



## D4.2 Distributed Trust Management Framework - Final version

| Document Identification |       |                 |            |
|-------------------------|-------|-----------------|------------|
| Status                  | Final | Due Date        | 28/02/2025 |
| Version                 | 1.0   | Submission Date | 05/03/2025 |

|                        |                              |                         |                         |
|------------------------|------------------------------|-------------------------|-------------------------|
| Related WP             | WP4                          | Document Reference      | D4.2                    |
| Related Deliverable(s) | D2.1, D2.2, D2.3, D2.4, D4.1 | Dissemination Level (*) | PU                      |
| Lead Participant       | VTT                          | Lead Author             | Erik Hieta-aho          |
| Contributors           | ATOS, UMU, CEA, QBE, UTH     | Reviewers               | Vitalii Demianets (NOR) |
|                        |                              |                         | Kaitai Liang (DTU)      |

| Keywords:                                                                |
|--------------------------------------------------------------------------|
| SSI, Seamless onboarding, behavioral authentication, side-channel attack |

### Disclaimer for Deliverables with dissemination level PUBLIC

This document is issued within the frame and for the purpose of the TANGO project. This project has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No. 101070052. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission. The dissemination of this document reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains. This deliverable is subject to final acceptance by the European Commission. This document and its content are the property of the TANGO Consortium. The content of all or parts of this document can be used and distributed provided that the TANGO project and the document are properly referenced. Each TANGO Partner may use this document in conformity with the TANGO Consortium Grant Agreement provisions.

## Document Information

| List of Contributors  |         |
|-----------------------|---------|
| Name                  | Partner |
| Erik Hieta-aho        | VTT     |
| Anni Karinsalo        | VTT     |
| Maike Jung            | CEA     |
| Nicolas Belleville    | CEA     |
| Apostolos Apostolaras | UTH     |
| Ross Little Armit     | ATOS    |
| Maria Hernandez       | UMU     |
| Niklas Palaghias      | QBE     |

| Document History |            |                      |                                        |
|------------------|------------|----------------------|----------------------------------------|
| Version          | Date       | Change editors       | Changes                                |
| 0.01             | 28/11/2024 | Erik Hieta-aho (VTT) | ToC draft                              |
| 0.02             | 21/02/2025 | Erik Hieta-aho (VTT) | Partner contributions. Internal Review |
| 0.03             | 26/02/2025 | Erik Hieta-aho (VTT) | Internal Review formatting edits       |
| 0.04             | 27/02/2025 | Erik Hieta-aho (VTT) | Adjusted figures                       |
| 0.05             | 04/03/2025 | Erik Hieta-aho (VTT) | Final figures updated                  |
| 1.0              | 05/03/2025 | Erik Hieta-aho (VTT) | FINAL VERSION TO BE SUBMITTED          |

| Quality Control     |                            |               |
|---------------------|----------------------------|---------------|
| Role                | Who (Partner short name)   | Approval Date |
| Deliverable leader  | Erik Hieta-aho (VTT)       | 04/03/2025    |
| Quality manager     | Jürgen Neises (FUJ_GE)     | 05/03/2025    |
| Project Coordinator | Tomás Pariente Lobo (ATOS) | 05/03/2025    |

|                       |                                                             |                       |    |                 |              |                      |
|-----------------------|-------------------------------------------------------------|-----------------------|----|-----------------|--------------|----------------------|
| <b>Document name:</b> | D4.2 Distributed Trust Management Framework - Final version |                       |    |                 | <b>Page:</b> | 2 of 58              |
| <b>Reference:</b>     | D4.2                                                        | <b>Dissemination:</b> | PU | <b>Version:</b> | 1.0          | <b>Status:</b> Final |

# Table of Contents

---

|                                                                             |    |
|-----------------------------------------------------------------------------|----|
| Document Information .....                                                  | 2  |
| Table of Contents .....                                                     | 3  |
| List of Tables.....                                                         | 5  |
| List of Figures .....                                                       | 6  |
| List of Acronyms.....                                                       | 7  |
| Executive Summary .....                                                     | 8  |
| 1 Introduction .....                                                        | 9  |
| 1.1 Purpose of the document.....                                            | 9  |
| 1.2 Relation to other project work.....                                     | 9  |
| 1.3 Structure of the document .....                                         | 9  |
| 2 WP4 in a nutshell.....                                                    | 10 |
| 3 Final Demonstrations.....                                                 | 12 |
| 3.1 Self-Sovereign Identity .....                                           | 12 |
| 3.1.1 Short Component description - SSI Agent for Issuer and Verifier ..... | 12 |
| 3.1.2 Short Component description - Wallet .....                            | 18 |
| 3.1.3 Final Demonstration description.....                                  | 22 |
| 3.1.4 Support for pilots .....                                              | 22 |
| 3.1.5 Final version of this component .....                                 | 22 |
| 3.2 Seamless onboarding.....                                                | 22 |
| 3.2.1 Short Component description .....                                     | 22 |
| 3.2.2 Demonstration description.....                                        | 33 |
| 3.2.3 Support for pilots .....                                              | 33 |
| 3.2.4 Final version of this component .....                                 | 33 |
| 3.3 User behavioural authentication.....                                    | 33 |
| 3.3.1 Short Component description .....                                     | 33 |
| 3.3.2 Demonstration description.....                                        | 35 |
| 3.3.3 Support for pilots .....                                              | 35 |
| 3.3.4 Final version of the component .....                                  | 36 |
| 3.4 Device behavioural authentication.....                                  | 36 |
| 3.4.1 Short Component description .....                                     | 36 |
| 3.4.2 Demonstration description.....                                        | 48 |
| 3.4.3 Support for pilots .....                                              | 52 |
| 3.4.4 Final version of this component .....                                 | 53 |
| 3.5 Side-channel attack hardening .....                                     | 54 |
| 3.5.1 Short Component description .....                                     | 55 |
| 3.5.2 Demonstration description.....                                        | 55 |

|                       |                                                                |                       |    |                 |              |                      |
|-----------------------|----------------------------------------------------------------|-----------------------|----|-----------------|--------------|----------------------|
| <b>Document name:</b> | D4.2 Distributed Trust Management Framework -<br>Final version |                       |    |                 | <b>Page:</b> | 3 of 58              |
| <b>Reference:</b>     | D4.2                                                           | <b>Dissemination:</b> | PU | <b>Version:</b> | 1.0          | <b>Status:</b> Final |

|       |                          |    |
|-------|--------------------------|----|
| 3.5.3 | Support for pilots ..... | 56 |
| 4     | Conclusions .....        | 57 |
| 5     | References .....         | 58 |

|                       |                                                                |                       |    |                 |              |                      |
|-----------------------|----------------------------------------------------------------|-----------------------|----|-----------------|--------------|----------------------|
| <b>Document name:</b> | D4.2 Distributed Trust Management Framework -<br>Final version |                       |    |                 | <b>Page:</b> | 4 of 58              |
| <b>Reference:</b>     | D4.2                                                           | <b>Dissemination:</b> | PU | <b>Version:</b> | 1.0          | <b>Status:</b> Final |

## List of Tables

---

|                                                                                 |    |
|---------------------------------------------------------------------------------|----|
| Table 1. WP4 components                                                         | 11 |
| Table 2. SSI Agent features                                                     | 18 |
| Table 3. SSI Wallet Features                                                    | 19 |
| Table 4. Pilots supported                                                       | 22 |
| Table 5. Final version of SSI components                                        | 22 |
| Table 6. Use Case: End User requests to be issued with a Verifiable Credential. | 23 |
| Table 7. Internal Architecture Functionalities                                  | 37 |
| Table 8. Server's Endpoints                                                     | 46 |
| Table 9. Summary of Updates and Improvements in the DCBA Component              | 53 |

|                       |                                                             |                       |    |                 |              |                      |
|-----------------------|-------------------------------------------------------------|-----------------------|----|-----------------|--------------|----------------------|
| <b>Document name:</b> | D4.2 Distributed Trust Management Framework - Final version |                       |    |                 | <b>Page:</b> | 5 of 58              |
| <b>Reference:</b>     | D4.2                                                        | <b>Dissemination:</b> | PU | <b>Version:</b> | 1.0          | <b>Status:</b> Final |

## List of Figures

|                                                                                    |    |
|------------------------------------------------------------------------------------|----|
| Figure 1. WP4 components and position in the TANGO architecture                    | 10 |
| Figure 2. Non-normative example of VC                                              | 13 |
| Figure 3. EmployeeCredential Example                                               | 14 |
| Figure 4. CustomerCredential Example                                               | 15 |
| Figure 5. ePassport VC Issuer Flow Diagram (part 1 of 2)                           | 16 |
| Figure 6. ePassport VC Issuer Flow Diagram (part 2 of 2)                           | 16 |
| Figure 7. EmployeeCredential VC Issuer Flow Diagram                                | 17 |
| Figure 8. M2M Flow                                                                 | 20 |
| Figure 9. ZKP Verifier flow                                                        | 21 |
| Figure 10. UCBA and PDP integration flow                                           | 34 |
| Figure 11. Continuous Authentication Flowchart                                     | 36 |
| Figure 12. System Architecture Components                                          | 37 |
| Figure 13. User Registration Process Pipeline                                      | 39 |
| Figure 14. User Login Process Pipeline                                             | 40 |
| Figure 15. Android Log Capture Process Pipeline                                    | 40 |
| Figure 16. Foreground Service                                                      | 41 |
| Figure 17. Android Log Capture: Managing Device Offline Scenarios                  | 42 |
| Figure 18. Android Log Capture: Managing Authentication Token Expiration Scenarios | 43 |
| Figure 19. User Logout Process Pipeline                                            | 44 |
| Figure 20. Android System Logs                                                     | 44 |
| Figure 21. Data Pre-Processing Phase                                               | 46 |
| Figure 22. 'Authenticator' Requesting Device Location Access                       | 48 |
| Figure 23. 'Authenticator' Requesting Superuser Rights                             | 49 |
| Figure 24. Login Page                                                              | 50 |
| Figure 25. Registration Page                                                       | 51 |
| Figure 26. System Deployment Architecture                                          | 52 |
| Figure 27. Comparison of Implementations                                           | 55 |

## List of Acronyms

| Abbreviation / acronym | Description                                                                    |
|------------------------|--------------------------------------------------------------------------------|
| ABC                    | Attribute-Based Cryptography                                                   |
| DCBA                   | Device Continuous Behavioral Authentication                                    |
| DID                    | Decentralized Identity                                                         |
| Dx.y                   | Deliverable number y belonging to WP x                                         |
| DSBA                   | Data Spaces Business Alliance                                                  |
| EBSI                   | European Blockchain Service Infrastructure                                     |
| EC                     | European Commission                                                            |
| eIDAS                  | EU regulation for electronic Identification, Authentication and trust Services |
| FE                     | Front End                                                                      |
| GAIA-X                 | Federated and secure Data Infrastructure                                       |
| IDSA                   | International Data Spaces Association                                          |
| IoT                    | Internet of Things                                                             |
| JWT                    | JSON Web Tokens                                                                |
| ML                     | Machine Learning                                                               |
| M2M                    | Machine-to-Machine                                                             |
| MQTT                   | Message Queueing Telemetry Transport                                           |
| NFC                    | Near Field Communication                                                       |
| OID4VP                 | OpenID Verifiable Presentation                                                 |
| OID4VCI                | OpenID Verifiable Credential Issuance                                          |
| OIDC                   | OpenID Connect                                                                 |
| PII                    | Person Identifiable Information                                                |
| SD                     | Selective Disclosure                                                           |
| SDK                    | Software Development Kit                                                       |
| SIOP                   | Self-Issued OpenID Provider                                                    |
| SSI                    | Self-Sovereign Identity                                                        |
| UI                     | User Interface                                                                 |
| VC                     | Verifiable Credential                                                          |
| VDR                    | Verifiable Data Registry                                                       |
| W3C                    | The World Wide Web Consortium                                                  |
| WP                     | Work Package                                                                   |
| ZKP                    | Zero-Knowledge Proof                                                           |

## Executive Summary

---

This report describes the TANGO distributed trust management framework and its components developed in WP4. This is the final version describing final versions of work that were carried out in different tasks of the aforementioned work package. Each task describes its component(s) and the final state achieved in the form of a demonstration.

The first chapter introduces the work conducted in WP4 and the contents of this deliverable. Chapter two covers all demonstrators created in WP4 and its five subtasks. All subchapters follow common structure: component(s) description, demonstration description, support for pilots and future integration. The first subchapter covers Self Sovereign Identity (Issuer, Verifier, Wallet), second subchapter covers Seamless Onboarding. The third and fourth subchapter cover Human Behavioral Authentication and Device Behavioral Authentication respectively. The last subchapter handles Hardening against Side-Channel Attacks.

Third chapter covers conclusions drawn from the final results of these demonstrations and next steps in the integration and piloting of this trust management framework for TANGO. WP4 is in completion and its components are being integrated into a common platform.

|                       |                                                             |                       |    |                 |              |                      |
|-----------------------|-------------------------------------------------------------|-----------------------|----|-----------------|--------------|----------------------|
| <b>Document name:</b> | D4.2 Distributed Trust Management Framework - Final version |                       |    |                 | <b>Page:</b> | 8 of 58              |
| <b>Reference:</b>     | D4.2                                                        | <b>Dissemination:</b> | PU | <b>Version:</b> | 1.0          | <b>Status:</b> Final |

# 1 Introduction

---

## 1.1 Purpose of the document

---

The purpose of this document is to describe all tools and demonstrations created in WP4 that together form the TANGO distributed trust framework. This work is ongoing and will be continued in other Work Packages in terms of integration work. The presented version here is the final version of this work package for the project. The document also covers features identified as missing and integration work.

## 1.2 Relation to other project work

---

These components are based on previous design and specification work conducted in WP2 and presented in the three deliverables D2.1, D2.2, and D2.3. They are the basis for the work in this work package.

Important input to this deliverable for shaping these solutions is based on:

- ▶ D2.2 User Needs and Requirements & Use Case Scenarios.
- ▶ D2.3 System Requirements and Specifications, Platform Architecture, and Privacy, Ethical, Social and Legal Impact Assessment.
- ▶ Architecture discussions to align with GAIA-X and IDSA and therefore analysis of different IDSA Connectors and their fit with TANGO.

The necessary components developed in this WP4, will in turn provide necessary functionality and services for the TANGO platform and the pilots later in the project.

## 1.3 Structure of the document

---

The second chapter goes through each task of the WP4 and components they provide for the framework. It covers all demonstrators created in WP4 in its five subtasks respectively. All sub-chapters follow common structure: component(s) description, demonstration description, support for pilots, and a description of the final version of the component.

The third chapter covers conclusions drawn from the current results of these demonstrations and next steps in the development of this trust management framework for TANGO.

|                |                                                             |                |    |          |       |         |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|---------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 9 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status: | Final |

## 2 WP4 in a nutshell

The purpose of WP4 is to offer a distributed trust framework with several functionalities providing components for the use of TANGO platform implemented in WP6 and various pilots deployed in WP7.

The objective of T4.1 is development of SSI component that allows authentication using verified identity. The objective of T4.2 is to develop a remote identity verification system based on identity documents. The objective of T4.3 is to provide security with continuous user-friendly authentication. The objective of T4.4 is to consistently identify the users based on various device behaviors. Finally, the objective of T4.5 is to develop a component for countermeasures against side-channel attacks.

It is crucial that access into the TANGO platform services will be secure and trusted. In addition, the various Pilot application areas such as public administration, smart hospitality, autonomous vehicles, smart manufacturing, banking and retail, serve the purpose of demonstrating the added value of the components' functionalities. For these purposes, seamless and user-friendly onboarding is offered via smooth identification [T4.2] process, after which the authentication into services is handled via personal credentials that are managed by the personal wallet [T4.1]. In order to improve trust in both the user and device identity, a method for providing continuous device authentication [T4.4] and another method for continuous user authentication [T4.3] are developed. For protecting the security of the TANGO-enabled services against side-channel attacks, a module implementing countermeasures by hardening compiler code [T4.5] used by the TANGO services is developed.

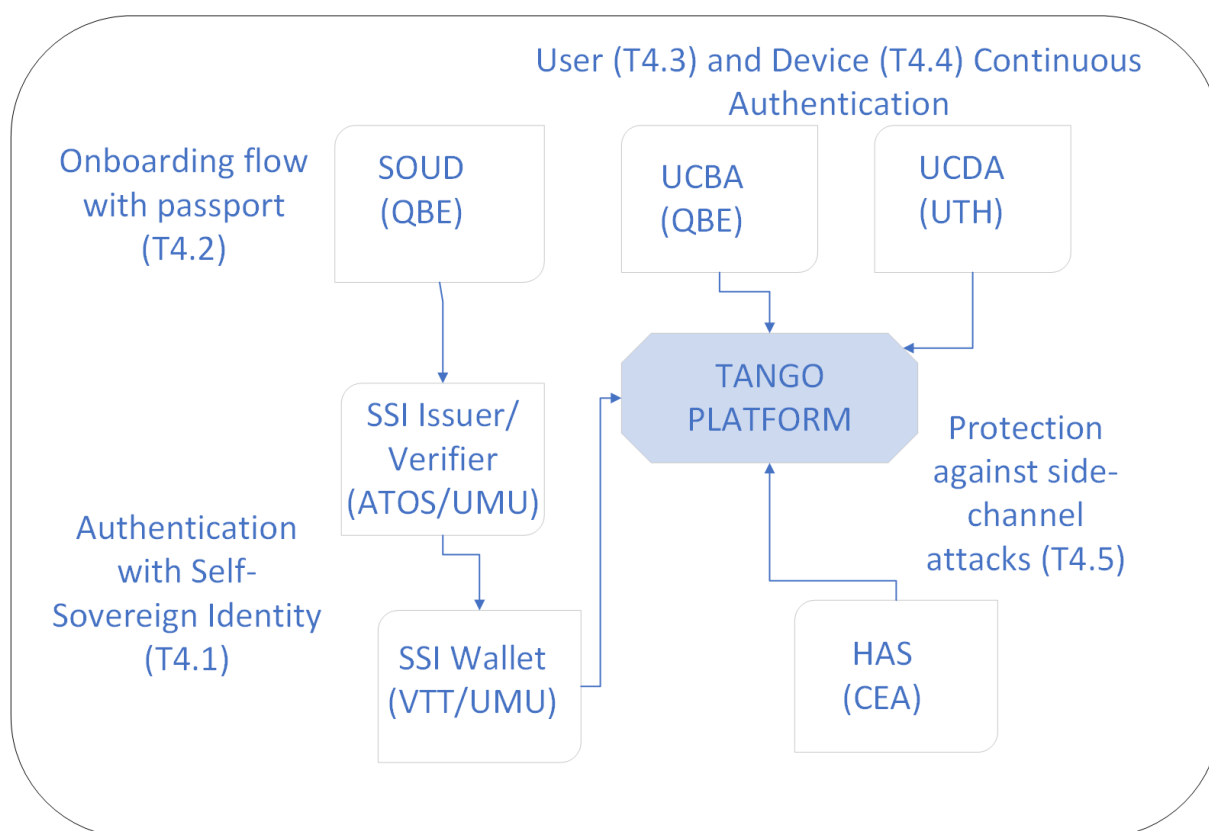


Figure 1. WP4 components and position in the TANGO architecture

A brief description of the main WP4 components, as well as a summary of what is new with respect to the delivery of the intermediate version (D4.1) is provided in Table 1.

|                |                                                             |                |    |          |     |         |          |
|----------------|-------------------------------------------------------------|----------------|----|----------|-----|---------|----------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          |     | Page:   | 10 of 58 |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0 | Status: | Final    |

Table 1. WP4 components

| Task | Component name                   | Short description                                                                                           | What's new                                                                                                                                                                                                                                                      |
|------|----------------------------------|-------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| T4.1 | SSI Agent – Issuer and Verifier  | SSI Agent that supports ePassport credential issuance and verification while deployed in the FIWARE DSC.    | Support for ePassport credentials and deployment in FIWARE DSC.                                                                                                                                                                                                 |
|      | SSI Wallet                       | SSI wallet with granular disclosure.                                                                        | The SSI wallet is integrated with the issuance and verification of ePassport credentials.                                                                                                                                                                       |
|      | SSI Agent + ZKP and M2M          | SSI issuer and wallet with zero-knowledge proof features and the ability to support IoT M2M authentication. | The IoT M2M authentication flow is a new addition.                                                                                                                                                                                                              |
|      | SSI Wallet + ZKP & M2M           | SSI wallet compatible with ZKP and IoT M2M                                                                  | The IoT M2M authentication flow is a new addition.                                                                                                                                                                                                              |
| T4.2 | Seamless onboarding              | e-passport app with related SDK                                                                             | Support for multiple issuers, integration with SSI wallet and issuer, support for Finnish passport, improving the screening and validation mechanisms of the app                                                                                                |
| T4.3 | User behavioral authentication   | User authentication application comprising of SDK and related backend                                       | Improvement of session management by adding a manual labelling process, addition a risk score feature, addition of an API for other TANGO components, addition of a similarity score feature                                                                    |
| T4.4 | Device behavioral authentication | Identification of the users based on various device behaviors                                               | Improved user interaction on the Authenticator app. Improved algorithms in terms of localization. Introducing ML model capabilities. Improved component in terms of continuity and data storage efficiency.                                                     |
| T4.5 | Side-channel attack hardening    | Research focused on developing countermeasures to mitigate side-channel attacks                             | Implementation of masked AES algorithm for smart hospitality pilot.<br><br>Acceleration of masked AES implementation (publication accepted and presented at SeHAS 2025).<br><br>Combination of hiding and masking countermeasures (publication in preparation). |

In the next sections, the document provides a detailed view of the final status of the implementation of these components with more details.

|                       |                                                             |                       |    |                 |              |                      |
|-----------------------|-------------------------------------------------------------|-----------------------|----|-----------------|--------------|----------------------|
| <b>Document name:</b> | D4.2 Distributed Trust Management Framework - Final version |                       |    |                 | <b>Page:</b> | 11 of 58             |
| <b>Reference:</b>     | D4.2                                                        | <b>Dissemination:</b> | PU | <b>Version:</b> | 1.0          | <b>Status:</b> Final |

## 3 Final Demonstrations

This section covers the demonstrations from each of the tasks within WP4. These demonstrations are planned to be presented in video format although they are based on dockerized components and therefore they could also be demonstrated live.

### 3.1 Self-Sovereign Identity

#### 3.1.1 Short Component description - SSI Agent for Issuer and Verifier

The SSI Agent implementation considers the scope as outlined in section 3.1 and is focused on persons for this deliverable. The following sub-sections will go over the specification and design of the SSI Agent supporting the issuer and verifier capabilities.

##### 3.1.1.1 Update on Standards

The following standard specifications provide an update to the previous deliverable of those followed in the TANGO SSI solution:

- ▶ W3C Verifiable Credentials Data Model [1] supporting JWT credentials and presentations.
- ▶ W3C Decentralized Identifiers [2] supporting DID:Key method.
- ▶ EBSI Conformance Guidelines [3]
- ▶ OID4VCI [4]
- ▶ OID4VP [5]
- ▶ W3C did:key method [6]
- ▶ EBSI DID Method for Natural Persons [7]
- ▶ EBSI VC Framework [8]

The challenge with the standards is noted with the integration between the various actors, primarily with the SSI Wallet and Agent, but also with the FIWARE DSC which follows certain cryptography and methods from the broad standard specifications, whereas TANGO SSI components follow the EBSI Guidelines and VC with more specific integration profiles to aid working interoperability.

##### 3.1.1.2 Verifiable Credentials

The following Verifiable Credential types are supported with example given below:

- ▶ Travellers and organization end users issued with ePassport VCs.
- ▶ Organization employees issued with EmployeeCredential VCs.
- ▶ Organization customers issued with CustomerCredential VCs.

#### ePassport Verifiable Credential example:

The ePassport Verifiable Credential is issued for a natural person and aims to simulate future EU Digital Identity Wallet approach to issue a person with their VerifiableID (in this case based on their ePassport information). A non-normative example is given in the following figure:

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ],
  "type": [
    "VerifiableCredential",
    "ePassport"
  ],
  "id": "vc:uself:agent:#8133274365875303553",
```

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 12 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

```

"credentialSchema": {
  "id": "https://api-conformance.ebsi.eu/trusted-schemas-registry/v2/schemas/z3MgU",
  "type": "FullJsonSchemaValidator2021"
},
"validFrom": "2024-03-26T11:58:51Z",
"issuer": "did:key:zDnaeixVSY9NWoxFetd9PPUgixxTQFa587HbuCrQFhU3tvjU9",
"issuanceDate": "2024-03-26T11:58:51Z",
"expirationDate": "2024-03-26T11:58:51Z",
"credentialSubject": {
  "id": "did:key:zDnaerDaTF5BXEavCrfrZEK316dpbLsfPDZ3WJ5hRTPFU2169",
  "address": "Seestraße 1",
  "passportNumber": "C7648XXXX",
  "familyName": "Doe",
  "givenName": "John",
  "dateOfBirth": "1830-10-01",
  "placeOfBirth": "Zug",
  "gender": "M",
  "authority": "Aarau AG",
  "nationality": "Schweiz",
  "countryCode": "CHE",
  "passportPhoto": "data:image/png;base64,iVBORw0KGgo...kJggg=="
  "dateOfExpiry": "2035-01-01",
  "dateOfIssue": "2025-01-01"
}
}

```

Figure 2. Non-normative example of VC

#### Organization Issuer Verifiable Credentials:

To support organizations employees and customers to connect to a Data Space Connector (DSC) the following Verifiable Credentials are supported:

- ▶ EmployeeCredential
- ▶ CustomerCredential

Non normative examples of these credentials to be issued in JWT format are shown below.

This follows the approach taken in the DSBA Technical Convergence recommendations [9] for its Verifiable Credentials.

EmployeeCredential example:

|                |                                                             |                |    |          |     |         |          |
|----------------|-------------------------------------------------------------|----------------|----|----------|-----|---------|----------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          |     | Page:   | 13 of 58 |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0 | Status: | Final    |

```
{
  "@context": [
    https://www.w3.org/2018/credentials/v1,
    https://example.com/issuer/context/credentials/employee/v1
  ],
  "id": https://example.com/issuer/credentials/6e14b8b8-87fa0014fe2a,
  "type": [
    "VerifiableCredential",
    "EmployeeCredential"
  ],
  "issuer": {
    "id": "did:key:zDnaeixVSY9NWoXfetd9PPUGixxTQFa587HbuCrQFhU3tvjU9"
  },
  "issuanceDate": "2022-03-22T14:00:00Z",
  "validFrom": "2022-03-22T14:00:00Z",
  "expirationDate": "2023-03-22T14:00:00Z",
  "credentialSubject": {
    "id": "did:key:zDnaerDaTF5BXEavCrFrZEK316dpbLsfPDZ3WJ5hRTPFU2169",
    "roles": [
      {
        "target": "did:key:zDnaerx9CtbPJ1q36T5Ln5wYt3MQYeGRG5ehnPAmxcf5mDZpv",
        "names": [
          "seller",
          "buyer"
        ]
      }
    ]
  },
  "name": "Jane Doe",
  "givenName": "Jane",
  "familyName": "Doe",
  "preferredUsername": "j.doe",
  "email": "janedoe@eviden.com"
}
```

Figure 3. EmployeeCredential Example

The EmployeeCredential is issued by the issuer organization that it will represent in the interactions with Data Space Connectors. The roles that the employee can carry out on the target DSC are included in the credential.

#### CustomerCredential example:

```
{
  "@context": [
    https://www.w3.org/2018/credentials/v1,
    https://example.com/issuer/context/credentials/customer/v1"
  ],
  "id": "example.com/issuer/credentials/25159389-8dd17b796ac0",
  "type": [
    "VerifiableCredential",
    "CustomerCredential"
  ],
}
```

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 14 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

```

"issuer": {
  "id": "did:key:zy1TZhaXgBZDvotDkL5257faiztiGiC2QtKLgpbnnEGta"
},
"issuanceDate": "2022-03-22T14:00:00Z",
"validFrom": "2022-03-22T14:00:00Z",
"expirationDate": "2023-03-22T14:00:00Z",
"credentialSubject": {
  "id": "did:key:z66ab5bca41bb45b78d242a46f0157b7d",
  "roles": [
    {
      "target": "z6MkhaXgBZDvotDkL5257faiztiGiC2QtKLgpbnnEGta2doK",
      "names": [
        "P.Info.gold"
      ]
    }
  ],
  "name": "Jane Doe",
  "givenName": "Jane",
  "familyName": "Doe",
  "preferredUsername": "j.doe",
  "email": "janedoe@example.com",
  "memberOf": {
    "@type": "Organization",
    "name": "Example Ltd",
    "url": "http://www.example.com/"
  }
}
}

```

Figure 4. CustomerCredential Example

The CustomerCredential is issued by the issuer organization that it is a client of and has powers on behalf of that organization to interact with Data Space Connectors. The roles that the customer can carry out on the target DSC are included in the credential.

### 3.1.1.3 Internal architecture

To complement the previous component architecture, shared in a previous deliverable, this section provides flow diagrams demonstrating the interwork between the various components of the solution.

#### ePassport VC Issuer Message Flow Sequence Diagram:

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 15 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

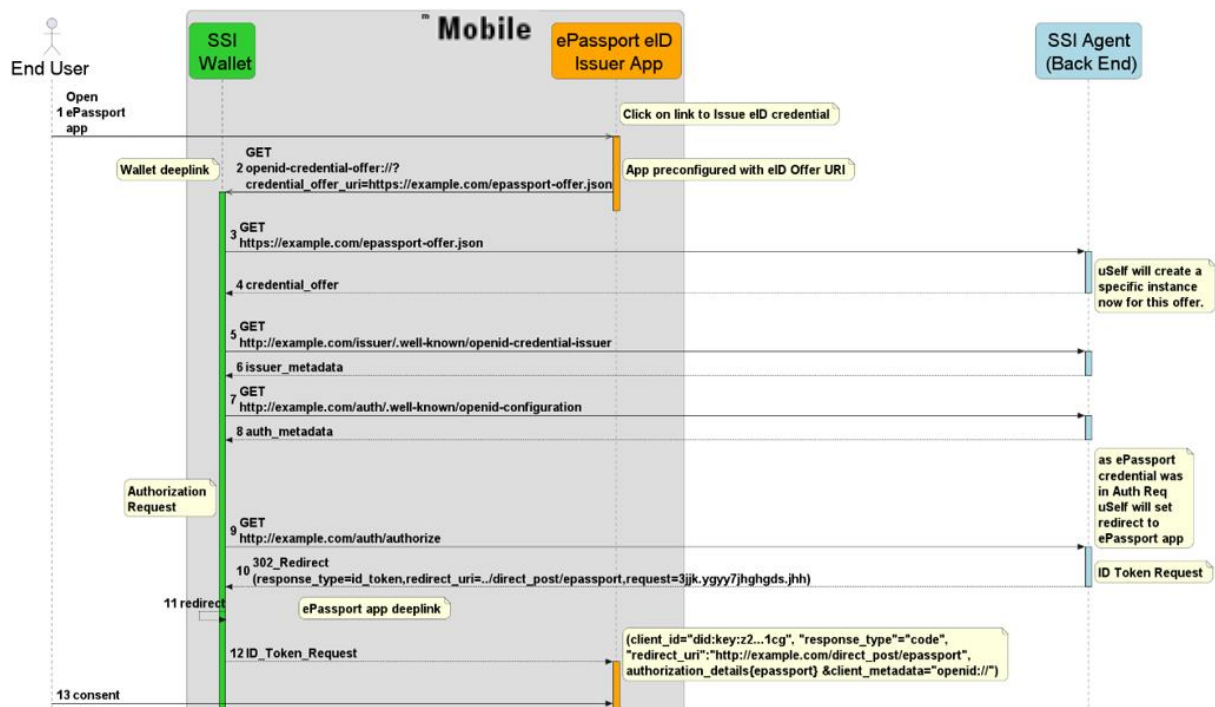


Figure 5. ePassport VC Issuer Flow Diagram (part 1 of 2)

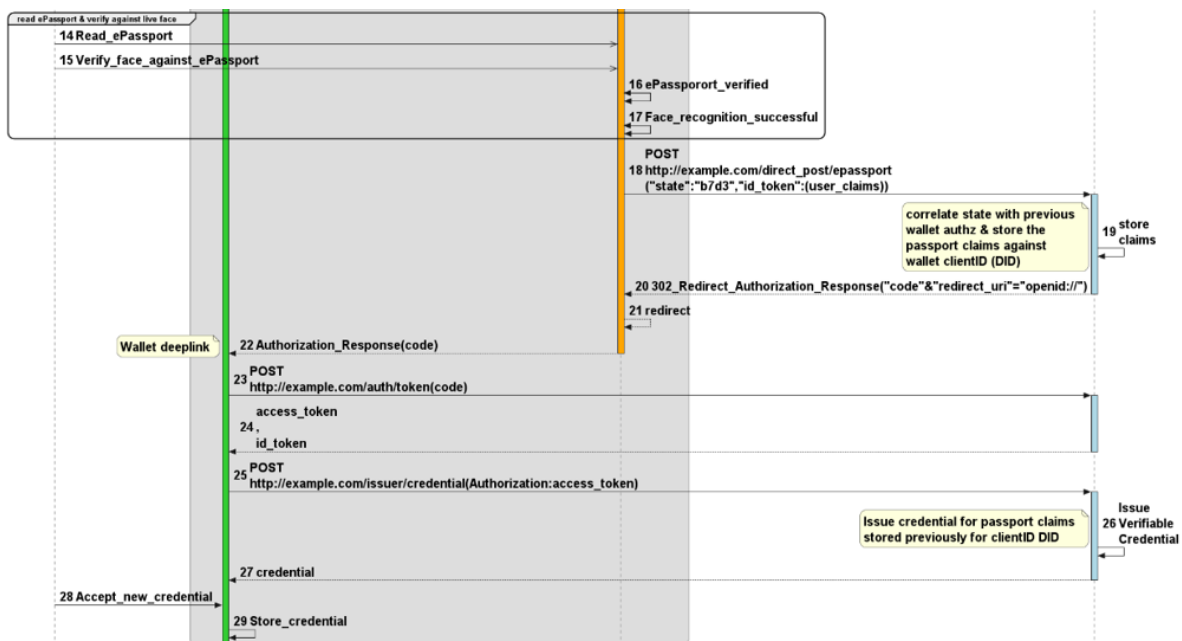


Figure 6. ePassport VC Issuer Flow Diagram (part 2 of 2)

### EmployeeCredential VC Issuer Message Flow Sequence Diagram:

Organizations maintain a database of their employees and end user customers based on their unique person identifier (ePassport or national eID). Upon requesting an employee or customer credential from an organization the end user must authenticate themselves with their ePassport or VerifiableID Verifiable Credential that was previously issued to them by a trusted authority. For this project it is issued by the ePassport app.

|                |                                                             |                |          |
|----------------|-------------------------------------------------------------|----------------|----------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version | Page:          | 16 of 58 |
| Reference:     | D4.2                                                        | Dissemination: | PU       |
| Version:       | 1.0                                                         | Status:        | Final    |

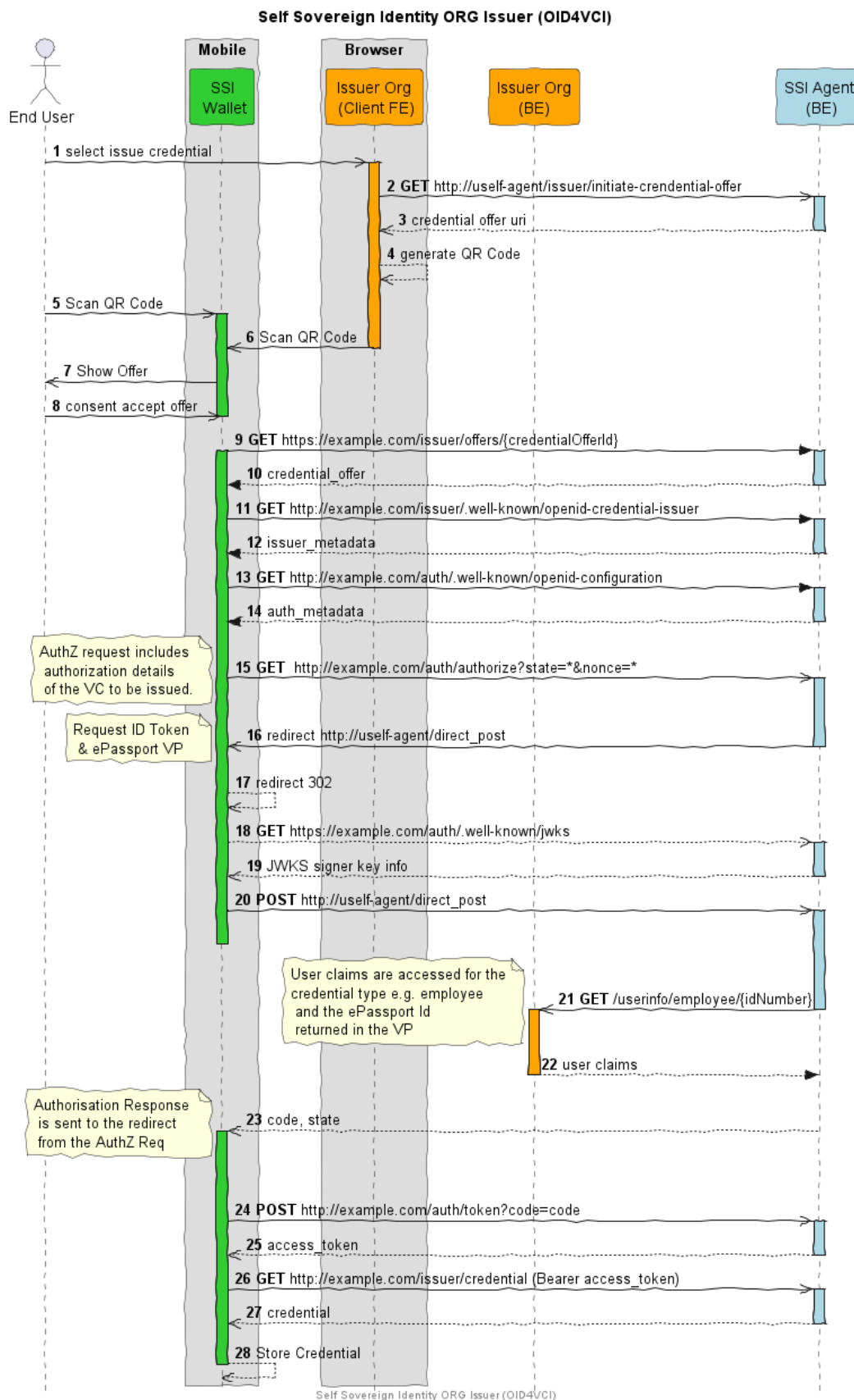


Figure 7. EmployeeCredential VC Issuer Flow Diagram

|                |                                                             |                |          |
|----------------|-------------------------------------------------------------|----------------|----------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version | Page:          | 17 of 58 |
| Reference:     | D4.2                                                        | Dissemination: | PU       |
| Version:       | 1.0                                                         | Status:        | Final    |

### 3.1.1.4 Implemented features

The main change from D4.1 is the introduction of the SSI Agent and the SSI Agent + ZKP&M2M in which each support a different Verifiable credential formatting.

#### SSI Agent features:

Table 2. SSI Agent features

| SSI Agents          | Features                                                                                                         |
|---------------------|------------------------------------------------------------------------------------------------------------------|
| SSI Agent           | JWT-VCs for issuing and verifying credentials following the EBSI Credential Framework                            |
| SSI Agent + ZKP&M2M | Agent with a dp-ABC module supports LDP-VCs with a ZKP algorithm and the machine-to-machine authentication flow. |

Further description of SSI Agent + ZKP&M2M is included in the following section of the SSI Wallet.

### 3.1.2 Short Component description - Wallet

In TANGO there will be two implementations of the SSI wallet. One SSI Wallet has granular disclosure implemented and supports JWT-VCs. The other SSI Wallet + ZKP M2M will include the zero-knowledge proof algorithm, machine to machine IoT implementation flow, and supports LDP-VCs.

#### 3.1.2.1 Internal architecture

The internal architecture is as described in D4.1. Here is a description of the architecture of the SSI Wallet with the FIWARE data space connector.

#### EmployeeCredential Verification on the FIWARE DSC

The following diagram shows the integration of the TANGO Wallet with the FIWARE DSC through the VCVerifier using an EmployeeCredential VC for authentication and authorization of an organization's representative.

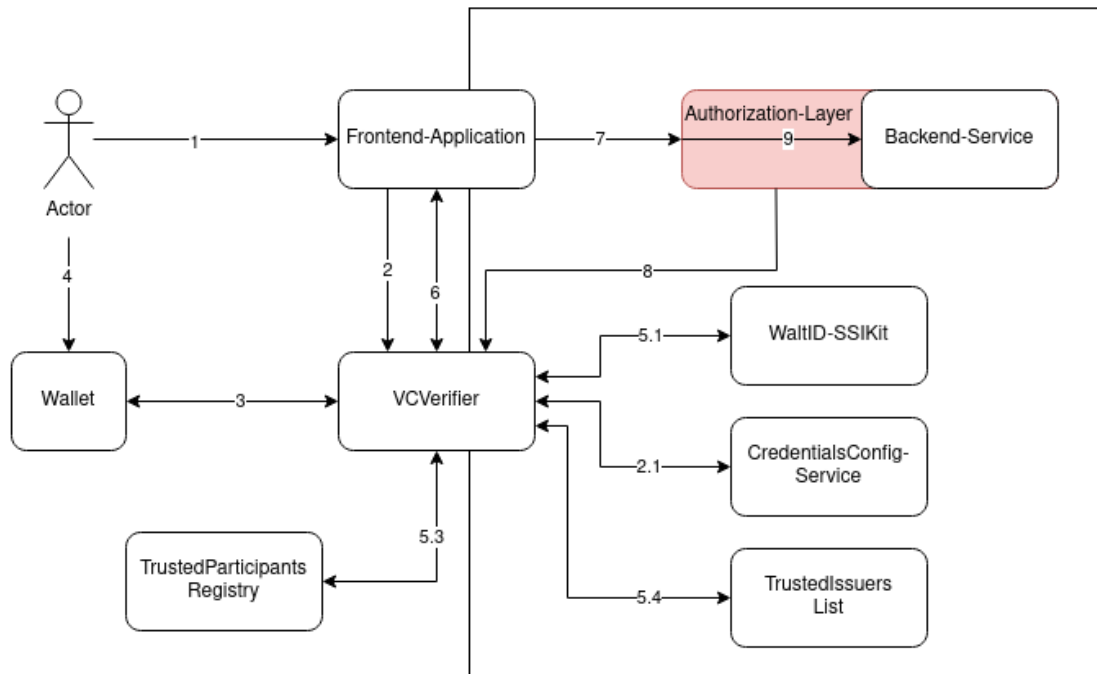


Figure 4. FIWARE DSC Interwork Flow Diagram [10]

### 3.1.2.2 Implemented features

Table 3. SSI Wallet Features

| SSI Wallet Implemented Features                        | New/Updates                             |
|--------------------------------------------------------|-----------------------------------------|
| M2M Issuance and Verification for IoT devices, LDP-VCs | The M2M flow is described below.        |
| Zero-Knowledge Proof, LDP-VCs                          | As described in D4.1                    |
| Granular Disclosure, JWT-VCs                           | Successfully implemented in SSI wallet. |

#### M2M Issuance and Verification Flows

##### Credential Registration and Issuance

This structured flow ensures secure and seamless credential issuance and storage for the IoT device. Below is a high-level summary of the key steps.

1. **Registration and Login in the Wallet (Steps 1-4)**
  - A new account is registered in the Wallet, storing essential information such as the DID and future credentials.
2. **Registration and Login in the Issuer (Steps 5-8)**
  - The Issuer registers a new user or device.
  - The Wallet logs into the Issuer to authenticate.
  - OAuth credentials (clientId and clientSecret) are received for further interactions.
3. **Updating Wallet Information (Steps 9-10)**
  - The Wallet updates its internal configuration with the received OAuth credentials.
4. **Wallet Learning Issuer Metadata (Steps 11-16)**
  - The Wallet retrieves Issuer metadata, including supported cryptographic algorithms and credential types.
5. **Credential Issuance Process (Steps 17-18)**
  - The Wallet initiates the credential issuance process by making an authentication request (/auth).
  - The Issuer responds with an authorization code (/code).
6. **Obtaining and Using the Authorization Code (Steps 21-24)**
  - The IoT device requests and receives the authorization code from the Issuer.
  - The code is passed to the Wallet.
7. **Finalizing the Credential Issuance (Steps 25-30)**
  - The Wallet exchanges the authorization code for an Access Token.
  - The Wallet requests the credential from the Issuer.
  - The IoT device receives the signed credential.
8. **Storing the Credential in the Wallet (Steps 31-32)**
  - The Wallet securely stores the issued credential.
  - The credential is available for future verification and management.

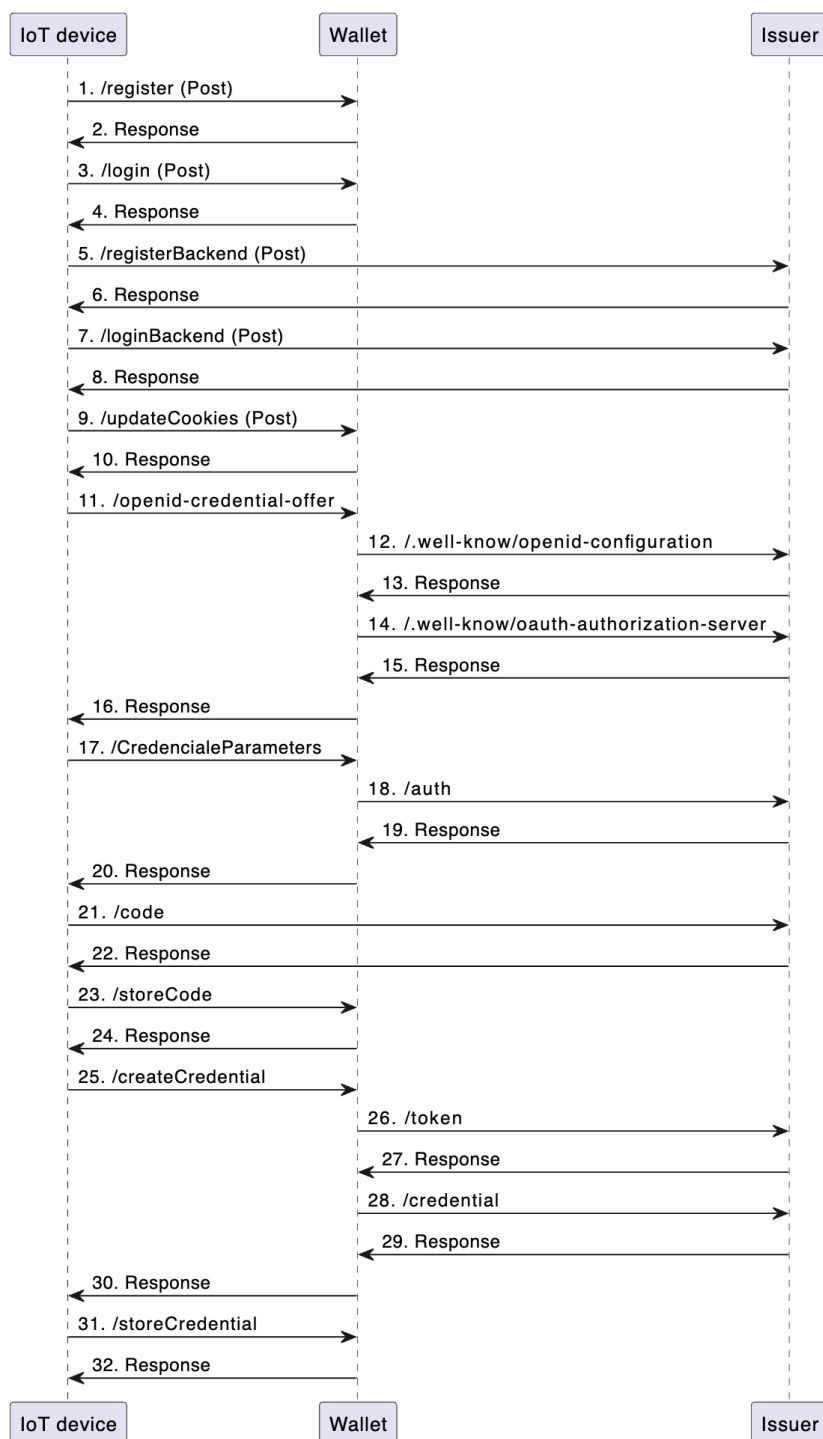


Figure 8. M2M Flow

## Verifiable Presentation Verification and Access Token Generation

This flow describes the process of obtaining an Access Token in JWT format for authenticating a user or device. The process involves interactions between the Verifier and the Wallet, utilizing a Verifiable Presentation (VP) that meets a pre-defined policy.

### 1. Initial Query to the Verifier (Steps 1-2)

- ▶ The Verifier is queried to retrieve necessary information about the Verifiable Presentation issuance process.
- ▶ **Objective:** Understand the policy, requirements, and expected formats for credential presentation.

### 2. Credential Derivation and Verifiable Presentation Generation (Steps 3-4)

- ▶ The IoT device interacts with the Wallet to derive the credential and generate a Verifiable Presentation.
- ▶ **Step 3:** The IoT device requests credential derivation from the Wallet, ensuring it aligns with the policy requirements.
- ▶ **Step 4:** The Wallet generates the Verifiable Presentation while ensuring:
  - Compliance with the policy defined by the Verifier.
  - Data integrity and confidentiality.
  - A cryptographic signature using Zero-Knowledge Proofs (ZKP) to guarantee validity.

### 3. Submitting the Verifiable Presentation and Authentication (Step 5)

- ▶ **Step 5:** The Verifiable Presentation is submitted to the Verifier for validation.
- ▶ The Verifier delegates the ZKP signature verification to a specialized entity (Verifier-zkp) that performs cryptographic checks.
- ▶ If successful, the Verifier responds with an **Access Token in JWT format**, which is then used for authentication.

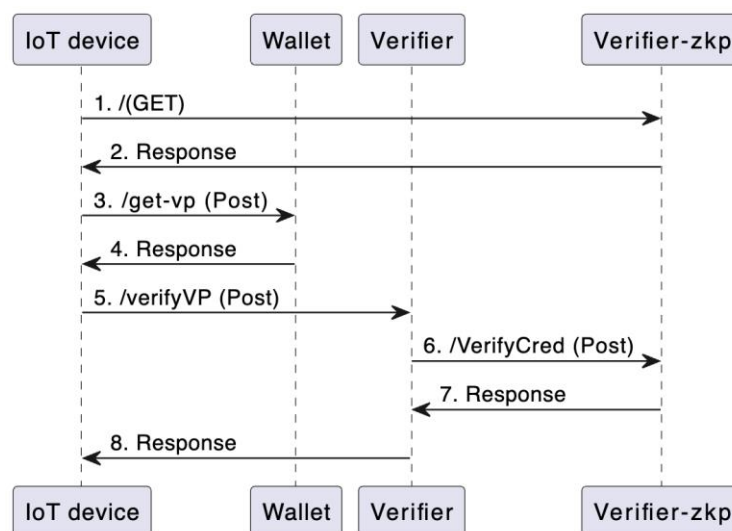


Figure 9. ZKP Verifier flow

#### 3.1.2.3 Software artifacts and position in the TANGO architecture

As described in D4.1. The SSI wallets will be provided within docker containers. The SSI agent and wallet will be implemented within the architecture as an authentication check point to enter the TANGO

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 21 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

platform. Listed below one can see in which pilots the ZKP&M2M will be implemented along with the SSI agent and wallet. The M2M issuance and verification flows are implemented in the SSI agent in conjunction with the ZKP protocol and LDP-VCs. The SSI wallet will also be packaged as an apk, enabling seamless use on android devices, as well as deep link functionality.

### 3.1.3 Final Demonstration description

#### 3.1.3.1 Docker/API description

As described from D4.1. An update being that there will be two implementations of the SSI wallet. One described as Wallet + ZKP&M2M which implements ZKP and M2M IoT flow and the other, SSI Wallet, that has granular disclosure implemented.

### 3.1.4 Support for pilots

Table 4. Pilots supported

| Pilot               | Company associated to pilot | Component features           |
|---------------------|-----------------------------|------------------------------|
| Smart Hospitality   | ANYSOL                      | SSI Agent & Wallet           |
| Autonomous Vehicles | IDIADA                      | SSI Agent & Wallet + ZKP&M2M |
| Smart Manufacturing | FMAKE                       | SSI Agent & Wallet + ZKP&M2M |
| Smart Manufacturing | RIAS                        | SSI Agent & Wallet           |
| Retail              | MET                         | SSI Agent & Wallet           |

Integration work has begun to ensure the SSI Agent and wallet work with the ePassport application to ensure seamless onboarding of customers and employees in each of the pilots. Further integration has begun with the user interface of the TANGO platform for seamless authentication in the pilots.

### 3.1.5 Final version of this component

Table 5. Final version of SSI components

| SSI Components               | Final Version of features                                                                 |
|------------------------------|-------------------------------------------------------------------------------------------|
| SSI Agent                    | Compliance with EBSI standards. Integrated with FIWARE DSC and ePassport issuer/verifier. |
| SSI Wallet                   | Granular Disclosure                                                                       |
| SSI Agent & Wallet + ZKP&M2M | Zero-knowledge proof functionality. IoT M2M authentication feature.                       |

The final version of the SSI component in TANGO includes various functionalities and is compliant with various standards and protocols.

## 3.2 Seamless onboarding

### 3.2.1 Short Component description

The Seamless Onboarding for Users and Devices component constitutes a vehicle to enable people and devices to become part of the TANGO ecosystem. In essence, it provides the elements that are required to securely onboard people and devices to the TANGO platform, enabling them to interact with the various components and functionalities. The components supports both users and devices. For the users side, a mobile app has been developed, the ePassport app, which enables users to remotely verify their identity with High Level of Assurance based on the eIDAS regulation, leveraging an existing identity document. The process involves an identity verification wizard which initially informs the user about

|                |                                                             |                |    |          |     |         |          |
|----------------|-------------------------------------------------------------|----------------|----|----------|-----|---------|----------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          |     | Page:   | 22 of 58 |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0 | Status: | Final    |

the process of identity verification that will take place, informs the user about the data that will be collected and process and retrieves the explicit consent of the user for the data collection and processing. Following the consent of the user, a four step verification process is taking place where the user is asked to: a) take a picture of the identity document and through optical character recognition to read the machine readable zone including the identity information of the person, b) to scan the NFC chip of the passport in order to securely retrieve the identity information and the facial image of the person store on the NFC chip, c) to take a selfie of the person to crosscheck the facial image of the person on the identity document with the person that is performing the identity verification process, while applying liveness detection to avoid any potential presentation attacks, d) perform validation of the information retrieved from the optical character recognition against the information presented through the NFC chip, to finally verify the identity of the person. For the devices, a unique identifier of the device will be generated by each of the devices, which will then be passed to the SSI component in order to generate verifiable credentials.

### 3.2.1.1 Internal architecture

In the D4.1, an initial working version of the Seamless Onboarding was created. The technology was tested through a host app that incorporated the SDK responsible for performing the seamless onboarding for users. The host app, i.e. the ePassport app, was fully functional in the D4.1 and operated with the sole purpose to perform and test the identity verification process offered by the Seamless Onboarding SDK.

In this deliverable D4.2, the primary changes that have occurred are related to the improvement of the SDK responsible for the seamless onboarding process based on the feedback received from the various stakeholders while testing the ePassport app, as well as during the integration of the Seamless onboarding with the SSI component. The overall internal architecture of the component stayed the same. The changes performed related to the architecture was with respect to the interactions of the component with other components such as the SSI Wallet and the SSI Issuers.

For the interaction of the Seamless onboarding with the SSI Wallet, a specific functionality was developed in order to enable the ePassport app to send but also listen to deep links from the SSI Wallet app. A functionality was developed to trigger the SSI Wallet through a deep link in order to create the credential offer. Once the credential offer has been created at the SSI Wallet, then a deep link triggers the ePassport app to initiate the identity verification process. For this functionality a specific deep link pattern was defined in collaboration with T4.1, which develops the SSI Wallet app. The technology was tested on multiple devices. Following the identity verification the information are forwarded to the selected SSI Issuer in order to issue the verifiable credentials based on the verified identity.

Table 6. Use Case: End User requests to be issued with a Verifiable Credential.

| Factor               | Description                                                                                                                                                                                                                                                                                                      |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Title</b>         | Request an ePasport Verifiable Credential                                                                                                                                                                                                                                                                        |
| <b>ID</b>            | UC_Onboarding_SSI_001                                                                                                                                                                                                                                                                                            |
| <b>Overview</b>      | An individual (natural person) can request an ePassport credential to be issued to their SSI Wallet.                                                                                                                                                                                                             |
| <b>Actor</b>         | Natural Person                                                                                                                                                                                                                                                                                                   |
| <b>Frequency</b>     | Infrequent                                                                                                                                                                                                                                                                                                       |
| <b>Trigger</b>       | End User accesses the ePassport Issuer App.                                                                                                                                                                                                                                                                      |
| <b>Preconditions</b> | 1.The user has downloaded the ePassport and SSI Wallet apps to their mobile device.<br>2. The user has an ePassport                                                                                                                                                                                              |
| <b>Main UC</b>       | 1. User selects to be issued with an ePassport Verifiable Credential on the ePassport App.<br>2. The user is requested to scan their ePassport on the mobile device.<br>3. The end user is requested to scan their face.<br>4. The ePassport App verifies the ePassport electronic authentication and the user's |

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 23 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

| Factor                | Description                                                                                                                                                                                                                                     |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | face biometrics against the electronic image stored on the ePassport.<br>5. Upon successful validation the ePassport claims are issued as a Verifiable Credential to the user.<br>6. The user accepts the ePassport credential in their wallet. |
| <b>Postconditions</b> | End User issued with an ePassport Verifiable Credential to their wallet.                                                                                                                                                                        |
| <b>Alternate UC</b>   | The user ePassport is not working and cannot be issued with an ePassport Verifiable Credential.                                                                                                                                                 |

### 3.2.1.2 Issuer Message Flow Sequence Diagram

The issuing of an ePassport credential is captured in Figure 5. ePassport VC Issuer Flow Diagram (part 1 of 2) and Figure 6. ePassport VC Issuer Flow Diagram (part 2 of 2), detailing the interwork between the mobile ePassport app, SSI Wallet and the SSI Agent.

**Flow Description with example messages:** The Figure 5 and Figure 6 sequence diagram's message flow is described below with non-normative example messages.

1. Open ePassport app and click on issue ePassport Credential
2. This initiates the following same-device request for an ePassport Verifiable Credential to the SSI Agent, as follows:

```
GET http://example.com/issuer/initiate-credential-
offer?redirect=true&credential_type=ePassport&client_id=epassport&credential_offer_endpoint=openid-
credential-offer://
```

Note: The client\_id is only used for tracking and not of much real importance at this step.

3. This returns a redirect to same-device *openid://* deeplink with credential offer uri.
4. The ePassport app performs redirect to the deeplink location.
5. The mobile wallet is invoked by a deeplink with a credential offer, as per the following example:

```
openid-credential-offer://?credential_offer_uri=http://example.com/issuer/offers/9135583383084935512
```

6. The credential offer uri is queried from the SSI Wallet

```
http://example.com/issuer/offers/9135583383084935512
```

7. The wallet receives the details of the credential offer:

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "credential_issuer": "https://example.com/issuer",
  "credentials": [
    {
      "format": "jwt_vc",
      "types": [
        "VerifiableCredential", "ePassport", "VerifiableAttestation", "VerifiableId"
      ],
      "trust_framework": {
        "name": "EBSI",
        "type": "Accreditation",
        "id": "https://api-pilot.ebsi.eu/trusted-issuers-registry/v4/issuers/did:key:z2dmzD81cgPx8Vki7Jbu"
```

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 24 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

```

    }
  },
  "grants": {
    "authorization_code": {
      "issuer_state": "tracker=123456789"
    }
  }
}

```

8. The user accepts the offer.
9. Get OpenID-credential-issuer metadata

GET <http://example.com/issuer/.well-known/openid-credential-issuer>

10. Return issuer metadata

```

{
  "credential_issuer": "https://example.com/issuer",
  "authorization_server": "https://example.com/auth",
  "credential_endpoint": "https://example.com/issuer/credential",
  "credentials_supported": [{
    "format": "jwt_vc",
    "types": ["VerifiableCredential", "employeeCredential", "customerCredential",
    "ePassport", "VerifiableAttestation", "VerifiableId"],
    "trust_framework": { "name": "EBSI", "type": "Accreditation", "https://api-pilot.ebsi.eu/trusted-issuers-registry/v4/issuers/did:key:z2dmzD81cgPx8Vki7Jbu":"" },
    "display": { "name": "EBSI Framework", "locale": "en-GB" }
  }]
}

```

11. Get issuer's authorization server metadata

GET [http:// credential-issuer.example.com/issuer/auth/.well-known/](http://credential-issuer.example.com/issuer/auth/.well-known/)

12. Return authorization server metadata

```

{
  "issuer": "https://example.com/auth",
  "redirect_uris": ["https://example.com/auth/direct_post"],
  "authorization_endpoint": "https://example.com/auth/authorize",
  "token_endpoint": "https://example.com/auth/token",
  "jwks_uri": "https://example.com/auth/auth/jwks",
  "scopes_supported": ["openid"],
  "response_types_supported": ["vp_token", "id_token", "code", "token"],
  "response_modes_supported": ["query"],
  "grant_types_supported": ["authorization_code"],
  "subject_types_supported": ["public"],
  "id_token_signing_alg_values_supported": ["ES256"],
  "request_object_signing_alg_values_supported": ["ES256"],
  "request_parameter_supported": true,
  "token_endpoint_auth_methods_supported": ["private_key_jwt"],
  "request_uri_parameter_supported": true,
  "request_authentication_methods_supported": { "authorization_endpoint": ["request_object"] },
}

```

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 25 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

```
{
  "vp_formats_supported":{"jwt_vp":{"alg":["ES256"]},
  "jwt_vc":{"alg":["ES256"]}},
  "subject_syntax_types_supported":["did:key","did:ebis"],
  "subject_syntax_types_discriminations":["did:key:jwk_jcs-pub","did:ebis:v1"],
  "subject_trust_frameworks_supported":["ebis"],
  "id_token_types_supported":["subject_signed_id_token","attester_signed_id_token"]
}
```

### 13. Authorization request

```
GET
https://example.com/auth/authorize?response_type=code&scope=openid&issuer_state=tracker=123456789&state=client-
state&client_id=did:key:z2dmzD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht1j9KbsEYvdrjxMjQ4tpnje9B
DBTzuNDP3knn6qLZERzd4bJ5go2CChoPjd5GAH3zpFJP5fuwSk66U5Pq6EhF4nKnHzDnznEP8fX99nZGgwb
Ah1o7Gj1X52Tdhf7U4KTK66xsA5r&authorization_details=[{"type":"openid_credential","format":"jwt_vc","ty
pes":["VerifiableCredential","ePassport"]}]&"redirect_uri"= "openid:
",&nonce=glkFFoisdfEui43&code_challenge=Yji0ZTQ4NTBhMzJmMmZhNjZkZDFkYzVhNzlhNGMyZDdjZ
DlkMTM4YT4NjcyMTA5M2Q2OWQ3YjNjOGJlZDBlMSAgLQo=&code_challenge_method=S256&client_
metadata={"vp_formats_supported":{"jwt_vp":{"alg":["ES256"]},"jwt_vc":{"alg":["ES256"]}},"response_types
_supported":["vp_token","id_token","token"],"authorization_endpoint":"ePassport-reader://"}
```

### 14. 302 Redirect back to SSI Wallet with a ID Token Request for ePassport included:

```
HTTP 302 Location: epassport-reader://?
client_id=https://credential-issuer.example.com/auth
&response_type=id_token
&scope=openid
&redirect_uri=https://example.com/auth/direct_post/epassport&
request=<<<<SEE DECODED REQUEST JWS BELOW>>>>
```

#### Decoded Request JWS:

-----  
JWT Header:

```
{
  "typ": "JWT",
  "alg": "ES256",
  "kid": "c4KrepJXzmBMW-qo2ntDCwkTgLM2Cb_5eabzkljTh_0"
}
```

-----  
JWT Payload:

```
{
  "iss":"https://credential-issuer.example.com/auth",
  "aud":"did:key:z2dmzD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht1j9KbsEYvdrjxMjQ4tpnje9BDBTzuN
DP3knn6qLZERzd4bJ5go2CChoPjd5GAH3zpFJP5fuwSk66U5Pq6EhF4nKnHzDnznEP8fX99nZGgwbAh1o7Gj
1X52Tdhf7U4KTK66xsA5r",
  "sub":"did:key:z2dmzD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht1j9KbsEYvdrjxMjQ4tpnje9BDBTzuN
DP3knn6qLZERzd4bJ5go2CChoPjd5GAH3zpFJP5fuwSk66U5Pq6EhF4nKnHzDnznEP8fX99nZGgwbAh1o7Gj
1X52Tdhf7U4KTK66xsA5r",
  "exp": 1589699162,
  "response_type": "id_token",
  "response_mode": "direct_post",
  "client_id": "https://example.com/auth",
}
```

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 26 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

```
"redirect_uri": "https://example.com/auth/direct_post/epassport",
"scope": "openid",
"state": "123456789",
"nonce": "n-0S6_WzA2Mj"
}
```

-----  
Signature  
-----

string

15. Redirect to ePassport app
16. ID Token Request received in ePassport App
17. Lookup the authorization server's signature received in the ID Token Request

GET https://example.com/auth/.well-known/jwks

18. Return the jwks

```
Jwks(keys=[JWKSerializable(kty=EC, crv=P-256, alg=ES256, x=N7Fpbhd_R-3p-
lMsaWcKffUg7au6KxX8mSrnYTElNsl, y=9FRC3snOiFoG4Z-yPSiFivTbvc0ytB2L0Rp06dGKGe8,
kid=z2dmzD81cgPx8Vki7Jbu#z2dmzD81cgPx8Vki7Jbu)])
```

19. User sees request to authenticate to ePassport App on mobile and consents
20. User successfully scans the ePassport by the app
21. User presents face for verification
22. ePassport verified
23. Face verified against ePassport image
24. ID Token Response (Direct Post) from Wallet to Authorization Server of the Issuer

POST /auth/direct\_post/epassport  
Host: https://example.com  
Content-Type: application/x-www-form-urlencoded  
state=123456789,  
&id\_token=<<<SEE DECODED ID\_TOKEN JWS BELOW>>>

#### Decoded ID\_Token JWS:

-----  
JWT Header:  
-----

```
{
  "typ": "JWT",
  "alg": "none"
}
```

-----  
JWT Payload:  
-----

```
{
  "iss": "did:key:z2dmzD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht1j9KbsEYvdrjxMjQ4tpnje9BDBTzuND
P3knn6qLZErzd4bJ5go2CChoPjd5GAH3zpFJP5fuwSk66U5Pq6EhF4nKnHzDnznEP8fX99nZGgwbAh1o7Gj1
X52Tdhf7U4KTK66xsA5r",
  "sub": "did:key:z2dmzD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht1j9KbsEYvdrjxMjQ4tpnje9BDBTzuN
DP3knn6qLZErzd4bJ5go2CChoPjd5GAH3zpFJP5fuwSk66U5Pq6EhF4nKnHzDnznEP8fX99nZGgwbAh1o7Gj1
X52Tdhf7U4KTK66xsA5r",
```

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 27 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

```

"did:key:z2dmzD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht1j9KbsEYvdrjxMjQ4tpnje9BDBTzuNDP3kn
n6qLZErzd4bJ5go2CChoPjd5GAH3zpFJP5fuwSk66U5Pq6EhF4nKnHzDnznEP8fX99nZGgwbAh1o7Gj1X52T
dhf7U4KTK66xsA5r",
"address": "Seestraße 1",
"passportNumber": "C7648556",
"familyName": "Little",
"givenName": "Ross",
"dateOfBirth": "1830-10-01",
"placeOfBirth": "Zug",
"gender": "M",
"authority": "Aarau AG",
"nationality": "Schweiz",
"countryCode": "CHE",
"passportPhoto": "data:image/png;base64,iVBORw0KGgo...kJggg=="
"dateOfExpiry": "1935-10-01",
"dateOfIssue": "11950-10-01",
"aud": "https://credential-issuer.example.com/auth",
"exp": 1589699360,
"iat": 1589699260,
"state": "123456789",
"nonce": "n-0S6_WzA2Mj"
}
-----
Signature
-----

```

25. Store ePassport credential Subject claims temporarily for the issuer session
26. Redirect Authorization Response to the SSI Wallet on the mobile

```

HTTP/1.1 302 Found
Location: openid://redirect?code=6892848073776569093&state=client-state

```

27. Process redirect at the ePassport app to send to SSI Wallet deeplink
28. Authorization Response received at the SSI Wallet
29. Token Request

```

POST /auth/token
Host: https://example.com
Content-Type: application/json

grant_type=authorization_code
&client_id=did:key:z2dmzD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht1j9KbsEYvdrjxMjQ4tpnje9BDBT
zuNDP3knn6qLZErzd4bJ5go2CChoPjd5GAH3zpFJP5fuwSk66U5Pq6EhF4nKnHzDnznEP8fX99nZGgwbAh1o
7Gj1X52Tdhf7U4KTK66xsA5r
&code=6892848073776569093
&code_verifier=rnzYLRUEKPDNYgAiZDCXo8yvd3qRvJ5J50GvM-YCmclWg8iziOSgj8oVJhCErXn3y4I

```

Note:

If the method is plain, then the authorization server needs only to check that the provided code\_verifier matches the expected code\_challenge string. If the method is S256, then the authorization server should take the provided code\_verifier and transform it using the same hash method, then comparing it to the stored code\_challenge string in the initial authorization request.

30. Token Response

```

HTTP Response Payload: Content-Type application/json

```

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 28 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

```
{
"token_type": "Bearer",
"expires_in": 1711454631639,
"c_nonce": "6892848073776569093",
"c_nonce_expires_in": 1711454631639,
"access_token": "SEE DECODED ACCESS TOKEN JWS BELOW"
```

#### Decoded Access Token Credential JWS:

-----  
JWT Header:

```
{
  "kid": "did:key:z2dmzD81cgPx8Vki7Jbu#z2dmzD81cgPx8Vki7Jbu",
  "typ": "JWT",
  "alg": "ES256"
}
```

-----  
JWT Payload:

```
{
  "aud": "https://example.com/issuer",
  "sub":
    "did:key:z2dmzD81cgPx8Vki7JbuMmFYrWPgYoytykUZ3eyqht1j9KbsEYvdrjxMjQ4tpnje9BDBTzuNDP3kn6qLZErzd4bJ5go2CChoPjd5GAH3zpFJP5fuwSk66U5Pq6EhF4nKnHzDnznEP8fX99nZGgwbAh1o7Gj1X52Tdhf7U4KTK66xsA5r",
  "iss": "https://example.com/auth",
  "claims": {
    "authorizationDetails": [
      {
        "type": "openid_credential",
        "format": "jwt_vc",
        "locations": [
          "https://example.com/issuer"
        ],
        "types": [
          "VerifiableCredential",
          "ePassport"
        ]
      }
    ],
    "cNonce": "6892848073776569093",
    "cNonceExpiresIn": 1711454631639,
    "clientId":
      "did:key:z2dmzD81cgPx8Vki7JbuMmFYrWPgYoytykUZ3eyqht1j9KbsEYvdrjxMjQ4tpnje9BDBTzuNDP3kn6qLZErzd4bJ5go2CChoPjd5GAH3zpFJP5fuwSk66U5Pq6EhF4nKnHzDnznEP8fX99nZGgwbAh1o7Gj1X52Tdhf7U4KTK66xsA5r"
  },
  "exp": 1811454631,
  "iat": 1711454331,
  "nonce": "6892848073776569093"
}
```

-----  
Signature

-----  
string

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 29 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

## 31. Credential Request

```
POST /issuer/credential
Host: https://example.com
Content-Type: application/json
Authorization: BEARER <<JWS Access Token obtained from the previous step>>
{
  "format": "jwt_vc",
  "types": [
    "VerifiableCredential", "ePassport"
  ],
  "proof": {
    "proof_type": "jwt",
    "jwt": "SEE DECODED CREDENTIAL REQUEST JWT PROOF BELOW"
  }
}
```

**Decoded Credential Request JWT PROOF:**

-----  
JWT Header:

```
-----
{
  "kid":
  "did:key:z2dmzD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht1j9KbsEYvdrjxMjQ4tpnje9BDBTzuNDP3kn
  n6qLZERzd4bJ5go2CChoPjd5GAH3zpFJP5fuwSk66U5Pq6EhF4nKnHzDnznEP8fX99nZGgwbAh1o7Gj1X52T
  dhf7U4Ktk66xsA5r#z2zD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht",
  "typ": "JWT",
  "alg": "ES256"
}
```

-----  
JWT Payload:

```
-----
{
  "iss":
  "did:key:z2dmzD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht1j9KbsEYvdrjxMjQ4tpnje9BDBTzuNDP3kn
  n6qLZERzd4bJ5go2CChoPjd5GAH3zpFJP5fuwSk66U5Pq6EhF4nKnHzDnznEP8fX99nZGgwbAh1o7Gj1X52T
  dhf7U4Ktk66xsA5r",
  "aud": "https://example.com/issuer",
  "nonce": "6892848073776569093",
  "iat": 1711454331
}
```

-----  
Signature

-----  
string

32. Issuer creates an ePassport Verifiable Credential

33. SSI Wallet receives an ePassport credential from the issuer.

HTTP Response Payload: Content-Type application/json

```
{ "format": "jwt_vc",
  "credential": "SEE DECODED VERIFIABLE CREDENTIAL ISSUED AS A JWS BELOW" }
```

**Decoded Verifiable Credential ISSUED AS A JWS:**

-----

|                |                                                             |                |    |          |       |               |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|---------------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 30 of 58      |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status: Final |

#### JWT Header:

```
{
  "kid": "did:key:z2pxbzD81cg123#z2pxbzD81cg123",
  "typ": "JWT",
  "alg": "ES256"
}
```

#### JWT Payload:

```
{
  "sub":
  "did:key:z2dmzD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht1j9KbsEYvdrjxMjQ4tpnje9BDBTzuNDP3kn
  n6qLZErzd4bJ5go2CChoPjd5GAH3zpFJP5fuwSk66U5Pq6EhF4nKnHzDnznEP8fX99nZGgwbAh1o7Gj1X52T
  dhf7U4KTK66xsA5r",
  "nbf": 1711454331,
  "iss": "did:key:z2pxbzD81cg123",
  "exp": 1711454331,
  "iat": 1711454331,
  "vc": {
    "@context": [
      "https://www.w3.org/2018/credentials/v1",
      "https://www.w3.org/2018/credentials/examples/v1"
    ],
    "type": [
      "VerifiableCredential",
      "ePassport"
    ],
    "id": "vc:uself:agent:#8133274365875303553",
    "credentialSchema": {
      "id": "https://api-conformance.ebsi.eu/trusted-schemas-
      registry/v2/schemas/z3MgUFUkb722uq4x3dv5yAJmnNmzDFeK5UC8x83QoeLJM",
      "type": "FullJsonSchemaValidator2021"
    },
    "validFrom": "2024-03-26T11:58:51Z",
    "issued": "2024-03-26T11:58:51Z",
    "issuer": "did:key:z2pxbzD81cg123",
    "issuanceDate": "2024-03-26T11:58:51Z",
    "expirationDate": "2024-03-26T11:58:51Z",
    "credentialSubject": {
      "id": "did:key:z2dmzD81cgPx8Vki7JbuuMmFYrWPgYoytykUZ3eyqht1j9KbsEYvdrjxMjQ4tpnje9BDBTzuND
      P3knn6qLZErzd4bJ5go2CChoPjd5GAH3zpFJP5fuwSk66U5Pq6EhF4nKnHzDnznEP8fX99nZGgwbAh1o7Gj1
      X52Tdhf7U4KTK66xsA5r",
      "address": "Seestraße 1",
      "passportNumber": "C7648556",
      "familyName": "Little",
      "givenName": "Ross",
      "dateOfBirth": "1830-10-01",
      "placeOfBirth": "Zug",
      "gender": "M",
      "authority": "Aarau AG",
      "nationality": "Schweiz",
      "countryCode": "CHE",
      "passportPhoto": "data:image/png;base64,iVBORw0KGgo...kJggg=="
      "dateOfExpiry": "1935-10-01",
      "dateOfIssue": "11950-10-01"
    }
  }
},
```

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 31 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

```
"jti": "vc:uself:agent:#8133274365875303553"
}
```

-----  
Signature  
-----

string

34. End user accepts the ePassport Verifiable Credential at the wallet

35. Credential is stored in the wallet

### 3.2.1.3 Implemented features

During the deliverable D4.1, the features foreseen by the description of work were developed. Some key features that were developed are a) the optical character recognition that is able through the camera of the smartphone to detect a passport and read the machine readable zone that includes the information of the passport, b) to implement the functionality that manages the NFC of the smartphone including the chip authentication and performs the reading of the identity information stored at the NFC chip, c) to further develop algorithms for face detection, face matching and liveness detection for the verification of the person performing the onboarding process against the identity document used in the verification process, d) to crosscheck the information retrieved from the optical character recognition and the NFC chip in order to verify the identity with High Level of Assurance.

Following the development of the initial prototype, various tests were performed by the engaged stakeholders in terms of the functionality of the ePassport component. The key features that were developed after the D4.2 are the following, in order to address the comments and feedback received during the testing of the technology:

- ▶ **Support of Finnish passport.** During the initial testing process the ePassport app was tested with real people against passports from United Kingdom, Sweden, Greece, Spain and Finland. During the tests with the Finnish passport a particular issue was observed when reading the NFC, which prohibited the users to finalize the identity verification process if a Finnish passport was used during the onboarding process.
- ▶ **Integration with SSI.** In order to enable the component to onboard seamlessly the users for TANGO, QBE performed integration of the SSI through enabling the ePassport app to be triggered from the SSI wallet but also to communicate with an SSI Issuer. Following the integration process, the ePassport app is able to receive the credential offer, verify the identity of the person and then forward the passport details to the respective Issuer.
- ▶ **Support of multiple issuers.** Considering the fact that in the TANGO project there are more than one Issuers that are being developed, a specific functionality was developed in order to enable the user to choose which issuer to use when initiating the onboarding process. Currently, the two Issuers supported are from ATOS and UMU.

### 3.2.1.4 Software artifacts

The Seamless Onboarding consists of two main artifacts:

- ▶ **ePassport app.** This is a host app that has integrated the user onboarding SDK which performs the identity verification. Apart from the hosting of the SDK, the ePassport app is responsible for the communication through deep links with the SSI Wallet developed in T4.1 both triggering and listening for deep links. The ePassport app provides also the capability of choosing the appropriate Issuer, at which the credentials will be created.
- ▶ **Seamless Onboarding SDK.** This constitutes an SDK developed through the Kotlin Multiplatform framework and triggers the wizard that performs the identity verification. The wizard initially informs the user about what the SDK performs, the data collected and then chooses which type of identity document and from which country the document has been issued. Following, through optical character recognition the SDK retrieves the information of the document and the similarly through

|                       |                                                                |                       |    |                 |              |                      |
|-----------------------|----------------------------------------------------------------|-----------------------|----|-----------------|--------------|----------------------|
| <b>Document name:</b> | D4.2 Distributed Trust Management Framework -<br>Final version |                       |    |                 | <b>Page:</b> | 32 of 58             |
| <b>Reference:</b>     | D4.2                                                           | <b>Dissemination:</b> | PU | <b>Version:</b> | 1.0          | <b>Status:</b> Final |

the NFC. A facial image is retrieved from the user in order to matching the image of the person visible at the document with the person that performs the onboarding process, and finally crosschecks the information of the document retrieved through the optical character recognition and the NFC chip.

The ePassport app with integrated Seamless Onboarding SDK interacts directly with the SSI Wallet and the SSI Issuer developed in T4.1.

### 3.2.2 Demonstration description

For the demonstration, a presentation has been created that guides the users step by step on how to perform the identity verification process and how to ensure the successful verification of the person. The presentation was shared with UPRC in order to forward the information to the respective pilot use cases in order to test and demonstrate the technology. The version 0.1.10 of the ePassport app was shared with the pilot use cases.

#### 3.2.2.1 Docker/API description

The API has remained the same as described in D4.1.

### 3.2.3 Support for pilots

The ePassport component app is planned to be shared with the following pilots:

- ▶ Retail
- ▶ Smart Hospitality
- ▶ Smart Manufacturing
- ▶ Autonomous Vehicles

The plan is currently to test the particular component initially as an individual identity verification mechanism, in order to ensure the correct implementation of the onboarding mechanism. The testing will take place through different types and versions of devices in order to ensure the compatibility across diverse devices. Following the testing of the ePassport app as an individual component, the testing through the pilots will continue as an integrated component with the SSI (T4.1). There the users will test the overall onboarding process to the SSI similar to the registration process to the TANGO platform.

### 3.2.4 Final version of this component

As the ePassport is tested by the various stakeholder potential improvement requests may arise based on the received feedback. While monitoring the performance of the ePassport app, these improvements will be incorporated in the app and the onboarding mechanism in order to enhance the functionality and robustness of the component. Each of the pilot sites will provide feedback and the potential refinements that may arise will be incorporated in the core version of the component.

## 3.3 User behavioural authentication

### 3.3.1 Short Component description

The User Continuous Behavioural Authentication is a component that focuses on strengthening the security of particular online systems while enhancing the user experience. The component leverages biometrics and behavioural patterns of the user to continuously authenticate the user while accessing a particular system. Behavioural patterns such as the way a person is moving, the way they are using a particular device, and potential transactional patterns are uniquely combined with biometrics in order to provide strong reassurance regarding if the account owner is the person that is accessing the system at any point of time. The component combines the aforementioned behavioural patterns of the user and generates a risk score which is exposed to the related system, in order to enable the relevant stakeholders to take the appropriate actions. The risk score can then be leveraged by other components such as the Security & Privacy, which can leverage the behavioural risk score as an additional attribute at the predefined access policies. In this way, in order to provide access to a particular system, the TANGO

|                       |                                                                |                       |    |                 |              |                      |
|-----------------------|----------------------------------------------------------------|-----------------------|----|-----------------|--------------|----------------------|
| <b>Document name:</b> | D4.2 Distributed Trust Management Framework -<br>Final version |                       |    |                 | <b>Page:</b> | 33 of 58             |
| <b>Reference:</b>     | D4.2                                                           | <b>Dissemination:</b> | PU | <b>Version:</b> | 1.0          | <b>Status:</b> Final |

ecosystem will provide strong security by not only leveraging password based policies, but also incorporating additional attributes in terms of the behavior of the user.

### 3.3.1.1 Internal architecture

The architecture of the system does not have major changes since the D4.1. The focus of the architectural modifications have been pointed primarily on the integration of the User continuous behavioural authentication component with the PEP and PDP described in deliverable D3.1 and D3.2. The attribute-based authentication of the user has provided the capability of incorporating additional contextual information in the authentication process in order to grant access to a user based on specific contextual information. The user continuous behavioural authentication component due to its nature in continuously assessing the user while accessing a particular application. Thus, in addition to the pre-default configured attributed of the defined policy, an exploration was performed of adding the user continuous behavioural authentication risk score to the PDP in order to generate the appropriate Capability Token, as shown in the figure below.

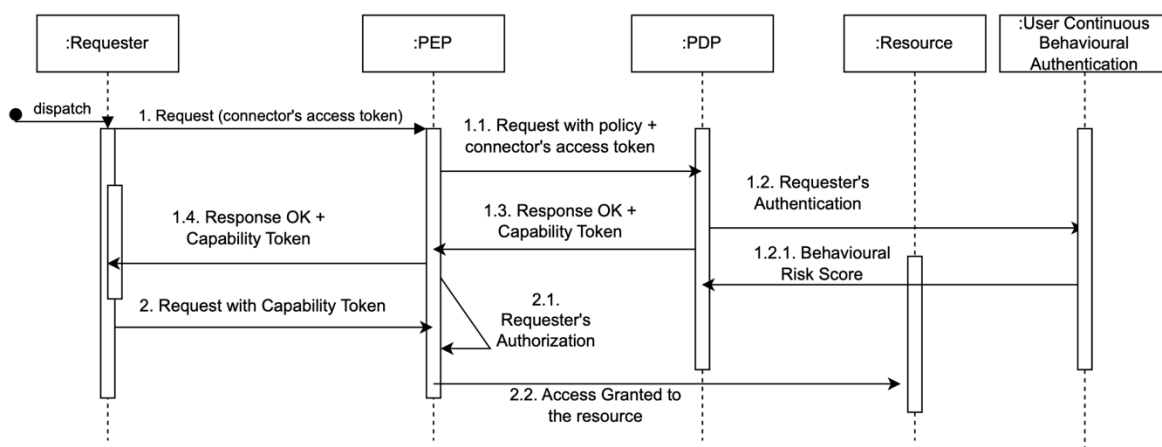


Figure 10. UCBA and PDP integration flow

### 3.3.1.2 Implemented features

The work implemented has been mainly focused on improving the performance of the models developed for the user continuous behavioural authentication.

- ▶ **Add manual labelling.** At the initial version of the component the system generated automatically an estimation of the legitimacy of the user. At the administration panel the admin user was initial only able to review the information regarding the particular session. Through additional work and through the testing process it was evident that there was a need to incorporate a mechanism to label the different sessions that may require further assessment. For this reason, a capability was created that enables the administrator to add a tag to the particular session. In this way, the administrator can label the session and further exploration of the session can take place, actions for improvement etc.
- ▶ **Expose the risk score.** A dedicated API was developed in order to enable the exposure of the risk score computed by the component to third party components. Initially the risk score was computed at the backend of the component and through a particular API with end-user authentication, was returning the result to the end-user's device. There is specific action that could take place such as presenting a lock screen. In order to assist and enable the communication with other components of the TANGO platform, a dedicated API was developed envisioned for communication between backend components and is not intended for use by end-user devices.
- ▶ **Similarity score.** The initial approach regarding the output of the system was to provide a binary result (legit/imposter). The decision for this approach was primarily in order to enable the developer to easily decide on the action to be taken. However, there were cases where a more fine-grained estimation was required. For this reason, the system was modified in order to provide a human

|                |                                                             |                |    |          |     |         |          |
|----------------|-------------------------------------------------------------|----------------|----|----------|-----|---------|----------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          |     | Page:   | 34 of 58 |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0 | Status: | Final    |

understandable score that showcases the similarity of the owner of the account against the user that has currently access to the system.

### 3.3.1.3 Software artifacts

The User Continuous Behavioural Authentication component comprises two software artifacts:

- ▶ The SDK that is responsive for the data collection and the data offloading to the backend component. This component supports both mobile and web interfaces. The SDK is integrated with the host apps that are about to be protected, in order to ensure that during the access of the user to the particular apps, that the component is able to continuously assess the user that is interacting with the particular host app.
- ▶ The backend component receives the behavioural data from the SDK and initiates the processing and analysis of the behavioural data to derive the behavioural data and estimate the behavioural score of the user. The backend component has been developed to be deployed on Kubernetes combined with docker images of each of the behavioural trait components. Considering the high cloud consumption requirements, the component has been deployed at the QBE cloud infrastructure.

### 3.3.2 Demonstration description

For the demonstration of the technology, a demo banking app has been developed in order to showcase the user experience but was also utilized for testing and validation of the user continuous behavioural authentication component. The demo banking app incorporates the common functionality provided by a banking app such as checking the balance of the account, providing a list of the transactions performed by the owner of the account (incoming and outbound), showing the details of a transaction that has been performed, adding a beneficiary to whom the money will be transferred to, performing a transaction to a selected beneficiary including specifying the amount and adding a justification to the transaction, some options and configuration of the app where the user can be informed about the user continuous behavioural authentication, the biometric and behavioural data can be deleted, the user continuous behavioural authentication can be enabled/disabled by the user, and some additional information regarding the privacy of the user. All these functionalities are commonly available at a banking app.

Following the development of the demo banking app, a guideline is currently being created in order to assist the end-users in the evaluation process of the user continuous behavioural authentication component. The presentation will be shared with the relevant stakeholders in order to enable them to download the demo app with integrated the user continuous behavioural authentication component, as well as to use the demo banking app and perform certain transactions. In this way, feedback will be provided to QBE to perform further refinements and improvements of the solution.

#### 3.3.2.1 Docker/API description

The API has remained the same as described in D4.1.

### 3.3.3 Support for pilots

The User Continuous Behavioural Authentication component is planned to be shared with the following pilots:

- ▶ Retail
- ▶ Smart Hospitality
- ▶ Smart Manufacturing
- ▶ Autonomous Vehicles

As the User Continuous Behavioural Authentication is deployed through an SDK, there is a need to integrate the SDK with a particular host app either mobile or web app, which will then be used by the targeted end-users. For the demonstration purposes, QBE developed an additional mobile host app that represents a demo banking app. This demo app was used for testing purposes and in order to collect behavioural data and assess the usage of the particular apps. This demo banking app is being shared

|                       |                                                             |                       |    |                 |              |                      |
|-----------------------|-------------------------------------------------------------|-----------------------|----|-----------------|--------------|----------------------|
| <b>Document name:</b> | D4.2 Distributed Trust Management Framework - Final version |                       |    |                 | <b>Page:</b> | 35 of 58             |
| <b>Reference:</b>     | D4.2                                                        | <b>Dissemination:</b> | PU | <b>Version:</b> | 1.0          | <b>Status:</b> Final |

with related stakeholders from Retail and Smart Hospitality in order to perform preliminary testing of the technology as an individual and isolated component. The component is being integrated with the T3.3 Security & Privacy where the extracted behavioural risk score will be used as an attribute to the attribute-based access policy.

### 3.3.4 Final version of the component

The User Continuous Behavioural Authentication component enables the continuous assessment of the user when accessing a mobile or a web application. During the access of the particular application the system collects various behavioural elements that can uniquely identify an individual, so combined they can offer a high degree of confidence regarding if the legitimate user is accessing the particular application. The component will be integrated across various application domains that are defined in the TANGO project, in order to provide strong reassurance regarding the authenticity of the user accessing the particular application. As the component integrates across the different applications on mobile or web, potential feedback may arise from the relevant stakeholders. The related feedback will be incorporated to provide further enhancements to the component addressing the needs of the end-users.

## 3.4 Device behavioural authentication

### 3.4.1 Short Component description

The device behavioural authentication mechanism provides continuous verification of a device's identity in real time, extending security beyond traditional user credentials like PINs or passwords. Once a user logs in with their credentials, the system begins monitoring the device's activity by collecting and analysing its system logs. These logs are continuously captured and transmitted to a server, where a Risk Assessment Engine—a Machine Learning model—analyses them to detect any deviations from the device's normal behaviour. If unusual activity is identified, the system may request additional user verification, such as re-entering the device's PIN. This approach ensures ongoing monitoring and helps maintain the security and integrity of the device during the entire user session, as illustrated in Figure 11.

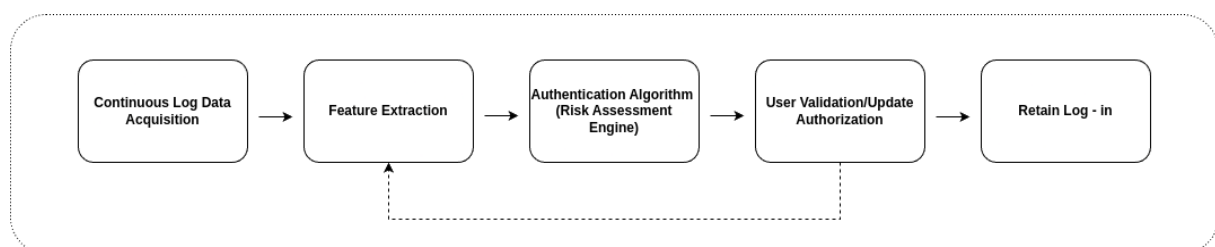


Figure 11. Continuous Authentication Flowchart

The main components of the implementation, as shown in the Figure 12:

- ▶ The Android Application ‘*Authenticator*’: This client-side component is an Android application developed in Java using Android Studio, currently installed on an Android 10.0 device (Lenovo Tablet). The application manages user interactions, such as login, registration, and logout, and runs a foreground service to continuously capture Android system logs and transmit them to the Server for further processing.
- ▶ The Backend Server: The server is deployed in the cloud and developed using JavaScript with the Node.js runtime environment. It manages user authentication, registration, and login/logout requests. Additionally, it continuously stores and analyzes the Android system logs of the device associated with the logged-in user.

|                |                                                             |                |    |          |     |         |          |
|----------------|-------------------------------------------------------------|----------------|----|----------|-----|---------|----------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          |     | Page:   | 36 of 58 |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0 | Status: | Final    |

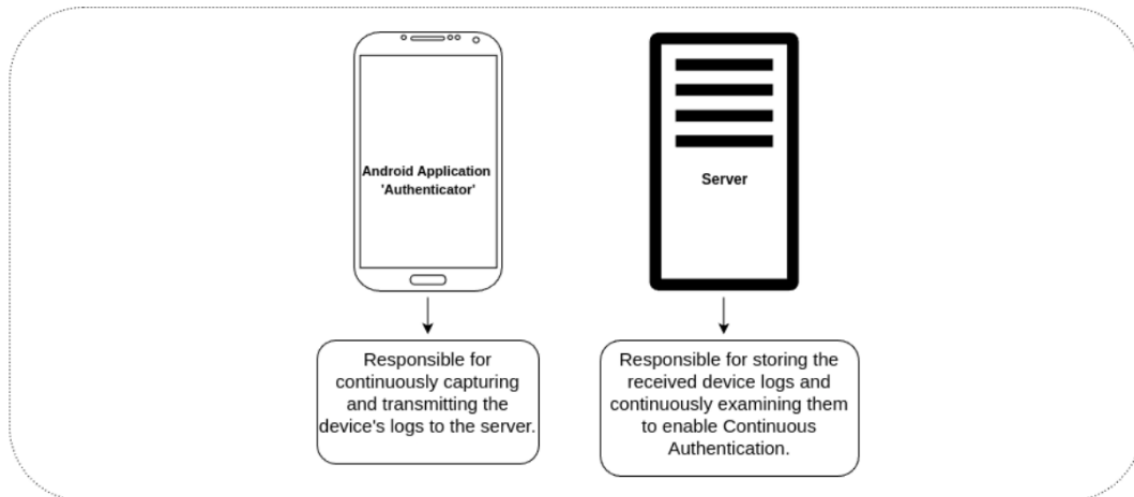


Figure 12. System Architecture Components

Important Note: The SSI concept, at the time it will be available, will be used for the authentication. For the development and testing purposes of the device behavioural authentication mechanism, a simple custom authentication mechanism was implemented, based on Public-Key Cryptography, which will be replaced by the SSI mechanism.

#### 3.4.1.1 Internal architecture

As mentioned in the previous deliverable D4.1, we encountered some **issues** with capturing device logs due to specific initialization requirements. Android permissions restrictions prevent third-party applications (i.e., applications not pre-installed on the device, such as ‘*Authenticator*’) from accessing system logs. They are only allowed to capture logs generated by the app itself. To resolve this, we rooted the Android devices. This process involved finding the original firmware of the devices and, through a complex procedure, rooting the stock firmware using a third-party application called ‘Magisk Manager’. Afterward, we flashed the corresponding rooted images to these devices. Once rooted, ‘*Authenticator*’ is granted access to the system logs. Therefore, the devices running our application must be rooted; otherwise, log collection cannot be performed, and the entire implementation will not function.

Proceeding with the **internal architecture**, it remains consistent with the design outlined in D4.1. The application communicates with the server by sending appropriate HTTP requests to the server’s URL endpoints. Currently, these requests are handled locally due to the absence of a public IP. The future implementation will utilize HTTPS protocol, incorporating SSL (Secure Sockets Layer) and TLS (Transport Layer Security) protocols, to encrypt the data transmission between users and server.

The current implementation supports key functionalities, presented in the Table 7.

Table 7. Internal Architecture Functionalities

| Functionality     | Description                                                                                                                          |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| USER LOGIN        | Users can log in using their credentials, which currently include a username and password for testing purposes.                      |
| USER REGISTRATION | Users can create an account by providing their credentials, which currently consist of a username and password for testing purposes. |
| USER LOGOUT       | Users can log out at any time by clicking the “LOGOUT” button.                                                                       |

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 37 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

| Functionality       | Description                                                                                                                             |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| ANDROID LOG CAPTURE | Upon user login, a foreground service is activated to capture the device's Android system logs and transmit them to the backend server. |

The primary objective of the system is to consistently identify the users based on various device behaviours. For the authentication, public-key cryptography is employed alongside a password mechanism, currently used for testing purposes. Once a user is authenticated, the Android application continuously captures and transmits Android system logs to the server. These logs provide valuable insights into device behaviour, including Network and Bluetooth connections, as well as Battery statistics.

As part of the smart manufacturing use case, DCBA will serve as an authentication and authorization mechanism for users regarding their physical access to factory premises. Leveraging the fact that we have already implemented the mechanism for extracting Wi-Fi metrics and other relevant data, we can enhance access control by utilizing these signals for user verification.

Since Android system logs do not provide location data, earlier implementations utilized GPS to retrieve the device's latitude and longitude at random intervals. However, this approach proved inefficient due to its significant impact on battery consumption. To address this, the application performs periodic scans and generates custom logs to capture RSSI (Received Signal Strength Indicator) signals from nearby Wi-Fi and Bluetooth Low Energy (BLE) access points or devices. This enables the implementation of a localization mechanism, allowing the application to estimate the device's location based on these RSSI signals.

#### 3.4.1.2 Implemented features

The implemented features remain consistent with the previous version, with minor adjustments made to improve the overall functionality. Specifically, we will examine and analyse the following:

- **USER REGISTRATION:** User attempts to create an account.
  - The users enter their credentials (username and password) to create an account. A registration request is then generated using these specific credentials.
  - If these credentials are valid:
    - The server generates and sends a registration challenge to the 'Authenticator', the application running on the user's device (client-side).
    - The 'Authenticator' generates a public-private key pair and signs the registration challenge using the private key, thereby creating the registration signature.
    - The registration signature, along with the corresponding public key associated with the user attempting to create the account and the original registration challenge, is sent back to the server for verification. The private key remains stored in the user's device internal storage, accessible only by the 'Authenticator' application.
    - The server verifies the registration signature using the registration challenge and the corresponding public key. If the signature is valid, the registration request is accepted, and the registration is successful. Additionally, a text file associated with the user is created in the device's internal storage to locally append logs when the device is offline (by all means, out of network). The URI of this file, along with the user's username, encrypted password, and device model name, are stored in the database. If the signature is invalid, the registration request is rejected.
  - If the credentials are invalid, the registration request is discarded, and the registration process fails.

The entire registration pipeline is illustrated in the Figure 13.

|                |                                                             |                |    |          |     |         |          |
|----------------|-------------------------------------------------------------|----------------|----|----------|-----|---------|----------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          |     | Page:   | 38 of 58 |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0 | Status: | Final    |

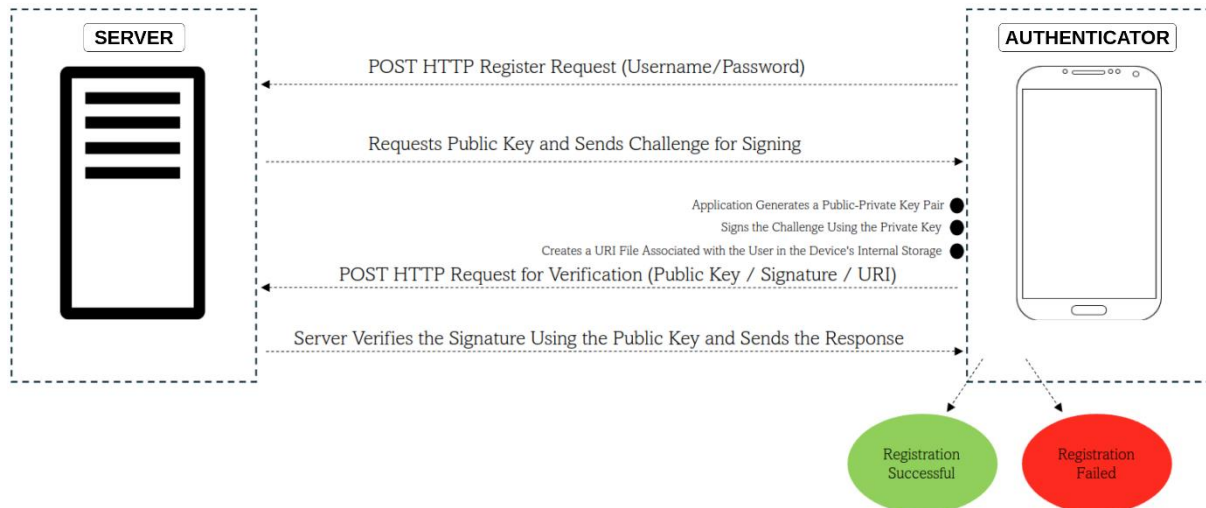


Figure 13. User Registration Process Pipeline

- **USER LOGIN:** User attempts to authenticate and log in.
  - The users enter their username and password to log in.
  - If these credentials are correct:
    - The server generates a login challenge and sends it to the 'Authenticator' to be signed using the private key of the corresponding user, which resides solely on the user's device and is accessible only through the 'Authenticator' application.
    - The 'Authenticator' retrieves the corresponding private key from its internal storage, signs the login challenge, and generates the login signature, which is then sent back to the server for verification.
    - The server searches the database for the corresponding user, retrieves his/her public key, signs the login challenge, and then compares the two signatures: the one sent by the 'Authenticator' app using the private key, and the one generated by the server using the corresponding public key.
    - If the user's signature is valid, the user is successfully authenticated and logged in. Therefore, an authentication token based on JSON Web Token (JWT) is generated to keep the user logged into the application for the duration of the session. If the signature is invalid, the authentication process fails.
  - If the credentials are incorrect, the login fails without any further steps being involved.

The entire login pipeline is illustrated in the Figure 14.

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 39 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

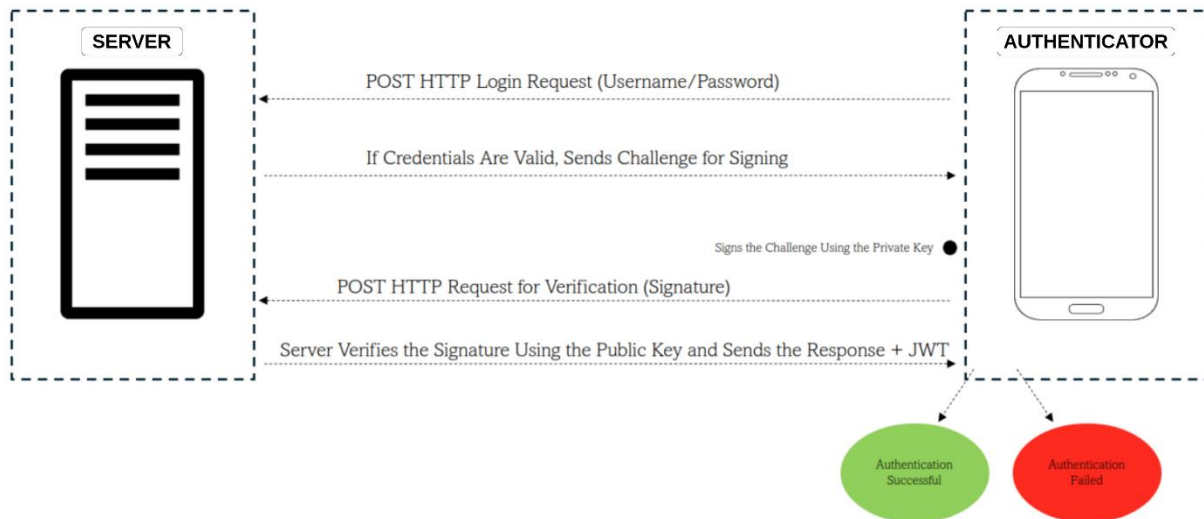


Figure 14. User Login Process Pipeline

- **ANDROID LOG CAPTURE:** Upon user login, the Android log capture process begins, and the logs are transmitted to the server in real-time.
  - Along with the logs, the user's username and authentication token are transmitted to the server for token verification, user authentication, and log association. If the authentication token is valid and not expired, the logs are stored in the database, anomaly detection is performed using the Risk Assessment Engine (ML model), and a status code is returned. On the other hand, if the authentication token is expired or an error occurs, an appropriate error message is provided. Figure 15 illustrates the entire log capture process pipeline.

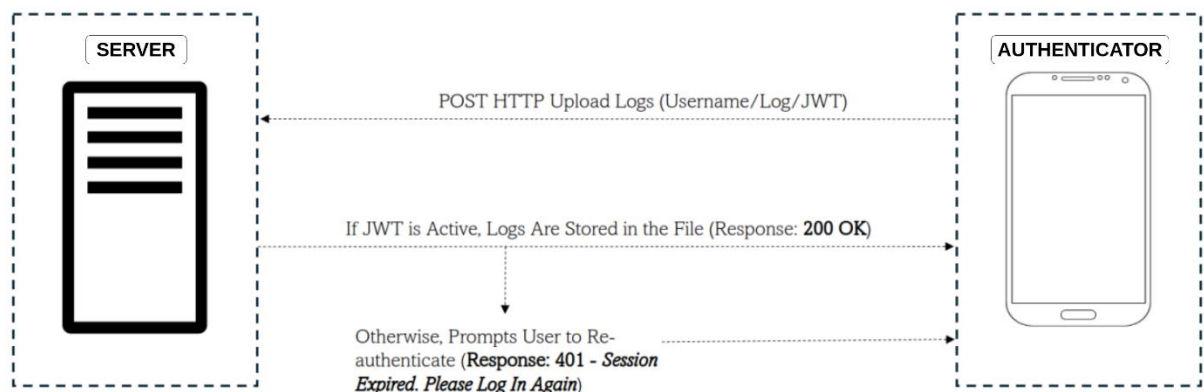


Figure 15. Android Log Capture Process Pipeline

- Upon logging into the 'Authenticator,' the user can interact with the device as usual. This is because a foreground service, started by the application and associated with the logged-in user, is responsible for capturing and filtering system logs to transmit them to the server, as it is depicted in the Figure 16.

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 40 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

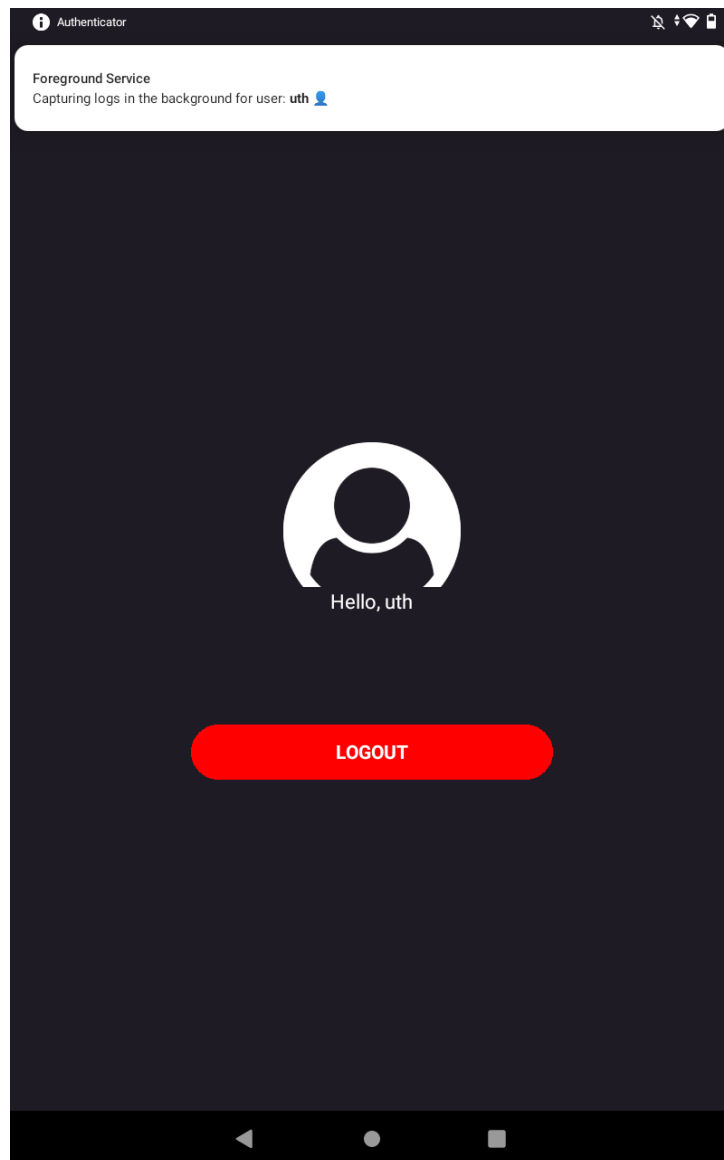


Figure 16. Foreground Service

- A crucial aspect is what happens when the device is offline or out of network coverage during the android log capture process, as we can see in the Figure 17. In such cases, the logs are stored in the text file created during user registration, as mentioned earlier. Once the device regains internet access, these stored logs are transmitted to the server along with any new logs.

|                |                                                             |                |    |          |     |         |          |
|----------------|-------------------------------------------------------------|----------------|----|----------|-----|---------|----------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          |     | Page:   | 41 of 58 |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0 | Status: | Final    |

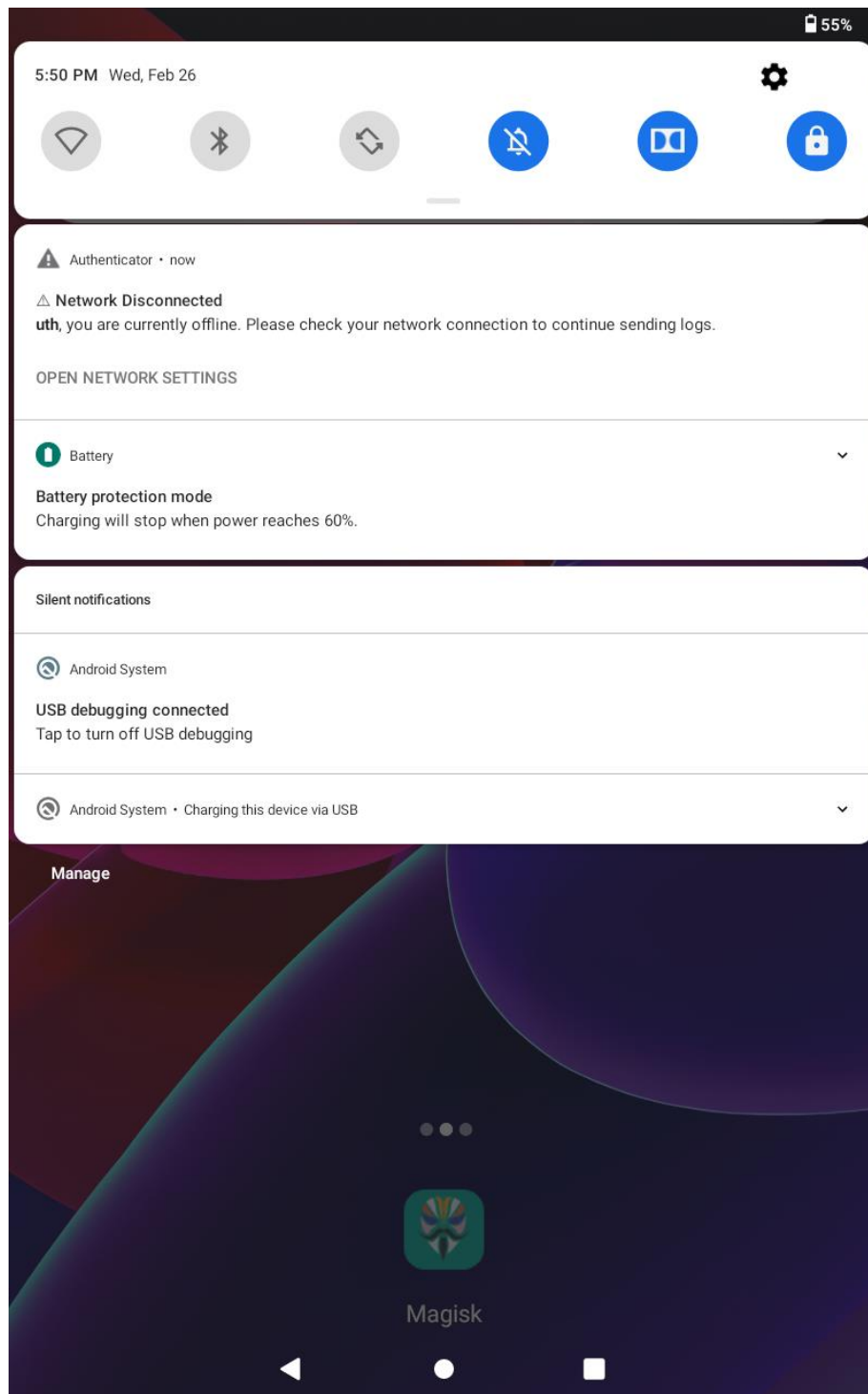


Figure 17. Android Log Capture: Managing Device Offline Scenarios

- In another scenario, the user's authentication token, which is used to maintain the user's login session in the application, may expire. When this happens, the Android log capture process stops, and the application prompts the user to re-login, as illustrated in Figure 18.

|                |                                                             |                |    |          |     |         |          |
|----------------|-------------------------------------------------------------|----------------|----|----------|-----|---------|----------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          |     | Page:   | 42 of 58 |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0 | Status: | Final    |

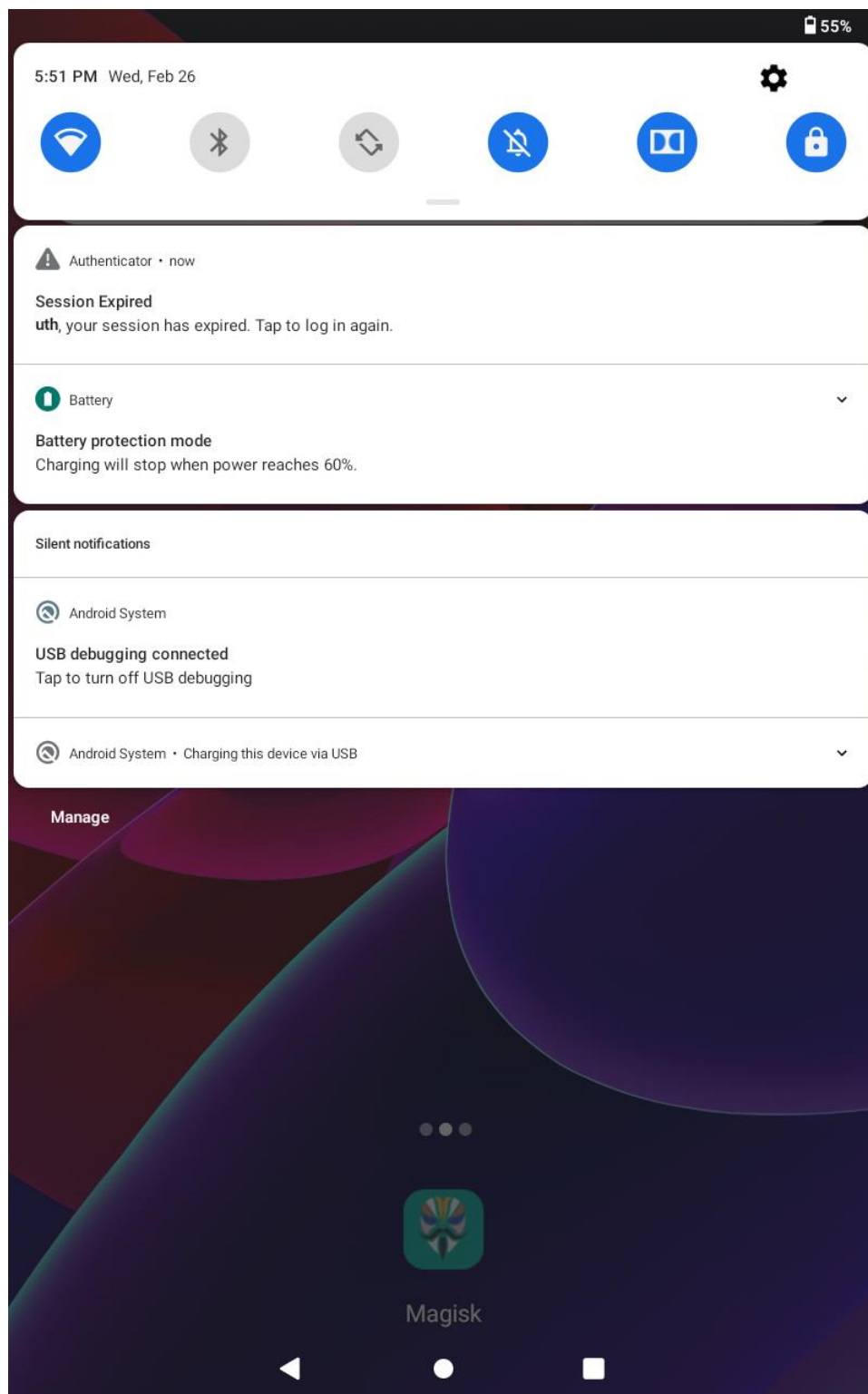


Figure 18. Android Log Capture: Managing Authentication Token Expiration Scenarios

- **USER LOGOUT:** User attempts to log out.
  - When user engages the 'LOGOUT' button, the 'Authenticator' sends a request to the server's designated endpoint, transmitting the username and the corresponding authentication token. The server verifies the token and, upon successful verification, removes the token from in-memory storage, effectively revoking the user's session. The server then confirms the successful logout, and

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 43 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

the 'Authenticator' clears the token from its memory as well. The logout process pipeline is depicted in the Figure 19.

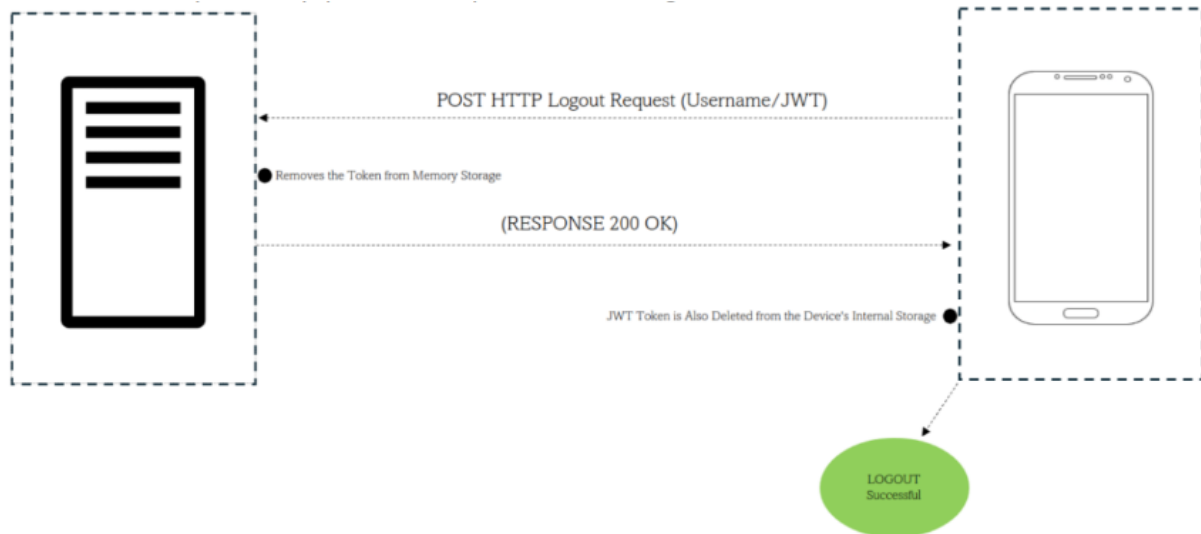


Figure 19. User Logout Process Pipeline

The system logs captured by the foreground service, transmitted to the server, and currently under examination, are shown in Figure 20.

| ANDROID LOGS                    |                     |               |                  |
|---------------------------------|---------------------|---------------|------------------|
| TAG                             | INTERFACE           | GENERATED BY  | USE CASE         |
| WifiDisplayController           | NETWORK             | ANDROID OS    | MACHINE LEARNING |
| NetworkinfoNotification         | NETWORK             | ANDROID OS    | MACHINE LEARNING |
| WifiService                     | NETWORK             | ANDROID OS    | MACHINE LEARNING |
| WifiConfigManager               | NETWORK             | ANDROID OS    | MACHINE LEARNING |
| netd                            | NETWORK             | ANDROID OS    | MACHINE LEARNING |
| WifiScanningService             | NETWORK             | ANDROID OS    | MACHINE LEARNING |
| WifiConnectivityManager         | NETWORK             | ANDROID OS    | MACHINE LEARNING |
| ConnectivityPacketTracker.wlan0 | NETWORK             | ANDROID OS    | MACHINE LEARNING |
| WifiNetworkSelector             | NETWORK             | ANDROID OS    | MACHINE LEARNING |
| WifiNetworkSelectorN            | NETWORK             | AUTHENTICATOR | LOCALIZATION     |
| DhcpClient                      | NETWORK             | ANDROID OS    | MACHINE LEARNING |
| MtkConnectivityService          | NETWORK             | ANDROID OS    | MACHINE LEARNING |
| WifiClientModeImpl              | NETWORK             | ANDROID OS    | MACHINE LEARNING |
| SettingsState                   | NETWORK - BLUETOOTH | ANDROID OS    | MACHINE LEARNING |
| AdapterState                    | BLUETOOTH           | ANDROID OS    | MACHINE LEARNING |
| CachedBluetoothDevice           | BLUETOOTH           | ANDROID OS    | MACHINE LEARNING |
| BluetoothDatabase               | BLUETOOTH           | ANDROID OS    | MACHINE LEARNING |
| BluetoothAdapterService         | BLUETOOTH           | ANDROID OS    | MACHINE LEARNING |
| BLeScannerN                     | BLUETOOTH           | AUTHENTICATOR | LOCALIZATION     |
| StatusBar                       | BATTERY             | ANDROID OS    | MACHINE LEARNING |
| LPPeService                     | BATTERY             | ANDROID OS    | MACHINE LEARNING |

Figure 20. Android System Logs

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 44 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

As observed, each log provides crucial information regarding the interfaces under examination, including Networking, Battery, and Bluetooth. Concerning Location, the 'Authenticator' application generates two custom logs as we mentioned earlier, as indicated in Figure 20. These logs specifically contain information about the RSSI signals received from nearby Wi-Fi and Bluetooth Low Energy access points and devices, which are used to perform the Localization Mechanism based on a predefined heatmap. Currently, the localization system is production-ready, with two algorithms available. The algorithm that yields the best results and is most suitable for the specific facility will be selected.

All other logs undergo a process called the 'Data Preprocessing Phase' before being fed into the Risk Assessment Service, which is the Machine Learning model. A simple and comprehensive illustration of the data preprocessing mechanism is provided in Figure 21, which demonstrates how raw Android logs are converted into numerical IDs. Then, we form sequences of these numerical IDs and feed them into the ML model, primarily built on a pre-trained **BERT** (Bidirectional Encoder Representations from Transformers) model with additional custom layers. The idea is that by training the model on sequences representing normal device operations, it will be able to detect abnormalities during evaluation, such as the appearance of unexpected logs. The model is still under development to further improve its performance.

|                       |                                                                |                       |    |                 |              |                      |
|-----------------------|----------------------------------------------------------------|-----------------------|----|-----------------|--------------|----------------------|
| <b>Document name:</b> | D4.2 Distributed Trust Management Framework -<br>Final version |                       |    |                 | <b>Page:</b> | 45 of 58             |
| <b>Reference:</b>     | D4.2                                                           | <b>Dissemination:</b> | PU | <b>Version:</b> | 1.0          | <b>Status:</b> Final |

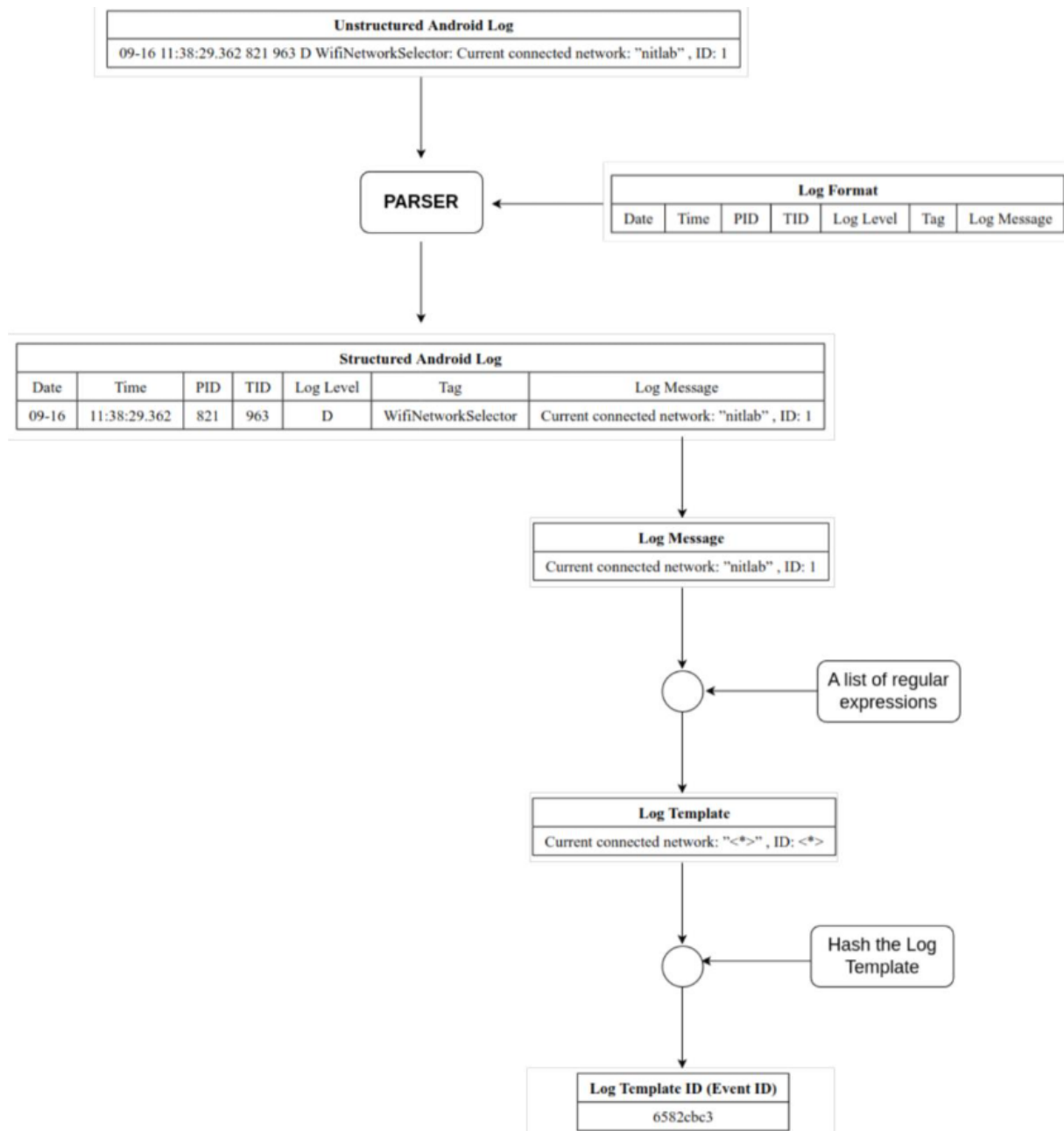


Figure 21. Data Pre-Processing Phase

### 3.4.1.3 Software artifacts

In this section the analysis of server's endpoints and their functionalities takes place, as we can see in the Table 8.

Table 8. Server's Endpoints

| ENDPOINT | TYPE | DESCRIPTION                        |
|----------|------|------------------------------------|
| /users   | GET  | Fetch all users from the database. |

|                |                                                             |                |    |          |     |         |          |
|----------------|-------------------------------------------------------------|----------------|----|----------|-----|---------|----------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          |     | Page:   | 46 of 58 |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0 | Status: | Final    |

| ENDPOINT                                     | TYPE        | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                |
|----------------------------------------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /users/registration/request                  | <b>POST</b> | Handle user registration requests by accepting the provided credentials (username and password). The server responds with a registration challenge and requests the user's public key and a registration signature, which is the challenge signed using the user's private key.                                                                            |
| /users/registration/request/upload/signature | <b>POST</b> | Upload the registration signature, along with the username and the public key. The server verifies the registration signature using the provided public key. If the signature is valid, the registration request is accepted, and the registration is successful. Otherwise, the registration fails.                                                       |
| /users/login/request                         | <b>POST</b> | Handle user login requests by verifying the provided username and password credentials. If the credentials are valid, a login challenge is sent to the user ('Authenticator' app) for signing with the private key, which is generated by the 'Authenticator' app during the registration request.                                                         |
| /users/login/request/upload/signature        | <b>POST</b> | Upload the login signature, along with the username. The server retrieves the user's public key from the database to verify the signature. If the signature is valid, the user is authenticated and logged in. The server then generates an authentication token to maintain the session, which is sent back to the user. Otherwise, authentication fails. |
| /users/android/log/capture                   | <b>POST</b> | Accepts system logs from users' Android devices for processing or analysis. The request includes an authentication token to verify if the user remains logged in or needs to log in again, and also the username of the user.                                                                                                                              |
| /users/authTokenValidation                   | <b>GET</b>  | Validates the provided authentication token to confirm its validity and ensure the user's session is active.                                                                                                                                                                                                                                               |

| ENDPOINT      | TYPE        | DESCRIPTION                                                                                                      |
|---------------|-------------|------------------------------------------------------------------------------------------------------------------|
| /users/logout | <b>POST</b> | Handles user logout requests by invalidating the authentication token and effectively ending the user's session. |

### 3.4.2 Demonstration description

Changes have been made to the user interface (UI) of the ‘*Authenticator*’ application for better interaction. Upon the first time entering the application, the user is asked for permission to access the device’s location, as Figure 22 illustrates. This is necessary for performing periodic scans of nearby access points and Bluetooth Low Energy devices to receive their RSSI signals and carry out the localization mechanism, estimating the user’s device location at any time. Therefore, the application requests superuser permissions from the user, as depicted in Figure 23, to access the system logs. This allows the application to capture and transmit the logs to the server for detecting deviations in the device's behaviour using the Risk Assessment Engine.

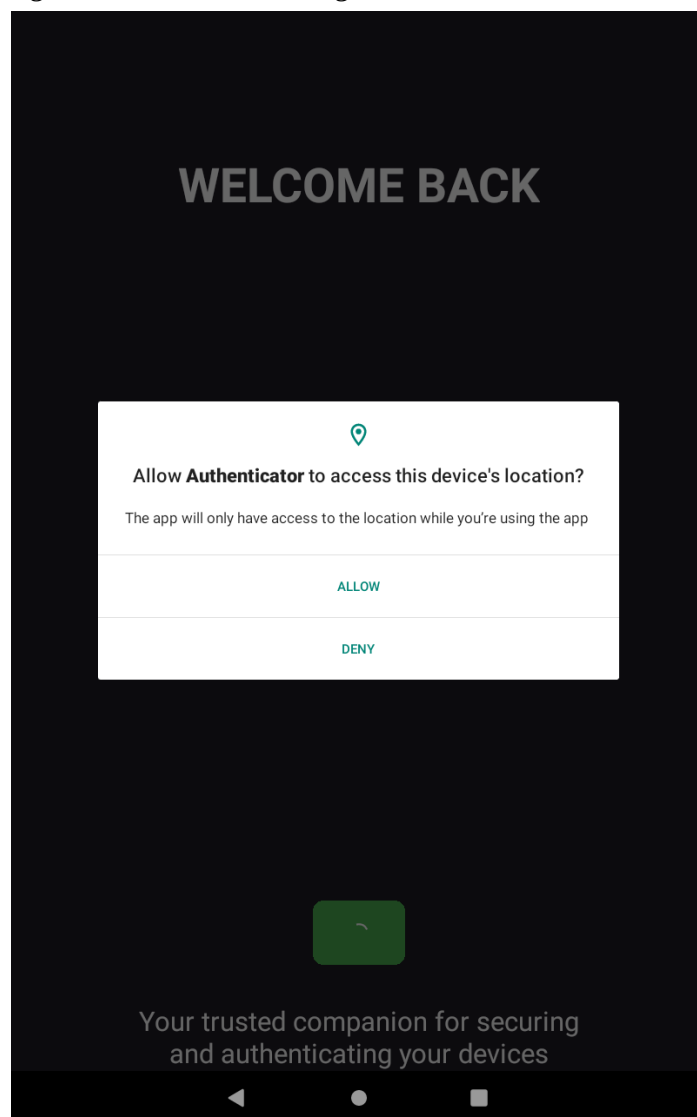


Figure 22. ‘Authenticator’ Requesting Device Location Access

|                       |                                                             |                       |    |                 |              |                      |
|-----------------------|-------------------------------------------------------------|-----------------------|----|-----------------|--------------|----------------------|
| <b>Document name:</b> | D4.2 Distributed Trust Management Framework - Final version |                       |    |                 | <b>Page:</b> | 48 of 58             |
| <b>Reference:</b>     | D4.2                                                        | <b>Dissemination:</b> | PU | <b>Version:</b> | 1.0          | <b>Status:</b> Final |

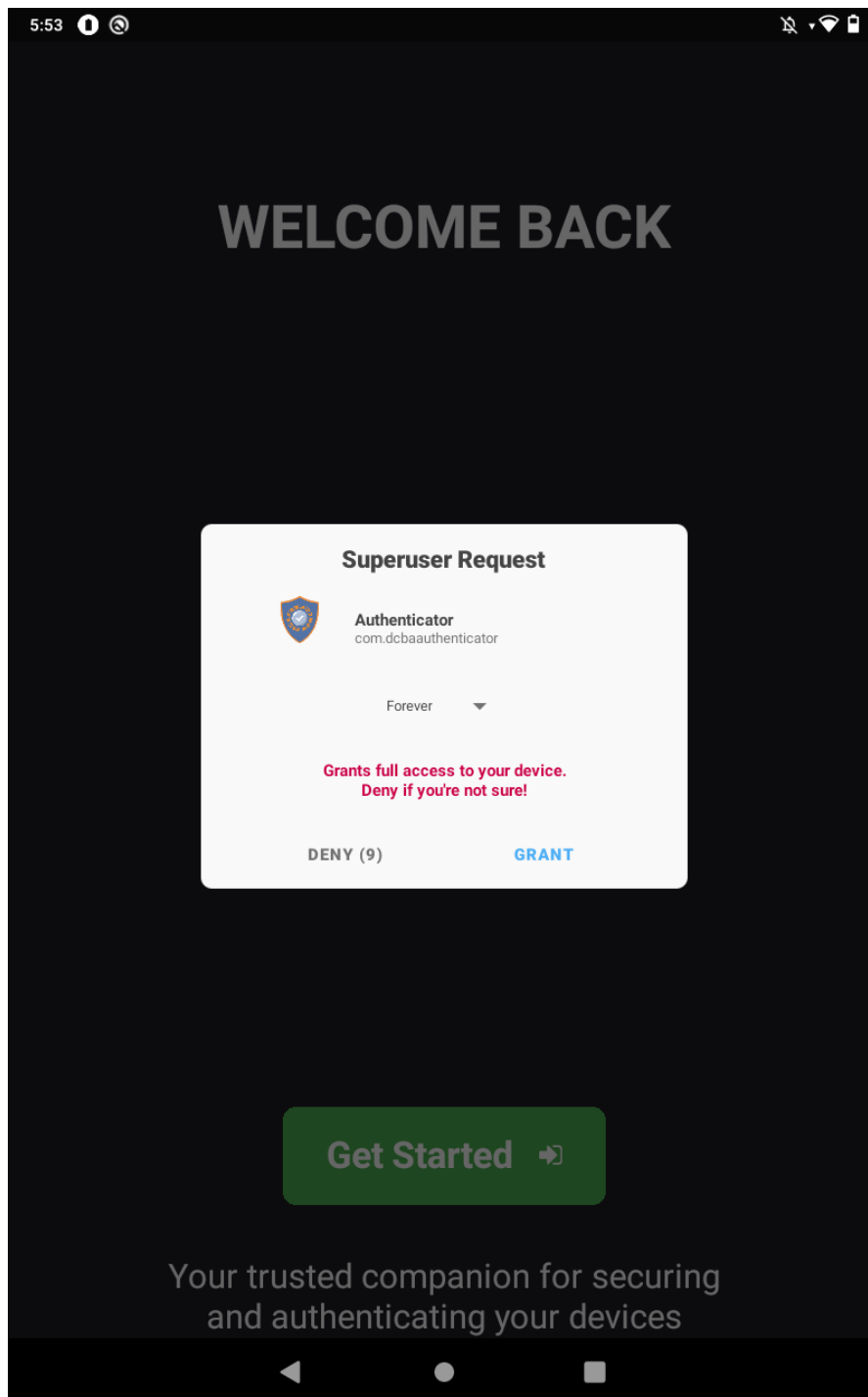
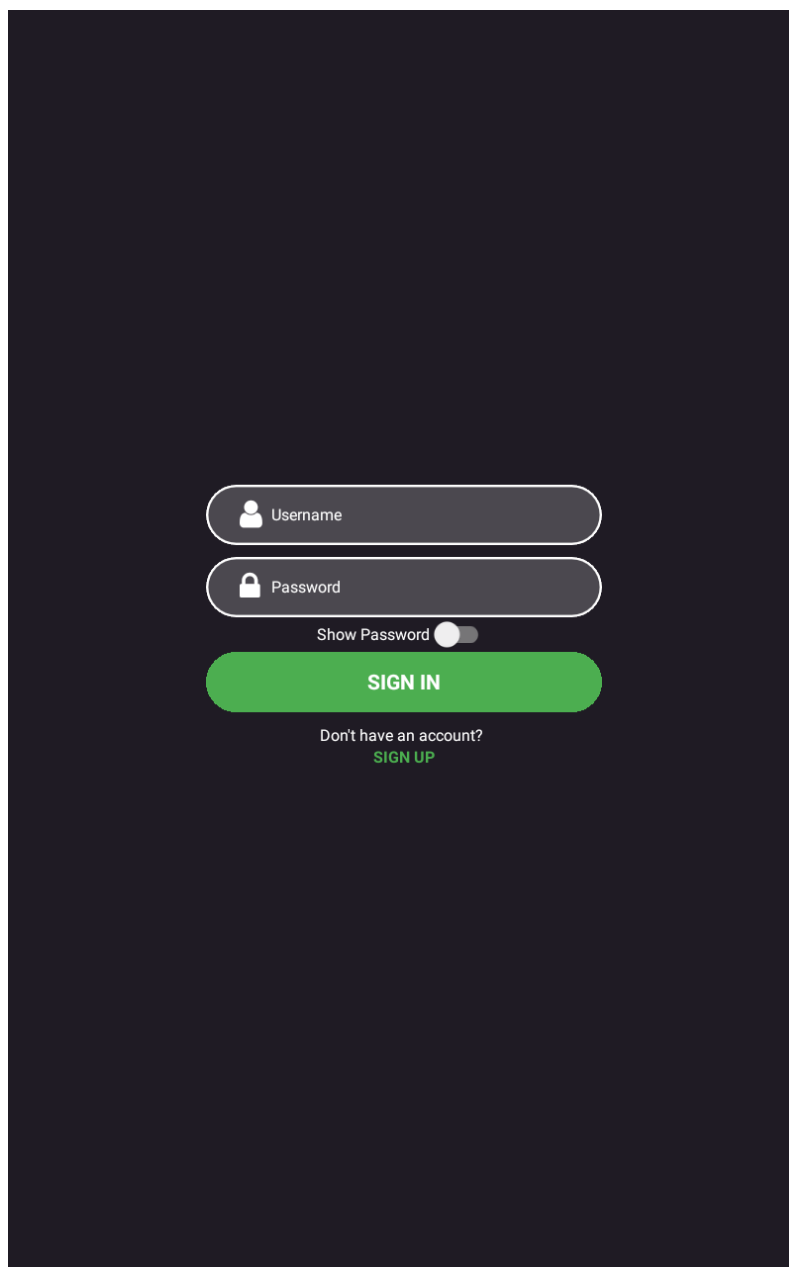


Figure 23. 'Authenticator' Requesting Superuser Rights

These two steps are only required once, when the user enters the application for the first time, serving as the initial setup. If the user denies these requests, the 'Authenticator' app prompts them with detailed instructions on how to re-enable the permissions. Without these essential steps, the Localization and Risk Assessment Engine cannot be performed.

Below are the main UI images of the application:

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 49 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |



The login page features a dark blue background with a central white login form. The form includes a 'Username' field with a person icon, a 'Password' field with a lock icon, and a 'Show Password' toggle switch. Below the password field is a prominent green 'SIGN IN' button. At the bottom of the form, there is a link for users who do not have an account, labeled 'SIGN UP'.

Figure 24. Login Page

|                |                                                             |                |    |          |       |               |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|---------------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 50 of 58      |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status: Final |

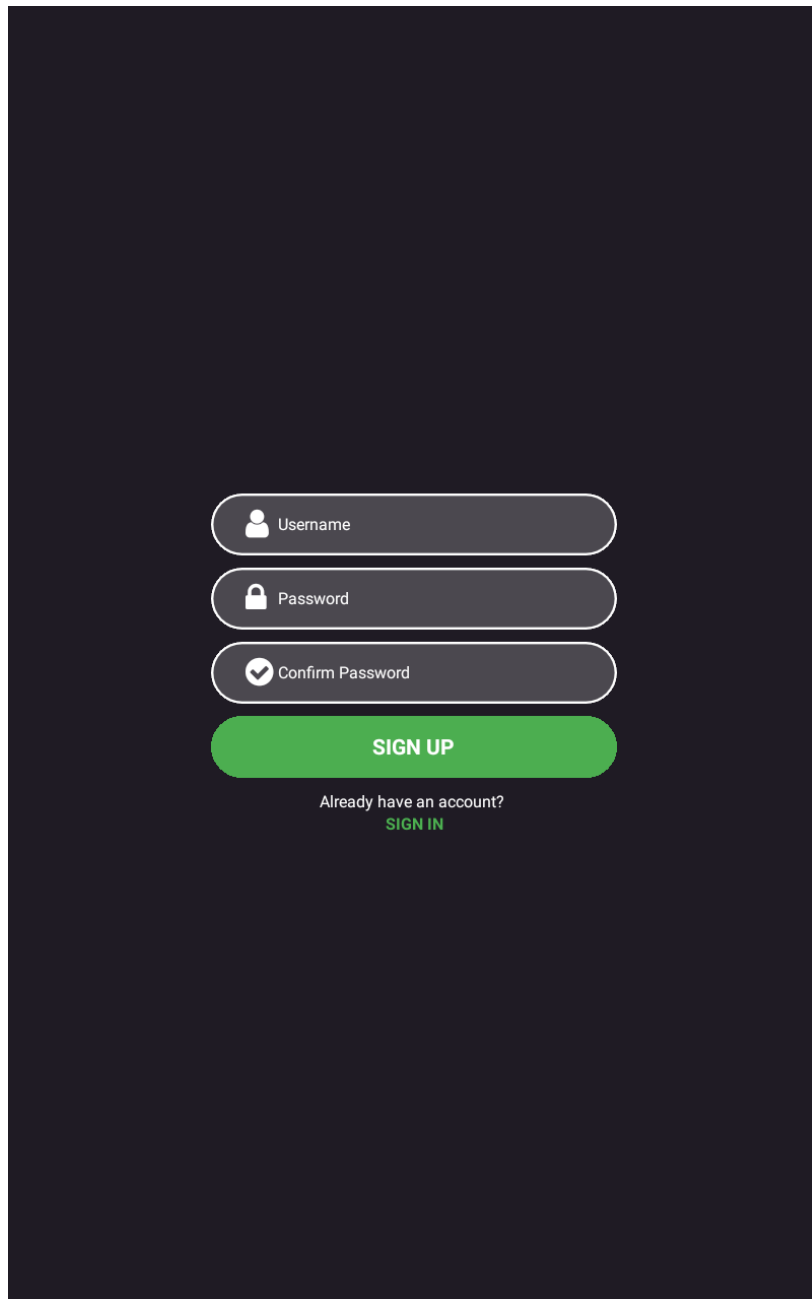

 The image shows a registration page with a dark blue background. In the center, there are three rounded rectangular input fields stacked vertically. The first field has a person icon and the text 'Username'. The second field has a lock icon and the text 'Password'. The third field has a checkmark icon and the text 'Confirm Password'. Below these fields is a large green rounded rectangular button with the text 'SIGN UP' in white. Underneath the button, the text 'Already have an account?' is displayed in a small, light blue font, followed by the text 'SIGN IN' in a green font.

Figure 25. Registration Page

The Figure 24 depicts the Login Page, while the Figure 25 shows the Registration. Figure 16 displays the username of the user who is currently authenticated and logged in. Also, the key functionality of the application is to manage Android log capture even when the device is offline, as shown in Figure 17. Additionally, a new feature has been added to check for expired authentication tokens during log capture. If the token expires during the log capture, the log capture stops, and the user is prompted to log in again, as illustrated in Figure 18.

#### 3.4.2.1 Docker/API description

As previously mentioned, the deployment consists of two primary components:

- Frontend Application: The Android application, ‘*Authenticator*’, designed to run **natively** on Android devices with a target Android version of 10.0 or higher. The application can be installed using its *.apk*

|                |                                                             |                |    |          |     |         |          |
|----------------|-------------------------------------------------------------|----------------|----|----------|-----|---------|----------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          |     | Page:   | 51 of 58 |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0 | Status: | Final    |

file, which is the standard installation file format for Android Operating Systems (OS). This file includes all the necessary components required for installing the application on a device.

- **Backend API:** Comprising the Server, and two databases. The server will now be deployed in the cloud, utilizing Docker technology for containerization. This ensures that the server operates within a containerized, isolated environment, allowing it to run reliably and without issues on any machine. Also, the server architecture incorporates two databases:
  - **MongoDB:** Used to store user data and their credentials.
  - **InfluxDB:** A time-series database designed to store and manage users' real-time logs, which are analyzed by the Risk Assessment Engine.

Both databases, along with the server, are containerized using Docker technology. The entire setup is managed through a *docker-compose.yml* file, which creates three containers (Server, MongoDB, and InfluxDB) within a fully isolated environment, forming a comprehensive API service, as Figure 26 illustrates.

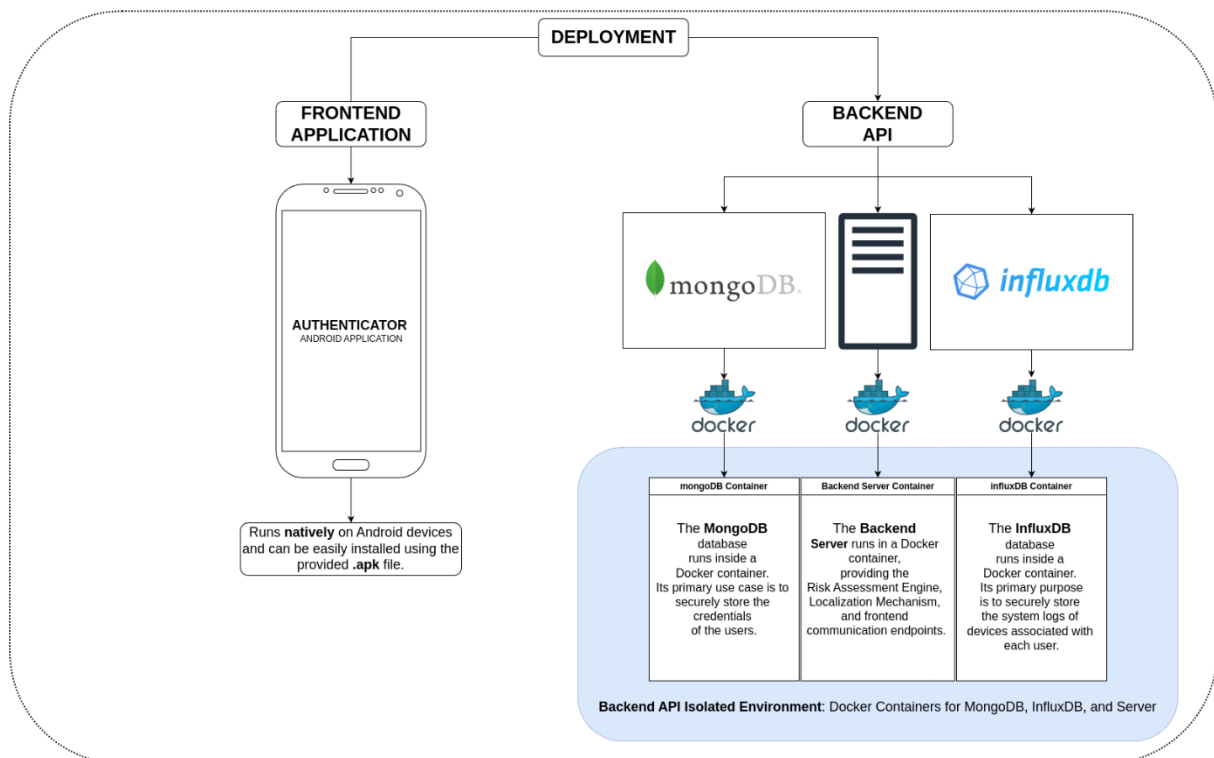


Figure 26. System Deployment Architecture

### 3.4.3 Support for pilots

The Device Continuous Behavioural Authentication support the following pilot:

- **Smart Manufacturing - Case 2 (RIA STONE).** The DCBA mechanism will assess the behavioural performance of a worker's device. The device is used by the workers to access specific regions in the factory. Not all the workers are allowed to enter or have physical access to all the factory's places. For security reasons, worker according to their roles and responsibilities are restricted to specific factory areas. Therefore, using the DCBA mechanism will allow the continuous authentication of the worker's device when the worker access permitted areas of the factory. If the device is placed or found in place where the worker is not allowed or permitted to have physical access, then an alert will be created to notify the responsible managers and the session will be terminated.

|                |                                                             |                |    |          |       |          |       |
|----------------|-------------------------------------------------------------|----------------|----|----------|-------|----------|-------|
| Document name: | D4.2 Distributed Trust Management Framework - Final version |                |    |          | Page: | 52 of 58 |       |
| Reference:     | D4.2                                                        | Dissemination: | PU | Version: | 1.0   | Status:  | Final |

### 3.4.4 Final version of this component

For better clarity and comprehensiveness, we summarize the key changes and updates made to the Devices Continuous Behavioural Authentication (DCBA) Component between the D4.1 and D4.2 deliverables in Table 9.

Table 9. Summary of Updates and Improvements in the DCBA Component

| TOPIC/ISSUE                                          | D4.1 (Previous State)                                                                                                                                                                                                                                                       | D4.2 (Current State)                                                                                                                                                                                                                                                                                        | STATUS/PROGRESS          |
|------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| ANDROID LOG CAPTURE                                  | To capture Android logs from the devices, an initial setup was required to establish a connection between the device and the server, ensuring they are on the <u>same network</u> .                                                                                         | We addressed this issue by rooting the Android devices, which enabled the collection of Android logs from any device, regardless of its network connection or the need for initial setup.                                                                                                                   | <b>DONE</b>              |
| LOCATION-RELATED LOGS FOR THE LOCALIZATION MECHANISM | Since the Android OS does not provide location information, previous implementations of our application relied on GPS to obtain the device's exact coordinates. However, this approach resulted in significant battery consumption, making it inefficient and unacceptable. | We resolved this issue by implementing periodic scans within our Android application to detect nearby Wi-Fi and Bluetooth access points and devices and their RSSI signals. This approach enables the localization mechanism, allowing the precise estimation of the device's location at any given moment. | <b>DONE</b>              |
| LOCALIZATION MECHANISM                               | The implementation had not commenced, as the focus was primarily on working with GPS coordinates.                                                                                                                                                                           | Two algorithms have been implemented. One estimates the device's area or room based on RSSI signals, while the other calculates the device's coordinates based on these signals. Both algorithms are functional, and the most suitable one will be selected for use.                                        | <b>DONE</b>              |
| RISK ASSESSMENT ENGINE (MACHINE LEARNING MODEL)      | The implementation of the machine learning model for detecting abnormalities in device behavior had not yet begun.                                                                                                                                                          | Currently, the main architecture has been established, which includes a machine learning model based on the pre-trained BERT model. Development is ongoing to enhance its accuracy.                                                                                                                         | <b>UNDER DEVELOPMENT</b> |

| TOPIC/ISSUE              | D4.1 (Previous State)                                                                                                                                                                                              | D4.2 (Current State)                                                                                                                                                                                                                          | STATUS/PROGRESS |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| LOGIN TIME               | No checks were implemented regarding login duration.                                                                                                                                                               | An authentication token has been added to manage the duration of user sessions.                                                                                                                                                               | <b>DONE</b>     |
| DEVICE OFFLINE SCENARIOS | In the previous implementation, the android application ‘Authenticator’ did not detect offline or network disconnections and continued attempting to transmit log data to the server while the device was offline. | Now, when the device is offline, logs are stored in the device’s text file created during the user’s successful registration. These logs are appended to this file and queued for transmission once the device reconnects to the network.     | <b>DONE</b>     |
| DATABASES                | In the previous implementation, file storage was used to save both logs and user credentials locally on the machine host.                                                                                          | Currently, we utilize MongoDB to store user credentials, and InfluxDB, a time-series database, to securely store the logs.                                                                                                                    | <b>DONE</b>     |
| DEPLOYMENT               | The Backend API was deployed locally on the host machine.                                                                                                                                                          | The Backend API now is ready to be deployed in the cloud, as shown in <a href="#">Figure 26</a> , utilizing Docker technology. These services run inside containers within an isolated environment, ensuring consistency across all machines. | <b>DONE</b>     |

### 3.5 Side-channel attack hardening

We continued our work on developing countermeasures against side-channel attacks, i.e., preventing attackers from extracting secret information through physical quantities, such as electromagnetic emissions from a chip. Although we focused on hiding countermeasures in the first part of the project, we continued with a stronger focus on masking countermeasures, providing complementary work.

As part of this effort, we implemented a masked AES for the smart hospitality pilot, ensuring its seamless integration and usability for project partners.

For this purpose, we studied how to provide a faster masked AES by combining several techniques, such as Boolean masking, multiplicative masking, bitslicing, and other specific optimizations, resulting in a faster implementation.

Additionally, we advanced our research on combining masking and hiding techniques. While this work is more theory-grounded, it is important to enable the safe combination of such countermeasures.

|                       |                                                             |                       |    |                 |              |                      |
|-----------------------|-------------------------------------------------------------|-----------------------|----|-----------------|--------------|----------------------|
| <b>Document name:</b> | D4.2 Distributed Trust Management Framework - Final version |                       |    |                 | <b>Page:</b> | 54 of 58             |
| <b>Reference:</b>     | D4.2                                                        | <b>Dissemination:</b> | PU | <b>Version:</b> | 1.0          | <b>Status:</b> Final |

### 3.5.1 Short Component description

Previous version focused on the combination of 2 hiding countermeasures: code polymorphism and loop shuffling. It showed promising performance and attack resistance.

In practice in products, hiding countermeasures are combined with masking countermeasures. To provide a complementary view to our first work, we studied how to mask AES efficiently, as well as how to combine masking with code polymorphism without threatening security.

In order to mask the AES efficiently, we adapted a technique combining Boolean with multiplicative masking from Genelle et al. [10] and accelerated it with recent optimization techniques such as bitslicing. In short, Genelle et al.' technique involves the computation of a secure function on the 8 bytes of the AES' state, that we accelerate the parallelism enabled by 32-bits Boolean instructions. We add other low-level optimizations to accelerate the most used part of the code. We compare our approach with 2 other implementations based on different masking approaches:

| Reference              | Implementation details                  | # of SecAnd or SecMul per round | Masking and t-test order | Max. abs. t-value | Size (kB) | Execution time (cycles) per exec. | Execution time (cycles) per plaintext |
|------------------------|-----------------------------------------|---------------------------------|--------------------------|-------------------|-----------|-----------------------------------|---------------------------------------|
| Cardoso dos Santos [5] | RP-based.                               | 32                              | 1st                      | 4.02              | 7.9       | 35645                             | 35645                                 |
|                        |                                         |                                 | 2nd                      | 5.17              | 12.7      | 68366                             | 68366                                 |
| Adomnicai et al. [3]   | Fixsliced. 256 bits ciphered at a time. | 32 (for 2 plaintexts)           | 1st                      | 4.01              | 7.3       | 34926                             | 17463                                 |
|                        |                                         |                                 | 2nd                      | 5.22              | 12.5      | 75209                             | 37604                                 |
| Our work               | Genelle et al. with partial bitslicing. | 4                               | 1st                      | 3.93              | 8.2       | 16922                             | 16922                                 |
|                        |                                         |                                 | 2nd                      | 5.08              | 12.5      | 31985                             | 31985                                 |

Figure 27. Comparison of Implementations

We consider 2 different masking and t-test order, i.e. 2 different security levels. We can notice how, for a given masking order, the results in security (max abs t-value) and size are similar across implementations. This lets us compare the execution time. Our implementation's execution time is largely faster than both other implementations. Yet, the implementation based on fixslicing actually ciphers 2 plaintexts at a time. When normalizing the execution time per plaintext, our implementation show similar execution time than the fixsliced one at first order, and shows a 1.18x improvement at second order. We note as well that our approach does not require to cipher several plaintexts simultaneously, and as such is more general. This work was published in IEEE Embedded Systems Letters (<https://ieeexplore.ieee.org/abstract/document/10798512/>) and presented at SeHAS 2025.

In addition, we are working on enabling the combination of masking countermeasure and code polymorphism countermeasure. Masking countermeasure is very sensitive to the order of operation: a functionally-correct code transformation may break the countermeasure. As such, the code polymorphism countermeasure, that is based on code transformations, cannot be used safely with masking countermeasures, despite the potential gain in security. We started a study to derive sufficient conditions under which a code transformation cannot break the security of a masked code. This study will allow us to adapt the code polymorphism countermeasure so that it can be safely combined with masking.

### 3.5.2 Demonstration description

We integrated our fast masked AES implementation (at 2nd order) into the OpenSSL library. By utilizing this 'hardened' OpenSSL library, each call of the AES\_encrypt() function executes the masked implementation instead of the standard encryption algorithm. This ensures that for every encryption operation, leakage—i.e., information that could be obtained through side-channel attacks—is significantly reduced.

This feature will be deployed in the Smart Hospitality use case, where sensors deployed in hotel rooms communicate with a cloud broker via a gateway. It will secure the communication between the gateway (Raspberry Pi) and the cloud broker. The communication relies on the Python MQTT client, which

|                       |                                                             |                       |    |                 |              |                      |
|-----------------------|-------------------------------------------------------------|-----------------------|----|-----------------|--------------|----------------------|
| <b>Document name:</b> | D4.2 Distributed Trust Management Framework - Final version |                       |    |                 | <b>Page:</b> | 55 of 58             |
| <b>Reference:</b>     | D4.2                                                        | <b>Dissemination:</b> | PU | <b>Version:</b> | 1.0          | <b>Status:</b> Final |

depends on the OpenSSL library. By replacing the standard OpenSSL library on the Raspberry Pi with our ‘hardened’ version, the communication can be effectively secured.

We tested the component and confirmed that it provides the same functionality and output as the original implementation when encrypting data.

From a security perspective, we are currently conducting side-channel measurements on the same Raspberry Pi model used in the pilot and plan to perform t-tests for leakage assessment.

### 3.5.3 Support for pilots

The component is used in the Smart Hospitality pilot. It hardens the AES implementation used in a raspberry PI gateway that interfaces with sensors and with the cloud broker. Its use is transparent to the end-user: the tourists coming to the hotel do not have any change in their experience. As such, the component helps enable the hotel features without compromising on security.

For this usage, the component has been embedded in a library in order to enable an easy integration for AnySolution: a replacement with our library directly enables deploying the component.

|                       |                                                             |                       |    |                 |              |                      |
|-----------------------|-------------------------------------------------------------|-----------------------|----|-----------------|--------------|----------------------|
| <b>Document name:</b> | D4.2 Distributed Trust Management Framework - Final version |                       |    |                 | <b>Page:</b> | 56 of 58             |
| <b>Reference:</b>     | D4.2                                                        | <b>Dissemination:</b> | PU | <b>Version:</b> | 1.0          | <b>Status:</b> Final |

## 4 Conclusions

In conclusion, the TANGO Distributed Trust Management Framework has successfully addressed the objectives set out in tasks T4.1, T4.2, T4.3, T4.4, and T4.5. Task T4.1 focused on the development of the Self-Sovereign Identity (SSI) component, which allows for authentication using verified identity credentials. This component has been integrated with the FIWARE Data Space Connector and supports ePassport credentials, enhancing the security and trustworthiness of the TANGO platform. Future work will focus on further refining the SSI component and ensuring its seamless integration with other parts of the TANGO ecosystem.

Task T4.2 aimed to develop a seamless onboarding process for users and devices. The ePassport app, which enables remote identity verification based on existing identity documents, has been successfully integrated with the SSI Wallet and Issuer, providing a smooth and secure onboarding experience. The ePassport app will continue to be tested and improved based on feedback from various stakeholders. The integration with the SSI component will be further enhanced to ensure a smooth and secure onboarding experience.

Task T4.3 focused on continuous user authentication through behavioural patterns. The User Continuous Behavioural Authentication component leverages biometrics and behavioural data to continuously assess the legitimacy of the user, providing a risk score that can be used by other components of the TANGO platform to enhance security. The User Continuous Behavioural Authentication component will be integrated across various application domains within the TANGO project. Feedback from stakeholders will be incorporated to provide further enhancements to the component.

Task T4.4 aimed to provide continuous device authentication based on various device behaviours. The Device Continuous Behavioural Authentication component captures and analyses system logs to detect deviations from normal device behaviour, ensuring ongoing monitoring and security throughout the user session. The Device Continuous Behavioural Authentication component will be further developed to improve its performance and accuracy. The localization mechanism will be refined, and the Risk Assessment Engine will be enhanced to detect abnormalities in device behaviour more effectively.

Finally, Task T4.5 focused on developing countermeasures against side-channel attacks. The implementation of a masked AES algorithm and the combination of masking and hiding techniques have significantly improved the security of the TANGO platform against such attacks. The focus will be on combining masking and hiding techniques to provide a more robust defense against side-channel attacks. The implementation of the masked AES algorithm will be further optimized, and additional research will be conducted to improve the security of the TANGO platform.

Overall, the components developed in these tasks have been successfully integrated into the TANGO platform, providing a robust and secure distributed trust management framework. Future work will focus on further refining these components and ensuring their seamless integration with other parts of the TANGO ecosystem.

|                       |                                                                |                       |    |                 |              |                      |
|-----------------------|----------------------------------------------------------------|-----------------------|----|-----------------|--------------|----------------------|
| <b>Document name:</b> | D4.2 Distributed Trust Management Framework -<br>Final version |                       |    |                 | <b>Page:</b> | 57 of 58             |
| <b>Reference:</b>     | D4.2                                                           | <b>Dissemination:</b> | PU | <b>Version:</b> | 1.0          | <b>Status:</b> Final |

## 5 References

1. <https://www.w3.org/TR/vc-data-model-2.0>
2. <https://www.w3.org/TR/did-core>
3. <https://hub.ebsi.eu/conformance>
4. [https://openid.net/specs/openid-4-verifiable-credential-issuance-1\\_0.html](https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html)
5. [https://openid.net/specs/openid-4-verifiable-presentations-1\\_0.html](https://openid.net/specs/openid-4-verifiable-presentations-1_0.html)
6. <https://w3c-ccg.github.io/did-method-key/>
7. <https://hub.ebsi.eu/vc-framework/did/natural-person>
8. <https://hub.ebsi.eu/vc-framework>
9. [https://data-spaces-business-alliance.eu/wp-content/uploads/dlm\\_uploads/Data-Spaces-Business-Alliance-Technical-Convergence-V2.pdf](https://data-spaces-business-alliance.eu/wp-content/uploads/dlm_uploads/Data-Spaces-Business-Alliance-Technical-Convergence-V2.pdf)
10. <https://github.com/FIWARE/VCVerifier/tree/main>
11. Genelle, L., Prouff, E., Quisquater, M. (2010). Secure Multiplicative Masking of Power Functions. In: Zhou, J., Yung, M. (eds) Applied Cryptography and Network Security. ACNS 2010. Lecture Notes in Computer Science, vol 6123. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-13708-2\\_13](https://doi.org/10.1007/978-3-642-13708-2_13)

|                       |                                                             |                       |    |                 |              |                      |
|-----------------------|-------------------------------------------------------------|-----------------------|----|-----------------|--------------|----------------------|
| <b>Document name:</b> | D4.2 Distributed Trust Management Framework - Final version |                       |    |                 | <b>Page:</b> | 58 of 58             |
| <b>Reference:</b>     | D4.2                                                        | <b>Dissemination:</b> | PU | <b>Version:</b> | 1.0          | <b>Status:</b> Final |