Identity Management in PUbLic SErvices

D2.9 [Implementation of basic system]

Lead Author: Xavier Martínez (GRAD)
With contributions from: Jaime Loureiro (GRAD), Pablo Dago Casas (GRAD) and all public administrations (ARH, ERTZ, GIJON, MOP, RVK, UC/IC)
Reviewer: Gianluca Markos (ICERT), Jakob Asmussen (ARH)

Deliverable nature:	Demonstrator (D)
Dissemination level: (Confidentiality)	Public (PU)
Delivery date:	29-07-2022
Version:	1.0
Total number of pages:	29
Keywords:	Configuration, deployment, instantiation, integration, interaction

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101004459

Executive summary

This document summarizes the instantiation of the identity management service-oriented basic system of IMPULSE. This includes all the integrations that a public administration has to perform to connect one of their services to IMPULSE, the configuration of the deployable component of the solution, known as the Enterprise Service, and the deployment of the component itself. It also describes the interaction of a Public Servant with the dashboard of the Enterprise Service, and brief summary of the different sessions that were held with the public administrations regarding the instantiation process.

The IMPULSE solution, as many other eID systems, has two main processes: the onboarding and the authentication. The onboarding process is performed by the natural persons who want to obtain and store an identity verifiable credential in their devices. The users can initiate this process by selecting their public administration from a list, reading a QR code, or clicking a deep link. The two last options of initiating the onboarding have to be integrated by the public administrations if they really want them to be available for the citizens, as they bring a better user experience. The authentication process is performed by the natural persons that want to authenticate themselves against a public administration to receive a service. In this case, it is mandatory that the public administration service has integrated one or both of the previously commented methods: reading a QR code or clicking a deep link. In this process, it also necessary that the public administration service is able to associate an attribute with the current user session of the web browser. For this service to know which user and which web browser to provide the service to after a successful authentication, it is necessary to expose an endpoint where it will receive this information.

To facilitate the whole instantiation process, a configuration tool that aims to automate the configuration and deployment of the Enterprise Service has been developed. This tool allows to configure common Enterprise Service information, Transport Layer Security, logging modes, public servant dashboard authentication, face matching service information and document verification service information. As the Enterprise Service is shared as a docker image, the configuration tool also has the capability of generating a valid docker run command to ease the generation of the container. Finally, this tool is also able to refresh the EBSI Verifiable Authorization of the configured Trusted Issuer, which is the verifiable credential used to receive permission to write information into the EBSI ledger.

Document information

Grant agreement No.	**101004459**	**Acronym**	**IMPULSE**
Full title	**Identity Management in PUbLic SErvices**		
Call	DT-TRANSFORMATIONS-02-2020		
Project URL	https://www.impulse-h2020.eu/		
EU project officer	Giorgio CONSTANTINO		

Deliverable	**Number**	D2.9	**Title**	Implementation of basic system
Work package	**Number**	WP2	**Title**	Co-creative design and piloting
Task	**Number**	T2.4	**Title**	IMPULSE basic system instantiation

Date of delivery	**Contractual**	M18	**Actual**	M18
Status	version 0.9		☐Final version	
Nature	☐Report ☒Demonstrator ☐Other ☐ORDP (Open Research Data Pilot)			
Dissemination level	☒Public ☐Confidential			

Authors (partners)	GRAD, ARH, ERTZ, GIJON, MOP, RVK, UC/IC			
Responsible author	**Name**	Xavier Martínez Luaña		
	Partner	GRAD	**E-mail**	xmartinez@gradiant.org

Summary (for dissemination)	This document summarizes the instantiation of the identity management service-oriented basic system of IMPULSE. This includes all the integrations that a public administration has to perform to connect one of their services to IMPULSE, the configuration of the deployable component of the solution, known as the Enterprise Service, and the deployment of the component itself. It also describes the interaction of a Public Servant with the dashboard of the Enterprise Service, and brief summary of the different sessions that were held with the public administrations regarding the instantiation process. The IMPULSE solution, as many other eID systems, has two main processes: the onboarding and the authentication. The onboarding process is performed by the natural persons who want to obtain and store an identity verifiable credential in their devices. The users can initiate this process by selecting their public administration from a list, reading a QR code, or clicking a deep link. The two last options of initiating the onboarding have to be integrated by the public administrations if they really want them to be available for the citizens, as they bring a better user experience. The authentication process is performed by the natural persons that want to authenticate themselves against a public administration to receive a service. In this case, it is mandatory that the public administration service has integrated one or both of the previously commented methods: reading a QR code or clicking a deep link. In this process, it also necessary that the public administration service is able to associate an attribute with the current user session of the web browser. For this service to know which user and which web browser to provide the service to after a successful authentication, it is necessary to expose an endpoint where it will receive this information. To facilitate the whole instantiation process, a configuration tool that aims to automate the configuration and deployment of the Enterprise Service has been developed. This tool allows to configure common Enterprise Service information, Transport Layer Security, logging modes, public servant dashboard authentication, face matching service information and document

	verification service information. As the Enterprise Service is shared as a docker image, the configuration tool also has the capability of generating a valid docker run command to ease the generation of the container. Finally, this tool is also able to refresh the EBSI Verifiable Authorization of the configured Trusted Issuer, which is the verifiable credential used to receive permission to write information into the EBSI ledger.
Keywords	Configuration, deployment, instantiation, integration, interaction

Version Log			
Issue Date	**Rev. No.**	**Author**	**Change**
10-03-2022	**0.1**	**Xavier Martínez Luaña**	**First Integration section**
30-03-2022	**0.2**	**Xavier Martínez Luaña, Pablo Dago Casas**	**Updates on Integration section and new Deployment section**
25-04-2022	**0.3**	**Xavier Martínez Luaña**	**New Instantiation section describing the configuration tool and new deployment method**
12-05-2022	**0.4**	**Xavier Martínez Luaña**	**New Interaction and Refreshing Trusted Issuer Sections**
29-06-2022	**0.5**	**Xavier Martínez Luaña, PAs**	**Updates on several sections after feedback from public administrations**
05-07-2022	**0.6**	**Xavier Martínez Luaña, PAs**	**Updates on several sections after more feedback from public administrations**
12-07-2022	**0.7**	**Xavier Martínez Luaña**	**New instantiation Activities section**
20-07-2022	**0.8**	**Xavier Martínez Luaña, Jaime Loureiro Acuña**	**Added missing sections (executive summary, list of figures, abbreviations, definitions, conclusions and references)**
21-07-2022	**0.9**	**Jaime Loureiro Acuña**	**First revision of the document prior to official internal review**
28-07-2022	**1.0**	**Xavier Martínez Luaña, Gianluca Markos, Jakob Asmussen**	**Official internal review**

Table of contents

List of figures

Abbreviations and acronyms

CPU: Central Processing Unit
DID: Decentralized IDentifier
GB: GigaByte
HTTP: HyperText Transfer Protocol
IT: Information Technology
JKS: Java KeyStore
JSON: JavaScript Object Notation
JWT: JSON Web Token
PA: Public Administration
PAS: Public Administration Service
QR Code: Quick Response Code
RAM: Random Access Memory
TLS: Transport Layer Security
URL: Uniform Resource Location

Definitions

Docker: It is an open-source project to automate the deployment of applications inside software containers, providing an additional layer of abstraction and automation of virtualization across multiple operating systems.

Dockerization: It is the process of packing, deploying and running applications using Docker containers.

1 Introduction

One of the main aspects to be taken into account when implementing IMPULSE is how the solution will be instantiated from the point of view of a public administration. These legal entities have a huge variety of environments, with different technologies, and with very diverse IT teams. This raised the need to create a simple installation process that emulates a plug & play system, so it could be adapted to almost every public administration in a relatively simple way. As it is reflected in section 6, the main idea behind the basic system instantiation was adapted to the needs that we continue to encounter from these institutions.

The two most important actions that we carried out to speed up this process was the dockerization of the Enterprise Service and the creation of the configuration tool. Some technical details of both are shown below:

Dockerization of the Enterprise Service:

Docker image details:

- **Base image**: openjdk:16-slim-bullseye
- **Enterprise Service** added as a JAR
- **Common configuration files** added (service-matrix.properties, fsStore.conf, signatory.conf)
- **Entrypoint**: java –jar EnterpriseService.jar
- **Size**: 882MB
- **Version**: 1.1.1

Configuration Tool:

A Configuration Tool has been created to facilitate the public administrations the configuration and deployment of the Enterprise Service. It has been created as Command Line Interface (CLI) using the clikt library [1]. It is designed to be interactive and self-explanatory, i.e., everything that can be done with the tool is explained within the tool itself.

```
Usage: ConfigurationTool [OPTIONS] COMMAND [ARGS]...

  It allows to onboard an EBSI DID, configure the Enterprise Service, help in
  the deployment and refreshes the Trusted Issuers' EBSI Verifiable
  Authorization.

Options:
  -h, --help  Show this message and exit

Commands:
  onboard     EBSI DID Onboarding related operations
  configure   Enterprise Service configuration
  credential  Verifiable Identity credential related operations
  deployment  Docker deployment related operations
```

Figure 1: Configuration Tool CLI.

2 Integration with a Public Administration Service

From the public administration perspective, the Enterprise Service of the IMPULSE solution is the key component to which they have to be integrated in order to use IMPULSE as an eID system.

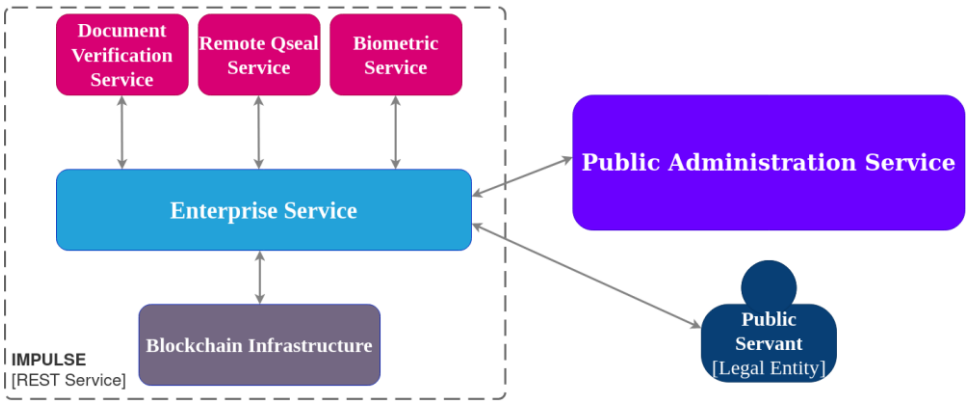

Figure 2: General Integration of a PA with the IMPULSE solution.

Here are described all the integrations needed to connect a Public Administration Service with the IMPULSE Enterprise Service. The IMPULSE Solution has two main flows: **onboarding** (user obtains an Identity Verifiable Credential), **authentication** (user authenticates using the Identity Verifiable Credential). All the integrations needed in both operations are explained below.

2.1 Onboarding

In the onboarding process it is not mandatory to do any integration, but a few can be addressed to give a better user experience to the citizens. When a user wants to make the onboarding, there are three possible options:

- The user selects their public administration from a list inside the IMPULSE application.
- The user uses the IMPULSE application to scan a QR Code in the public administration website.
- The user clicks on a Deep Link in the public administration website, using a web browser installed on their mobile device, which redirects them to the IMPULSE application.

Since the first option doesn't require any integration, it's always available, and the two others are completely optional. For the public administrations to address these optional ways of initiating the onboarding process, they have to make the Public Administration Service to connect to the Enterprise Service, requesting the content of the QR Code/ Deep Link. The request is in fact an HTTP GET Operation to the endpoint **/users-onboarding/v1/session**. The response of these endpoint is a JSON like the next one:

```
{
    "authenticationRequest":
"openid://?response_type=id_token&client_id=did%3Aebsi%3AzkqSHiqQSH1jk6U6
846ebmu&scope=openid+did_authn&request=eyJraWQiOiI5NjA2MDZhYjA0NjQ0ZDhlYm
Y3MWZiYTkzYzlhYjBlZSIsInR5cCI6IkpXVCIsImFsZyI6IkVkRFNBIn0.eyJhdXRoZW50aWN
hdGlvblJlcXVlc3RKd3QiOnsiYXV0aEhlYWRlciI6eyJ0eXAiOiJKV1QiLCJhbGciOiJFZERT
QSIsImp3ayI6eyJrdHkiOiJPS1AiLCJjcnYiOiJFZDI1NTE5IiwidXNlIjoic2lnIiwia2lkI
joiOTYwNjA2YWIwNDY0NGQ4ZWJmNzFmYmE5M2M5YWIwZWUiLCJ4IjoiZHliaDgtYTZxY2l0bj
JwQzhFSE44V1hpbUJJcmJSM1JfanczQ1RvWXVvMCIsImFsZyI6IkVkRFNBIn19LCJhdXRoUmV
xdWVzdFBheWxvYWQiOnsic2NvcGUiOiJvcGVuaWQgZGlkX2F1dGhuIiwiY2xhaW1zIjp7Imlk
VG9rZW4iOnsidmVyaWZpZWRDbGFpbXMiOm51bGx9fSwiaXNzIjoiZGlkOmVic2k6emtxU0hpc
VFTSDFqazZVNjg0NmVibXUiLCJyZXNwb25zZV90eXBlIjoiaWRfdG9rZW4iLCJyZWdpc3RyYX
Rpb24iOnsiYWNjZXNzX3Rva2VuX2VuY3J5cHRpb25fZW5jX3ZhbHVlc19zdXBwb3J0ZWQiOls
iQTEyOEdDTSIsIkEyNTZHQ00iXSwiandrc191cmkiOiIiLCJyZWRpcmVjdF91cmlzIjpbIiJd
LCJyZXF1ZXN0X29iamVjdF9zaWduaW5nX2FsZyI6WyJFUzI1NksiLCJFZERTQSJdLCJhY2Nlc
3NfdG9rZW5fc2lnbmluZ19hbGciOlsiRVMyNTZLIiwiRWREU0EiXSwiYWNjZXNzX3Rva2VuX2
VuY3J5cHRpb25fYWxnX3ZhbHVlc19zdXBwb3J0ZWQiOlsiRUNESC1FUyJdLCJpZF90b2tlbl9
zaWduZWRfcmVzcG9uc2VfYWxnIjpbIkVTMjU2SyIsIkVkRFNBIl0sInJlc3BvbnNlX3R5cGVz
IjoiaWRfdG9rZW4ifSwibm9uY2UiOiIzYTEzOWY2Mi1kZWQ3LTQxZTAtYmU4YS1lYWQ0Njg2M
mNmMjAiLCJjbGllbnRfaWQiOiJkaWQ6ZWJzaTp6a3FTSGlxUVNIMWprNlU2ODQ2ZWJtdSJ9fS
wic2NvcGUiOiJvcGVuaWQgZGlkX2F1dGhuIiwiY2FsbGJhY2siOiJodHRwOlwvXC80MC43MS4
1Ny45ODo4MDgwXC91c2Vycy1vbmJvYXJkaW5nXC92MVwvYXV0aGVudGljYXRpb24tcmVzcG9u
c2VzIiwicmVzcG9uc2VfdHlwZSI6ImlkX3Rva2VuIiwibm9uY2UiOiIzYTEzOWY2Mi1kZWQ3L
TQxZTAtYmU4YS1lYWQ0Njg2MmNmMjAiLCJjbGllbnRfaWQiOiJkaWQ6ZWJzaTp6a3FTSGlxUV
NIMWprNlU2ODQ2ZWJtdSJ9.V0craly-40Md9waPfl-WtDkpzaBvtIGzLz9CJAE-
uj59WojMyUxcaQG8hWnectoCvco-bX4ZtP770Wgy1a3EBQ&nonce=3a139f62-ded7-41e0-
be8a-ead46862cf20"
}
```


The **authenticationRequest** field is the text string that has to be extracted and converted into a QR Code or a Deep Link.

2.1.1 Show a QR Code

To convert this **authenticationRequest** field into a QR Code, the Public Administration Service will have to make use of any library supported by their programming language that serves for the conversion of text strings into QR codes.

The following is a compilation of some of the libraries for this task from among the most popular web development programming languages:

- Javascript/Typescript: **node-qrcode** [2]
- Python: **python-qrcode** [3]
- Java: **zxing** [4]
- C#: **QRCoder** [5]

Once the conversion is complete, the QR entered on the web page would look like the figure 3:

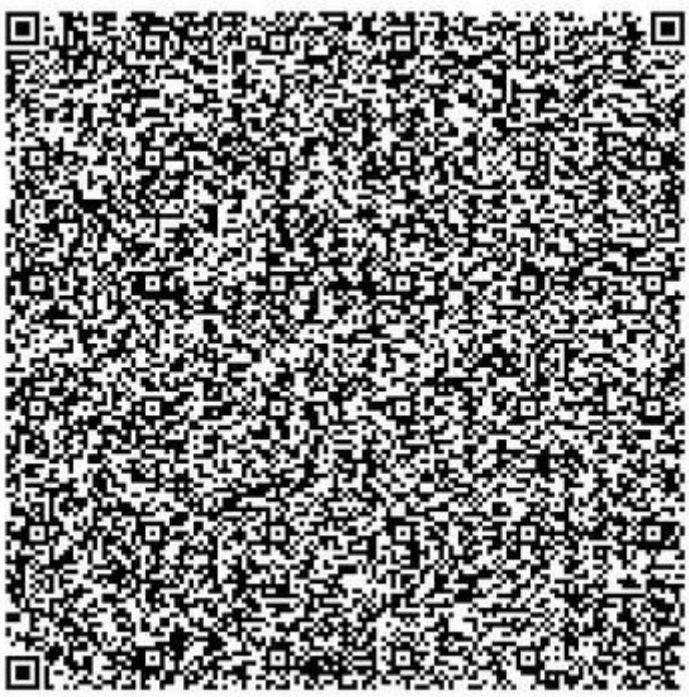

Figure 3: QR shown in the PA Web Application.

2.1.2 Show a Deep Link

To convert this **authenticationRequest** field into a Deep Link, the Public Administration Service will have to make use of any library supported by their programming language that serves for the encoding of UTF-8 text strings into Base64Url strings.

The following is a compilation of some of the libraries for this task from among the most popular web development programming languages:

- Javascript/Typescript: **base64url** [6]
- Python: **base64** [7]
- Java: **Base64** [8]
- C#: **base64urlencoder** [9]

After converting the **authenticationRequest** field into a Base64Url string, the result must be joined with the prefix **impulse://impulse_app/onboarding?request=**. The final Deep Link will look like the next one:

impulse://impulse_app/onboarding?request=b3BlbmlkOi8vP3Jlc3BvbnNlX3R5cGU9aWRfdG9rZW4mY2xpZW50X2lkPWRpZCUzQWVic2klM0F6a3FTSGlxUVNIMWprNlU2ODQ2ZWJtdSZzY29wZT1vcGVuaWQrZGlkX2F1dGhuJnJlcXVlc3Q9ZXlKcmFXUWlPaUk1TmpBMk1EWmhZakEwTmpRMFpEaGxZbVkzTVdaaVlUa3pZemxoWWpCbFpTSXNJblI1Y0NJNklrcFhWQ0lzSW1Gc1p5STZJa1ZrUkZOQkluMC5leUpoZFhSb1pXNTBhV05oZEdsdmJsSmxjWFZsYzNSS2QzUWlPbnNpWVhWMGFFaGxZV1JsY2lJNmV5SjBlWEFpT2lKS1YxUWlMQ0poYkdjaU9pSkZaRVJUUVNJc0ltcDNheUk2ZXlKcmRIa2lPaUpQUzFBaUxDSmpjbllpT2lKRlpESTFOVEU1SWl3aWRYTmxJam9pYzJsbklpd2lhMmxrSWpvaU9UWXdOakEyWVdJd05EWTBOR1E0WldKbU56Rm1ZbUU1TTJNNVlXSXdaV1VpTENKNElqb2laSGxpYURndFlUWnhZMmwwYmpKd1F6aEZTRTQ0VjFocGJVSkpjbUpTTTFKZmFuY3pRMVJ2V1hWdk1DSXNJbUZzWnlJNklrVmtSRk5CSW4xOUxDSmhkWFJvVW1WeGRXVnpkRkJoZVd4dllXUWlPbnNpYzJOdmNHVWlPaUp2Y0dWdWFXUWdaR2xrWDJGMWRHaHVJaXdpWTJ4aGFXMXpJanA3SW1sa1ZHOXJaVzRpT25zaWRtVnlhV1pwWldSRGJHRnBiWE1pT201MWJHeDlmU3dpYVhOeklqb2laR2xrT21WaWMyazZlbXR4VTBocGNWRlRTREZxYXpaVk5qZzBObVZpYlhVaUxDSnlaWE53YjI1elpWOTBlWEJsSWpvaWFXUmZkRzly

Wlc0aUxDSnlaV2RwYzNSeVlYUnBiMjRpT25zaVlXTmpaWE56WDNSdmEyVnVYMlZ1WTNKNWNI
UnBiMjVmWlc1algzWmhiSFZsYzE5emRYQndiM0owWldRaU9sc2lRVEV5T0VkRFRTSXNJa0V5TlRaS
FEwMGlYU3dpYW5kcmMxOTFjbWtpT2lJaUxDSnlaV1JwY21WamRGOTFjbWx6SWpwYklpSmRMQ0
p5WlhGMVpYTjBYMjlpYW1WamRGOXphV2R1YVc1blgyRnNaeUk2V3lKRlV6STFOa3NpTENKRlpF
UlRRU0pkTENKaFkyTmxjM05mZEc5clpXNWZjMmxuYm1sdVoxOWhiR2NpT2xzaVJWTXlOVFpMS
Wl3aVJXUkVVMEVpWFN3aVlXTmpaWE56WDNSdmEyVnVYMlZ1WTNKNWNIUnBiMjVmWVd4bl
gzWmhiSFZsYzE5emRYQndiM0owWldRaU9sc2lSVU5FU0MxRlV5SmRMQ0pwWkY5MGIydGxibDl6Y
VdkdVpXUmZjbVZ6Y0c5dWMyVmZZV3huSWpwYklrVlRNalUyU3lJc0lrVmtSRk5CSWwwc0luSmxjM
0J2Ym5ObFgzUjVjR1Z6SWpvaWFXUmZkRzlyWlc0aWZTd2libTl1WTJVaU9pSTRPRFZrT0ROaFppMD
BOV0kxTFRReU1qa3RZbVV4TXkwMk5HUmlZVEJqT0RKaU1qY2lMQ0pqYkdsbGJuUmZhV1FpT2lKa
2FXUTZaV0p6YVRwNmEzRlRTR2x4VVZOSU1XcHJObFUyT0RRMlpXSnRkU0o5ZlN3aWMyTnZjR1
VpT2lKdmNHVnVhV1FnWkdsa1gyRjFkR2h1SWl3aVkyRnNiR0poWTJzaU9pSm9kSFJ3T2x3dlhDODB
NQzQzTVM0MU55NDVPRG80TURnd1hDOTFjMlZ5Y3kxdmJtSnZZWEprYVc1blhDOTJNVnd2WVhW
MGFHVnVkR2xqWVhScGIyNHRjbVZ6Y0c5dWMyVnpJaXdpY21WemNHOXVjMlZmZEhsd1pTSTZJb
WxrWDNSdmEyVnVJaXdpYm05dVkyVWlPaUk0T0RWa09ETmhaaTAwTldJMUxUUXlNamt0WW1Ve
E15MDJOR1JpWVRCak9ESmlNamNpTENKamJHbGxiblJmYVdRaU9pSmthV1E2WldKemFUcDZhM0Z
UU0dseFVWTklNV3ByTmxVMk9EUTJaV0p0ZFNKOS5jUGE0NVZFeVJPOTRaVmFtRXV3ZHA5dVIx
bVNlUnlYWGtkWk9lcURMUWVMR1F2OHpLd0pPUmcxUmNmOUtpUHBRUlZVaVVWbzZ6Tzlqd0hja
kRmVTZBdyZub25jZT04ODVkODNhZi00NWI1LTQyMjktYmUxMy02NGRiYTBjODJiMjc

2.2 Authentication

In the authentication process it is mandatory to do some integrations. Since the Enterprise Service is in charge of the authentication mechanism, but the user accesses the service through a web browser, it's absolutely necessary for the Public Administration to display a QR, a Deep Link or both to the user. This also implies that the Public Administration Service will have to open (and protect) an endpoint to receive notifications of which user has successfully logged in.

For the public administrations to address one (or both) of these ways of initiating the Authentication process, they have to make the Public Administration Service to connect to the Enterprise Service, requesting the content of the QR Code/ Deep Link. The request is in fact an HTTP GET Operation to the endpoint **/verification/v1/session**. The response of these endpoint is a JSON like the next one:


```
{
    "sessionId": "c2b3e0f5-89b8-4d8f-8325-20228af1b8b5",
    "presentationRequest":
"openid://?response_type=id_token&client_id=did%3Aebsi%3AzkqSHiqQSH1jk6U6
846ebmu&scope=openid+did_authn&request=eyJraWQiOiI5NjA2MDZhYjA0NjQ0ZDhlYm
Y3MWZiYTkzYzlhYjBlZSIsInR5cCI6IkpXVCIsImFsZyI6IkVkRFNBIn0.eyJhdXRoZW50aWN
hdGlvblJlcXVlc3RKd3QiOnsiYXV0aEhlYWRlciI6eyJ0eXAiOiJKV1QiLCJhbGciOiJFZERT
QSIsImp3ayI6eyJrdHkiOiJPS1AiLCJjcnYiOiJFZDI1NTE5IiwidXNlIjoic2lnIiwia2lkI
joiOTYwNjA2YWIwNDY0NGQ4ZWJmNzFmYmE5M2M5YWIwZWUiLCJ4IjoiZHliaDgtYTZxY2l0bj
JwQzhFSE44V1hpbUJJcmJSM1JfanczQ1RvWXVvMCIsImFsZyI6IkVkRFNBIn19LCJhdXRoUmV
xdWVzdFBheWxvYWQiOnsic2NvcGUiOiJvcGVuaWQgZGlkX2F1dGhuIiwiY2xhaW1zIjp7Imlk
VG9rZW4iOnsidmVyaWZpZWRDbGFpbXMiOnsidmVyaWZpY2F0aW9uIjp7ImV2aWRlbmNlIjp7I
mRvY3VtZW50Ijp7ImNyZWRlbnRpYWxTY2hlbWEiOnsiaWQiOnsidmFsdWUiOiJodHRwczpcL1
wvYXBpLnByZXByb2QuZWJzaS5ldVwvdHJ1c3RlZC1zY2hlbWFzLXJlZ2lzdHJ5XC92MVwvc2N
oZW1hc1wvMHg2MDllMmNhMjIzMzI1M2U3NGUwZGNhOGRjOTUyNjAwYjYxZDQ2YWZmYzMxOGFh
NGE3MGEyOGViNzE4OTUyZTNlIiwiZXNzZW50aWFsIjp0cnVlfX0sInR5cGUiOnsidmFsdWUiO
lsiVmVyaWZpYWJsZUNyZWRlbnRpYWwiLCJWZXJpZmlhYmxlSWQiXSwiZXNzZW50aWFsIjp0cn
VlfX0sInR5cGUiOnsidmFsdWUiOiJ2ZXJpZmlhYmxlX2NyZWRlbnRpYWwiLCJlc3NlbnRpYWw
iOm51bGx9fSwidHJ1c3RfZnJhbWV3b3JrIjoiRUJTSSJ9fX19LCJpc3MiOiJkaWQ6ZWJzaTp6
a3FTSGlxUVNIMWprNlU2ODQ2ZWJtdSIsInJlc3BvbnNlX3R5cGUiOiJpZF90b2tlbiIsInJlZ
2lzdHJhdGlvbiI6eyJhY2Nlc3NfdG9rZW5fZW5jcnlwdGlvbl9lbmNfdmFsdWVzX3N1cHBvcn
```

RlZCI6WyJBMTI4R0NNIiwiQTI1NkdDTSJdLCJqd2tzX3VyaSI6IiIsInJlZGlyZWN0X3VyaXM
iOlsiIl0sInJlcXVlc3Rfb2JqZWN0X3NpZ25pbmdfYWxnIjpbIkVTMjU2SyIsIkVkRFNBIl0s
ImFjY2Vzc190b2tlbl9zaWduaW5nX2FsZyI6WyJFUzI1NksiLCJFZERTQSJdLCJhY2Nlc3Nfd
G9rZW5fZW5jcnlwdGlvbl9hbGdfdmFsdWVzX3N1cHBvcnRlZCI6WyJFQ0RILUVTIl0sImlkX3
Rva2VuX3NpZ25lZF9yZXNwb25zZV9hbGciOlsiRVMyNTZLIiwiRWREU0EiXSwicmVzcG9uc2V
fdHlwZXMiOiJpZF90b2tlbiJ9LCJub25jZSI6ImMyYjNlMGY1LTg5YjgtNGQ4Zi04MzI1LTIw
MjI4YWYxYjhiNSIsImNsaWVudF9pZCI6ImRpZDplYnNpOnprcVNIaXFRU0gxams2VTY4NDZlY
m11In19LCJzY29wZSI6Im9wZW5pZCBkaWRfYXV0aG4iLCJjYWxsYmFjayI6Imh0dHA6XC9cLz
QwLjcxLjU3Ljk4OjgwODBcL3ZlcmlmaWNhdGlvblwvdjFcL2F1dGhlbnRpY2F0aW9uLXJlc3B
vbnNlcyIsInJlc3BvbnNlX3R5cGUiOiJpZF90b2tlbiIsIm5vbmNlIjoiYzJiM2UwZjUtODli
OC00ZDhmLTgzMjUtMjAyMjhhZjFiOGI1IiwiY2xpZW50X2lkIjoiZGlkOmVic2k6emtxU0hpc
VFTSDFqazZVNjg0NmVibXUifQ.HXiTSArbXm-
6YCxFTAJuBx17xRAG6GVLKbp7UFbiYsk2PiOdwq2p8YNS9aSLAtv3ouY3OJzBOu7oLMn5O8cK
CQ&nonce=c2b3e0f5-89b8-4d8f-8325-20228af1b8b5"
}

The **presentationRequest** field is the text string that has to be extracted and converted into a QR Code or a Deep Link.

2.2.1 Show a QR Code

It is exactly the same procedure as in section 2.1.1, except that the converted text string is this time the **presentationRequest** field instead of the **authenticationRequest** field.

2.2.2 Show a Deep Link

It is exactly the same procedure as in section 2.1.2, except in two aspects:

- The converted text string is this time the **presentationRequest** field instead of the field **authenticationRequest**.
- After converting the **presentationRequest** field into a Base64Url string, the result must be joined with the prefix **impulse://impulse_app/authentication?request=**.

2.2.3 Handle the user session

Since the Authentication process occurs between the IMPULSE Application and the Enterprise Service, but the service is provided through other means (web service, locker...), it's mandatory to handle the user session using the **sessionId** field received in the endpoint **/verification/v1/session**. And this **sessionId** has to be linked in some way with user session inside the device where the service is being provided. This will imply that later on the Public Administration Service will be able to identify which device needs to receive the provided service.

The common scenario for the public administrations is that their web application handles the user sessions using some of the next mechanisms: cookies, JWTs, HTTP sessions, etc. The only required integration here is to associate this **sessionId** field to the user session, using the mechanism already in place. It is up to the public administration to decide how long this **sessionId** field is valid and how often it should be refreshed.

2.2.4 Exposing the endpoint to receive the notification

After a citizen has successfully authenticated to the Enterprise Service, this component has to communicate to the Public Administration Service that a user has logged in. This communication will be in fact an HTTP POST Operation to one endpoint exposed from the Public Administration Service for this purpose. This endpoint must receive two parameters:

1. **credentialSubject**: JSON that contains the personal attributes of the Identity Verifiable Credential presented by the citizen. This is the field used by the Public Administration Service to identify the citizen that has logged in. Example:


```
{
"firstName": "John",
    "familyName": "Smith",
    "dateOfBirth": "01-01-1900",
    "personaIdentifier": "12345678A",
    "placeOfBirth": "Fakeland",
    "currentAddress": "Fake Street 123",
    "gender": "male"
}
```


2. **sessionId**: String associated to a user session inside the Public Administration Service. This is the field used by the Public Administration Service to identify the browser/device that needs to receive the service.


```
c2b3e0f5-89b8-4d8f-8325-20228af1b8b5
```

3 Instantiation of the Enterprise Service

The instantiation process is thought to be very straightforward for any public administration that wants to deploy IMPULSE and connect to it. The Enterprise Service, which is the core element that will need to be instantiated by the public administrations, has been dockerized to facilitate the sharing and deployment of the component. To ease the configuration of this docker image prior to its deployment, a Configuration Tool has been created along with a manual that explains every step needed to perform an integration and instantiation of the IMPULSE solution.

3.1 Resources

1. Docker Image of the Enterprise Service.
    - It is shared through a robot account (only reading permissions) of a private docker repository (Harbor) owned by GRAD.
2. Configuration Tool alongside the configuration files.
    - **alice.properties**: Face Matching Service configuration file.
    - **dashboard.properties**: Public Servant Dashboard configuration file.
    - **enterprise-service.properties**: Enterprise Service common configuration file.
    - **logging.properties**: Logging configuration file.
    - **tls.properties**: Transport Layer Security configuration file.
    - **tree.properties**: Document Verification Service configuration file.
    - **trusted-issuer.properties**: Trusted Issuer configuration file.
    - **service-matrix.properties**: WaltId SSIKit Services configuration file.
    - **fsStore.conf**: WaltId SSIKit File System Store configuration file.
    - **signatory.conf**: WaltId SSIKit Signatory Service configuration file.
3. Data folder.
    - To facilitate the instantiation for the first pilot we have already onboarded one different DID for every public administration, and contacted EBSI's Service Desk to register these DIDs as Trusted Issuers and ESSIF Onboarding Services. This means that each public administration receives a pre-configured **data/** folder with everything they need to setup IMPULSE.
4. Manual "How to IMPULSE: Integration, Instantiation and Interaction".

3.2 Requirements

- Java 16+
- Docker 20+
- Virtual CPUs: 2+
- RAM: 8GB+

3.3 Configurations

In the following sections, all of the possible configurations of the Enterprise Service will be addressed, regardless of whether they are mandatory or optional.

3.3.1 Enterprise Service Common Configuration

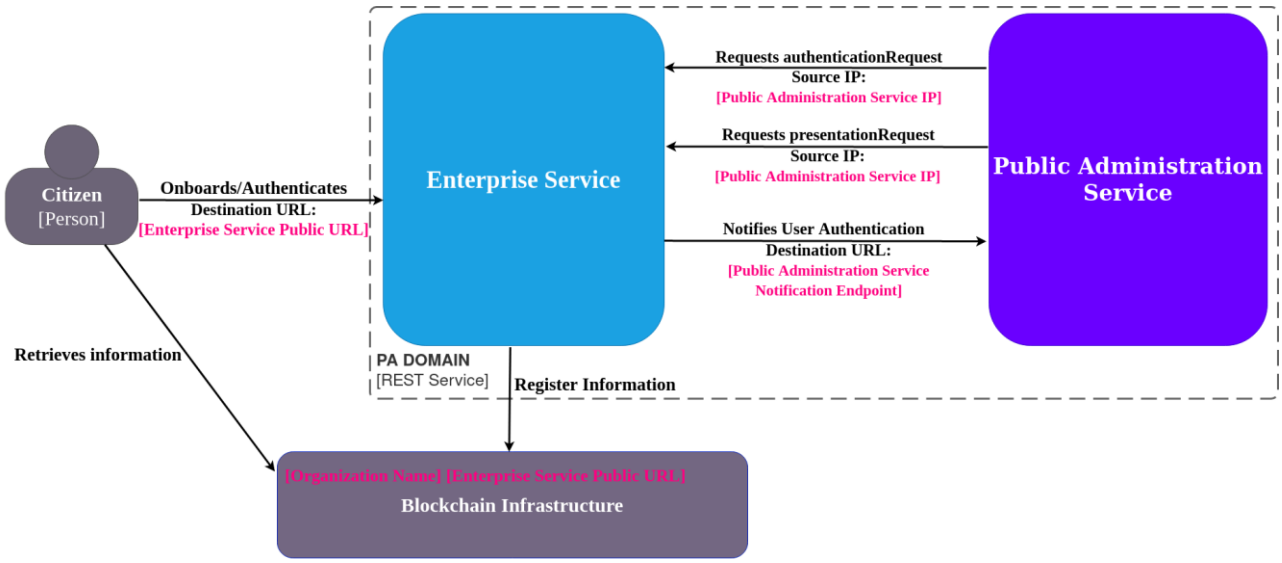

Figure 4: Overview of the role of the Enterprise Service Common configurable attributes.

The next attributes are all the possible configurations related to the common information of the Enterprise Service.

- **Organization Name**: It's not really an important value for now, but it should be the name of the public administration deploying this Enterprise Service.
    - Example: Gradiant - Centro Tecnolóxico de Galicia
    - Example: GRADIANT

- **Enterprise Service Public URL**: The public URL where the Enterprise Service is exposed. It has to be the complete URL (with port in case it differs from the default ones), and it can contain a domain or an IP address.
    - Example: https://www.gradiant.org/enterprise-service
    - Example: https://www.gradiant.org:8443/enterprise-service
    - Example: http://1.1.1.1
    - Example: http://1.1.1.1:8080

** This attribute also serves to configure the context path of the Enterprise Service. For instance, if you configure an URL like the one in the first example, the context path will be "/enterprise-service".

- **Public Administration Service Notification Endpoint**: The public/private URL where the Public Administration Service has the endpoint exposed. It has to be the complete path to the endpoint.
    - Example: https://www.gradiant.org/callback
    - Example: http://1.1.1.1:8080/notification

- **Public Administration Service IP**: The IP Address of the component that is requesting the content of the QR Code / Deep Link to the Enterprise Service. It's important to correctly set this value because the Enterprise Service filters the requests to the endpoints /users-onboarding/v1/session and /verification/v1/session to the IP Address that you set here. It can be a specific IP or a range of IPs. It's important to clarify that the IP Address to set here, it must be the one that appears in the request sent to the Enterprise Service. This implies that if you have any intermediary component that changes the IP Source Address of the request (like a proxy), this value must be set to the changed value or range.

- Example: 172.16.0.2
- Example: 1.1.1.1
- Example: 10.3.0.0/12
- Example: 192.168.1.0/24

** It can also be an IPv6 address or range.

3.3.2 TLS Configuration

The next attributes are all the possible configurations related to the Java Key Store PKCS12 information to enable TLS in the Enterprise Service.

- **TLS Enabled**: YES or NO to enable TLS in the enterprise service. Note that TLS can be configured and be disabled at the same time for testing purposes.
  - Example: yes
  - Example: NO

- **Path to JKS PKCS12**: The relative or absolute path to the Java Keystore PKCS12 that contains the certificate and key to be used as part of the TLS configuration.
  - Example: ./impulse.jks
  - Example: /home/impulse/keystores/impulse.jks

- **JKS Password**: Password that unlocks the Java Keystore.
  - Example: changeit

- **Key Alias**: The alias of the key corresponding to the certificate inside the JKS.
  - Example: impulse

- **Key Password**: Password that unlocks the Key inside the Java Keystore, identified by the key alias.
  - Example: changeit

3.3.3 Dashboard Configuration

The next attributes are all the possible configurations related to the Public Servant Dashboard Authentication mechanism.

- **Username**: The username that the Public Servant will use to login in the dashboard.
  - Example: admin
  - Example: john_doe

- **Password**: The password that the Public Servant will use to login in the dashboard.
  - Example: changeit

- **Two Factor Authentication enabled**: Password that unlocks the Java Keystore.
  - Example: yes
  - Example: NO

- **Email**: The email account that will receive a TOTP (Time-based One-Time Password) to perform a two-factor authentication by the Public Servant.

- Example: johndoe@impulse.com

3.3.4 Logging configuration

The next attributes are all the possible configurations related to the logging of the Enterprise Service.

- **Level**: DEBUG or INFO levels available. The DEBUG level is meant to be used for finding problems in the integration and instantiation, while the INFO level is intended to be used in the piloting of the project (since it gives a high-level overview of user interactions with the Enterprise Service).
  - Example: DEBUG
  - Example: info

- **File Path**: The absolute or relative path to the generated log file (inside the Docker container).
  - Example: ./impulse.log

  ** This path is actually pointing to a route inside the container.

3.3.5 Face Matching Configuration

The next attributes are all the possible configurations related to the Face Matching Service information.

- **URL**: The complete URL where the Face Matching Service API is available.
  - Example: https://apis.alicebiometrics.com/face

3.3.6 Document Verification Configuration

The next attributes are all the possible configurations related to the Document Verification Service information.

- **URL**: The complete URL where the Document Verification Service is available.
  - Example: https://impulse.tree-api.com

3.4 Deployment

Before deploying the Enterprise Service, it is necessary to obtain the docker image to run the container. For doing this, we first need to login into the GRAD private repository where the image is stored. After successfully logged in with a robot account provided to the public administrations, it is now possible to get the docker image of the Enterprise Service.

After acquiring the image, the deployment process is pretty simple as the configuration tool is able to prepare a docker run command with every file needed for the instantiation.

Example of prepared run command:

```
docker run -p 8080:8080 –name enterprise-service -v
/home/impulse/data:/data -v /home/impulse/config:/config
harbor.gradiant.org/syp-impulse-pr-01349/enterprise-service
```

It is worth mentioning that it can be modified adding options like -d (detached), changing the port, changing the container-name, etc.

** The Enterprise Service can be deployed in any available port, but it must coincide with the configured in the **Enterprise Service Public URL** attribute, explained in section 3.3.1.

4 Interaction with the Public Servant Dashboard

The Public Servant Dashboard is a component of the Enterprise Service that allows an operator to manually accept onboarding requests from the citizens. Every request is sent to this dashboard after it has been handled by the biometric authentication services, including some information in it about the results obtained.

4.1 Access

This dashboard will be available as a web application on route **/tasks**. After the first access in a web browser, basic user/password authentication will be requested. These values must have been configured according to section 3.3.3.

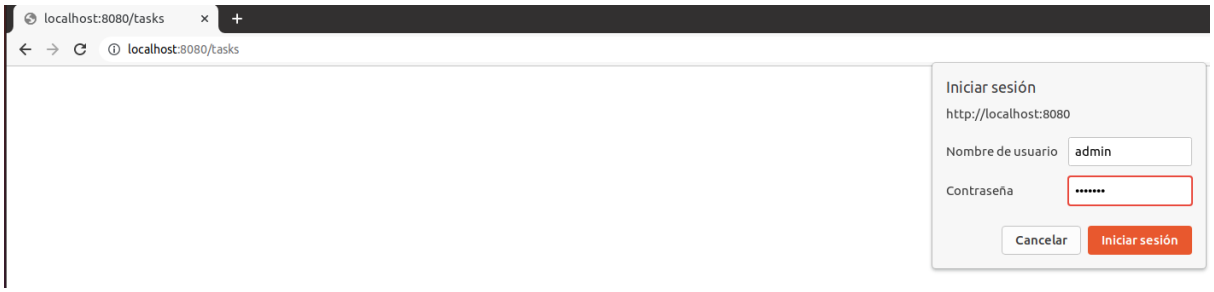

Figure 5: Basic Login in the Public Servant Dashboard.

If the dashboard has been configured to have a two-factor authentication, the next screen will appear.

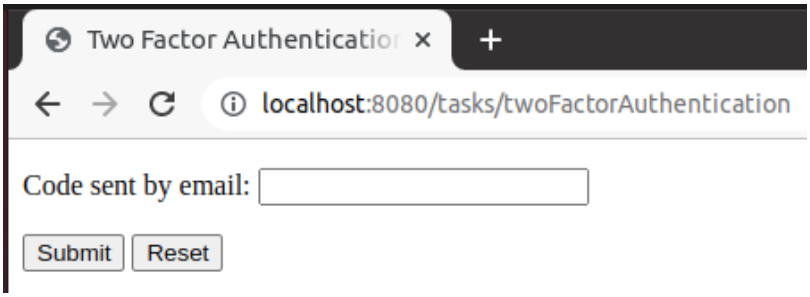

Figure 6: Two-factor authentication method in the Public Servant Dashboard.

One valid code that lasts 15 minutes is sent to the previously configured email (section 3.3.3). You can resend a valid code refreshing the page.

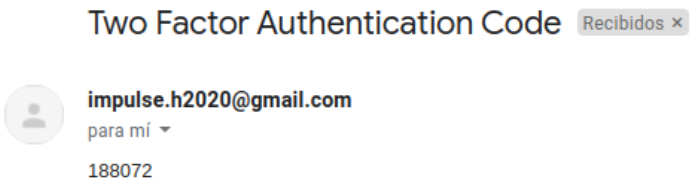

Figure 7: Two-factor authentication code in the public servant's email.

4.2 Usage

After login completely, the dashboard will appear, and it consists of a list of requests with the names of the citizens and the validations performed.

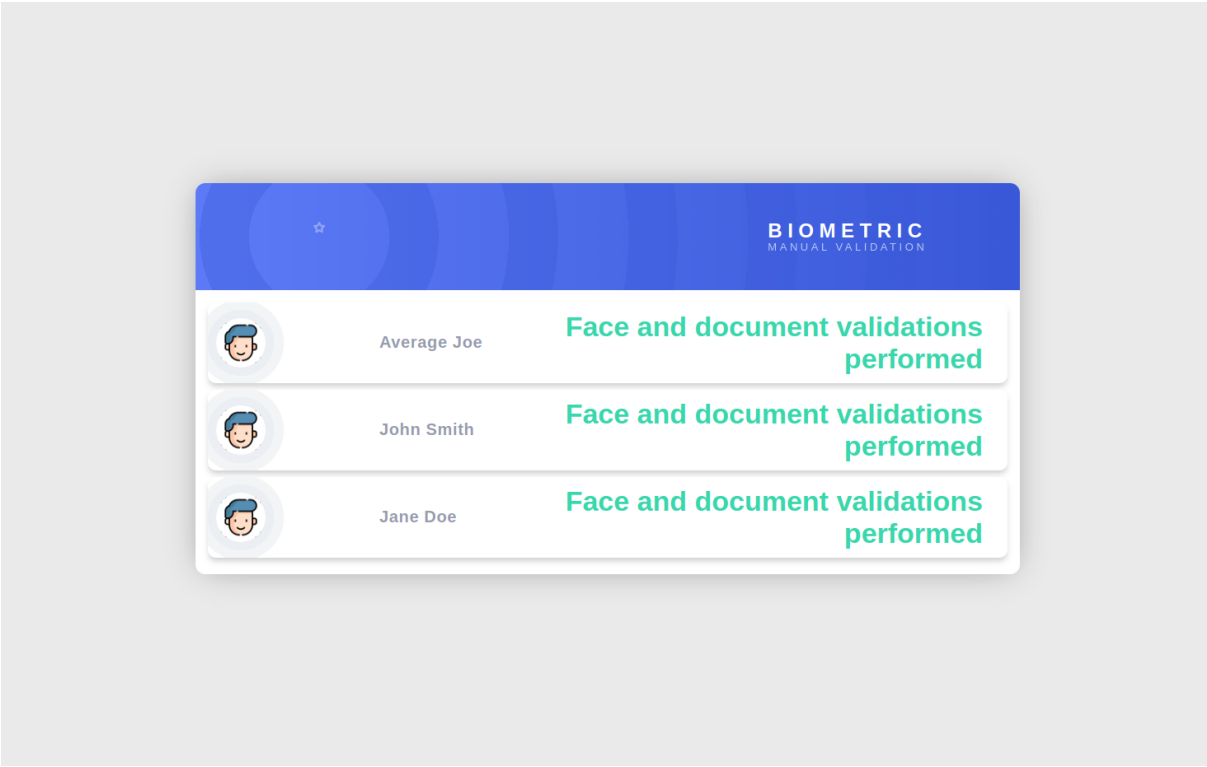

Figure 8: Batched onboarding requests in the Public Servant Dashboard.

After clicking on any request, the specific page with all the information it contains will be shown to the Public Servant. First, we will see the identifier of the task, the user information, the description of the face matching validation service result, and the description of the document validation service result. Then, the images sent by the user will appear, being the selfie **[10]** on the left, the front picture of the ID Document in the middle, and the back picture of the ID Document (in case there is one) on the right. Below the images, there are three buttons that correspond to the actions that the dashboard operator can perform. The request can be accepted or rejected, being possible to go back to the task list and leave it for later.

Task Identifier: zx7sDozmy7LvUdFTjzQnYhV

Name: Jane Doe

Personal Identifier: 12345678A

Gender: Male

Date of Birth: 2022-07-31

Place of birth: Fakeland

Current Address: Fake Address 123

Nationality: SPANISH

Document Type: ID_CARD

Expiration Date: 2022-07-31

Face Matching Description: Face of the person correctly validated

Document Validation Description: Document validated successfully

Accept Reject Back

Figure 9: Detailed onboarding request in the Public Servant Dashboard.

5 Refreshing the Trusted Issuer

The verifiable authorizations issued to each public administration Trusted Issuer have a validity of 6 months. This means that at least every 6 months, one member IT department of the public administration should stop the deployed docker container, update this verifiable authorization and redeploy it again. If this verifiable authorization expires, the public administration won't be able to change the URL of the Enterprise Service, so this is something that is recommended to be done once every 3 months.

The method of updating this EBSI Verifiable Authorization is pretty simple, as the configuration tool has the functionality to handle this process. The only requirement is to get a session token from the EBSI Users Onboarding Page.

To get this, the operator that is refreshing the Trusted Issuer has to enter the page: https://app.preprod.ebsi.eu/users-onboarding/.

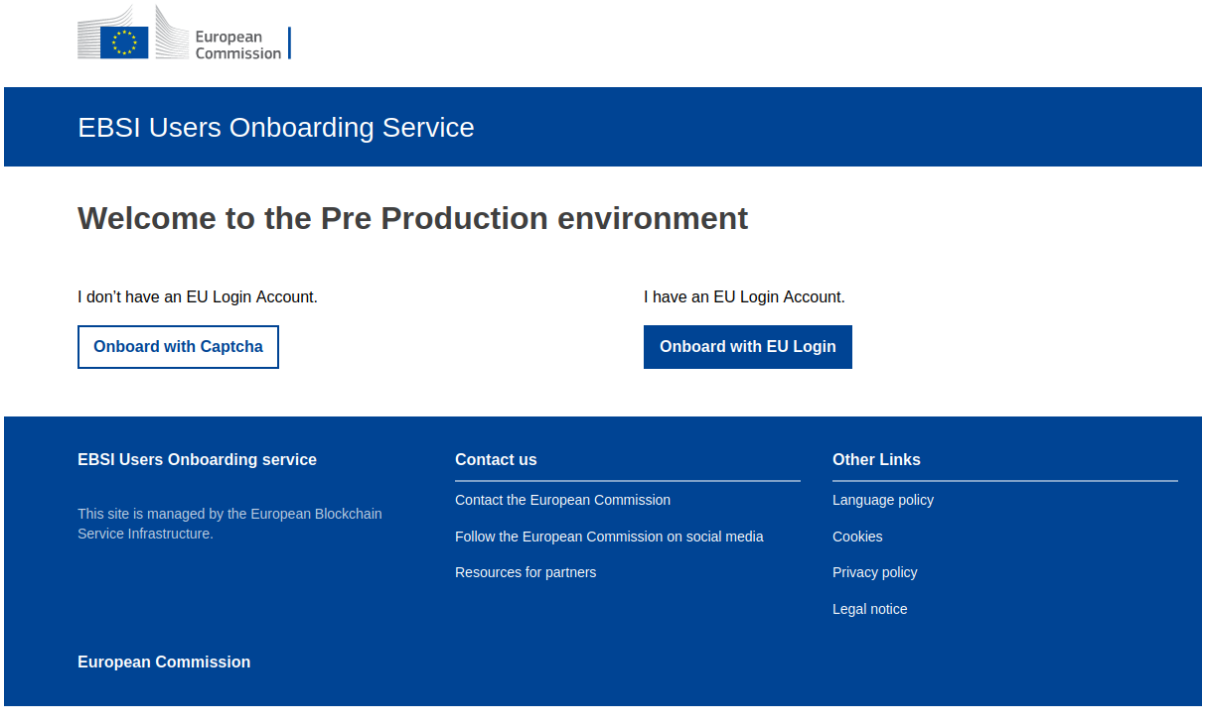

Figure 10: EBSI Onboarding Service page for the pre-production environment.

Once inside, they will have to click on "Onboard with Captcha".

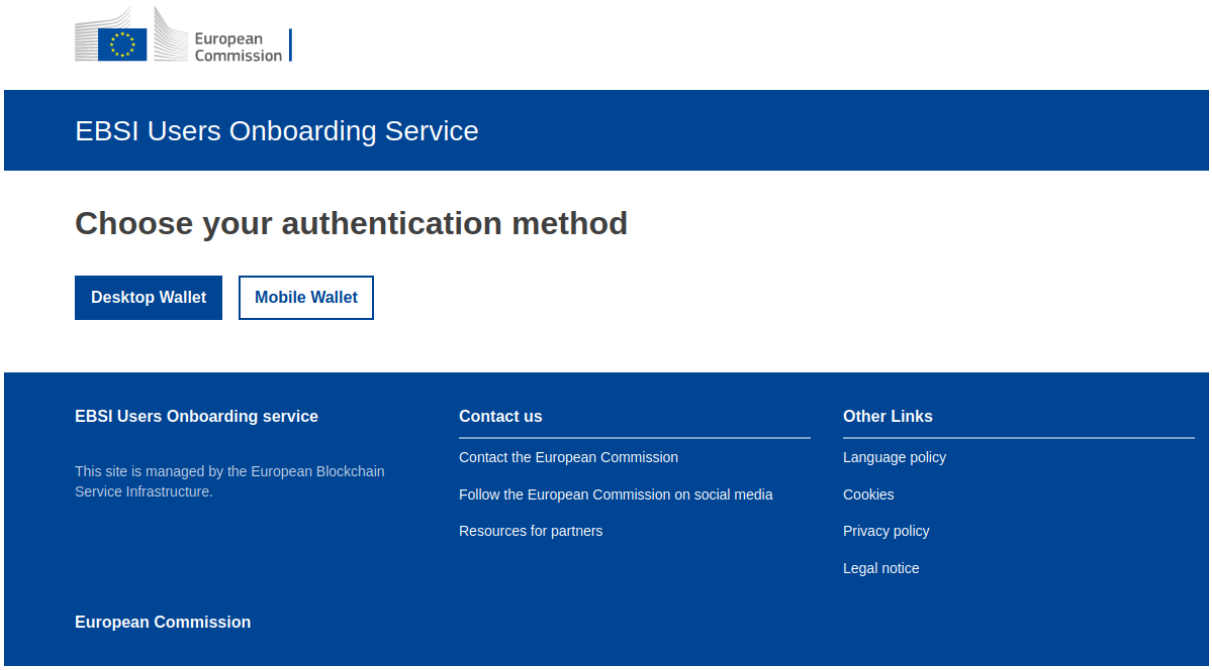

Figure 11: Captcha onboarding in the EBSI Onboarding Service.

Then, they will click on "Onboard with Captcha" which will make the session token appear.

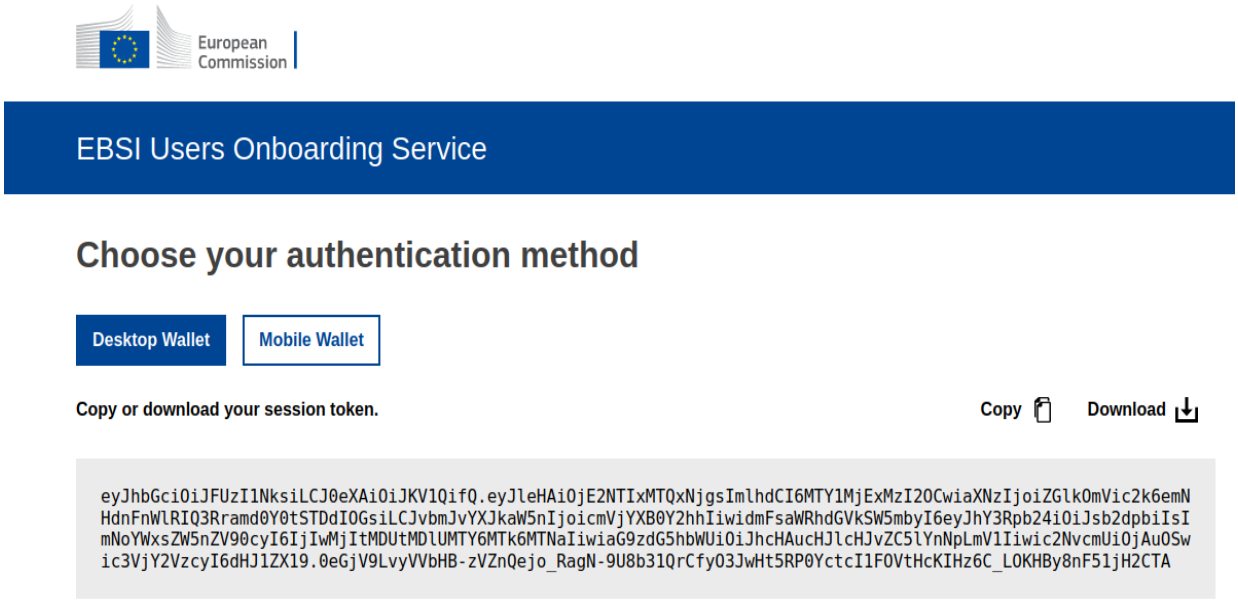

Figure 12: Onboarding session token obtained from the EBSI Onboarding Service.

After copying it to the clipboard, it can be introduced in the Configuration Tool to update the EBSI Verifiable Authorization.

6 Instantiation/Integration Activities

6.1 First Prototype Session

In February 2022, GRAD organized an online meeting with the public administrations of the Consortium to show the first prototype of the IMPULSE solution. It included a presentation of the message flow of the solution, an explanation of the points of the flow in which the public administrations were involved, a compilation of the requirements needed for the complete integration with the Enterprise Service, and a live demo of the solution. In this demonstration, the entire onboarding and authentication processes were shown, and then, GRAD answered all the questions that arose from the participants.

From this session, GRAD received some feedback which led to a series of changes in the solution itself, and its configuration/deployment:

- Dockerization of the Enterprise Service was addressed after receiving some concerns about the deployment process.
- The notification system via email was changed to a PUSH Notification system due to some concerns about the need to collect the emails of the users.
- An internal evaluation on the use of Legal Entity Verifiable Credentials for the Verifiers was addressed to avoid the need of the Verifiers to be Trusted Issuers. It will be implemented in the second round of IMPULSE.
- An internal evaluation on the possibility of cross-border scenarios among the public administrations that use IMPULSE as an eID system. It will be implemented in the second round of IMPULSE.
- The first version of the Configuration Tool was made to facilitate the onboarding of the public administrations' DIDs as Trusted Issuers.

6.2 F2F IMPULSE Instantiation

In March 2022, a F2F meeting was held in Vigo between the people from the Consortium. Among the different presentations, activities, or workshops carried out, GRAD made a live instantiation process of the Enterprise Service with a Dummy Public Administration Service that was developed and integrated by ICERT. Prior to the demo, the integrations that are necessary to connect a PAS with IMPULSE were explained. In this live demonstration of a real instantiation, an Azure Machine with Ubuntu 18.04 was created, an SSH connection was established with the machine, the configuration tool was used to onboard a DID in EBSI, the DID was requested to be registered as Trusted Issuer through the EBSI Service Desk, and both Dummy PAS Service and Enterprise Service were built as Docker images and deployed. Then, a demo of the authentication process was made to show that the instantiation and integration was successful. Finally, GRAD answered all the questions that came up among the participants.

From this session, GRAD received some feedback which led to a series of changes in the solution itself, and its configuration/deployment:

- The Deep Link method to initiate an onboarding or authentication was developed due to some concerns about the possibility of performing the two IMPULSE main processes only with the mobile device. This was made for the specific citizens that want to receive the PAS service in a web browser on the same device where the IMPULSE application is installed, without the need of using a second one (PC, laptop, tablet, etc.).
- The Docker images build step was removed from the instantiation process to simplify it. Instead, a pre-built image is stored in a private Docker repository owned by GRAD, and shared through a robot account with only reading permissions.
- The DID Onboarding and Trusted Issuers registration steps were removed, as we could just simply make all the registrations ourselves (GRAD), and then share the pre-configured wallet to each public administration for this first round of IMPULSE.

- The functionalities of the Configuration Tool were expanded to cover every possible configurable area (organization name, URLs, TLS, logging, dashboard, face matching service, document verification service).
- The manual "How to IMPULSE: Integration, Instantiation and Interaction" was made to cover everything that the public administrations needed to know to correctly perform the Instantiation phase.

7 Conclusions

With the creation of the Docker image of the Enterprise Service, the implementation of the configuration tool, and the preparation of the manual “How to IMPULSE: Integration, Instantiation and Interaction”, we have successfully fulfilled the requirements of the public administrations to deploy the IMPULSE solution.

GRAD has conducted several meetings with the PAs to solve particular issues of each one. The feedback obtained by the PAs in several meetings allowed to improve the instantiation automation mechanisms as well the documentation to facilitate the deployment of the solution. As a result, GRAD has built up an initial process for the basic IMPULSE system instantiation which will be used by the PAs for the first piloting activities.

Finally, it is important to remark that throughout this process some issues have already been identified to address during the second round of the project, and they will be included in the final IMPULSE system instantiation.

References

[1] [AJ Alt, 2022] https://ajalt.github.io/clikt/

[2] [Ryan Day, 2022] https://github.com/soldair/node-qrcode

[3] [Lincoln Loop, 2021] https://github.com/lincolnloop/python-qrcode

[4] [Zxing, 2022] https://github.com/zxing/zxing

[5] [Rafael Herrmann, 2021] https://github.com/codebude/QRCoder

[6] [Brian J Brennan, 2020] https://github.com/brianloveswords/base64url

[7] [Python, 2021] https://docs.python.org/es/3/library/base64.html#base64.urlsafe_b64encode

[8] [Oracle, 2022] https://docs.oracle.com/javase/8/docs/api/java/util/Base64.html#getUrlEncoder--

[9] [Microsoft, 2022] https://docs.microsoft.com/en-us/dotnet/api/microsoft.identitymodel.tokens.base64urlencoder?view=azure-dotnet

[10] [dDara, 2022] https://www.flaticon.com/free-icon/woman_949666