, considering both the technical and operational perspectives, the final user expectations regarding data privacy and the ACROSS platform integration strategy. Furthermore, an initial design of the framework was included. Then, a first prototype of the data governance framework [2], the so-called minimum viable product was developed to check the initial design and based on the validation results, the final framework architecture [3] was designed. Next, an intermediate version of the Data Governance Framework was implemented and deployed. The final version of the Data Governance Framework includes the changes requested by the use cases regarding usability and the solution to the issues arising during the validation phase.



This report contains the documentation needed to use the final version of the Data Governance Framework including the user manual, the integration guide, and the installation guide.



Table of Contents

1	INTRODUCTION	1
1.1	CONTEXT	1
1.2	PURPOSE AND SCOPE	2
1.3	APPROACH FOR WORK PACKAGE AND RELATION TO OTHER WORK PACKAGES AND DELIVERABLES	3
1.4	STRUCTURE OF THE DELIVERABLE	4
2	PERSONAL DATA MANAGEMENT RELATED CONCEPTS	6
2.1	PERSONAL DATA	6
2.2	GDPR: GENERAL DATA PROTECTION REGULATION.....	8
2.2.1	<i>Explicit consent</i>	8
2.2.2	<i>GDPR User rights</i>	9
2.3	SERVICES	10
2.4	CONSENT.....	11
2.5	DATA USAGE POLICIES	12
3	ACROSS DATA GOVERNANCE FRAMEWORK USER MANUAL.....	14
3.1	END USER'S WORKFLOW: DATA GOVERNANCE FRAMEWORK SCENARIO DESCRIPTION	14
3.2	MAIN CHANGES FROM THE INTERMEDIATE VERSION	17
3.2.1	<i>Functionality</i>	17
3.2.2	<i>Usability</i>	17
3.2.3	<i>Integration with ACROSS platform User Interface, the citizen front end</i>	17
3.3	TRANSPARENCY DASHBOARD USER GUIDE.....	18
3.3.1	<i>Dashboard</i>	20
3.3.2	<i>Event logs</i>	20
3.3.3	<i>Managing services</i>	21
3.3.4	<i>Managing consents</i>	23
3.3.5	<i>Managing data usage policies</i>	25
3.3.6	<i>Managing personal data: My personal data view</i>	26
4	ACROSS DATA GOVERNANCE FRAMEWORK INTEGRATION GUIDE	28
4.1	SERVICE CATALOGUE API USED BY THE FRAMEWORK.....	29
4.2	CONSENT MANAGEMENT	30
4.2.1	<i>Consent-manager > Consent: API to manage consents</i>	30
4.2.2	<i>Consent Manager > Event-log: API to manage event logs</i>	31



4.3	USAGE CONTROL MANAGEMENT	31
4.3.1	<i>Usage Control > Enforcement API</i>	31
4.3.2	<i>Usage Control > Policy rules: API to retrieve supported policy rules</i>	31
4.3.3	<i>Usage Control > Policies: API for policies management</i>	32
4.4	USER JOURNEY SERVICE ENGINE API	32
4.4.1	<i>Citizen front-end API controller</i>	33
4.4.2	<i>Modelling tool API controller</i>	34
4.4.3	<i>Log API controller</i>	34
5	INSTALLATION MANUALS	36
5.1	DATA GOVERNANCE FRAMEWORK INSTALLATION.....	36
5.1.1	<i>Configuring keycloak Identity and Access Manager</i>	37
5.1.2	<i>Installing Transparency Dashboard</i>	38
5.1.3	<i>Installing Usage Control</i>	40
5.2	USER JOURNEY SERVICE ENGINE INSTALLATION	43
5.2.1	<i>Configuring keycloak Identity and Access Manager</i>	44
5.2.2	<i>Installing UJSE</i>	44
6	CONCLUSIONS AND FUTURE PLANS.....	47
6.1	CONCLUSIONS	47
6.2	FUTURE PLANS.....	47
6.2.1	<i>European context in personal data management</i>	48
6.2.2	<i>Data spaces personal data management</i>	48
6.2.3	<i>Personal data governance framework future plans.</i>	50
7	REFERENCES	52
ANNEX I	53



List of Figures

FIGURE 1 DPV TAXONOMY FOR PERSONAL DATA	7
FIGURE 2 OVERVIEW OF LEGAL BASIS CONCEPTS IN DPV	8
FIGURE 3 LOGIN WINDOW	18
FIGURE 4 TRANSPARENCY DASHBOARD MAIN WINDOW	19
FIGURE 5 DASHBOARD WINDOW	20
FIGURE 6 EVENT LOGS WINDOW	20
FIGURE 7 AVAILABLE SERVICES WINDOW	21
FIGURE 8 AVAILABLE SERVICES FILTER	22
FIGURE 9 SERVICE INFORMATION WINDOW	22
FIGURE 10 CONSENT MANAGEMENT WINDOW	24
FIGURE 11 PERSONAL DATA CATEGORIES PERMISSIONS AND USAGE POLICIES ASSOCIATED TO THE CONSENT	24
FIGURE 12 DATA USAGE POLICY BROWSE WINDOW	25
FIGURE 13 ADD POLICY WINDOW EXAMPLES: ONE POLICY COMPOSED BY SEVERAL POLICY RULES	26
FIGURE 14 PERSONAL DATA VIEW INTERFACE: LIST OF SERVICES USING THE “PHONE NUMBER” PERSONAL DATA CONCEPT	26
FIGURE 15 PERSONAL DATA VIEW INTERFACE: SELECTING A PERSONAL DATA CATEGORY	27
FIGURE 16 ACROSS PERSONAL DATA GOVERNANCE FRAMEWORK SCENARIO	28
FIGURE 17 DATA GOVERNANCE FRAMEWORK COMPONENTS INTERACTION	36
FIGURE 18 LOGIN WINDOW	40
FIGURE 19 UJSE COMPONENTS INTERACTION	43
FIGURE 20 GAIA-X DATA EXCHANGE SERVICES SPECIFICATIONS CONCEPTUAL MODEL	50

List of Tables

TABLE 1 ACROSS WORKFLOW	15
-------------------------------	----



List of Terms and Abbreviations

Abbreviation	Definition
DPO	Data Protection Officer
CPSV-AP	Core Public Service Vocabulary Application Profile
ABC	Attribute Based Credentials
GDPR	General Data Protection Regulation
DGA	Data Governance Act
PIMS	Personal Information Management System
SDGR	Single Digital Gateway Regulation
OOP	Once-only principle
CRUD	Create, Read, Update and Delete
DPV	Data Privacy Vocabulary
Transparency Dashboard (TD)	The name of the Data Governance Framework graphical user interface application



1 Introduction

1.1 Context

One of the ACROSS objectives is **to ensure the protection of personal data (and documents) and its compliance with GDPR and other relevant regulations, especially when shared between organizations.**

This objective has been fulfilled by designing and implementing a private/personal data governance framework where data subjects can control the use of their personal data empowering them.

ACROSS offers the citizen the possibility of defining which public and private organization will be allowed to *access which data and for what purpose* through the **ACROSS Data Governance Framework**. The main aim is to give the citizen the chance of **govern the access to** their data, profiting from a set of usage policies that implement levels of access and they can be the **sovereign owner** of such data.

The **data governance framework**, that allows users to

- 1) monitor which data are available and how they are used or how it has been accessed,
- 2) to control: add, delete, or change data, provide, or block access to public bodies (where permitted or enforced by the law) businesses or data brokers, giving individuals the power to determine how their data can be used.

From a technical point of view the Data governance framework includes:

- 3) A “private/personal data” governance platform including a Personal data management Web-application - the Transparency Dashboard - which provides a user interface to define manage and control the use of personal data. (Data portal)
- 4) A set of APIs/libraries to interact with the ACROSS platform.

The governance framework has been based on existing initiatives and techniques.

1. **MyData**¹ model for human-centered personal data management and processing and MyData operator concept²
2. Built on the approach adopted in CaPe solution³ for personal data management,
3. Include generic **data usage policies**⁴ when the private data needs to be transferred among several stakeholders (IDSA Data Sovereignty)

¹ <https://mydata.org/>

² <https://mydata.org/mydata-operators/>

³ <https://github.com/OPSILab/Cape>

⁴ <https://internationaldataspaces.org/data-sovereignty-updated-position-paper-on-data-usage-control-in-the-ids/>
Deliverable D3.5 Implementation of the ACROSS Data Governance framework for data sovereignty - Final 959157 — ACROSS — H2020-SC6-GOVERNANCE-2018-2019-2020



D3.1 (Design of the ACROSS Data Governance framework for data sovereignty – Initial) provided an accurate description of the ACROSS data governance framework requirements, along with a first draft of technical architecture, modules, and APIs. Besides, a set of relevant baseline technologies that will be used for the implementation of the data governance framework were described and their applicability to ACROSS was analysed.

These following conclusions from D3.1 have driven the evolution of the data governance framework design and its implementation.

- The ACROSS Personal Data Governance Framework is part of the so-called ACROSS platform, useful in the context of cross-border public/private services for the citizens. However, it has been designed to be an independent framework useful also for the individuals to manage their personal data according to the GDPR in any other context.
- ACROSS data governance framework does not cover the following functionalities included in other initiatives:
 - Secure Data Storage and data minimization techniques: These will be provided by personal wallet technologies being developed by other projects.
 - Secure Data transfer among services
- ACROSS extends the MyData operator with data usage control based on data usage policies

Then, a first prototype of the data governance framework [2], the so-called minimum viable product was developed to check the initial design and based on the validation results, the final framework architecture [3] was designed. Based on the final design, an intermediate version of the Data Governance Framework including all the expected functionality was implemented and deployed. The final version of the Data Governance Framework includes the changes requested by the use cases regarding usability and the solution to the issues arising during the validation phase.

1.2 Purpose and Scope

This deliverable includes the implementation of the final version of the ACROSS Personal Data Management Framework covering the complete set of functionalities included in D3.2 “Design of the ACROSS Data Governance framework for data sovereignty – Final”. The intermediate version was evaluated by the use cases to produce D3.5: Implementation of the ACROSS Data Governance framework for data sovereignty – Final.

The deliverable includes the information needed to use this final version including:

- A description of the personal data management related concepts



- The user manual
- The integration manual
- The installation manual

Finally, some conclusions are presented along with a set of next steps.

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

The goal of WP3 is to design, implement and deploy a “private/personal data” governance framework that allows the citizens to control how their data and their activities are created or used by businesses, governments, or data brokers, giving individuals the power to determine how their data can be used. The governance framework has been based on existing solutions such as MyData model for human-centred personal data management and processing, but it will also include generic data usage policies when the private data needs to be transferred among several stakeholders.

The services from this WP will be integrated into the platform created in WP5 and will demonstrate the functionality of the use cases in WP6.

WP5 aims at providing the architectural and implementation aspects for the delivery of the ACROSS tools considering the full range of requirements for such service. The design of the ACROSS platform has driven the design and implementation of the various components produced in the context of WP3, WP4 & WP5.

The ACROSS Personal Data Governance Framework is part of the so-called ACROSS platform (defined in WP5), useful in the context of cross-border public/private services for the citizens. However, it has been designed to be an independent framework which can be used also for individuals to manage their personal data according to the GDPR in any other context. Furthermore, it extends the MyData operator concept with Data usage policies enforcement.

WP2 and WP6 together defined the so-called user journeys based on the results several interviews with people from the three pilot countries. The aim of the interview process is to form potential user journeys, building on initial ideas. User journeys can include services, actions and phases. Eventually to result in concrete (socio-technical) requirements for the ACROSS platform modules. A specific section about Data privacy issues was included in the questionnaire in order to gather requirements for the Data Governance Framework.

The final design was based on the results of the usability test findings for the first version of the platform (D3.3) that have been gathered in D6.2 Use case evaluation and impact assessment – Initial. **User tests** via structured interviews and a **co-creation workshop** have helped in collecting valuable feedback about



the initial version of the ACROSS platform from real users, as well as stakeholders from pilot countries and EU institutes who are involved in relevant national or European projects. **A set of recommendations for the evolution of ACROSS solution** have been reported, mostly focus on optimizing already implemented features as well as making previous requirements more specific.

Furthermore, some new functionalities have been included in the final design and have been implemented in the intermediate and final versions of the Data Governance Framework:

1. Data usage policies enforcement for Data access control
 - a. Data usage policy editor
 - i. Create, modify and delete policies associated to services.
 - b. Rest interface for policy enforcement (Usage control)
2. Rest interface for data consent definition from the User Journey Service Engine (UJSE)
3. My Personal Data view.

The following previous results of the project have been considered:

- D3.1 Design of the ACROSS Data Governance framework for data sovereignty – Initial [1]
- D3.3 Implementation of the ACROSS Data Governance framework for data sovereignty – Initial [2]
- D3.2 Design of the ACROSS Data Governance framework for data sovereignty – Final [3]
- D2.4 Report for cross-border service gap analysis – Final [4]
- D6.2 Use case evaluation and impact assessment – Initial [5]
- D5.2 System Architecture & Implementation Plan – Final [6]
- D3.4 Implementation of the ACROSS Data Governance framework for data sovereignty – intermediate [7]

Finally, an in-depth analysis of the ACROSS data governance framework has been performed by the use cases and some recommendations about user experience, usability and verticalization of the Personal Data Framework to be used in the context of the ACROSS platform have been implemented.

1.4 Structure of the Deliverable

This deliverable has been structured in the following sections:

Section 1 includes the context, purpose, and scope of the document.

Section 2 objective is helping the end user to understand the rationale behind the Personal Data Framework, the main concepts, and functionalities of the framework.

[Deliverable D3.5 Implementation of the ACROSS Data Governance framework for data sovereignty - Final 959157 — ACROSS — H2020-SC6-GOVERNANCE-2018-2019-2020](#)



Section 3 contains a complete user manual of the Transparency dashboard (TD), which is the graphical user interface of the ACROSS personal data management framework.

Section 4 contains the integration guide i.e., the APIs exposed by the ACROSS data governance framework back-end that will be used by external applications to interact with the framework. It also includes the User Journey Services Engine (UJSE) API.

Section 5 includes the installation manuals of the current version of the Data Governance Framework and the User journey services engine.

Finally, some conclusions are drawn together with recommendations for future work.



2 Personal data management related concepts

Next, some important concepts regarding personal data management are defined. To have a clear understanding of this concepts is paramount to use the Data governance framework.

2.1 Personal data

The European Commission⁵ defines personal data as “*any information about an identified or identifiable person, also known as the **data subject***”. Different pieces of information, which collected can lead to the identification of a particular person, also constitute personal data. Even personal data that has been de-identified, encrypted or pseudonymised but can be used to re-identify a person remains personal data and falls within the scope of the General Data Protection Regulation (GDPR).

Personal data includes information such as their:

- name
- address
- ID card/passport number
- income
- cultural profile
- Internet Protocol (IP) address
- data held by a hospital or doctor (which uniquely identifies a person for health purposes).

Certain types of sensitive personal data are subject to additional protection under the GDPR. These are listed under Article 9 of the GDPR as “**special categories**” of personal data. The special categories are:

1. Personal data revealing racial or ethnic origin.
2. Political opinions.
3. Religious or philosophical beliefs.
4. Trade union membership.
5. Genetic data and biometric data processed for the purpose of uniquely identifying a natural person.
6. Data concerning health.
7. Data concerning a natural person’s sex life or sexual orientation.
8. Personal data related to criminal convictions and offences unless this is authorised by EU or national law.

⁵ https://commission.europa.eu/law/law-topic/data-protection/reform/what-personal-data_en

Processing of these special categories is prohibited, except in limited circumstances.

In order to homogenise the personal data models, the ACROSS Personal Data Framework will use the personal data model defined on DPV-PD: Extended Personal Data categories for DPV ⁶ taxonomy and classes.

DPV provides broad top-level personal data categories adapted from the taxonomy contributed by EnterPrivacy⁷. The top-level concepts in this taxonomy refer to the nature of information (financial, social, tracking) and to its inherent source (internal, external). Each top-level concept is represented in the DPV vocabulary as a Class and is further elaborated by subclasses for referring to specific categories of information - such as preferences or demographics.

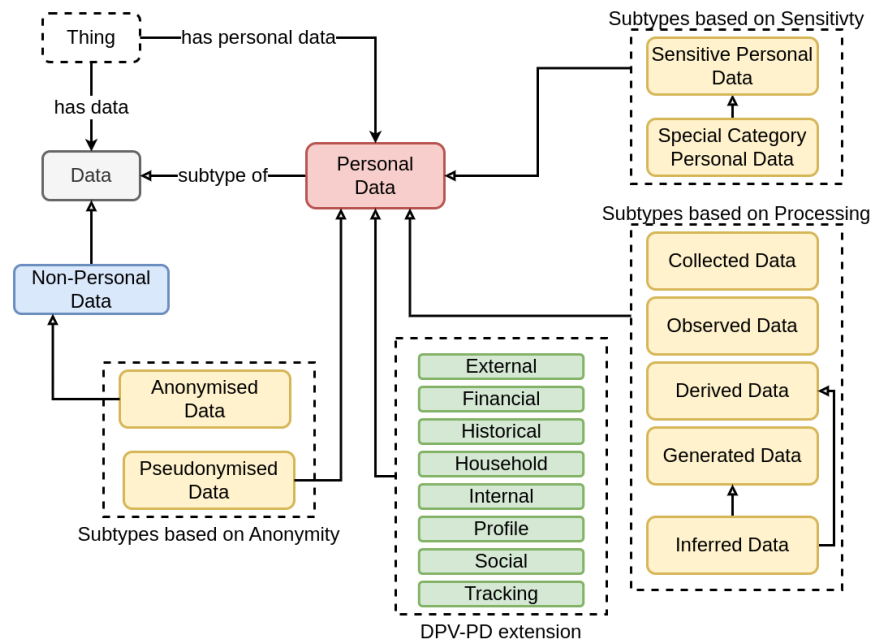


Figure 1 DPV taxonomy for personal data⁸

DPV's list of concepts is not universal nor exhaustive and includes 206 personal data concepts⁹. The ACROSS Data Governance Framework uses these concepts to describe the personal data used by the Services.

⁶ [DPV-PD: Extended Personal Data categories for DPV \(w3c.github.io\)](https://w3c.github.io/dpv-pd/)

⁷ <https://enterprivacy.com/wp-content/uploads/2018/09/Categories-of-Personal-Information.pdf>

⁸ <https://enterprivacy.com/wp-content/uploads/2018/09/Categories-of-Personal-Information.pdf>

⁹ [DPV-PD: Extended Personal Data categories for DPV \(w3c.github.io\)](https://w3c.github.io/dpv-pd/)



2.2 GDPR: General Data Protection Regulation

The GDPR is an EU data privacy law that went into effect May 25, 2018. It is designed to give individuals more control over how their data is collected, used, and protected online. It also binds organizations to strict new rules about using and securing the personal data they collect from people, including the mandatory use of technical safeguards like encryption and higher legal thresholds to justify data collection.

There are two main features of the GDPR the final user of the ACROSS data management framework must understand.

- Explicit consent
- GDPR User rights

2.2.1 Explicit consent

According to de GDPR, a company using personal data need to have a “legal base”. DPV¹⁰ provides the following categories of legal bases based on GDPR Article 6: **consent of the data subject**, contract, compliance with legal obligation, protecting vital interests of individuals, legitimate interests, public interest, and official authorities.

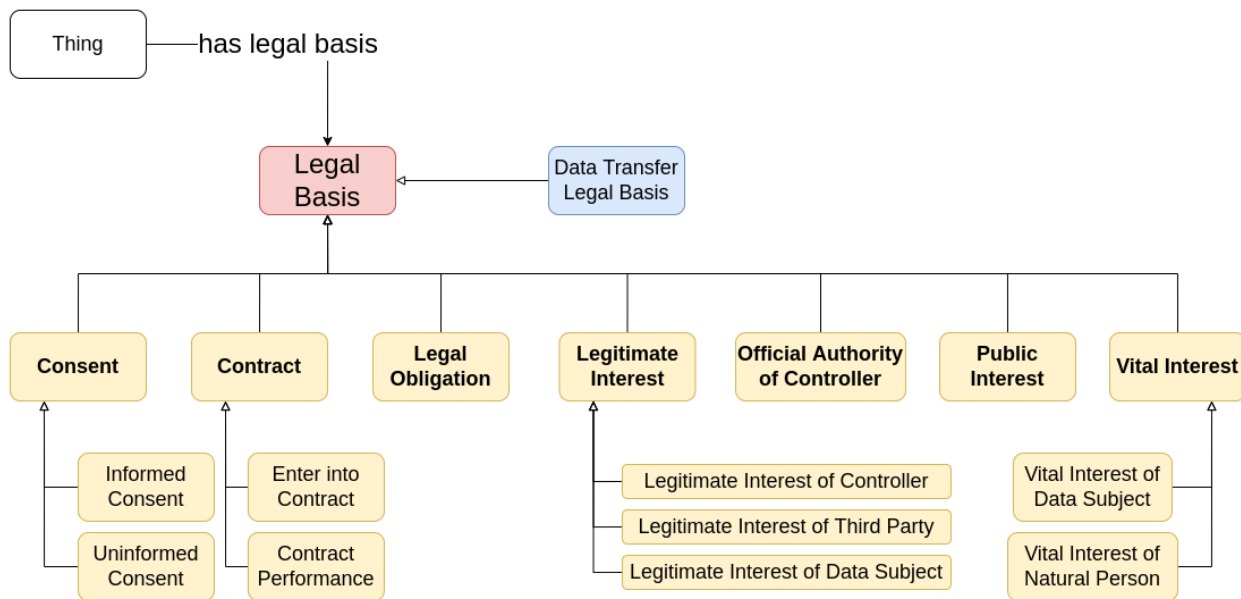


Figure 2 Overview of Legal Basis concepts in DPV

¹⁰ [Primer \(w3c.github.io\)](https://w3c.github.io/Primer/)



The ACROSS Personal Data Framework provide the individual the means to manage Consents, so that public bodies, businesses, or data brokers are able to get explicit consent to use the personal data of an individual. Section 2.4 elaborates more on the concept of consent and how it is used by the ACROSS data governance framework.

2.2.2 GDPR User rights

GDPR provides several rights to the data subject, whose applicability depends on the context and nature of processing taking place. DPV lists these rights at an abstract level as concepts along with their origin in specific clauses of the GDPR. The following list provides information about the user rights along with the way in which the Personal Data Framework facilitates the exercise of the rights by the individual.

- **Information** about the processing of your personal data
This information is included in the description of the service, although by now only one category of processing is included.
- **Obtain access to** the personal data held about you.
This right is not provided by the framework, but the email of the DPO is included in the service description so the user can write requesting the personal data the service provider holds.
- Ask for incorrect, inaccurate, or incomplete personal data to be **corrected**.
This right is not provided by the framework, but the email of the DPO is included in the service description so the user can request to exercise this right.
- Request that personal **data be erased** when it's no longer needed or if processing it is unlawful.
This right is not provided by the framework, but the email of the DPO is included in the service description so the user can write requesting the personal data the service provider holds.
- **Object** to the processing of your personal data for marketing purposes or on grounds relating to your particular situation.
The user can reject the consent if the processing of the personal data includes some specific purpose. Another way is by writing an email to the DPO.
- Request the **restriction** of the processing of your personal data in specific cases.
This right is not provided by the framework, but the email of the DPO is included in the service description so the user can request to exercise this right.
- Receive your personal data in a machine-readable format and send it to another controller ('**data portability**')
This right is not provided by the framework, but the email of the DPO is included in the service description so the user can request to exercise this right.



- Request that decisions based on **automated processing** concerning you or significantly affecting you and based on your personal data are made by natural persons, not only by computers. You also have the right in this case to express your point of view and to contest the decision.

This right is not provided by the framework, but the email of the DPO is included in the service description so the user can request to exercise this right.

2.3 Services

The ACROSS Data Governance Framework uses the concept of service in a broad sense to refer to the services offered by a company to the user. Therefore, the user is able to give the consent to use personal data to each specific service.

The whole list of services available are managed by another component of the ACROSS platform, the Service Catalogue[8]. In order to register a service in the Service Catalogue the service provider has to introduce the information described in the CPSV-AP¹¹ standard with some extensions.

CPSV-AP captures fundamental characteristics of a service, such as the name, description, organisation, output, etc. Public administrations and service providers can therefore use this approach to describe their services and guarantee a level of cross-domain and cross-border interoperability at European, national, and local level.

In the context of the Personal Data Management Framework the main information provided to the final user to support the process of definition of the consent is:

- **Personal data** requested by the services in two categories:
 - **Mandatory:** Information necessary to offer the service.
 - **Optional:** Additional information that can be used to improve the service, for example offering a greater degree of personalization
- **Organization** which provides the service.
- **Purpose:** represent the *reason* or *justification* for processing of personal data. Purposes are organised within DPV¹² based on how they relate to the processing of personal data in terms of several factors, such as: management functions related to information (e.g., records, account, finance), fulfilment of objectives (e.g., delivery of goods), providing goods and services (e.g., service provision), intended benefits (e.g. optimisations for service provider or consumer), and legal compliance. However,

¹¹ [About Core Public Service Vocabulary Application Profile | Joinup \(europa.eu\)](#)

¹² [Primer \(w3c.github.io\)](#)



according to the DPV purposes do not have a strict structure as used in real-world use-cases, it is important to note the following for real-world implications of Purpose:

- There is no universal definition for what constitutes a 'purpose' or what attributes are associated with it.
- There are several distinct ways to model purposes, e.g., as a 'goal' such as 'Delivery of Ordered Goods'; or as a statement explaining the processing of personal data, e.g., 'Sending newsletters to email'.
- DPV does not define requirements for what is a 'valid purpose' as these are defined externally, e.g. in laws such as [\[GDPR\]](#) Article.5-1b where purposes are required to be 'explicit and legitimate'.
- Purposes have contextual interpretations within their application and domains i.e., depending on how they are used in a use-case). For example, ServiceProvision is interpreted distinctly across the use-cases of an online website, a goods delivery outlet, and a medical centre - even if they use the same term or wording.

Following from the above, most use-cases would need to extend one of the concepts within DPV's purpose taxonomy to ensure its purpose descriptions are specific and understandable within the context of that use-case. Therefore, DVP suggest, where possible and appropriate, to create a customised purpose as required within the use-cases by extending or subtyping one or several purposes from the DPV taxonomy and to provide a human readable description to assist in its accurate interpretation. Following this recommendation, ACROSS have defined a restricted set of purposes, closer to the real-world use of terms.

- **Processing:** DPV's taxonomy of processing concepts reflects the variety of terms used to denote processing activities or operations involving personal data, such as those from GDPR Article.4-2 definition of processing. Real-world use of terms associated with processing rarely uses this same wording or terms, except in cases of specific domains and in legal documentation. On the other hand, common terms associated with processing are generally restricted to: collect, use, store, share, and delete. DPV provides a taxonomy that aligns both the legal terminologies such as those defined by GDPR with those commonly used. In ACROSS, only the common terms are used, so that it facilitates the usability from the final user point of view.

2.4 Consent

Consent is an important legal basis given its emphasis on individual empowerment and control, as well as the attention and relevance it receives from being part of direct interactions with individuals. In order to facilitate the definition of consents, the consent model used includes only the following information:



- **User Id:** The identifier of the user in the personal data framework
- **Service Id:** The identifier of the service in the service catalogue
- **Personal data permissions** for each personal data category needed by the service, both required and optional.
- **Status of the service:** It can be Granted, Denied or Withdrawn (See more information in section 3.1)

2.5 Data usage policies

ACROSS data governance framework focuses on permission management, providing a way for people to define the specific data that can be shared (or disclosed) between parties, for which purposes, and for how long. Furthermore, it will go beyond basic permission management including the possibility to define data usage policies for more fine-grained usage control. In general, the overall goal is to enforce usage restrictions for data after access has been granted.

A data usage policy could be used to constraint the location in which data is used, to enforce the deletion of the personal data after a specific period or to modify the personal during the data transfer process.

In fact, some of the defined data usage policies can be used to enforce the some GDPR rights and principles. For example, the rule that allows to modify the data in transit could be used to ensure the data minimization principle.

The complete data usage policy enforcement functionality is performed by a specific software (the IDS connectors) and some of the rules can only be applied by the consumer connector, so it can be used only if the services (both public and private services) deploy the IDS connectors technology for data transfer.

However, the ACROSS personal data government framework strategy is to minimize the service providers adaptation needed to use the framework and it will assume the responsibility of performing data usage policies management and enforcement. Therefore, not real “data usage control” can be applied, only a restricted set of data usage policies providing data access rules. Each service is associated with a data usage policy that can be composed by one or several policy rules.

The following rules can be used:

- **N times usage:** This policy rule restricts the numeric count of using your data by a specified data consumer (provider side). The number of times used is updated and consulted via an external end point.



- **Duration usage:** allows data usage for a specified time period. For example, an instantiated policy from this policy class may allow a Data Consumer to use the data for a duration of three months. The permitted period may start from a given date and time.
- **Usage during interval:** This policy rule provides data usage within a specified time interval (start + end date)

Other rules that can only be enforced by the service provider receiving the personal data have not been implemented, specifically the Role-restricted Data Usage and Purpose-restricted Data Usage rules.

The following management functionalities are provided for the end user:

- Data usage policies browsing and filtering.
- Data usage policy create/modify/delete functionalities.



3 ACROSS Data governance framework user manual

The ACROSS Data governance framework defines a completely new way to manage personal data. Nowadays, the common way of dealing with personal data that we share with companies is to give the necessary permissions to each of the companies/services that use the data in a one-by-one basis using the applications provided by them.

This approach has several drawbacks from the final user perspective. To manage the personal data related permissions, it is necessary to use many different web sites and find the way to exercise the personal data rights. Furthermore, it is almost impossible to remember the whole set of applications/services holding and using our personal data.

The ACROSS Data Governance Framework offers the citizens a centralized way to manage all the personal data they share, defining which public and private organization will be allowed to access which data and for what purpose. Furthermore, the citizens will be able to see what services/applications and companies are using each specific personal data concept, e.g., their address or their age.

In summary, the data governance framework allows users to:

- 1) monitor which data are available and how they are used or how it has been accessed,
- 2) control: add, delete, or change data, provide, or block access to public bodies (where permitted or enforced by the law) businesses or data brokers, giving individuals the power to determine how their data can be used.

Next section provides an overview of the main concepts related to the personal data management.

3.1 End user's workflow: Data governance framework scenario description

This section describes the workflow the users must follow to use final implementation of the ACROSS Data Governance framework for data sovereignty. Three types of users are envisaged: Administrator, End User (Citizen) and Service provider.

1. **Administrator:** Users management. Register and manage new users including end users and services providers. This functionality is provided by the external security package.
2. **Service provider:** Service description and registration. Each service provider has to register the services using the CSPV-AP extended model.
3. **End User:** The ACROSS Personal data framework facilitates the individual to perform the end-to-end process of consent management. In order to use all the functionalities, a workflow has been designed and consists of the following steps:



- a. Select services → Select the services the user is going to use.
- b. Consent Management → Define the personal data to be used by each selected service.
- c. Data Usage policies management → Define the data usage policies applicable the data to be used by each service. This is an optional step.
- d. Monitor the data usage for each service.
- e. Monitor the services using a specific personal data category.

Next table summarizes the functionalities covered by each step.

Table 1 ACROSS workflow

Step	Functionality
Service description and registration	<p>This is an initial step covered by the Service Catalogue and it is the responsibility of the service providers.</p> <p>The service model is the CPSV-AP extended with:</p> <ul style="list-style-type: none"> • personal data related data model: Purpose and processing • Information about Personal data needed by the service, including optional data. • Information about the REST service channel if available.
Login	<p>End User logged in the ACROSS Personal data governance framework. Keycloak is used as the IDM and two options are provided. The login can be performed by using the eIDAS framework and the respective citizens eIDs (e.g., Latvian eID) or use a local Keycloak user registered by the data governance framework administrator. The second option has been implemented mainly for testing purposes.</p>
Service Selection	<p>Using the Services window, the user selects the services to be used among the whole list of services registered in the catalogue. Consents and data usage policies can be defined and assigned only to previously selected services.</p>
Consent management	<p>For each selected service the user can create a consent that define the personal data categories that can be shared. All the mandatory personal data categories must be allowed to use the service.</p> <p>For each Consent the end User can change the Status of the relative Consent:</p> <ul style="list-style-type: none"> • Grant: Enables a new consent or a previously denied Consent • Deny: disable the Consent • Withdraw: revoke the Consent, a new one must be given by the Consenting phase.



Step	Functionality
Data Usage policies management	For each selected service the user can define a data usage policy composed by one or more policy rules that restricts the access to personal in a more fine-grained way.
Monitor data usage notifications and Activity Logs	Once the user has defined the applicable consents and data usage policies (optional), all the data transfers among services are tracked by the ACROSS data governance framework as Event Logs that can be viewed by the Data Subject via the Transparency dashboard (TD). Furthermore, all the interactions of the user with the Transparency dashboard are logged, so that the user is able to consult them.



3.2 Main changes from the intermediate version

3.2.1 Functionality

This is the list of new functionalities that have been added in the final version:

- **Internationalization:** User interface currently support the three different languages of the use cases. Though the TD is ready to provide the user interface in any other language.
- **eIDAS integration.** It is possible to log into the TD using the eIDAS nodes.
- **Privacy policy.** Now the transparency dashboard includes a revised version of the privacy policy needed to manage the private user information.
- **Information button added to every window.** Include help pages to describe the processes and concepts included in the consent management process.
- The **consent definition window** includes now the following options: “Select only required”, “Select all”, and “Unselect all”.

3.2.2 Usability

Some usability improvements have been implemented since the intermediate version.

- Reorganization of the main menu to reflect the workflow associated with consent management.
- Homogenize the user interface to have the same look and feel of the Citizen FE.
- The Action buttons have been situated at the left of the window to make them easier to find.
- Homogenise the field names along the user interface. For example, in page “Consents management” the “Personal data” column has been named “Consent”.

3.2.3 Integration with ACROSS platform User Interface, the citizen front end

Even though the personal data framework has been designed to be used as a personal data intermediary not only in the context of the ACROSS platform it has been decided to adapt it to facilitate the user experience providing a homogeneous look and feel and some transparent cross application links. The objective of this adaptation is to provide the user with an integrated solution regarding cross country services and user journeys execution.

Three main features have been added in order to provide such us integrated platform:

1. **New Left menu harmonized with the Citizen front-end.** The same left menu is used in the Citizen front end and the Transparency dashboard.
2. **Language integrated.** Selected language in the Citizen front end is also applied in the TD.

3. **Browser management:** It allows the user to navigate through the ACROSS platform user interfaces (both the Transparency Dashboard and the Citizen front end) using the browser navigation buttons.
4. **Dashboard is now the first page.** The about page has been removed.

3.3 Transparency dashboard user guide

The Transparency dashboard is the Web portal for citizens (acting as Data Subjects) to manage the Services they use and their related consents and get an overview of what is happening with their Personal Data. Next figure shows the main login window. The user registration is done previously by the Data Governance Framework administrator.

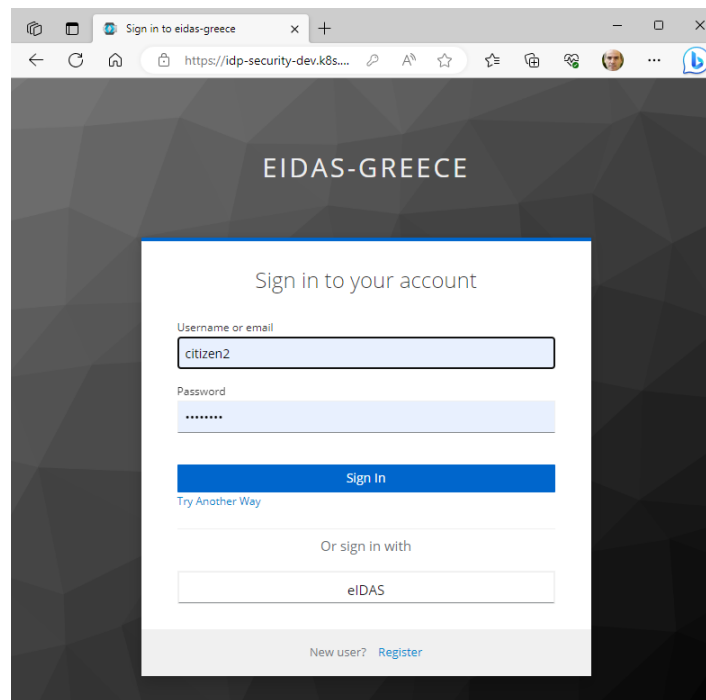


Figure 3 Login window

Next figure shows the Data Governance framework main window. The main page structure is common for all pages:

- Left frame with the vertical menu to access the framework functions.
- The page header providing access to notifications, language selection and logout.
- A central frame which provides the interface for each functionality. In all the windows a new “information” icon has been included with detailed information about the interface.
- The page footer allows the user to access the Privacy policy of the Data Governance Framework

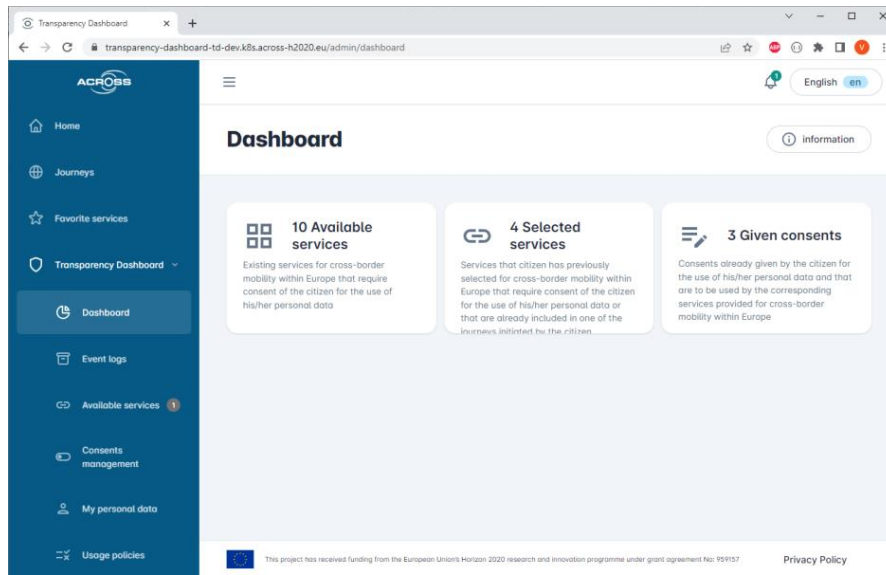


Figure 4 Transparency dashboard main window

The end user by interacting with the Transparency Dashboard, will be able to:

Interact with the citizen front end

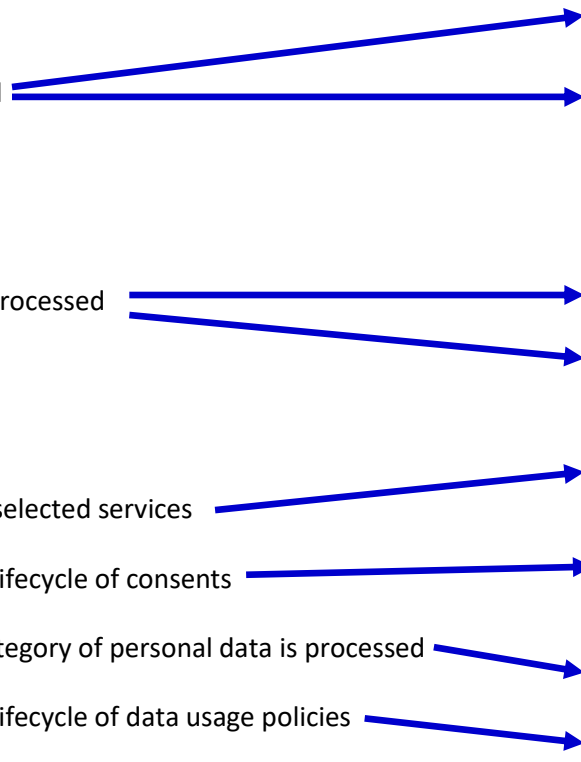
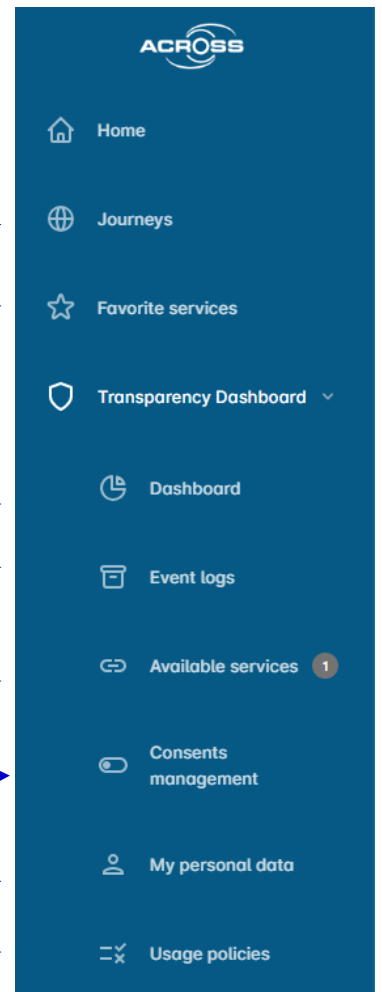
Monitor the personal data being processed

Get an overview and manage the selected services

Get an overview and manage the lifecycle of consents

Get an overview of where each category of personal data is processed

Get an overview and manage the lifecycle of data usage policies





3.3.1 Dashboard

In the Dashboard section (image below), the Transparency dashboard provides to the Data Subject an overview of the available, the selected services and given consents.

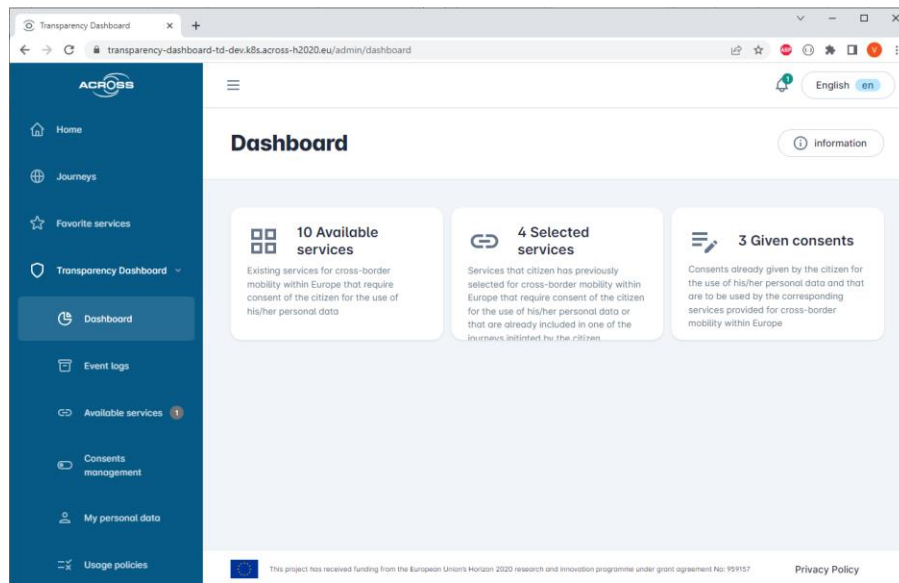


Figure 5 Dashboard window

3.3.2 Event logs

The event log window

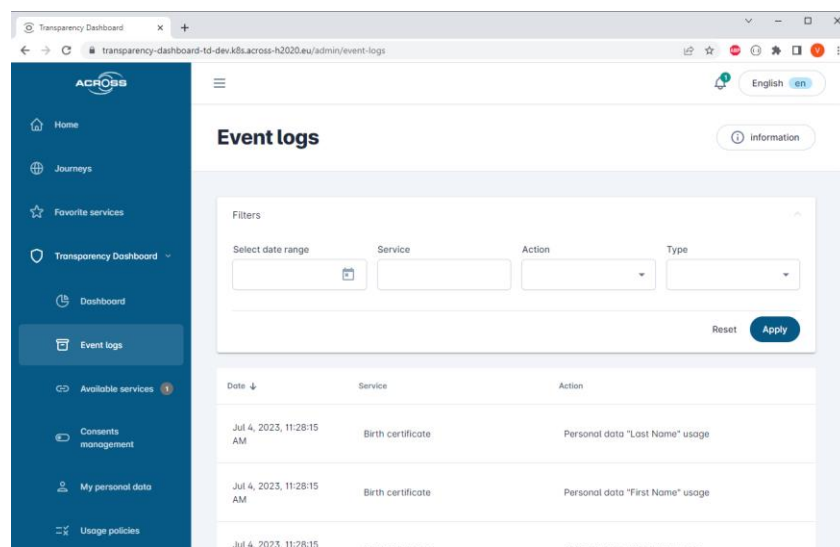


Figure 6 Event logs window

3.3.3 Managing services

The available services window provides the list of services registered in the Service Catalogue. The following management functionalities are provided:

- Services browsing and filtering.
- View services detailed information.
- Select/Unselect services.

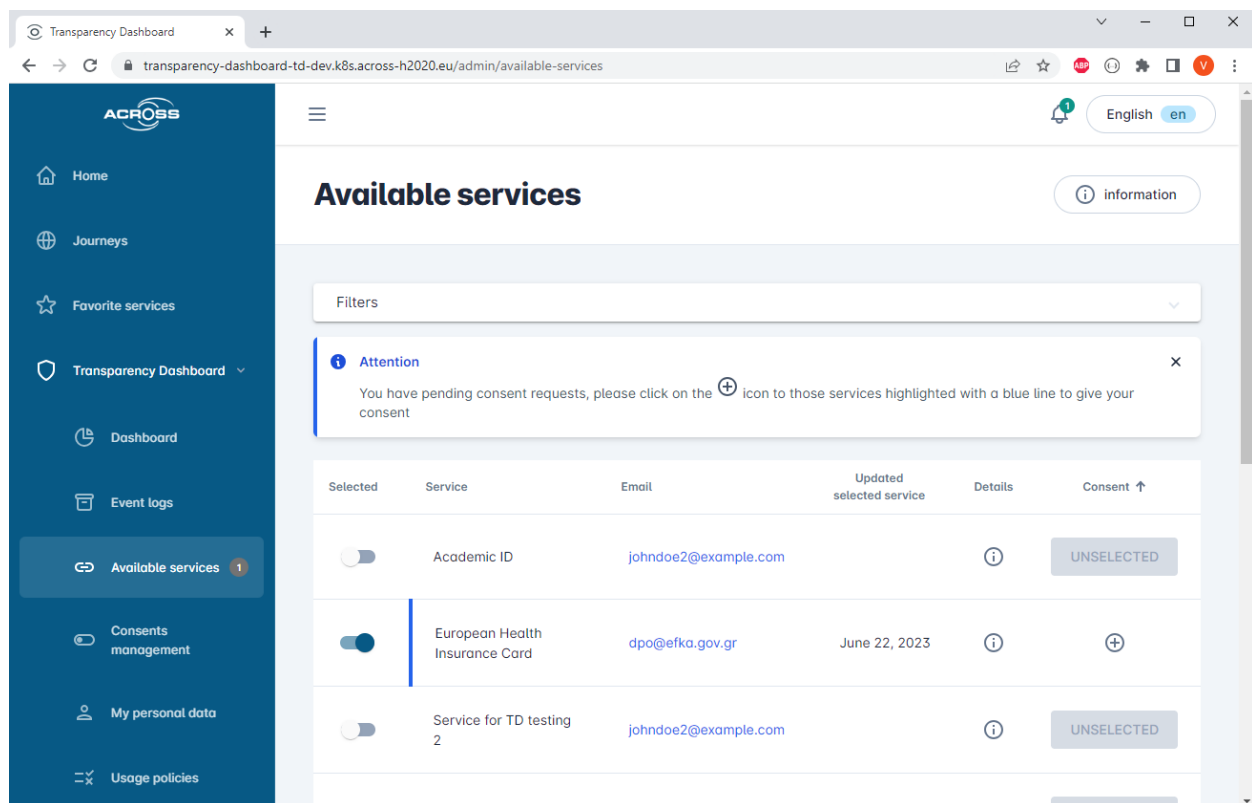


Figure 7 Available services window

This window allows the user to browse all the available services using a list format. For each service the following information is presented to the user:

- **Selected:** The service status (selected or unselected).
- **Service:** The service name.
- **Email:** the email of the DPO of the service provider.
- **Updated selected service:** date of the last service update.
- **Details:** Link to the detailed information about the service.
- **Consent:** Related consent status. A + icon allows the user to define the related consent.

Service list can be filtered out by: Country, Purpose category, Selected services and Service name.

Available services

information

Figure 8 Available services filter

The user can also access more detailed information about the service and select/unselect services. Next figure provides an example of the window providing information about one specific service:

Figure 9 Service information window

This window also provides more detailed information about the service provider and the service characteristics via the “Data controller” and “More info” buttons.



3.3.4 Managing consents

The Consent management window provides access and management functionalities for the consents. The following management functionalities are provided:

- Consents browsing and filtering.
- Consents create/delete functionalities.
- Consents status management.

This window allows the user to browse all the defined consents using a list format. For each service the following information is presented to the user:

- The service name and description.
- The status of the Consent.
- The date of the last update of the consent.
- A button to define the permissions for each personal data category requested by the service.
- A direct link to the usage policies defined for that service.

Consent list can be filtered out by Status, Status date and Personal data category.

Finally, a special button “Revoke all consents” is available to revoke all the consents.

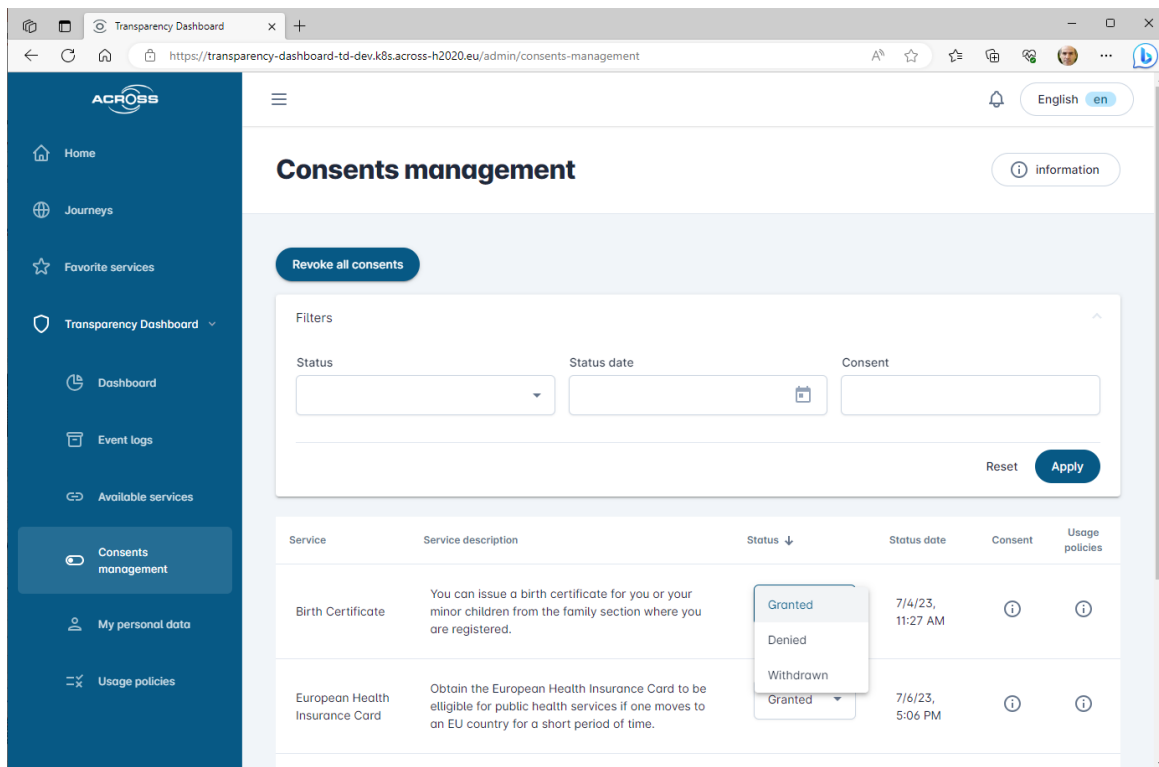


Figure 10 Consent management window

Next figure shows the window to define the specific permissions for each personal data category requested by the service and the window showing the related usage policies.

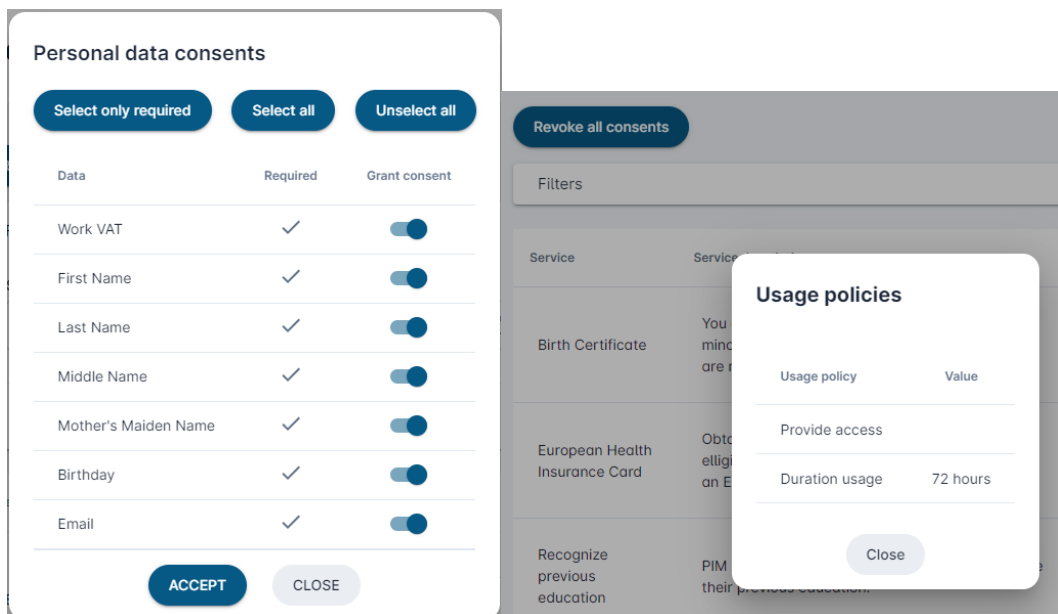


Figure 11 Personal data categories permissions and usage policies associated to the consent

3.3.5 Managing data usage policies

The Policy templates window provides access and management functionalities for the data usage policies. The following management functionalities are provided:

- Data usage policies browsing.
- Data usage policy create/modify/delete functionalities.

Each service is associated with a data usage policy that can be composed by one or several policy rules.

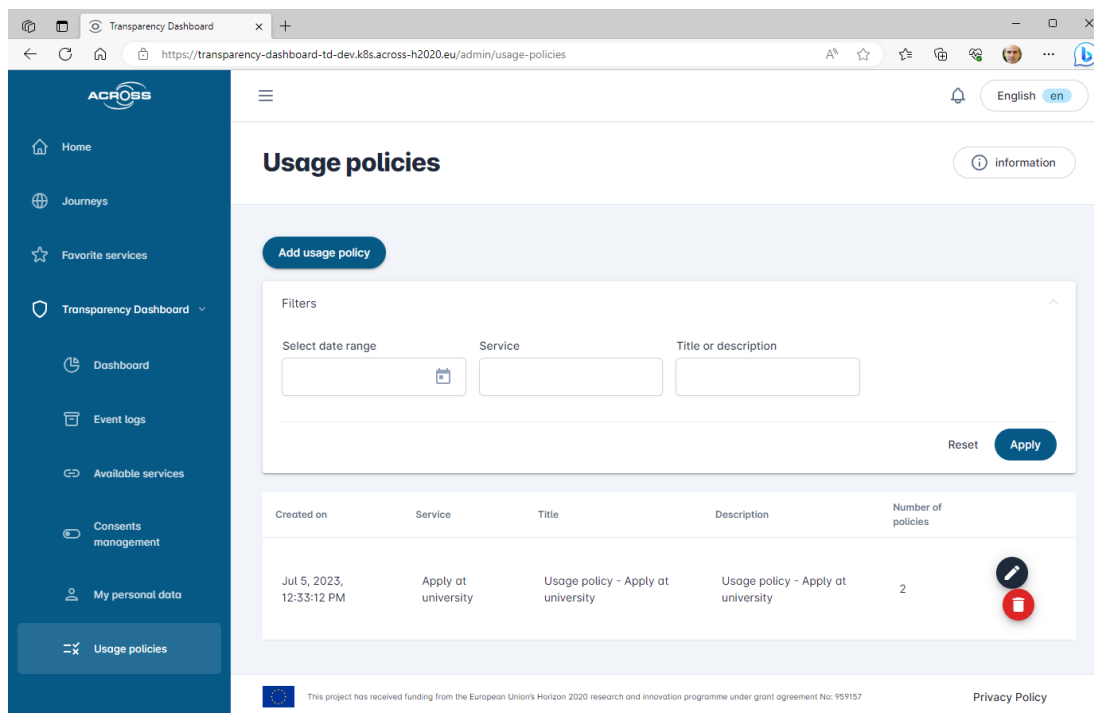


Figure 12 Data usage policy browse window.

Usage policy list can be filtered out by date range, service and title or description.

Next figure shows a policy example with several policy rules.

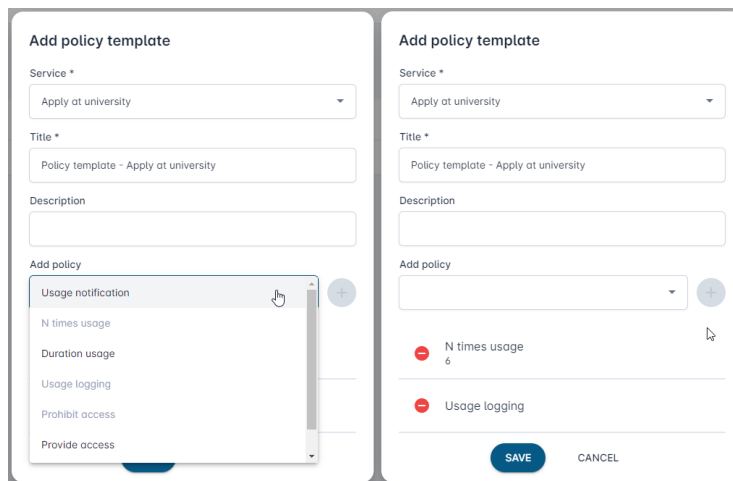


Figure 13 Add policy window examples: one policy composed by several policy rules

3.3.6 Managing personal data: My personal data view

This functionality allows the user to view the list of services that are using a specific personal data concept. The framework will give the service provider a way to map its specific data model with the DPV Personal Data Categories data model. In this way, it would be possible to have a vision of where each type of data is being used in different services.

Next figures show the user interface of My Personal Data view. From this interface is possible to revoke or grant the consent to use a specific data category by a service.

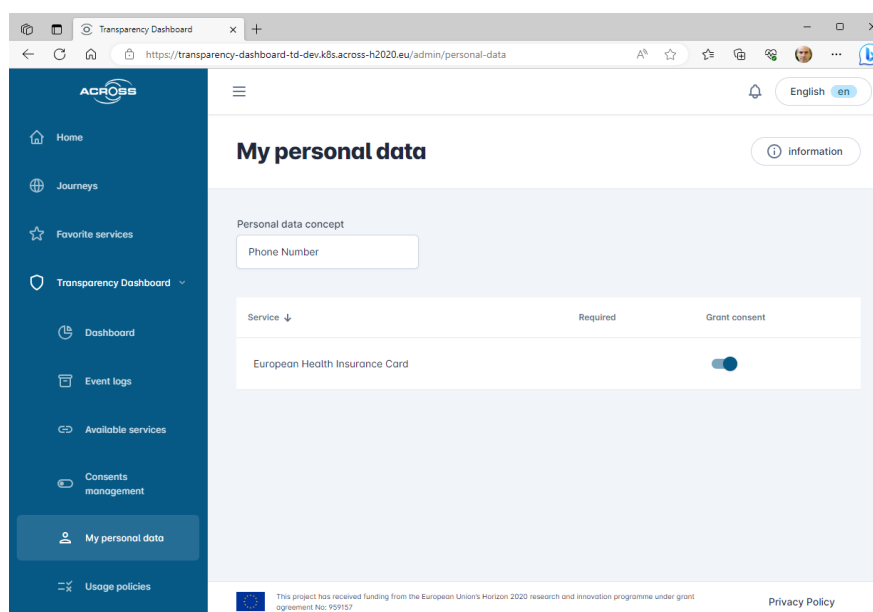


Figure 14 Personal data view interface: list of services using the “Phone number” personal data concept



In this example the “Phone number” personal data category is requested by “European Health Insurance Card” service, but it is not mandatory, and the user has granted its use by the service.

Next window shows the interface to select a specific personal data category.

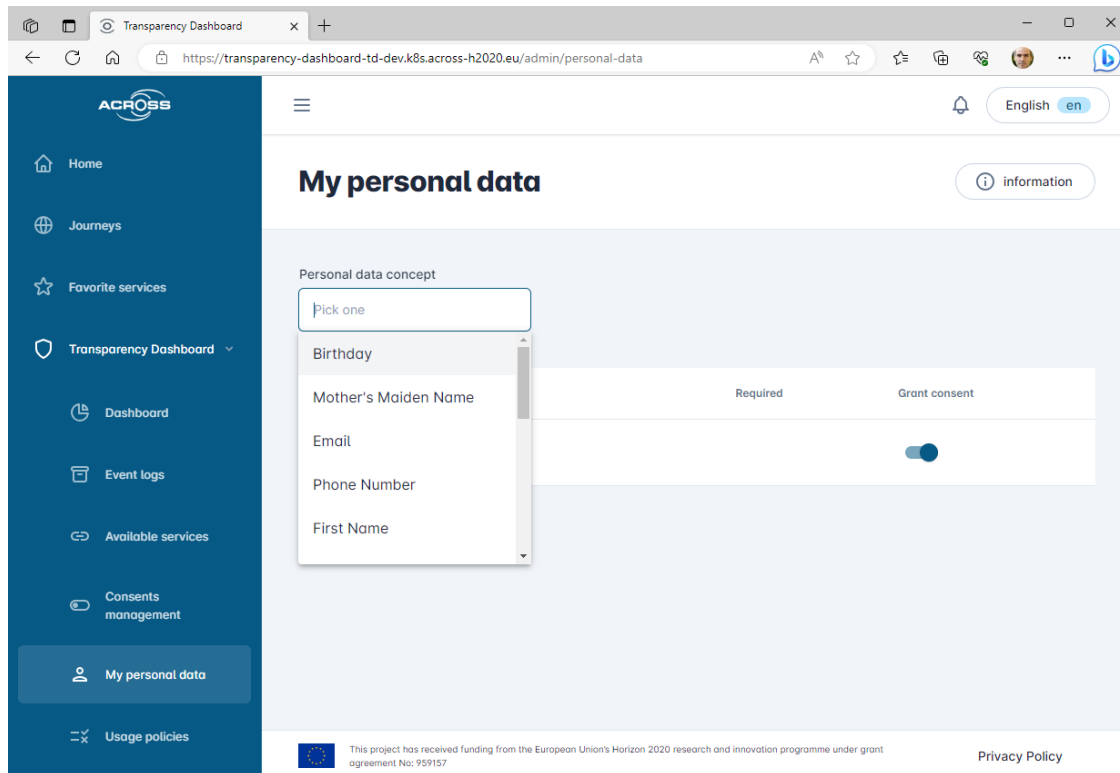


Figure 15 Personal data view interface: selecting a personal data category.

4 ACROSS Data governance framework integration guide

The ACROSS Personal Data Governance Framework is part of the so-called ACROSS platform[6], useful in the context of cross-border public/private services for the citizens. However, it has been designed to be an independent framework useful also for the individuals to manage their personal data according to the GDPR in any other context.

Any application or service can interact with the Personal Data Governance Framework to ensure that it is using personal data in accordance with the permissions and restrictions set by the end user.

Next figure shows the whole ACROSS scenario with the main actors, components, and interactions.

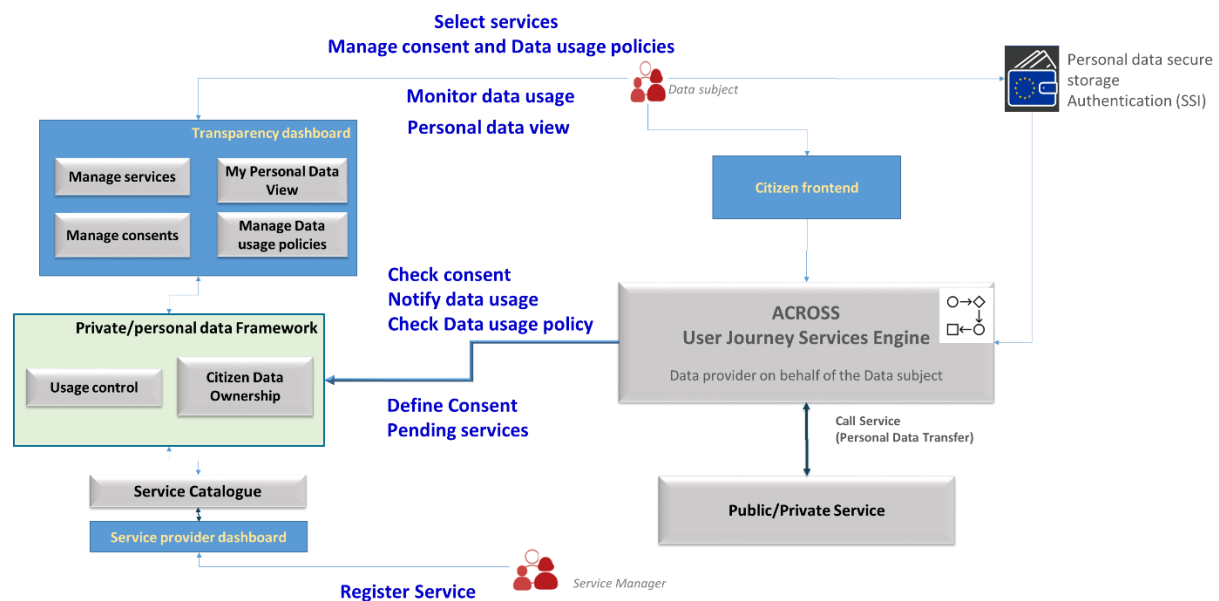


Figure 16 ACROSS Personal data governance framework scenario

The following logical interfaces have been defined:

- **Register Service:** First, each service provider must register the services using the CSPV-AP extended model. This API is provided by the Service Catalogue, which is an independent module not included in the Data Governance Framework.
- **Check consent:** The UJSE request to check the permissions to use personal data of a specific user by a service.
- **Check Data usage policy:** The UJSE request to enforce the data usage policies defined by a user for a service.
- **Pending services:** The UJSE inform the framework about the services included in a new workflow.



- **Notify data usage:** Each time a service is used the UJSE notifies the Data Governance framework about the personal data usage for logging and audit functions.
- **Define consent:** If a service called by the UJSE doesn't have the consent to use the personal data, this interface gives the user the possibility of creating the consent directly from the USJE.

Keycloak and eIDAS are used for Identity and Access Management instead of the SSI Authentication included in the figure, which is the most promising technology for the future.

4.1 Service Catalogue API used by the framework

The Service Catalogue API provides several REST services to register a new service and to get information about the service, e.g.: if it makes use of personal data, and which personal data it uses. Next figure shows the API used by the Data Governance framework to interact with the Service Catalogue. The complete Service Catalogue API is described in [Service Catalogue API documentation](#)

Servers
https://service-catalogue-server-dev.k8s.across-h2020.eu/service-catalogue - Generated server url

Service model Service model Description APIs to get and manage service model descriptions. ^

GET	/api/v2/services	Get all the Service model descriptions.	∨
PUT	/api/v2/services	Update Service model description, by replacing the existing one	∨
POST	/api/v2/services	Create a new Service model description.	∨
DELETE	/api/v2/services	Delete Service model description by Service Id.	∨
POST	/api/v2/services/many	Create new Service model descriptions.	∨
GET	/api/v2/services/time	Get Service time by serviceId.	∨
GET	/api/v2/services/specified/title	Get the Service model descriptions by specified Service Title.	∨
GET	/api/v2/services/specified/location	Get the Service model descriptions by specified Service Location.	∨
GET	/api/v2/services/specified/keyword	Get the Service model descriptions by specified Service Keywords.	∨
GET	/api/v2/services/specified/**	Get the Service model descriptions by specified Service Ids.	∨
GET	/api/v2/services/json/**	Get the Service model description by Service Id.	∨
GET	/api/v2/services/isPersonalDataHandling	Get the Service model descriptions is handling personal data	∨
GET	/api/v2/services/isPersonalDataHandling/count	Get the count of the Service model descriptions is personal data handling.	∨



4.2 Consent management

4.2.1 Consent-manager > Consent: API to manage consents

The Consent Manager provides the following API to manage the consents, that is, to give, deny or withdraw consents and to get information about the consents already given, by different criteria.

personal-data-consents-controller		^
PUT	/api/rest/v1/personal-data-consent	∨ 🔒
consents-controller		^
GET	/api/rest/v1/consent/{id}	∨ 🔒
PUT	/api/rest/v1/consent/{id}	∨ 🔒
PUT	/api/rest/v1/consent/{id}/personal-data	∨ 🔒
GET	/api/rest/v1/consent	∨ 🔒
POST	/api/rest/v1/consent	∨ 🔒
POST	/api/rest/v1/consent/user/{userId}/services	∨ 🔒
POST	/api/rest/v1/consent/user/{userId}/services/without-consent	∨ 🔒
POST	/api/rest/v1/consent/user/{userId}/services/status/{status}	∨ 🔒
POST	/api/rest/v1/consent/external-consents	∨ 🔒
GET	/api/rest/v1/consent/user	∨ 🔒
GET	/api/rest/v1/consent/user/{userId}/status/{status}	∨ 🔒
GET	/api/rest/v1/consent/user/{userId}/service/check-status-consent/{status}	∨ 🔒
GET	/api/rest/v1/consent/user/totals	∨ 🔒
GET	/api/rest/v1/consent/user/status/{status}	∨ 🔒
GET	/api/rest/v1/consent/user/status/not-null	∨ 🔒
GET	/api/rest/v1/consent/user/services	∨ 🔒
GET	/api/rest/v1/consent/user/service-selected/{selectedService}	∨ 🔒
GET	/api/rest/v1/consent/user/count/pending/{pending}	∨ 🔒
GET	/api/rest/v1/consent/test	∨ 🔒
GET	/api/rest/v1/consent/revoke-all	∨ 🔒



4.2.2 Consent Manager > Event-log: API to manage event logs

The Consent Manager provides the following API to log events (applied actions on consents), and to retrieve those logs.

event-logs-controller ^

POST	/api/rest/v1/consent-event-log/user/{userId}/service/usage	▼	🔒
GET	/api/rest/v1/consent-event-log/user	▼	🔒

4.3 Usage control management

4.3.1 Usage Control > Enforcement API

The Usage Control component provides the enforcement mechanism to apply usage policies according to previously defined data usage policies. This enforcement can be invoked by the following API.

Across Usage Control 1.0 OAS3
/across/AcrossDataUsage/1.0/v3/api-docs
Api Documentation
Apache 2.0

Servers: /across/AcrossDataUsage/1.0 Authorize 🔒

enforce-usage-controller ^

POST	/datausage/enforce	usageControlUse	▼	🔒
------	--------------------	-----------------	---	---

admin-controller ^

POST	/datausage/admin/resetNumAccess	resetNumAccess	▼	🔒
GET	/datausage/admin/access	getAccess	▼	🔒

4.3.2 Usage Control > Policy rules: API to retrieve supported policy rules

The Usage Control component provides the following API to retrieve the policy rules supported by the framework.

policy-rule-controller ^

GET	/api/rest/v1/policy-rule	▼	🔒
-----	--------------------------	---	---



4.3.3 Usage Control > Policies: API for policies management

The Usage Control component provides the following API to manage the usage policies defined by the end user.

usage-policies-controller		^
PUT	/api/rest/v1/usage-policy/{id}	⌵ 🔒
DELETE	/api/rest/v1/usage-policy/{id}	⌵ 🔒
POST	/api/rest/v1/usage-policy	⌵ 🔒
GET	/api/rest/v1/usage-policy/user	⌵ 🔒
GET	/api/rest/v1/usage-policy/user/{userId}/service/{serviceId}/check-data-usage-policies	⌵ 🔒
GET	/api/rest/v1/usage-policy/user/available-services	⌵ 🔒
GET	/api/rest/v1/usage-policy/user/available-services/count	⌵ 🔒

4.4 User journey service engine API

The User Journey Services Engine (UJSE) is the component that carries out the execution of the services defined in a journey workflow.

This component has two main responsibilities:

1. It manages machine-readable orchestration descriptions from the User Journey Modelling Tool and instantiates them into User Journey Services (which are the concrete executable workflows) to the frontend components - and thus the citizens.
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.

For doing so, the following steps are carried out:

- Once the journey workflow is modelled via the User Journey Modelling Tool (UJMT), it is uploaded to the UJSE, via the API offered by the UJSE. The workflow will be composed of phases. Each phase will be composed of actions and each action will be composed of one or more services. The services can be of several types:
 - an offline service to be carried out by the citizen, by contacting directly with the service provider by phone, email, etc.
 - an external web page, where the citizen will be able to carry out the service.



- a synchronous REST service, to be executed via the Citizen Front End (Citizen FE) and the UJSE.
- an asynchronous REST service, to be executed via the Citizen FE and the UJSE
- Then, the citizen will be able to instantiate one of the workflows via the Citizen Front End, which will invoke the UJSE API to carry out this instantiation.
- Once a workflow is instantiated, the citizen will be able to execute the services inside it via the citizen FE.
- If the service is a REST service, the Citizen FE will invoke the UJSE API to carry out its execution, providing the input data specified by the citizen. The UJSE, will first invoke the Transparency Dashboard Backend, to determine if the citizen has given consent to use his/her personal data for the corresponding service, and then it will invoke the Usage Control component to do the enforcement of the usage policies specified by the citizen. If everything is correct, the UJSE will invoke the service, passing as input the data provided by the citizen and for which the citizen has given consent to use. The output of the service execution will be provided by the UJSE to the Citizen FE, so that it can be shown to the citizen.
- The UJSE will log all the activities carried out by the citizen inside the workflows. E.g.: instantiation of a workflow, change of the status of an action/service, etc.

The User Journey Services Engine provides three main APIs:

- **Citizen front-end API:** to interact with the Citizen front-end.
- **Modelling tool API:** to interact with the User Journey Modelling Tool.
- **Log API:** providing logging functionality.

The following sections present the UJSE backend developed.

The detailed description of the APIs is included in ANNEX I

4.4.1 Citizen front-end API controller

It collects all the API/REST that Citizen Front End component can use to work with workflows, execute services that are inside of a concrete workflow, etc.



Across UJSE 1.0 OAS3
/across/1.0v3/api-docs
Api Documentation
Apache 2.0

Servers
/across/1.0 Authorize

citizen-frontend-api-controller

- PUT** /workflowExecutionManagement/workflowName rename workflow name
- PUT** /workflowExecutionManagement/setWorkflowStatus setWorkflowStatus
- PUT** /workflowExecutionManagement/setServiceStatus setServiceStatus
- PUT** /workflowExecutionManagement/setServiceStatusAsynchronous setServiceStatusAsynchronous
- PUT** /workflowExecutionManagement/setActionStatus setActionStatus
- PUT** /workflowExecutionManagement/killWorkflow kill a workflow
- POST** /workflowExecutionManagement/executeService execute step
- POST** /workflowExecutionManagement/createUserWorkflowInstance create a new workflow instance
- GET** /workflowExecutionManagement/getWorkflowStatus getWorkflowStatus
- GET** /workflowExecutionManagement/getUserWorkflowsInstances get list of workflow instances of a user
- GET** /workflowExecutionManagement/getUserWorkflowInstanceById get an existing workflow instance
- GET** /workflowExecutionManagement/getServiceInfo get service info
- DELETE** /workflowExecutionManagement/{workflowInstanceId} delete a workflowInstance

4.4.2 Modelling tool API controller

These are all the API/REST services that User Journey Modelling Tool can use for the workflow administration including operations like, add, update, delete and get a specific workflow by its identifier.

modelling-tool-api-controller

- GET** /workflowManagement getAllWorkflows
- POST** /workflowManagement addOrUpdateWorkflow
- GET** /workflowManagement/getWorkflowById/{workflowId} getWorkflowById
- DELETE** /workflowManagement/{workflowId} delete a Workflow giving workflowId

4.4.3 Log API controller

It collects the services to create a log or to access the already created logs by the User Journey Services Engine component.



log-api-controller		^
GET	/logManagement get logs with no pagination	⌵ 🔒
POST	/logManagement save log	⌵ 🔒
GET	/logManagement/Paged get logs with pagination	⌵ 🔒

5 Installation manuals

The installation of the Data Governance framework and of the User Journey Services Engine are detailed in the following sections.¹³

5.1 Data Governance framework installation

This section provides the description of the installation of the Data Governance framework.

The installation involves the deployment of several components that make up the Data Governance framework. The interaction among these components is shown in the following figure:

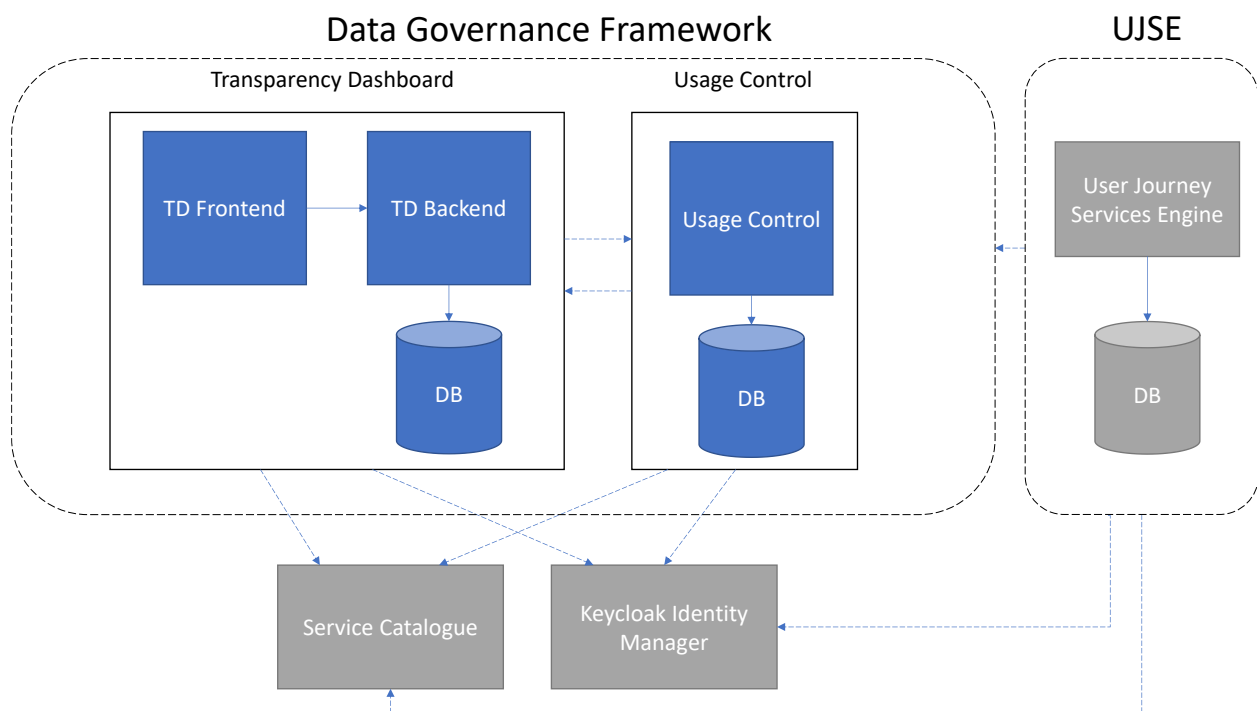


Figure 17 Data governance framework components interaction

Moreover, these components communicate with the following external modules:

- Keycloak Identity Manager: used for end-user and software clients' authentication.
- Service Catalogue: to get information about the services.

¹³ In the current version there are some links to a [private git](#) repository. The final of the software will be published in a [public repository as open source](#).



- User Journey Services Engine (UJSE): the engine asks the Data Governance Framework if the citizen has given consent to the use of his/her personal data and if the usage policies specified by the citizen are accomplished.

Each of the components in the Data Governance Framework correspond to a docker container. These docker containers are the following:

- Transparency Dashboard component:
 - transparency_dashboard_frontend docker container
 - transparency_dashboard_backend docker container
 - MySQL database docker container
- Usage Control component:
 - Datausage docker container
 - PostgreSQL database docker container

The deployment process is explained in the following sections.

5.1.1 Configuring keycloak Identity and Access Manager

First of all, a [Keycloak](#) server must be installed because the Data Governance framework components interact with the Keycloak Identity Manager. Specifically, the Open Id Connect protocol upon the OAuth2 Authentication will be used to authenticate the end users that access to the framework through the Transparency Dashboard Frontend, and to authenticate any software client that wants to use the API-s provided by the Transparency Dashboard Backend and the Usage Control component.

The following steps must be carried out to configure Keycloak:

- Create a realm called “across-dev”.
- Register 3 clients with the following client ID-s:
 - “trans-db-be”:
 - Client protocol: openid-connect
 - Access type: bearer-only
 - Create a role inside the client called “citizen”.
 - “trans-db-fe”:
 - Client protocol: openid-connect
 - Access type: public
 - Create a role inside the client called “citizen”.
 - “usagecontrol”:



- Client protocol: openid-connect
- Access type: bearer-only
- Create a role inside the client called “citizen”.
- Create at least one user in this realm. E.g: “citizen1” and from the “Role Mappings” section, assign the following Client Roles to the user:
 - “trans-db-be” client-> “citizen” Role
 - “trans-db-fe” client-> “citizen” Role
 - “usagecontrol” client-> “citizen” Role

5.1.2 Installing Transparency Dashboard

The Transparency Dashboard consists of the following components/docker containers:

- MySQL database: this component is where all the information related to the consents is stored.
- Transparency Dashboard Backend: this component manages the process of giving/withdrawing consents and logging the activities related to the consents. It also receives notifications related to new services to be invoked by the end-user and it accesses the Service Catalogue to get the information of the personal data required by each service. This component is implemented as Spring Boot services.
- Transparency Dashboard Frontend: this component is the GUI for the end user to manage his consents and to check the actions he has carried out so far with his consents. This component is implemented with Fuse Angular.

These components are installed as docker containers, and the files to do this dockerization are located in the same GitLab repo where the source code is located, that is, at <https://git.code.tecnalia.com/across/private/citizen-front-end/transparency-dashboard/transparency-dashboard-ui> . The dockerization is carried out using the following files:

- docker-compose.yml file located at <https://git.code.tecnalia.com/across/private/citizen-front-end/transparency-dashboard/transparency-dashboard-ui/-/blob/main/docker-compose.yml> .
- mysql.env file to configure the database configuration parameters, and which is located at <https://git.code.tecnalia.com/across/private/citizen-front-end/transparency-dashboard/transparency-dashboard-ui/-/blob/main/mysql.env> .
- transparency_dashboard_backend.env file to configure the Transparency Dashboard Backend (database connection string, Keycloak authorization and URL-s for accessing the Service Catalogue



- and the Usage Control component), and which is located at https://git.code.tecnalia.com/across/private/citizen-front-end/transparency-dashboard/transparency-dashboard-ui/-/blob/main/transparency_dashboard_backend.env .
- environment.ts file that contains the URL-s of the Service Catalogue, Usage Control component and Keycloak, and which is located at https://git.code.tecnalia.com/across/private/citizen-front-end/transparency-dashboard/transparency-dashboard-ui/-/blob/main/transparency_dashboard_frontend/environment.ts .
 - Dockerfile of Transparency Dashboard Backend, located at https://git.code.tecnalia.com/across/private/citizen-front-end/transparency-dashboard/transparency-dashboard-ui/-/blob/main/transparency_dashboard_backend/Dockerfile .
 - Dockerfile of Transparency Dashboard Frontend, located at https://git.code.tecnalia.com/across/private/citizen-front-end/transparency-dashboard/transparency-dashboard-ui/-/blob/main/transparency_dashboard_frontend/Dockerfile .

The following steps must be followed to install the Transparency Dashboard component:

- Install [Git](#), if not installed in your system.
- Get the source code from the repository, by executing the following command at the command prompt:

```
git clone https://git.code.tecnalia.com/across/private/citizen-front-end/transparency-  
dashboard/transparency-dashboard-ui.git  
  
cd transparency-dashboard-ui
```

- Change, if required, the values of the following properties in “transparency_dashboard_backend.env” file, so that they point to the correct Keycloak Authentication Server used, the Usage Control component and the Service Catalogue:
 - `SPRING_SECURITY_OAUTH2_RESOURCESERVER_JWT_ISSUER_URI=https://keycloak-security-dev.k8s.across-h2020.eu/realms/across-dev`
 - `KEYCLOAK_AUTH_SERVER_URL=https://keycloak-security-dev.k8s.across-h2020.eu`
 - `KEYCLOAK_REALM=across-dev`
 - `URL_DATA_USAGE=https://uc-dev.k8s.across-h2020.eu/across/AcrossDataUsage/1.0/`
 - `URL_SERVICE_CATALOG=https://service-catalogue-server-v1.k8s.across-h2020.eu/service-catalogue/api/v2/services`
- Change, if required, the values of the following properties in “environment.ts” file located at the “transparency-dashboard-ui/transparency_dashboard_frontend” folder, so that they point to the

correct Service Catalogue, Usage Control component, Transparency Dashboard Backend, and Keycloak Authentication Server:

- publicServicesApiUrl: "https://service-catalogue-server-v1.k8s.across-h2020.eu/service-catalogue/api/v2/services"
 - dataUsageApiUrl: "https://uc-dev.k8s.across-h2020.eu/across/AcrossDataUsage/1.0"
 - consentManagementApiUrl: "https://transparency-dashboard-be-td-dev.k8s.across-h2020.eu"
 - authApiUrl: "https://keycloak-security-dev.k8s.across-h2020.eu",
 - keycloakUrl: <https://keycloak-security-dev.k8s.across-h2020.eu>
 - keycloakRealm: across-dev
- Install [docker-compose](#) software, if not installed in your system.
 - Execute the following command at the command prompt to run the docker-compose file, which will bring up all the containers of the Transparency Dashboard:

```
docker-compose up -d
```

Now, every component of the Transparency Dashboard will be running. To access the frontend, write the following URL in a Web Navigator at the machine where the installation has been made: <http://localhost:4200/admin>.

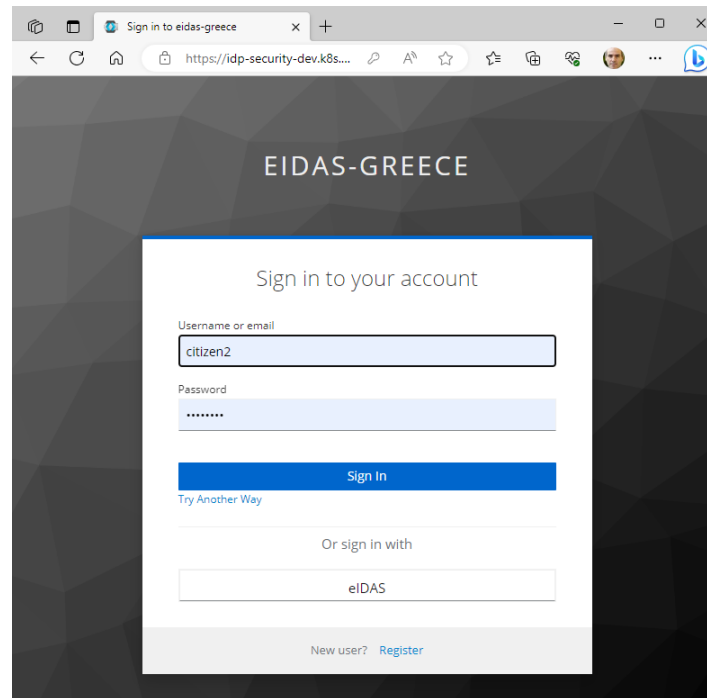


Figure 18 Login window

5.1.3 Installing Usage Control

The Usage Control consists of the following components/docker containers:



- PostgreSQL database: this component is where all the information related to the usage policies is stored.
- Datausage: this component offers an API to manage the usage policies defined by the end user to specify how his personal data can be used (time interval in which the data can be used, number of times the data can be used, etc.) and to do the enforcement of these usage policies. This component is implemented as Spring Boot services.

These components are installed as docker containers, and the files to do this dockerization are located in the same GitLab repo where the source code is located, that is, at <https://git.code.tecnalia.com/across/private/citizen-data-ownership-and-usage-control/usage-control/usagecontrol> . The dockerization is carried out using the following files:

- docker-compose.yml file located at <https://git.code.tecnalia.com/across/private/citizen-data-ownership-and-usage-control/usage-control/usagecontrol/-/blob/main/docker-compose.yml> .
- postgres.env file to configure the database configuration parameters, and which is located at <https://git.code.tecnalia.com/across/private/citizen-data-ownership-and-usage-control/usage-control/usagecontrol/-/blob/main/postgres.env> .
- datausage.env file to configure the Usage Control (database connection string, keycloak authorization and URL-s for accessing the Service Catalogue and the Transparency Dashboard Backend component), and which is located at <https://git.code.tecnalia.com/across/private/citizen-data-ownership-and-usage-control/usage-control/usagecontrol/-/blob/main/datausage.env> .
- Dockerfile of the Usage Control, located at <https://git.code.tecnalia.com/across/private/citizen-data-ownership-and-usage-control/usage-control/usagecontrol/-/blob/main/Dockerfile> .

The following steps must be followed to install the Usage Control component:

- Install [Git](#), if not installed in your system.
- Get the source code from the repository, by executing the following command at the command prompt:

```
git clone https://git.code.tecnalia.com/across/private/citizen-data-ownership-and-usage-control/usage-control/usagecontrol.git  
  
cd usagecontrol
```



- Change, if required, the values of the following properties in “datausage.env” file, so that they point to the correct Data Usage component, Transparency Dashboard Backend, Service Catalogue and Keycloak Authentication Server used:
 - URL_DATA_USAGE_PIP_ENDPOINT=https://uc-dev.k8s.across-h2020.eu/across/AcrossDataUsage/1.0/datausage/admin/access
 - URL_CONSENT_MANAGER=https://transparency-dashboard-be-td-dev.k8s.across-h2020.eu/api/rest/v1
 - URL_SERVICE_CATALOG=https://service-catalogue-server-v1.k8s.across-h2020.eu/service-catalogue/api/v2/services
 - KEYCLOAK_AUTH_SERVER_URL=https://keycloak-security-dev.k8s.across-h2020.eu
 - KEYCLOAK_REALM=across-dev
 - SPRING_SECURITY_OAUTH2_RESOURCESERVER_JWT_ISSUER_URI=https://keycloak-security-dev.k8s.across-h2020.eu/realms/across-dev
- Install [docker-compose](#) software, if not installed in your system.
- Execute the following command at the command prompt to run the docker-compose file, which will bring up all the containers of the Transparency Dashboard:

```
docker-compose up -d
```

Now, every component of the Usage Control will be running. To see the API offered by this component, write the following URL in a Web Navigator at the machine where the installation has been made: <http://localhost:8080/across/AcrossDataUsage/1.0/swagger-ui/index.html?configUrl=/across/AcrossDataUsage/1.0/v3/api-docs/swagger-config>.

5.2 User Journey Service Engine installation

This section provides the description of the installation of the User Journey Services Engine (UJSE).

The installation involves the deployment of several components that make up the UJSE: the Engine and a database. The interaction among these components is shown in the following figure:

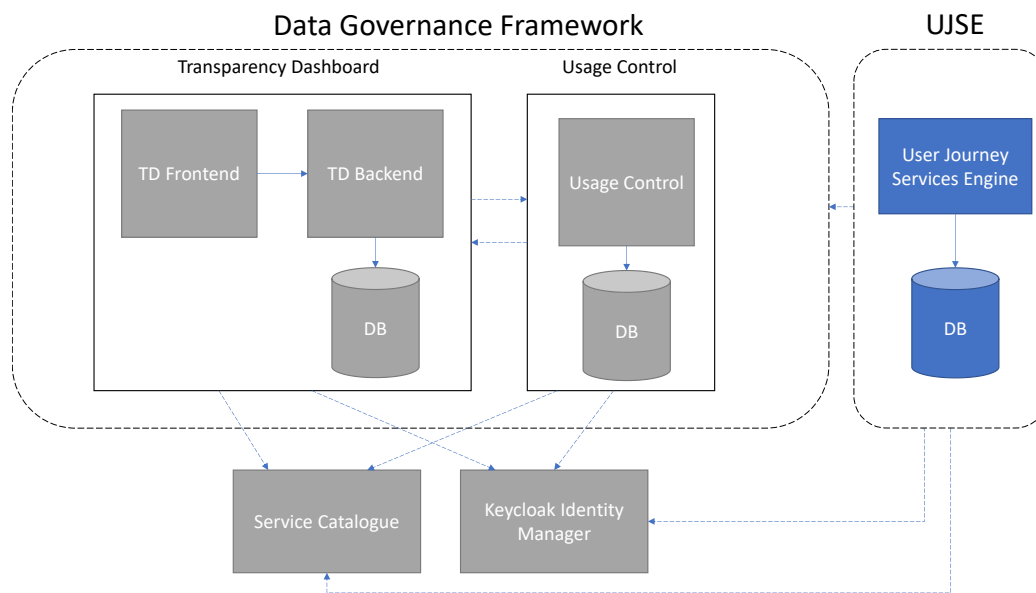


Figure 19 UJSE components interaction

Moreover, these components communicate with the following external modules:

- Keycloak Identity Manager: used for end-user and software clients' authentication.
- Service Catalogue: to get information about the services.
- Data Governance Framework: to check if the citizen has given consent to the use of his/her personal data and if the usage policies specified by the citizen are accomplished.

Each of the components in the UJSE corresponds to a docker container. These docker containers are the following:

- UJSE docker container
- PostgreSQL database docker container

The deployment process is explained in the following sections.



5.2.1 Configuring keycloak Identity and Access Manager

First of all, a [Keycloak](#) server must be installed because the UJSE interacts with the Keycloak Identity Manager. Specifically, the Open Id Connect protocol upon the OAuth2 Authentication will be used to authenticate the software clients that want to use the API-s provided by the engine.

The following steps must be carried out to configure Keycloak:

- Create a realm called “across-dev”.
- Register 1 client with the following client ID:
 - “ujse”:
 - Client protocol: openid-connect
 - Access type: bearer-only
 - Create a role inside the client called “citizen”.
- Create at least one user in this realm. E.g: “citizen1” and from the “Role Mappings” section, assign the following Client Role to the user:
 - “ujse” client-> “citizen” Role

5.2.2 Installing UJSE

The UJSE consists of the following components/docker containers:

- **PostgreSQL database:** this component is where all the information related to the workflow instances is stored.
- **UJSE:** this component offers an API to upload new workflows defined by the User Journey Modelling Tool, to instantiate workflows and to execute the services defined inside them. It accesses the Service Catalogue to get all the information required to execute the services. It also accesses the Transparency Dashboard Backend, to determine if the citizen has given consent to the use of his/her personal data when executing the services, and the Usage Control component, to verify if the usage policies specified by the citizen are fulfilled. It also logs the activities carried out with the workflow instances. E.g.: instantiation of a workflow, change of an action/service status, etc. This component is implemented as Spring Boot services.

These components are installed as docker containers, and the files to do this dockerization are located in the same GitLab repo where the source code is located, that is, at <https://git.code.tecnalia.com/across/private/user-journey-service-delivery/userjourneyservicesengine>.

The dockerization is carried out using the following files:



- docker-compose.yml file located at <https://git.code.tecnalia.com/across/private/user-journey-service-delivery/userjourneyservicesengine/-/blob/main/docker-compose.yml> .
- postgres.env file to configure the database configuration parameters, and which is located at <https://git.code.tecnalia.com/across/private/user-journey-service-delivery/userjourneyservicesengine/-/blob/main/postgres.env> .
- ujse.env file to configure the UJSE (database connection string, Keycloak authorization and URL-s for accessing the Service Catalogue, the Transparency Dashboard and the Usage Control components), and which is located at <https://git.code.tecnalia.com/across/private/user-journey-service-delivery/userjourneyservicesengine/-/blob/main/ujse.env> .
- Dockerfile of UJSE, located at <https://git.code.tecnalia.com/across/private/user-journey-service-delivery/userjourneyservicesengine/-/blob/main/Dockerfile> .

The following steps must be followed to install the UJSE component:

- Install [Git](#), if not installed in your system.
- Get the source code from the repository, by executing the following command at the command prompt:

```
git clone https://git.code.tecnalia.com/across/private/user-journey-service-delivery/userjourneyservicesengine.git

cd userjourneyservicesengine
```

- Change, if required, the values of the following properties in “ujse.env” file, so that they point to the correct Keycloak Authentication Server used, the Transparency Dashboard component, Usage Control component and the Service Catalogue:
 - SERVICE_CATALOG_URI_GETSERVICEINFOENG=https://service-catalogue-server-v1.k8s.across-h2020.eu/service-catalogue/api/v2/services/json/**
 - SERVICE_CATALOG_URI_GETSERVICESINFOENG=https://service-catalogue-server-v1.k8s.across-h2020.eu/service-catalogue/api/v2/services/specified/**?
 - SERVICE_CATALOG_URI_GETPERSONALDATASERVICESENG=https://service-catalogue-server-v1.k8s.across-h2020.eu/service-catalogue/api/v2/services/isPersonalDataHandling
 - TRANSPARENCY_DASHBOARD_MAIN_URI= https://transparency-dashboard-be-td-dev.k8s.across-h2020.eu/api/rest/v1
 - SPRING_SECURITY_OAUTH2_RESOURCESERVER_JWT_ISSUER_URI=https://keycloak-security-dev.k8s.across-h2020.eu/realms/across-dev
 - KEYCLOAK_AUTH_SERVER_URL=https://keycloak-security-dev.k8s.across-h2020.eu
 - KEYCLOAK_REALM=across-dev
- Install [docker-compose](#) software, if not installed in your system.
- Execute the following command at the command prompt to run the docker-compose file, which will bring up all the containers of the Transparency Dashboard:



```
docker-compose up -d
```

Now, every component of the UJSE will be running. To see the API offered by this component, write the following URL in a Web Navigator at the machine where the installation has been made: <http://localhost:8080/across/1.0/swagger-ui/index.html?configUrl=/across/1.0/v3/api-docs/swagger-config/> .



6 Conclusions and future plans

6.1 Conclusions

ACROSS offers the citizen the possibility of defining which public and private organization will be allowed to *access which data and for what purpose* through the **ACROSS Data Governance Framework**. The main aim is to give the citizen the chance of **govern the access to** their data, profiting from a set of usage policies that implement levels of access and they can be the **sovereign owner** of such data.

The ACROSS Personal Data Governance Framework is part of the so-called ACROSS platform, useful in the context of cross-border public/private services for the citizens. However, it has been designed to be an independent framework useful also for the individuals to manage their personal data according to the GDPR in any other context.

ACROSS data governance framework does not cover the following functionalities included in other initiatives:

- **Secure Data Storage and data minimization techniques:** These will be provided by personal wallet technologies being developed by other projects.
- **Secure Data transfer among services.** There already exist secure data transfer protocols implemented, and the concept of Data Spaces is defining new ways of data transfer providing also data sovereignty functionalities to the data owners, both natural and legal persons.

The **service catalogue** functionality is a fundamental part of the personal data framework, since it provides the list of services available to be used by the user, along with all the associated information. However, the catalogue has been designed and implemented as a building block so that the personal data framework is designed to be used with any available catalogue provided that it implements the services catalogue API defined by the framework.

The final version of Data Governance Framework already covers the whole set of required functionalities.

6.2 Future plans

In this final section a brief review of the personal data management trends is presented, including some European initiatives and the personal data management in the context of Data Spaces. The current trends shows that the personal data framework developed can be adapted to be used not only in ACROSS, but in a wider scope. These trends will drive the future evolution and exploitation of the Personal Data Management framework.



6.2.1 European context in personal data management

The concept of Personal Data Governance framework defined in ACROSS is perfectly aligned with the current European legislation and the main initiatives in the field of Personal and Private data management, including:

- **GDPR:** General Data Protection Regulation.
- **Data Governance Act** is a legislative proposal of the European Commission that aims to create a framework which will facilitate data-sharing, fostering the availability of data for use by increasing trust in data intermediaries and by strengthening data-sharing mechanisms across the EU, allowing personal data to be used with the help of a ‘personal data-sharing intermediary’, designed to help individuals exercise their rights under the General Data Protection Regulation (GDPR). The ACROSS Personal Data Governance Framework is perfectly aligned with the **personal data-sharing intermediary** concept
- **Tech Dispatch published by the European Data Protection Supervisor:** ACROSS Data Governance Framework can unambiguously be qualified as a Personal Information Management System (PIMS). As described in greater detail in a 2021 Tech Dispatch published by the European Data Protection Supervisor[9] *“PIMS are products and services that help individuals to have more control over their personal data. PIMS enable individuals themselves to manage and control their online identity.*
- **MyData initiative:** MyData is a global non-profit organisation about digital rights awarded by the European Commission for being one of the most impactful initiatives contributing towards a human centric, trustworthy and sustainable internet. MyData operator is an actor that provides infrastructure for human-centric personal data management and governance. ACROSS Data Governance Framework shares its main capabilities regarding permission and service management and it can be considered a proto-operator. Furthermore, ACROSS extends the MyData operator with data usage control based on data usage policies.

More detailed information about these initiatives and legislation can be found in D3.1 [1]

6.2.2 Data spaces personal data management

Some of the main initiatives in Europe regarding Data Spaces, like IDSA and GAIA-X, are now including in their conceptual model and reference architecture, modules in charge of providing personal data management to be compliant with European legislation (GDPR).

The ACROSS personal data governance framework developed



6.2.2.1 IDSA personal data working group

The initial IDSA Reference Architecture focused on data sharing among companies, and personal data management and protection was not explicitly mentioned. However, recently a new working group has been created to include personal data management in the context of IDS.

Furthermore, two ways of integrating the ACROSS personal data management framework with the IDS architecture have been analysed. (See D3.1[1] and D3.2[3] for more information)

1. **Data usage policies management:** A new user interface has been designed to allow the citizen to define and manage data usage policies associated to services, including the possibility to restrict even more the personal data usage. Furthermore, a new API, **Check Data usage policy** has been added to the personal data governance framework to enforce the data usage policies.
2. New specific **personal data usage policy** to be added to the list of IDS policies. This new policy would allow the IDS connectors to assure that the citizen has given the consent to transfer the personal data from the data provider to the data consumer. However, this option would imply to use an IDS connector to transfer the data, both in the provider and the consumer side.

6.2.2.2 GAIA-X data transfer architecture

The GAIA-X Data Exchange Services Specifications¹⁴ includes in the conceptual model the concept of data licensor and the data usage consent, both including personal data management information:

- **Data licensor:** *A natural or legal participant (not necessarily a Gaia-X participant) who own usage rights for some data. It can be a data subject as per GDPR for personal data or a primary owner of non-personal data (i.e. not liable to GDPR).*
- **Data Usage Consent:** *A legally binding agreement made by and between a Data Producer and Data Provider specifying the terms and conditions to the usage of the data. Data Usage Consent constitutes a legal usage consent for data which are subject to GDPR.*

According to the GAIA-X Data Exchange Services Specifications: *“If a specific Data License is attached to the data (for instance when the data is liable to GDPR), then this co-signed Data Usage Consent constitutes a legal usage consent and must refer to the explicit license rights from the Data Licensor (data subject as per GDPR or data owner). Signed Data Usage Consents are notarized by a **Trusted Consent Authority**.”*

The ACROSS personal data governance framework could be extended to play this important role in the GAIA-X framework.

¹⁴ [Data Exchange Services Specifications - GXFSv2 - Data Exchange - 22.10 Release \(gaia-x.eu\)](#)
Deliverable D3.5 Implementation of the ACROSS Data Governance framework for data sovereignty - Final 959157 — ACROSS — H2020-SC6-GOVERNANCE-2018-2019-2020

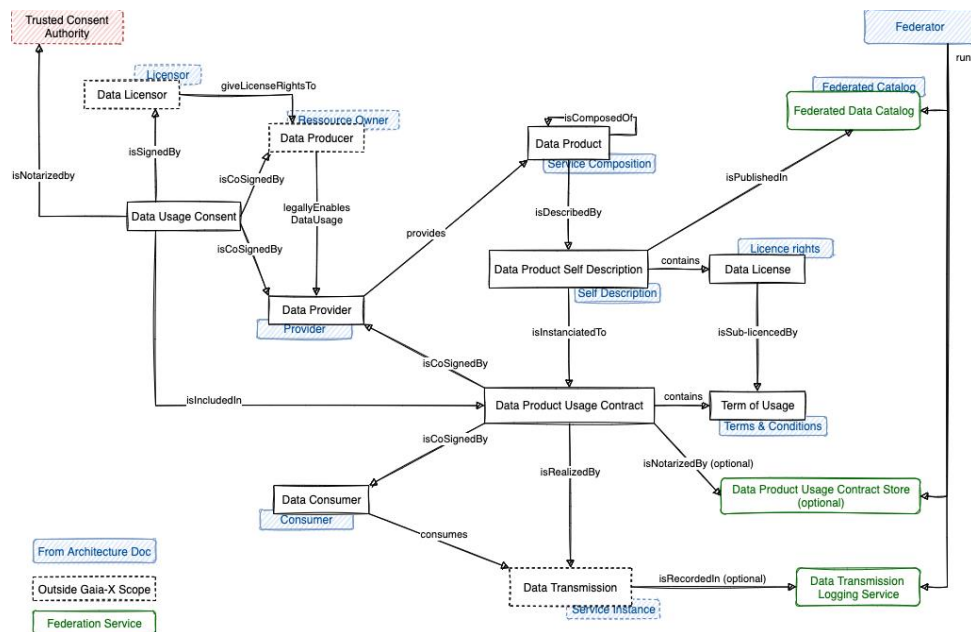


Figure 20 GAIA-X Data Exchange Services Specifications conceptual model

6.2.3 Personal data governance framework future plans.

Two versions of the current ACROSS Personal data governance framework are envisaged: The version integrated within the ACROSS platform and the isolated version ready to be used as a building block to provide personal data management functionalities in other contexts.

The verticalization of the ACROSS personal data framework to be used in the context of the ACROSS platform facilitates the user experience in the context of the ACROSS uses cases. Therefore, this version includes some specific user interface features:

1. **Harmonised look and feel.** The page structure, menus and colour palette have been harmonised.
2. **Integrated language management.** Selected language in the Citizen front end is also applied in the Transparency Dashboard.
3. **Browser management:** It allows the user to navigate through the ACROSS platform user interfaces (both the Transparency Dashboard and the Citizen front end) using the browser navigation buttons.
4. **Dashboard is now the first page.** The about page has been removed.

On the other hand, the ACROSS personal data framework has been designed to be used as an isolated version ready to be used as a building block, playing the role of “personal data intermediary” to be integrated with any service using personal data.



The isolated version would have its own distinct look and feel and the integration with external applications would be based only in APIs. The reasons for that are twofold:

- It is not possible to adapt the user interface to all the services or applications using the personal data framework.
- Since the personal data framework provides a very important and distinct functionality to the individual, it is better to provide a specific UI with its own look and feel.

Two of the envisaged uses of the personal data management framework as a building block could be:

- Implementation of a MyData operator as a general GDPR compliance tool.
- Implementation of a personal data consent management framework in the context of dataspace.



7 References

- [1] D3.1 Design of the ACROSS Data Governance framework for data sovereignty – Initial
- [2] D3.3 Implementation of the ACROSS Data Governance framework for data sovereignty – Initial
- [3] D3.2 Design of the ACROSS Data Governance framework for data sovereignty – Final
- [4] D2.4 Report for cross-border service gap analysis – Final
- [5] D6.2 Use case evaluation and impact assessment– Initial
- [6] D5.2 System Architecture & Implementation Plan – Final
- [7] D3.4 Implementation of the ACROSS Data Governance framework for data sovereignty – Intermediate
- [8] D4.6 Micro Proxies and services catalogue – Final
- [9] TechDispatch #3/2020 – Personal Information Management Systems, 6 January 2021, from the European Data Protection Supervisor; see https://edps.europa.eu/data-protection/our-work/publications/techdispatch/techdispatch-32020-personal-information_en



ANNEX I

Present tables give a description of each API developed inside of User Journey Services Engine:

citizen-front-end-api-controller

REST API	Description
PUT /workflowExecutionManagement /workflowName	Update the name of the workflow instance. Input parameters: <ul style="list-style-type: none">- workflowInstanceId: identifier of the workflow instantiated by the user- name: new name for the workflow Interaction with other components: Citizen Front End
PUT /workflowExecutionManagement /setWorkflowStatus	Set the status to the workflow instance specified by the parameter. Input parameters: <ul style="list-style-type: none">- workflowInstanceId: identifier of the workflow instantiated by the user- status: status for the workflow Interaction with other components: Citizen Front End
PUT /workflowExecutionManagement /setServiceStatus	Set the status of a service that belongs to a specific workflow instance, phase and action. Input parameters: <ul style="list-style-type: none">- workflowInstanceId: identifier of the workflow instantiated by the user- phaseId: phase id.- actionId: action id.- serviceId: service id.- status: status for the service Interaction with other components: Citizen Front End



<p>PUT /workflowExecutionManagement /setStatusAsynchronous</p>	<p>Set the status of a service that belongs to a specific workflow instance, phase and action.</p> <p>Input parameters:</p> <ul style="list-style-type: none">- status: status for the service- token: token includes information of workflowInstanceId + phaseId + actionId + serviceId- serviceOutput: URL that contains the service execution output <p>Interaction with other components: Citizen Front End</p>
<p>PUT /workflowExecutionManagement /setStatus</p>	<p>Set the status of an action that belongs to a specific workflow instance and phase.</p> <p>Input parameters:</p> <ul style="list-style-type: none">- workflowInstanceId: identifier of the workflow instantiated by the user- phaseId: phase id.- actionId: action id.- status: status for the service <p>Interaction with other components: Citizen Front End</p>
<p>PUT /workflowExecutionManagement/ killWorkflow</p>	<p>Kill a concrete workflow instance.</p> <p>Input parameters:</p> <ul style="list-style-type: none">- workflowInstanceId: identifier of the workflow instantiated by the user <p>Interaction with other components: Citizen Front End</p>
<p>POST /workflowExecutionManagement/ executeService</p>	<p>Execute a service that it is included in a workflow instance.</p> <p>Input parameters:</p> <ul style="list-style-type: none">- workflowInstanceId: identifier of the workflow instantiated by the user



	<ul style="list-style-type: none">- phaseId: phase id.- actionId: action id.- serviceId: service id.- token: combination of workflowInstanceId + phaseId + actionId + serviceId <p>Interaction with other components: Citizen Front End</p>
POST /workflowExecutionManagement/createUserWorkflowInstance	<p>Create an instance of a workflow.</p> <p>Input parameters:</p> <ul style="list-style-type: none">- countryOrigin: origin country- countryDestination: destination country- workflowType: workflow type (study/work)- name: name of the workflow <p>Interaction with other components: Citizen Front End</p>
GET /workflowExecutionManagement/getWorkflowStatus	<p>Informs about the status of the specified workflow instance.</p> <p>Input parameters:</p> <ul style="list-style-type: none">- workflowInstanceId: identifier of the workflow instantiated by the user <p>Interaction with other components: Citizen Front End</p>
GET /workflowExecutionManagement/getUserWorkflowInstances	<p>Retrieves all the workflow instances for the logged in user. The user identifier is retrieved from the Identity Server.</p> <p>Input parameters:</p> <ul style="list-style-type: none">- countryOrigin: origin country- countryDestination: destination country- workflowType: workflow type (study/work)



	<p>Interaction with other components: Citizen Front End</p>
<p>GET /workflowExecutionManagement /getUserWorkflowInstanceById</p>	<p>Retrieves a specific workflow instance. The user identifier is retrieved from the Identity Server.</p> <p>Input parameters:</p> <ul style="list-style-type: none"> - workflowInstanceId: identifier of the workflow instantiated by the user <p>Interaction with other components: Citizen Front End</p>
<p>GET /workflowManagement getServiceInfo</p>	<p>Get information about the specified service.</p> <p>Input parameters:</p> <ul style="list-style-type: none"> - serviceId: service id. <p>Interaction with other components: Citizen Front End</p>
<p>DELETE /workflowManagement/{workflowInstanceId}</p>	<p>Delete a workflow instance given its id.</p> <p>Input parameters:</p> <ul style="list-style-type: none"> - workflowInstanceId: id. of the workflow instance to be removed <p>Interaction with other components: Citizen Front End</p>

modelling-tool-api-controller

REST API	Description
<p>GET /workflowManagement getAllWorkflows</p>	<p>Get information about all the workflows uploaded into the UJSE.</p> <p>Input parameters:</p> <ul style="list-style-type: none"> - None <p>Interaction with other components: Citizen Front End</p>
<p>POST /workflowManagement add/update workflow</p>	<p>Upload new or updated workflow to UJSE.</p> <p>Input parameters:</p> <ul style="list-style-type: none"> - File: workflo file in json format



	Interaction with other components: User Journey Modelling Tool
GET /workflowManagement/getWorkflowById/{workflowId}	Retrieve information about a specific workflow. Input parameters: <ul style="list-style-type: none">- workflowId: workflow id. Interaction with other components: User Journey Modelling Tool
DELETE /workflowManagement/{workflowId}	Delete a workflow given its identifier. Input parameters: <ul style="list-style-type: none">- workflowId: workflow id. Interaction with other components: User Journey Modelling Tool

log-api-controller

REST API	Description
GET /logManagement	Get information about logs without pagination. Input parameters: <ul style="list-style-type: none">- workflowInstanceId: identifier of the workflow instantiated by the user- phaseId: phase id.- actionId: action id.- serviceId: service id. Interaction with other components: Citizen Front End
POST /logManagement	Add a new log that corresponds to a specific workflowInstanceId, phaseId, actionId and serviceId. Input parameters: <ul style="list-style-type: none">- description: information regarding the log.- workflowInstanceId: identifier of the workflow instantiated by the user- phaseId: phase id.- actionId: action id.



	<ul style="list-style-type: none">- serviceId: service id. Interaction with other components: Citizen Front End
GET /logManagement/Paged	Get information about logs with pagination. Input parameters: <ul style="list-style-type: none">- workflowInstancelId: identifier of the workflow instantiated by the user- phaseId: phase id.- actionId: action id.- serviceId: service id.- pageNo: page number- pageSize: number of results per page. Interaction with other components: Citizen Front End