# Software Requirements Specification

for

# Gaia-X Federation Services Authentication/Authorization IDM.AA

## Published by

eco – Association of the Internet Industry (eco – Verband der Internetwirtschaft e.V.)
Lichtstrasse 43h
50825 Cologne, Germany

## Copyright

# Table of Contents

# List of Figures

# List of Tables

# 1. Introduction

To get general information regarding Gaia-X and the Gaia-X Federation Services please refer to [TAD] and [PRD].

## 1.1. Document Purpose

The purpose of the document is to specify the requirements of the Identity Management and Trust Subcomponent “Authentication/Authorization” with the intention of a European wide public tender for implementing this software. Main audience for this document is attendees of the public tender, which are able to supply an open-source software solution for the area of open id connect, oAuth2 and other identity management solutions with the purpose to extend existing identity management solutions with SSI functionality.

## 1.2. Product Scope

The scope of the document is to describe the components for “Authorization & Authentication” which shall deliver core functionalities for authorization, access management and authentication as well as services around it, to Gaia-X Participants with the purpose to join the trustful environment of the ecosystem. This document does not describe how to replace an already established IAM System within a Gaia-X participant environment or how to operate it within the Gaia-X participant environment. There is also no claim to replace any existing IAM or to provide a solution for managing users in the enterprise environment of the Gaia-X participant.

In contrast, the product delivers necessary components to integrate with an existing IAM environment, based on the standard protocols, which would enable such IAM to operate in Gaia-X ecosystems.

## 1.3. Definitions, Acronyms and Abbreviations

The IDM and Trust Architecture Overview Document [IDM.AO] MUST be considered and applied as the core technical concept that includes also the Terminology and Glossary.

All requirements from other documents are referenced by [IDM.<document-id>.XXXXX] as defined in the chapter “Methodology” in the document [IDM.AO].

## 1.4. References

| | |
|---|---|
| [BCP OAuth2] | **T. Lodderstedt, J. Bradley, A. Labunets, D. Fett (2020), OAuth 2.0 Security Best Current Practice draft-ietf-oauth-security-topics-16** |

| | |
|---|---|
| | https://tools.ietf.org/html/draft-ietf-oauth-security-topics-16 (Status: 03-17-2021) |
| [BDD] | **Specflow (n.D.), Getting Started with Behavior Driven Development** |
| | https://specflow.org/bdd/ (Status 03-18-2021) |
| [CryptoLen] | **Damien Giry, Prof. Jean-Jacques Quisquater (2020), Cryptographic Key Length Recommendation** |
| | https://www.keylength.com/en (Status 03-18-2021) |
| [DID SIOP] | **DIF Working Group (n.d.), Self-Issued OpenID Connect Provider DID Profile v0.1** |
| | https://identity.foundation/did-siop/ (Status: 02-18-2021) |
| [DIDComm.Msg] | **Daniel Hardman (n.D.), DIDComm Messaging** |
| | https://identity.foundation/didcomm-messaging/spec/ (Status 02-26-2021) |
| [EUCS] | **European Union Agency for Cybersecurity (ENISA) (2020), EUCS – Cloud Services Scheme** |
| | https://www.enisa.europa.eu/publications/eucs-cloud-service-scheme (Status: 03-29-2021) |
| [FIPS-140-2] | **NIST (2001) FIPS 140--2, Security Requirements for Cryptographic Modules** |
| | http://csrc.nist.gov/publications/PubsFIPS.html#140-2 (Status 03-18-2021) |
| [IDM.AO] | **Gaia-X WP1[1] (2021), Architecture Overview** |
| | Please refer to annex “GX_IDM_AO” |
| [ISO25000] | **ISO 25000 Portal (n.d.), ISO/IEC 25010** |
| | https://iso25000.com/index.php/en/iso-25000-standards/iso-25010 (Status: 03- |

[1] Please refer to appendix C for an overview and explanation of the Work Packages (WP).

| | |
|---|---|
| | 17-2021) |
| [IDM.TSA] | **Specification for Gaia-X Federation Service Identity & Trust - Trust Services API** |
| | Please refer to annex “SRS_GXFS_IDM_TSA” |
| [OIDC.Core] | **N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, C. Mortimore (2014), OpenID Connect Core 1.0 incorporating errata set 1** |
| | https://openid.net/specs/openid-connect-core-1_0.html (Status: 03-11-2021) |
| [OIDC.CIBA] | **G. Fernandez, F. Walter, A. Nennker, D. Tonge, B. Campbell (2020), OpenID Connect Client Initiated Backchannel Authentication Flow - Core 1.0 draft-03** |
| | https://openid.net/specs/openid-client-initiated-backchannel-authentication-core-1_0.html (Status: 03-11-2021) |
| [OIDC.Conformance] | **OpenID Connect Working Group, OpenID Foundation (2018), OpenID Connect Conformance Profiles v3.0** |
| | https://openid.net/wordpress-content/uploads/2018/06/OpenID-Connect-Conformance-Profiles.pdf (Status: 03-11-2021) |
| [OIDC.Discovery] | **N. Sakimura, J. Bradley, M. Jones, E. Jay (2014), OpenID Connect Discovery 1.0 incorporating errata set 1** |
| | https://openid.net/specs/openid-connect-discovery-1_0.html (Status: 03-11-2021) |
| [NF.SPBD] | **Gaia-X Federation Service Non-functional Requirements Security & Privacy by Design** |
| | Please refer to annex “GXFS_Nonfunctional_Requirements_SPBD” |
| [PRD] | **Gaia-X, European Association for Data and Cloud, AISBL (2021): Gaia-X Policy Rules Document** |
| | Please refer to annex “Gaia-X_Policy Rules_Document_2104” |

| | |
|---|---|
| [RFC6749] | **Internet Engineering Task Force (IETF) (2012), The OAuth 2.0 Authorization Framework** |
| | https://tools.ietf.org/html/rfc6749/ (Status 02-26-2021) |
| [RFC7519] | **Internet Engineering Task Force (IETF) (2015), JSON Web Token (JWT)** |
| | https://tools.ietf.org/html/rfc7519/ (Status 02-26-2021) |
| [RFC6750] | **Internet Engineering Task Force (IETF) (2012), The OAuth 2.0 Authorization Framework: Bearer Token Usage** |
| | https://tools.ietf.org/html/rfc6750 (Status: 03-11-2021) |
| [RFC7591] | **Internet Engineering Task Force (IETF) (2015), OAuth 2.0 Dynamic Client Registration Protocol** |
| | https://tools.ietf.org/html/rfc7591 (Status: 03-11-2021) |
| [RFC7807] | **Internet Engineering Task Force (IETF) (2016), Problem Details for HTTP APIs** |
| | https://tools.ietf.org/html/rfc7807 (Status: 03-11-2021) |
| [RFC5789] | **Internet Engineering Task Force (IETF) (2010), PATCH Method for HTTP** |
| | https://tools.ietf.org/html/rfc5789 (Status: 03-11-2021) |
| [RFC7231] | **Internet Engineering Task Force (IETF) (2014), Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content** |
| | https://tools.ietf.org/html/rfc7231 (Status: 03-11-2021) |
| [RFC3161] | **C. Adams, P. Cain, D. Pinkas, R. Zuccherato (2001), Internet X.509 Public Key Infrastructure, Time-Stamp Protocol (TSP)** |
| | https://www.ietf.org/rfc/rfc3161.txt (Status: 03-17-2021) |
| [RFC2119] | **Network Working Group (1997) Key words for use in RFCs to Indicate Requirement Levels** |

| | |
|---|---|
| | https://tools.ietf.org/html/rfc2119 (Status 03-18-2021) |
| [SOG-IS] | **SOG-IS Crypto Working Group (2020), SOG-IS Crypto Evaluation Scheme - Agreed Cryptographic Mechanisms** |
| | https://www.sogis.eu/documents/cc/crypto/SOGIS-Agreed-Cryptographic-Mechanisms-1.2.pdf (Status 03-18-2021) |
| [TR02102-1] | **BSI (2020), Cryptographic Mechanisms: Recommendations and Key Lengths BSI TR-02102-1** |
| | https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf?__blob=publicationFile&v=2 (Status 03-18-2021) |
| [TR02102-2] | **BSI (2020), Cryptographic Mechanisms: Recommendations and Key Lengths: Use of Transport Layer Security (TLS) BSI TR-02102-2,** |
| | https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-2.pdf?__blob=publicationFile&v=2 (Status 03-18-2021) |
| [TDR] | **Gaia-X Federation Services Technical Development Requirements** |
| | Please refer to annex “GXFS_Technical_Development_Requirements” |
| [TAD] | **Gaia-X, European Association for Data and Cloud, AISBL (2021): Gaia-X Architecture Document** |
| | Please refer to annex “Gaia-X_Architecture_Document_2103” |

***Table 1**: References*

## 1.5. Document Overview

The document describes the product perspective, functions, and constraints. It furthermore lists the functional and non-functional requirements and defines the system features in detail. The listed requirements are binding. Requirements as an expression of normative specifications are identified by a unique ID in square brackets (e.g. **[IDM.ID.Number]**) and the keywords MUST, MUST NOT, SHOULD, SHOULD NOT, MAY, corresponding to RFC 2119 [RFC2119], are written in capital letters (see also [IDM.AO] - Methodology).

# 2. Product Overview

## 2.1. Product Perspective

The product is divided in several modules, which enable a standardized IAM system integration to operate and exchange identity and identity data within the Gaia-X environment and trust framework. The definition of an IAM system as such is not a part of this specification, however it is required to present a working integration of the delivered software modules with an existing open-source OIDC/OAuth2 compatible solution, such examples are Keycloak, Gluu, WSO2, etc. to Authenticate and Authorize in usage of self-sovereign identity concepts. The preference of the solution SHOULD be determined and aligned with the requirements of the other work packages: especially Catalogue, Portal and Organization Credential Manager.

The modules form altogether the IAM SSI Adoption Shell, which encapsulates the functionality needed for SSI-based interactions and protocols, translating them into flows and data formats understandable by the Standard IAM. The main components of this shell are the SSI OIDC Broker and the SSI IAT Provider which support the SSI based user login and the SSI based dynamic client registration.

To visualize the product's cooperation, the functionality was drawn in the figure below (**Figure 1**) as an ArchiMate cooperation view together with the business process of user login and client registration.

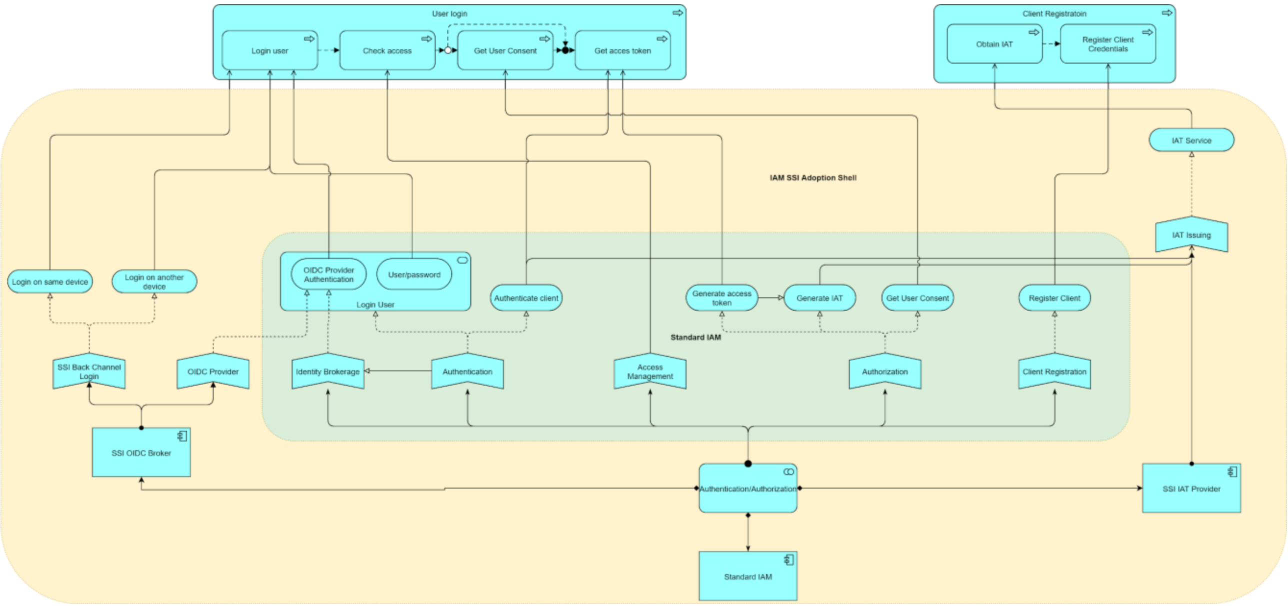


***Figure 1**: Sketch of the Functional Overview of the SSI Adoption Shell*

As seen in **Figure 2**, the target of the product is to adopt an existing IAM by adding SSI based modules to the standard OpenID Connect and OAuth2 compliant IAM. The expected behavior for end users and applications is described as the following:

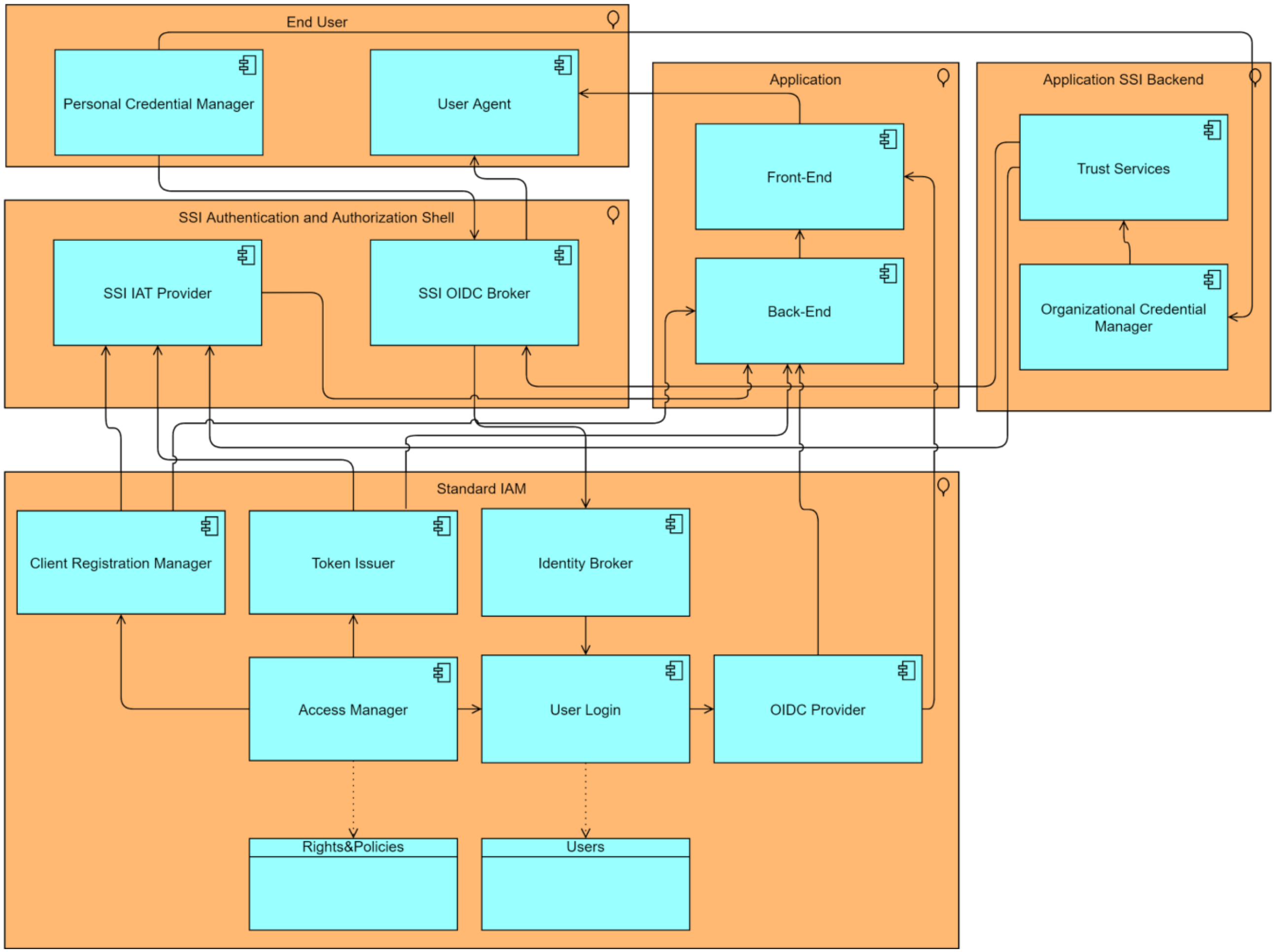


***Figure 2**: Sketch of the Behavior of the SSI Adoption Shell*

**Figure 2** shows that the additional modules have a hard connection to the SSI based application backend which must work as a kind of SSI sidecar for the standard IAM. All actions by the user or by any Back end is triggered in this direction. The "Trust Services" and the "Organizational Credential Manager" are separate lots of the IDM&Trust Package and not scope of this product. The interaction of the product in this direction must be exclusively over standard internet HTTP protocol calls.

## 2.2. Product Functions

Authentication and Authorization components are anchored into the big picture of the IAM architecture for Gaia-X [IDM.AO] and form together a core Gaia-X IAM building block.

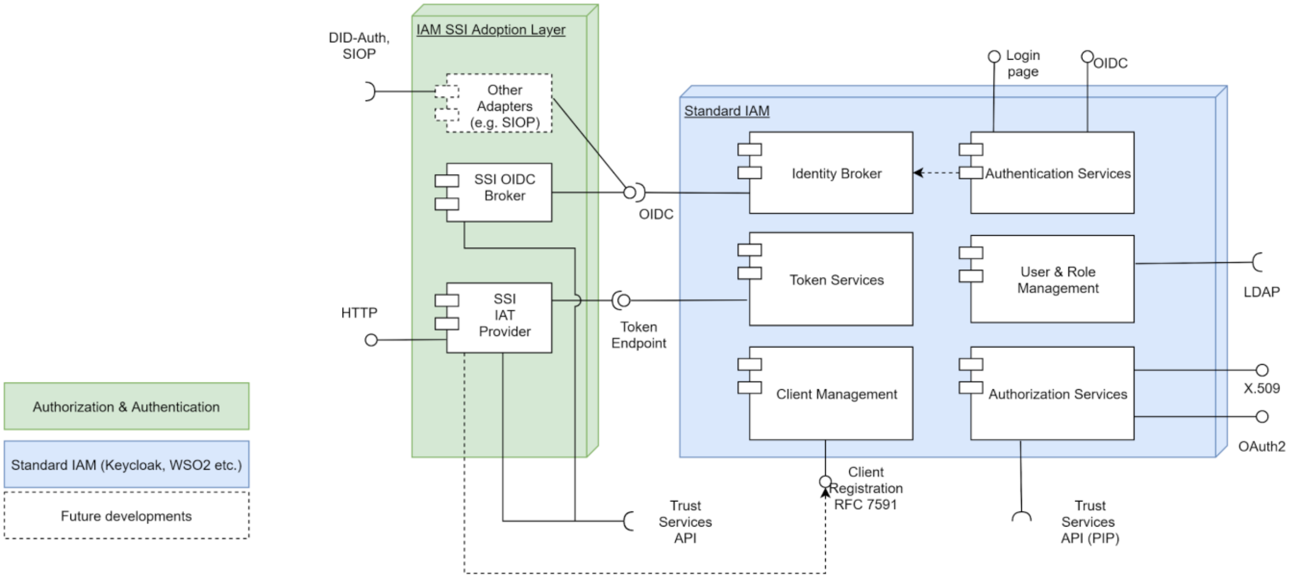


***Figure 3**: Product Functions*

The core functions are:

- SSI based Issuing of Initial Access Tokens (IAT)
- SSI based Login

Both functions are realized with a backchannel authentication which is triggered over the trust services API. This API provides a stateless policy evaluation API which connects the relevant back-channel authentication points over HTTP in the background. The API delivers all information which has to be proofed on the client side over the backchannel. This is inspired by the principle of OpenID Connect Client Initiated Backchannel Authentication [OIDC.CIBA]. The user agents have in this case to poll to evaluate the login state of their action by a kind of ticket ID. Pull and Ping flows are out of scope. In both cases the users have to trigger the proof over the backchannel, or the proof is timed out.

- Other functions like DID SIOP [DID SIOP]can follow in the future if the specification is done by the OpenID foundation. Especially the DID SIOP flow is currently in an early version and just yet pushed to the OpenID foundation workgroups for specification of version two.

## 2.3. Product Constraints

⏩ IDM.AA.00001 **The document IDM.AO is the common basis for this functional specification**

The architecture document "IDM.AO" [IDM.AO] is an essential part of this specification and a prerequisite for understanding the context. The specifications and requirements from the Architecture Document MUST be considered during implementation. ⏪

## 2.4. User Classes and Characteristics

| *User Class* | *Description* | *Frequency* | *Expertise* | *Privilege Level* | *Product Usage* |
|---|---|---|---|---|---|
| Administrator | The administrator installs the product and configures it in the existing IAM system. | Low | High | High | Maintenance |
| External Users | External users are using the product for login. | High | Low | Low | SSI Based Login |
| External APIs | External APIs can use the product to authenticate via SSI to obtain an IAT for client registration. | Low | High | Low | SSI Based Client Registration |

***Table 2**: User Classes and Characterstics*

## 2.5. Operating Environment

Please refer to [TDR] for further binding requirements regarding the operating environment.

⏩ IDM.AA.00002 **TLS Protected Endpoints**

To protect the product endpoint(s), it’s necessary to support a network infrastructure e.g., load balancers/proxies which MUST support TLS encryption. The encryption MUST meet the requirements listed in the chapter for security requirements. ⏪

⏩ IDM.AA.00003 **NTP Server on Stratum Level**

The product MUST be operated within an environment connected to a trusted stratum level 2/3 NTP Server. ⏪

## 2.6. User Documentation

Please refer to [TDR] for further requirements regarding documentation.

⏩ IDM.AA.00004 **Participant Administration Documentation**

The documentation MUST contain:

- Installation Manuals

- Cryptographic Initialization (if applicable)
- Description of Deployment/Compile Process
- Description of the Automatic Tests / Verification
- How to build the products from source code ⏪

⏩ IDM.AA.00005 **Participant Documentation**

The documentation MUST contain:

- Short Software Description (why and for what, when to use, how use, where to use)
- Usage guide
- GDPR design decisions
- Security concept
- Operations concept
- FAQ
- Keyword Directory ⏪

## 2.7. Assumptions and Dependencies

The main assumptions behind this SSI adoption shell are that the shell is running in the same security domain as the standard OIDC compatible IAM system. All components MUST be built on top of the OAuth2 [RFC6749] and OIDC standard [OIDC.Core].

An understanding of the overall Gaia-X architecture and philosophy is necessary. Please refer to [TAD].

## 2.8. Apportioning of Requirements

All Features have to be delivered in the first version.

# 3. Requirements

Further binding requirements can be found in [TDR].

## 3.1. External Interfaces

⏩ IDM.AA.00006 **Trust Services API**

The Trust Services API is an API which evaluates policies in the background. In this scenario the API is used to check the proof state of the given presentationID to find out if the proof is valid or not. The API is returning a REST Codes and JSON structure with the associated proof information. ⏪

⏩ IDM.AA.00007 **OIDC Interface**

The OIDC Interface provides a standard OIDC flow which follows the open id connect protocol 1.0. ⏪

⏩ IDM.AA.00008 **Token Interface**

The token interface is a token endpoint of a standard IAM system and provides functionalities to obtain token with different token flows. This token endpoint MUST follow the OAuth2 authorization framework spec [RFC6749]. ⏪

### 3.1.1. User Interfaces

The SSI OIDC Broker interface provides a web page for login with a QR Code. (see more in the functional chapter below)

### 3.1.2. Software Interfaces

The product described within this specification does not expose any specific software interfaces other than communication interfaces covered in 3.1.3. Communications Interfaces.

The choice of operating system, database system, tools and libraries are left open to the provider, as long as other requirements within this specification, especially those described in 2.3. Product Constraints and 2.5. Operating Environment are fulfilled.

### 3.1.3. Communications Interfaces

#### 3.1.3.1. SSI OIDC Broker

3.1.3.1.1. Offered Interfaces

⏩ IDM.AA.00009 **OIDC Provider**

SSI OIDC Broker offers OpenID Connect provider interface for integration of authentication and authorization flows with any compatible application or identity brokering IAM system.

All offered HTTP interfaces are following [OIDC.Core] and [OIDC.Discovery] as appropriate and specified in the feature description part of this specification. ⏪

3.1.3.1.2. Consumed Interfaces

⏩ IDM.AA.00010 **Trust Services API**

SSI OIDC Broker consumes policy evaluation interfaces of Trust Services API [IDM.TSA] for back channel DIDComm-based authentication and authorization. ⏪

### 3.1.3.2. SSI IAT Provider

3.1.3.2.1. Offered Interfaces

⏩ IDM.AA.00011 **SSI Client Registration Auth API**

SSI IAT Provider offers a HTTP-based REST API which allows for obtaining [RFC7591] compliant Initial Access Token (IAT) for client registration at an OAuth2 Authorization Server.

Open API specification: SSI Client Registration Auth API.yml ⏪

3.1.3.2.2. Consumed Interfaces

⏩ IDM.AA.00012 **Trust Services API**

SSI OIDC Broker consumes policy evaluation interfaces of Trust Services API [IDM.TSA] for back channel DIDComm-based authentication and authorization. ⏪

⏩ IDM.AA.00013 **OAuth2 token interface**

SSI OIDC Broker connects to a standard IAM OAuth2 endpoint, using client credentials grant with appropriate scope, to generate new [RFC7591] Initial Access Tokens. ⏪

## 3.2. Functional

### 3.2.1. Common

⏩ IDM.AA.00014 **Credential Based Access Control (CrBac)**

The SSI adoption shell SHOULD be able to dynamically reload credentials for access decisions, for instance the current identity wants a “sales” action and is currently just logged in with a “visitor” permission. The CrBac SHOULD be able to resolve this by requesting new credentials. This might be achieved either by a renewed authentication and authorization flow triggered by the application (via SSI OIDC Provider), or via an asynchronous process, which SHOULD be done over standard IAM outgoing PIP interface towards Trust Services or the It MAY be realized over additional components within the architecture, but the Standard IAM MUST NOT be modified (excepting configuration, plugins or supported extensions).

Acceptance Criteria

1) During a resource access, it must be demonstrated that new credentials can be transmitted to get access
2) The documentation must describe what to configure in the demonstration IAM ⏪

⏩ IDM.AA.00015 **External PIP Integration**

It MUST be demonstrated how an external PIP with asynchronous behavior can be integrated in the authorization services of a standard IAM system. It MAY be demonstrated with additional components. ⏪

### 3.2.2. SSI OIDC Broker

⏩ IDM.AA.00016 **Standard open source IAM Package**

Together with the Adoption Shell, the product MUST include a working integration with one basic open source IAM system. The selection has to be compliant with the Gaia-X policy and rules and any choice that supports the OAuth2 [RFC7591] standard functions as Client Registration [RFC7591] , Token Issuing (minimum code and client credential flow), authorization and permission handling.

Acceptance Criteria

1) One working IAM System with the Adoption Shell
2) Documentation of proper configuration of both Adoption Shell and chosen IAM system
3) “All in one” package, ready for installation

⏪

⏩ IDM.AA.00017 **OpenID Provider Configuration Information**

SSI OIDC Provider MUST offer an OpenID Provider Configuration Information endpoint as specified in [OIDC.Discovery]. All SSI-related scopes and claim types SHOULD be exposed through this endpoint for dynamic discovery and configuration. The SSI OIDC Provider MUST fulfill the requirements of OpenID Provider Publishing Configuration Information profile as in [OIDC.Conformance].

Acceptance Criteria

1) Documentation of self-conducted conformance testing of OpenID Provider Publishing Configuration Information profile or an official certification from OpenID Foundation
2) Documentation how the Well-Known Configuration has to be setup

3) Demonstrate a working dynamic configuration of a standard IAM OIDC broker with OpenID Provider Configuration Information of SSI OIDC Provider

4) Explanation in the operations concept how to configure it

⏪

⏩ IDM.AA.00018 **OpenID Connect Implicit profile**

SSI OIDC Provider MUST fulfill all the requirements of Implicit OpenID Provider profile as defined in [OIDC.Conformance].

Acceptance Criteria

1) Documentation of self-conducted conformance testing of Implicit OpenID Provider profile or an official certification from OpenID Foundation

2) Demonstrate a working integration with a standard IAM brokering OIDC Authorization to SSI OIDC Provider with OpenID Connect Implicit profile configuration.

3) The documentation must describe what to configure in the demonstration IAM

⏪

⏩ IDM.AA.00019 **OAuth 2.0 Security Best Current Practice**

SSI OIDC Provider SHOULD employ all relevant measures for security of OAuth2.0 framework [BCP OAuth2].

Acceptance Criteria:

4) Documentation of applied measures as per [BCP OAuth2]

⏪

⏩ IDM.AA.00020 **SSI Login Page**

The OIDC Provider MUST contain a customizable Standard Login web page integrated into OIDC Authorization flow and endpoint as per [OIDC.Core] which shows an QR Code for login using another device hosting Personal Credential Manager as well as a button with the same link being able to open Personal Credential Manager on the same device. The button MUST trigger a registered application or page on the new browser tab, without interruption of the count-down and the status polling process.

The SSI Login Page MUST display a progress bar which counts down a configurable time (e.g., 30 seconds).

The login page MUST poll in the background the configured login state URL with a given request identifier for the configured amount of time.

If the request is successful or timed-out during the waiting time the SSI Login Page MUST redirect the User Agent back to the IAM system redirect_url with appropriate response parameters as per [OIDC.Core].

This login page MUST be styleable over CSS or html template system to customize the look and feel at the installation and configuration time. The login page should support internationalization and follow either browser settings or OIDC parameters defined for this purpose. OIDC parameters take precedence. Minimum set of supported languages: English, German, French.

Acceptance Criteria

1) The login page presents an QR Code and the Button Link with an SSI invitation link/proof request
2) The login page redirects with an error response after the time-out
3) The login page polls in the background the login state and redirects to the given URL after a successful login
4) The login page must deliver with one specific Look & Feel aligned with Gaia-X Portal UX. Information regarding Look & Feel of the Gaia-X Portal UX will be provided by eco.
5) The login page must support required languages and display according to the browser settings or OIDC authorization request parameters as per precedence definition
6) The documentation must describe Look & Feel customization options and their configuration

⏪

⏩ IDM.AA.00021 **QR Code Generation**

The QR code contains the content of the SSI invitations/proofs, which MUST be obtained from an external URL with the values for scope and a value for “Namespace”. Scope Values MUST be extracted from the authorize request. The URL request body SHOULD follow the following pattern matching the requirements of [IDM.TSA] Appendix B GetLoginProofInvitation Policy:

```
{
        “scope”:
        [
                “openid”,
                "gaia-x.user",
                "gaia-x.role",
                "gaia-x.organization.membership",...
        ]
        "Namespace": "AdministrationLogin", (Client)
        "sub": "did:example:123123123", (Optional reference)
        "not_older_than": "2021-04-01T01:23:45.678+00:00", (Optional ttl of the required proof)
        "max_age": 3600 (Maximum Authentication Age. Specifies the allowable elapsed time in seconds since the
last time the End-User was actively authenticated)
}
```

The response body follows the following pattern:

```
{
        "presentationID":"....",
        "link":"..."
}
```

The link content has to be generated in a QR Code. PresentationID needs be securely stored in the browser session, so that it's available for the [IDM.AA.00028] Login State Background Polling process and not revealed outside of this context as it represents a secure token to get identity data.

Acceptance Criteria

1) The button and the QR Code contain the link content rendered as QR Code
2) The QR Code must be readable by a smartphone
3) The presentation ID is not inside the QR Code or button link

◂◂

### ▸▸ IDM.AA.00022 **Login State Background Polling**

During the shown Login page, a background task MUST poll for the state of the current presentationID created in [IDM.AA.00027] QR Code Generation. This HTTP request MUST execute in the background a request for the state with the following pattern in the request body to Trust Services API [IDM.TSA] Appendix B GetLoginProofResult Policy:

```
{
        “presentationID”:”....”,
}
```

The response body SHOULD follows the following pattern:

1) No State: HTTP 204
2) Done: HTTP 200

```
{
        “iss”: “...”,
        “sub”: “...”,
        “claim1”: “...”
        ...
        “claimx”: “...”
}
```

3) Error response: HTTP 40x

The content in the response MUST be available in the IAM system after the successful response. This MUST be realized by continuing the standard OIDC flow and forming a valid id_token including the claims from the response.

Unsuccessful response shall be followed with an option to retry the process with [IDM.AA.00027] QR Code Generation or to fail the process and redirect back to the IAM with appropriate OIDC failure response.

Acceptance Criteria

1) Login Token with the contained claims and scopes
2) Correctly signed token
3) Redirect to IAM System on Success
4) Offer Retry or redirect back with failure to IAM System on Error

◂◂

▸▸ IDM.AA.00023 **Session handling and scope elevation**

SSI OIDC Provider MUST employ configurable session handling allowing multiple authorization requests for the same identity being seamlessly handled without a need to authenticate the user again for a pre-defined time period of session validity. id_token_hint parameter as specified in [OIDC.Core] SHOULD be supported.

In case additional scopes are required the SSI OIDC Provider MUST conduct a proof request for the additional Verifiable Credentials without a need to build a new connection with an invitation QR Code/Link. The option to create a new Link MUST be available, however.

Session handling related parameters of [OIDC.Core], like prompt or max_age shall be respected and translated into appropriate proof requests to assure required functionality.

The proof is realized with the same methods and policies as described in [IDM.AA.00027] QR Code Generation and [IDM.AA.00028] Login State Background Polling using optional sub and max_age parameters.

Acceptance Criteria:

1) session handling by consecutive authentication requests with id_token_hint do not impose new authentication and authorization and authenticate the user in a seamless manner
2) a request for additional proofs is conducted and added to id_token in case the requested scope of authorization is wider than previously
3) Session duration is configurable at installation / deployment time

◄◄

### 3.2.3. SSI IAT Provider

►► IDM.AA.00024 **Offer SSI Client Registration Auth API**

An API as per [IDM.AA.00017] SSI Client Registration Auth API API MUST be offered to enable initiation and polling for the result of SSI-based issuance of IAT for Dynamic Client Registration.

Acceptance Criteria

1) SSI Client Registration API is offered and documented

◄◄

►► IDM.AA.00025 **Policy based authorization**

The SSI IAT Provider MUST integrate with Trust Services to conduct policy authorization checks of the client trying to obtain an Initial Access Token (IAT). IAT MUST not be issued unless the policy evaluation allows for that operation.

The following Policies have to be executed by this authorization flow:

- [IDM.TSA] Appendix B GetIatProofInvitation  - flow initiation
- [IDM.TSA] Appendix B GetIatProofResult Policy - polling for a result

Acceptance Criteria

2) IAT is issued only after successful evaluation of the Policy by Trust Services
3) negative evaluation of the Policy by Trust Services results in no IAT issued and an appropriate error response

IDM.AA.00026 **Standard IAM Compatibility**

The issued Initial Access Token MUST be compatible with the Client Registration Endpoint of the docked standard IAM. It MUST be possible to register with this IAT client as defined in [RFC7591].

Acceptance Criteria

1) A dynamic client with an IAT can be registered in the IAM system over the client registration endpoint.

IDM.AA.00027 **Client Registration**

The IAT Provider MUST be registered as an OAuth2 [RFC6750] Client using client credential grant within the Standard IAM to obtain Initial Access Tokens of the System.

Acceptance Criteria

1) A dynamic client with a software statement can be registered in the IAM system over the client registration endpoint.

## 3.3. Other Nonfunctional Requirements

IDM.AA.00028 **Security Hardening**

The whole adoption shell is security relevant, and it has to be defined in the security concept how these components can be more secured and what kind of steps to do.

### 3.3.1. HTTP Requirements

IDM.AA.00029 **HTTPS**

All HTTP Endpoints MUST be protected by TLS 1.2 (all protocol version numbers SHOULD be superseded by upcoming standards) Each endpoint of the product MUST support TLS certificates which are configurable by the administrator of the system.

IDM.AA.00030 **HTTP Protocol Definitions**

All HTTP Endpoints MUST follow [RFC7231] and [RFC5789], but it MAY be chosen what of the protocols is necessary to realize the functionality. For problem reports the [RFC7807] MUST be used in combination with Standard HTTP Error Codes.

### 3.3.2. Configuration

IDM.AA.00031 **Configuration**

All components MUST support one of the major configuration formats (yaml, json, ini, environment variables) wherever configuration is required. If environment variables are overwriting an actively set configuration, a warning SHOULD be logged.

### 3.3.3. Logging Requirements

IDM.AA.00032 **Data Minimization**

From GDPR perspective the product MUST NOT log data which is related to personal information. (e.g., User Names, Birth Dates etc. ) The product MUST only log data, which is relevant to technical operations, except for the purpose that, in the event of an incident, enable reconstruction of the sequence of the message exchange for establishing the place and the nature of the incident. The data shall be stored for a period of time in accordance with national requirements and, as a minimum, shall consist of the following elements:

(a) node's identification

(b) message identification

(c) message data and time

All logged data/information MUST be documented in the GDPR design decisions for a GDPR review.

⏪

⏩ IDM.AA.00033 **Logging Frameworks**

The product MUST support logging frameworks e.g., graylog, fluentD or logstash to support logging and analysis by enterprise infrastructures. The supported framework MAY be chosen for the first version, but it MUST support potentially the most common open-source logging solutions. The final solution MUST be aligned with the other subcomponents. It MUST be sketched in the operations concept how the support of multiple solutions is given in the future.

⏪

### 3.3.4. Monitoring Requirements

⏩ IDM.AA.00034 **Monitoring Frameworks**

The product MUST support monitoring frameworks e.g., Grafana to support the analysis of incoming data by the enterprise infrastructures. The supported framework MAY be chosen for the first version, but it MUST support potentially the most common monitoring solutions. (e.g., Zabbix)  The final solution MUST be aligned with the other subcomponents. It MUST be sketched in the operations concept how the support of multiple solutions is given in the future. ⏪

⏩ IDM.AA.00035 **Alerting Frameworks**

Additional to the Monitoring Frameworks an Alerting framework (e.g., Prometheus or Cloud Based) MUST/MAY be in place at least in the System nodes to promptly communicate to e.g., System Administrators or owners the occurrence of an event in form of a security incident or application/system malfunction or anomaly. ⏪

### 3.3.5. Performance Requirements

⏩ IDM.AA.00036 **Performance Scalability**

The performance of the product MUST be scalable. This MUST be demonstrated in a load demonstration example. The optimal scalability SHOULD be in the best case a linear behavior of minimum 50% more performance by each additional instance. ⏪

⏩ IDM.AA.00037 **Performance by Design**

The product SHOULD be designed and implemented in a way, that the implementation is non-blocking and performance oriented. It SHOULD be a microservice architecture, but it MAY follow other concepts. The decision MUST be documented. ⏪

### 3.3.6. Safety Requirements

⏩ IDM.AA.00038 **Recovery Point Objective (RPO)**

The RPO for the product MUST be 0 for a single and multiple instance(s). It MAY be higher by configuration or deployment, decided by the user. ⏪

⏩ IDM.AA.00039 **Recovery Time Objective (RTO)**

The RTO for the product MUST be one Minute for a single instance. For multiple instances the RTO MUST be 0. ⏪

⏩ IDM.AA.00040 **Mitigation of Single Point of Failure threats**

Critical components in the Gaia-X Ecosystem MUST be identified and strategies to warranty their availability and scalability MUST be implemented. ⏪

### 3.3.7. Security Requirements

#### 3.3.7.1. General Security Requirements

Each Gaia-X Federation Service SHALL meet the requirements stated in the document "Specification of non-functional Requirements Security and Privacy by Design" [NF.SPBD]. Federation Services specific requirements will be documented in the next chapter.

#### 3.3.7.2. Service Specific Security Requirements

This chapter will describe the service specific requirements, which will extend the requirements defined in the chapter above.

⏩ IDM.AA.00041 **Cryptographic Algorithms and Cipher Suites**

Cryptographic algorithms and TLS cipher suites SHALL be chosen based on the recommendation from the German Federal Office for Information Security (BSI) or SOG-IS. These recommendations and the recommendations of other institutions and standardization organization are quite similar [CryptoLen]. The recommendations can be found in the technical guidelines TR 02102-1 [TR02102-1] and TR 02102-2 [TR02102-2] or SOG-IS Agreed Cryptographic Mechanisms [SOG-IS]. ⏪

⏩ IDM.AA.00042 **Digital Certificates**

For digital certificates and cryptographic signatures in the context, the major requirements on cryptographic algorithms and key length MUST meet the definitions in the following table (as of 2020):

| Signature Algorithm | Key size | Hash function |
|---|---|---|
| EC-DSA | Min. 250 Bit | SHA-2 with an output length ≥ 256 Bit or better |
| RSA-PSS (recommended)<br>RSA-PKCS#1 v1.5 (legacy) | Min. 3000 Bit RSA Modulus (n) with a public exponent e > 2^16 | SHA-2 with an output length ≥ 256 Bit or better |
| DSA | Min. 3000 Bit prime p<br>250 Bit key q | SHA-2 with an output length ≥ 256 Bit or better |

***Table 3**: Requirements on cryptographic algorithms and key length*

Named curves SHALL be used for EC-DSA (e.g., NIST-p-256). ⏪

⏩ IDM.AA.00043 **TLS Certificate Validity Periods**

In general, the recommended validity period for a certificate used in the system should be one year or less. Under some circumstances (for example RootCA) the certificate validity can be extended. Certificate owners MUST ensure that valid certificates are renewed and replaced before their expiration to prevent service outages. ⏪

⏩ IDM.AA.00044 **Security by Design**

The software security MUST be from the beginning a design principle. Means separation of concerns, different administrative roles, especially for private key material and separate access to the data MUST be covered from the first second. It MUST be described in the security concept, what are the different security risks of the product and how they are mitigated (e.g., by Threat Modeling Protocols) ⏪

⏩ IDM.AA.00045 **Installation of Critical Security Updates**

Node operators SHALL deploy security critical updates without undue delay. ⏪

⏩ IDM.AA.00046 **Avoid HTTP Request Smuggling**

To avoid Request Smuggling attacks, the product MUST implement a standard which handles this kind of attack by design, because the attack vector results in an insufficient implementation of the header handling. The chosen way to handle it MUST be shared to the other implementers of all other subcomponents within IDM & Trust and MUST be described in the security concept.

IDM.AA.00047 **HTTP Pentesting**

All HTTP parts of the product has to be pen tested, for the following criteria:

1) Unauthorized Access to the System MUST be tested
2) Unauthorized Actions MUST be triggered without a user action
3) Endpoints MUST be tested for HTTP smuggling attack vectors
4) If a datastore is present over HTTP, illegal data access MUST be tested

It's RECOMMENDED to test more attack vectors and document it for the purpose to mitigate it in later versions.

IDM.AA.00048 **Storage of Secrets**

The storage of secret information such as private keys MUST take place in state-of-the-art secure environments to protect secret data confidentiality and integrity. Examples of this are Secure Enclaves, TPMs, HSM or Secure Vaults. In case (Personal) Agents are not equipped with a secure storage it MAY also be possible to store the secrets in a third party (e.g., Cloud) provider (e.g., Secure Wallet) that MUST provide overall the same level of security as the aforementioned methods.

IDM.AA.00049 **Secret Distribution and Usage**

The product MUST ensure interoperability of cryptographic primitives and components by public standards and MUST use secure state of the art methods to create and import secrets into the secure storage, as well as performing cryptographic operations (e.g., encryption or digital signatures). For Key distribution, state of the art DKMS methods MUST be implemented.

IDM.AA.00050 **Support for Potential Requirements for Secret Storages**

Devices that hold cryptographic information and perform cryptographic functions MUST be compliant with the standard PKCS #11. Moreover, the products MUST be potentially eligible for a [FIPS-140-2] or ETSI/Common Criteria certification with the minimum-security level necessary to operate securely in the Gaia-X ecosystem. Security Levels in FIPS-140-2 range from 1 to 4. Current HSM Cloud Service offerings (AWS, Azure, GCP) are Level 3.

⏩ IDM.AA.00051 **Secure Timestamps**

All timestamps MUST be issued according to [RFC3161]. ⏪

⏩ IDM.AA.00052 **Special Availability and Scalability Requirements for Secret Storage Components**

Secret Storage components play a central role in storage, encryption, and digital signing in the Gaia-X ecosystem, thus they can become a single point of failure for a Gaia-X participant, for example an organization. Therefore, methods and procedures to ensure the availability and scalability of the Secret Storage functionality MUST be implemented. ⏪

### 3.3.8. Software Quality Attributes

⏩ IDM.AA.00053 **Quality Aspects**

The software MUST meet the following requirements:

- The quality standards MUST meet ISO 25010 [ISO25000]
- Robustness / Reliability
- Performance
- Availability must be 24/7
- Interoperability with the other work packages
- Security
- Adaptability / expandability
- Maintainability and Code Quality
- Scalability

Major security concerns regarding design and implementation MUST be documented and highlighted to the steering board. Minor security concerns SHALL be documented and mitigated. ⏪

### 3.3.9. Business Rules

Business Rules are defined by external components, especially by [IDM.TSA].

## 3.4. Compliance

⏩ IDM.AA.00054 **GDPR Audit Logging**

All GDPR relevant access to personal relevant data MUST be logged for a later audit. ⏪

⏩ IDM.AA.00055 **GDPR Data Processing**

Is it necessary to process person-relevant data, it MUST be earmarked to a clearly defined business process, which has to be described in the GDPR design decisions. All person relevant data MUST be deleted after the processing, if applicable. ⏪

## 3.5. Design and Implementation

Please also refer to [TDR] for further requirements.

### 3.5.1. Installation

⏩ IDM.AA.00056 **Micro Service Architecture**

For a better scale out and decentralization, the product architecture MUST a micro service architecture. ⏪

⏩ IDM.AA.00057 **Independent Service**

All Adoption Shell functions MUST be separately deployable web applications, which communicate with other systems either over exposed HTTP-based APIs or consume HTTP-based APIs. The components MUST NOT be built into any particular IAM system in a way of extension or plugin, so that it’s always possible to connect them to any IAM system which is offering the needed APIs.

Acceptance Criteria

1) All functions of the Adoption Shell are deployable as standalone web applications ⏪

### 3.5.2. Distribution

⏩ IDM.AA.00058 **Config Data Distribution**

The product SHOULD support a global data distribution of config data to synchronize configurations between multiple regions in the world. Built-in synchronization technology (asynchronous and synchronous) MAY be used. ⏪

### 3.5.3. Maintainability

⏩ IDM.AA.00059 **Micro Service Architecture**

For a better scale out, maintainability and decentralization, the product architecture MUST have a micro service architecture. Each microservice MUST NOT be limited on the lines of code or number of days to implement it. The service “size” SHOULD be oriented on the fine granular business capabilities. (e.g., Order, ListMenu, Payment). ⏪

⏩ IDM.AA.00060 **Domain Driven Design**

To support the micro service architecture within the maintainability, it MUST be declared a domain model before realization. The software description MUST explain which domain model was chosen, which services contain it and how it scales. This MUST be documented in the public code repository to support future enhancements for new developers. ⏪

### 3.5.4. Operability

⏩ IDM.AA.00061 **FTE Estimation**

The product MUST be designed so that over scripts and tools one FTE within a Month SHOULD host and operate the product without any third-party help. It MUST be sketched in the operations concept how this can be achieved. If this target is not reachable it MAY be explained and described why the effort is higher and appropriate. ⏪

### 3.5.5. Interoperability

⏩ IDM.AA.00062 **Interoperability of IT security features and algorithms**

The following interoperability requirements of the respective IT security features and algorithms MUST be ensured across the system components:

- Interoperability of crypto algorithms and protocols (including the novel peer-reviewed ones through the established bodies and communities)
- Interoperability of secure secret transfer protocols (such as the holistic usage of PKCS#11 for HSM communication, etc.)
- Format interoperability of crypto material (such as the holistic usage of PKCS#12 for relevant cases) ⏪

# 4. System Features

## 4.1. SSI Back Channel login

### 4.1.1. Description

The feature provides the capability to login over an QR code, and an SSI Backchannel provided by the trust services API. This provider enables the user to use his personal SSI wallet for login to a protected resource.

The provider itself must be configured over a standard oidc identity provider configuration within an IAM System.

### 4.1.2. Stimulus/Response Sequences

#### 4.1.2.1. User Authentication

The purpose of this flow is the authentication of the user and retrieval of trustworthy identity data. The flow starts with an application (Relying Party) which initiates OpenID Connect authorization flow through its internal IAM system, requesting normal OIDC scopes as well as specific Gaia-X scopes, which would be translated into a set of proofs requested from the holder and its organization. The result of this flow is an authenticated identity as well as a set of claims resulting from the given set of proofs. The proof request and response process will be conducted by Trust Services API [IDM.TSA].

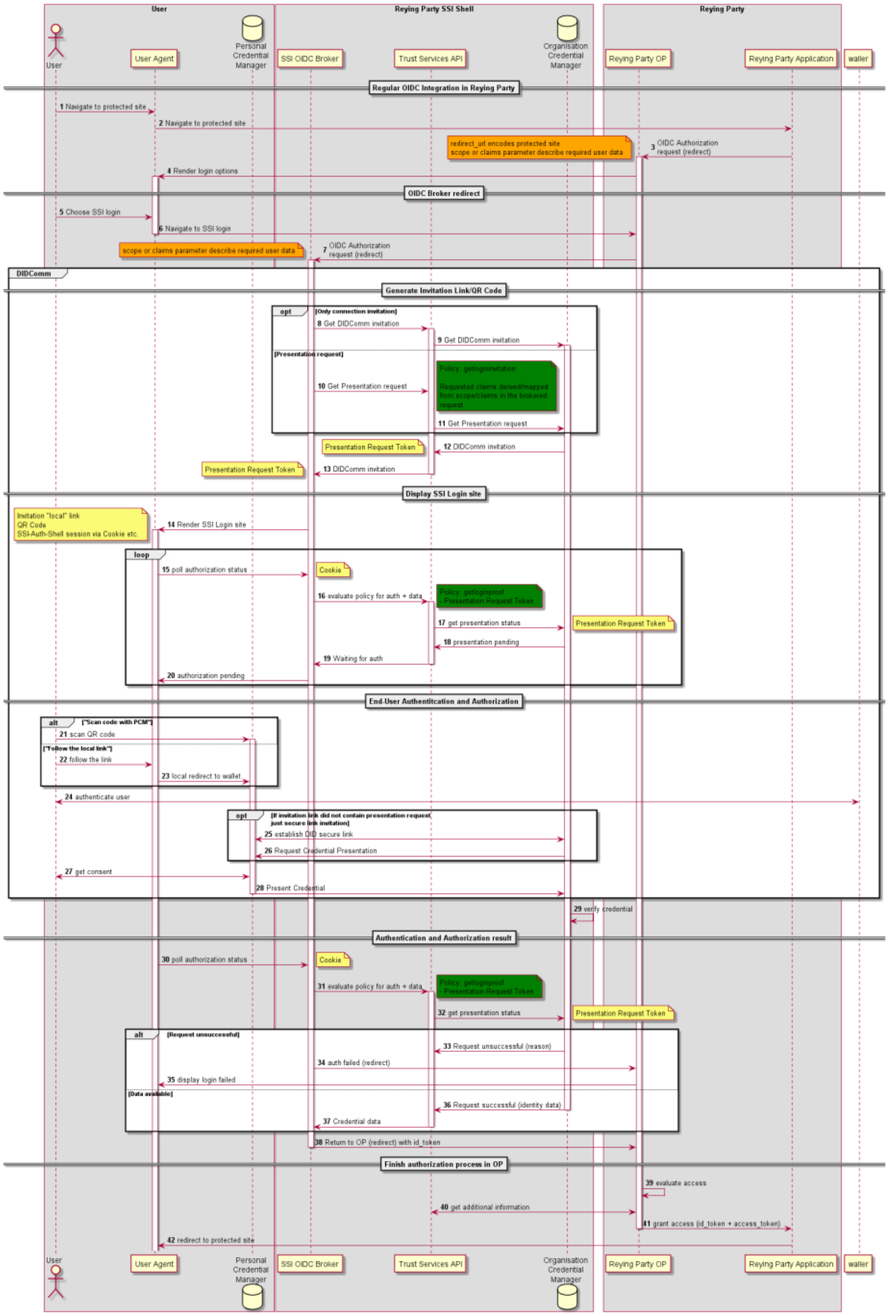

User
Reying Party SSI Shell
Reying Party
User
User Agent
Personal Credential Manager
SSI OIDC Broker
Trust Services API
Organisation Credential Manager
Reying Party OP
Reying Party Application
waller
Regular OIDC Integration in Reying Party
1 Navigate to protected site
2 Navigate to protected site
redirect_url encodes protected site
scope or claims parameter describe required user data
3 OIDC Authorization request (redirect)
4 Render login options
OIDC Broker redirect
5 Choose SSI login
6 Navigate to SSI login
scope or claims parameter describe required user data
7 OIDC Authorization request (redirect)
DIDComm
Generate Invitation Link/QR Code
opt
[Only connection invitation]
8 Get DIDComm invitation
9 Get DIDComm invitation
[Presentation request]
Policy: getlogininvitation
Requested claims derived/mapped from scope/claims in the brokered request
10 Get Presentation request
11 Get Presentation request
Presentation Request Token
12 DIDComm invitation
Presentation Request Token
13 DIDComm invitation
Display SSI Login site
Invitation "local" link
QR Code
SSI-Auth-Shell session via Cookie etc.
14 Render SSI Login site
loop
15 poll authorization status
Cookie
16 evaluate policy for auth + data
Policy: getloginproof
- Presentation Request Token
17 get presentation status
Presentation Request Token
18 presentation pending
19 Waiting for auth
20 authorization pending
End-User Authentitcation and Authorization
alt
["Scan code with PCM"]
21 scan QR code
["Follow the local link"]
22 follow the link
23 local redirect to wallet
24 authenticate user
opt
[If invitation link did not contain presentation request just secure link invitation]
25 establish DID secure link
26 Request Credential Presentation
27 get consent
28 Present Credential
29 verify credential
Authentication and Authorization result
30 poll authorization status
Cookie
31 evaluate policy for auth + data
Policy: getloginproof
- Presentation Request Token
32 get presentation status
Presentation Request Token
alt
[Request unsuccessful]
33 Request unsuccessful (reason)
34 auth failed (redirect)
35 display login failed
[Data available]
36 Request successful (identity data)
37 Credential data
38 Return to OP (redirect) with id_token
Finish authorization process in OP
39 evaluate access
40 get additional information
41 grant access (id_token + access_token)
42 redirect to protected site
User
User Agent
Personal Credential Manager
SSI OIDC Broker
Trust Services API
Organisation Credential Manager
Reying Party OP
Reying Party Application
waller

***Figure 4**: SSI Back Channel login*

### 4.1.2.2. User Authentication roaming between two applications and security domains

The purpose of this flow is the authentication and authorization with the same user identity when moving between two applications and security domains. This flow also illustrates how one application may dynamically gain access to resources hosted with another application in a trustworthy manner.

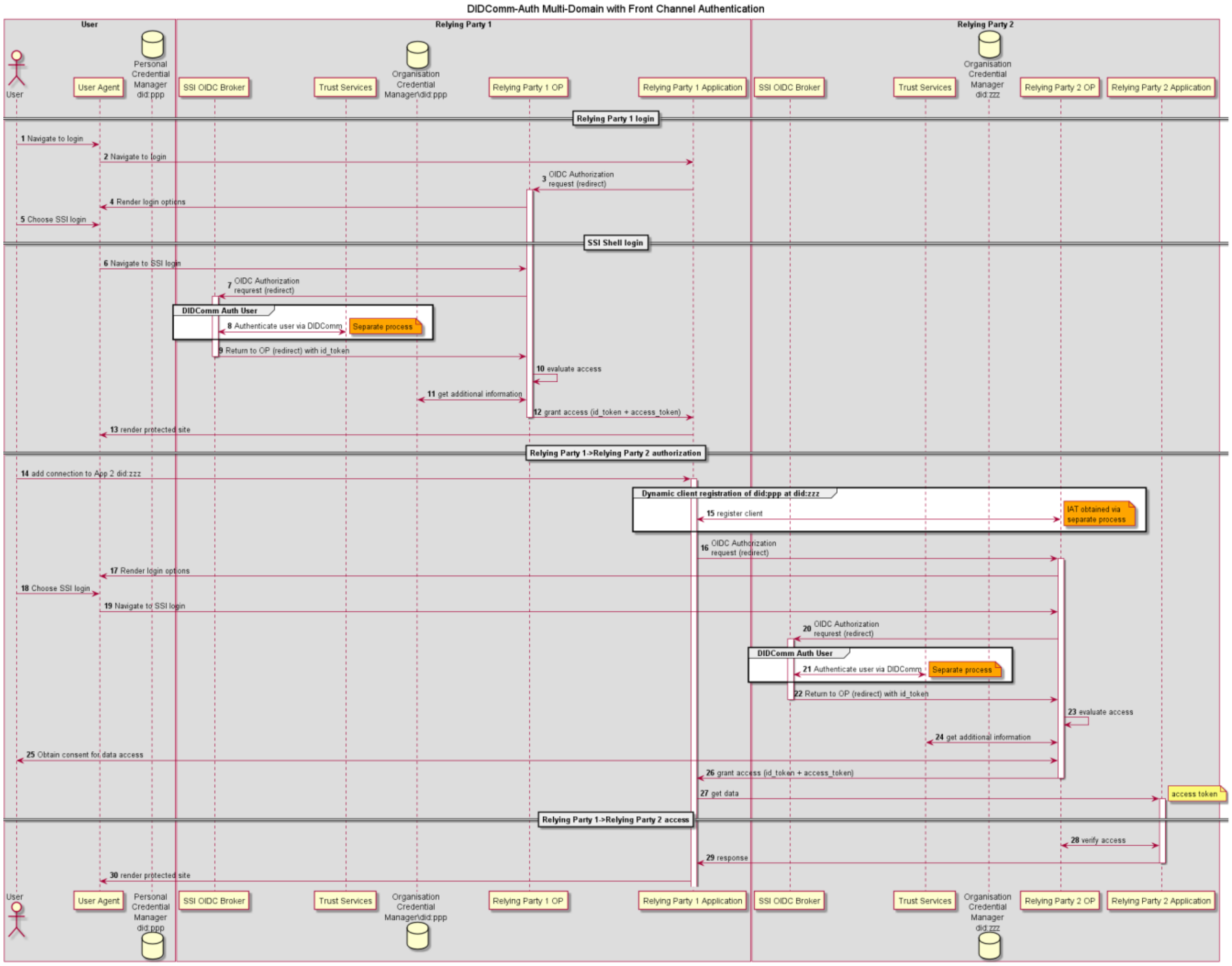

DIDComm-Auth Multi-Domain with Front Channel Authentication
User
Relying Party 1
Relying Party 2
User
User Agent
Personal Credential Manager did:ppp
SSI OIDC Broker
Trust Services
Organisation Credential Manager\did:ppp
Relying Party 1 OP
Relying Party 1 Application
SSI OIDC Broker
Trust Services
Organisation Credential Manager did:zzz
Relying Party 2 OP
Relying Party 2 Application
Relying Party 1 login
1 Navigate to login
2 Navigate to login
3 OIDC Authorization request (redirect)
4 Render login options
5 Choose SSI login
SSI Shell login
6 Navigate to SSI login
7 OIDC Authorization requrest (redirect)
DIDComm Auth User
8 Authenticate user via DIDComm
Separate process
9 Return to OP (redirect) with id_token
10 evaluate access
11 get additional information
12 grant access (id_token + access_token)
13 render protected site
Relying Party 1->Relying Party 2 authorization
14 add connection to App 2 did:zzz
Dynamic client registration of did:ppp at did:zzz
15 register client
IAT obtained via separate process
16 OIDC Authorization request (redirect)
17 Render login options
18 Choose SSI login
19 Navigate to SSI login
20 OIDC Authorization requrest (redirect)
DIDComm Auth User
21 Authenticate user via DIDComm
Separate process
22 Return to OP (redirect) with id_token
23 evaluate access
24 get additional information
25 Obtain consent for data access
26 grant access (id_token + access_token)
27 get data
access token
Relying Party 1->Relying Party 2 access
28 verify access
29 response
30 render protected site

***Figure 5****: Cross domain DIDComm-based Front-Channel Authentication*

### 4.1.3. Functional Requirements

| *Functional Requirement* | *Protocol* | *Parallelism* |
|---|---|---|
| *[IDM.AA.00014] Credential Based Access Control (CrBac…* | Trust Service API | Multi-user<br>Multi-session |
| *[IDM.AA.00016] Standard open source IAM Package* | OIDC | Multi-user<br>Multi-session |
| *[IDM.AA.00017] OpenID Provider Configuration Information* | OIDC | |

| | | |
|---|---|---|
| *[IDM.AA.00018] OpenID Connect Implicit profile* | OIDC | Multi-user<br>Multi-session |
| *[IDM.AA.00019] OAuth 2.0 Security Best…* | OIDC | Multi-user<br>Multi-session |
| *[IDM.AA.00020] SSI Login Page* | HTTP, Trust Service API | Multi-user<br>Multi-session |
| *IDM.AA.00021] QR Code Generation* | Trust Service API | Multi-user<br>Multi-session |
| *[IDM.AA.00022] Login State Background Polling* | HTTP | Multi-user<br>Multi-session |
| *[IDM.AA.00023] Session Handling* | Trust Service API | Multi-user<br>Multi-session |

***Table 4**: Functional Requirements SSI Back Channel login*

## 4.2. IAT Issuing

### 4.2.1. Description

The feature is handled over the IAT Provider which checks in the background over policies the trust relationship before the issuing of an Initial Access Token (IAT). The IAT can be used later for a dynamic client registration as defined in [RFC7591]. The IAT can also be used for other purposes later, because this IAT is bound to a proof request. The sequence diagram below shows the basic concept of how to. The process/feature MAY be modified if there are security considerations or any other optimization. The final how to, MUST be described in the final concepts.

## 4.2.2. Stimulus/Response Sequences

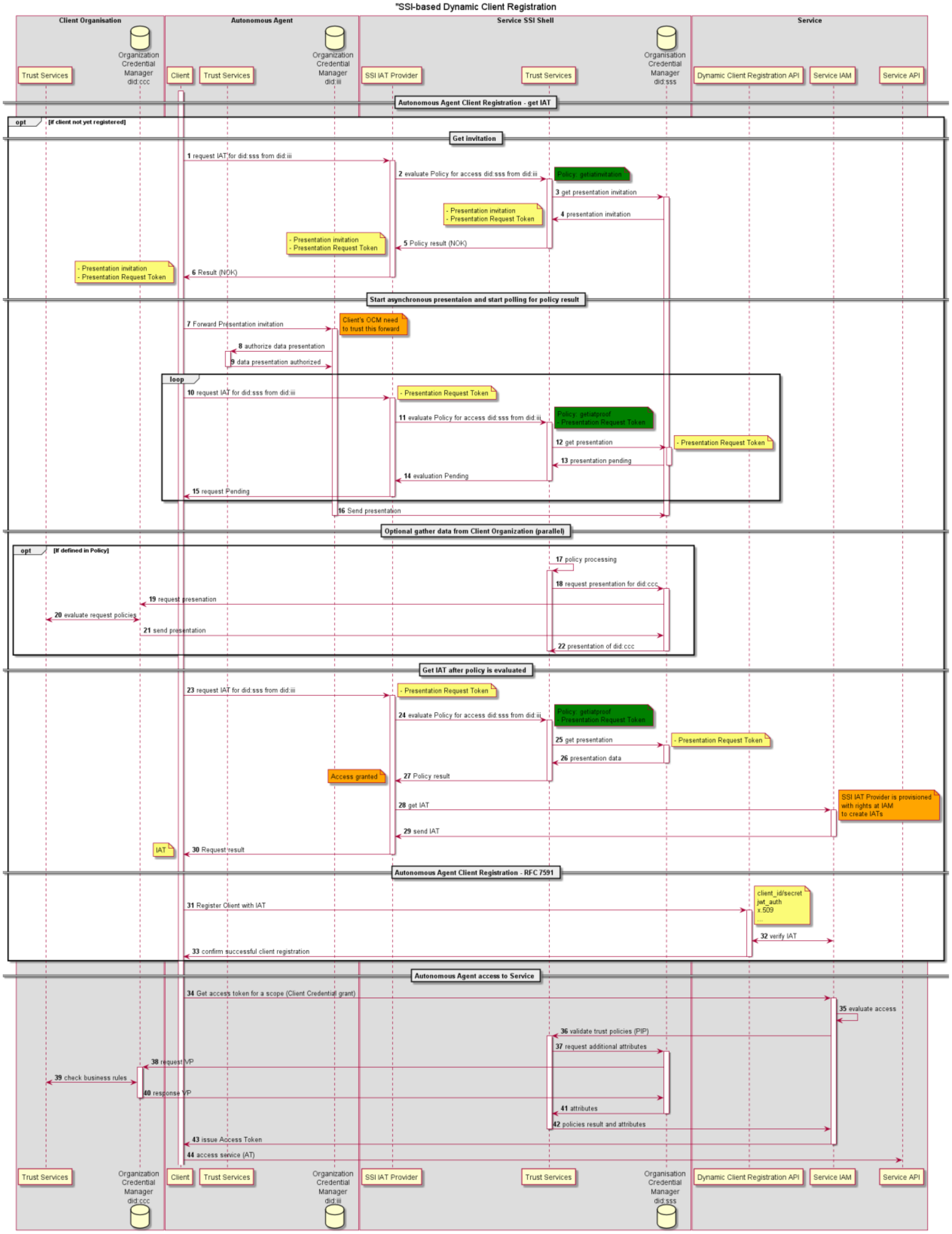

"SSI-based Dynamic Client Registration
Client Organisation
Autonomous Agent
Service SSI Shell
Service
Trust Services
Organization Credential Manager did:ccc
Client
Trust Services
Organization Credential Manager did:iii
SSI IAT Provider
Trust Services
Organisation Credential Manager did:sss
Dynamic Client Registration API
Service IAM
Service API
Autonomous Agent Client Registration - get IAT
opt [if client not yet registered]
Get invitation
1 request IAT for did:sss from did:iii
2 evaluate Policy for access did:sss from did:iii
Policy: getiatinvitation
3 get presentation invitation
4 presentation invitation
- Presentation invitation
- Presentation Request Token
5 Policy result (NOK)
- Presentation invitation
- Presentation Request Token
6 Result (NOK)
- Presentation invitation
- Presentation Request Token
Start asynchronous presentaion and start polling for policy result
7 Forward Presentation invitation
Client's OCM need to trust this forward
8 authorize data presentation
9 data presentation authorized
loop
10 request IAT for did:sss from did:iii
- Presentation Request Token
11 evaluate Policy for access did:sss from did:iii
Policy: getiatproof
- Presentation Request Token
12 get presentation
- Presentation Request Token
13 presentation pending
14 evaluation Pending
15 request Pending
16 Send presentation
Optional gather data from Client Organization (parallel)
opt [If defined in Policy]
17 policy processing
18 request presentation for did:ccc
19 request presenation
20 evaluate request policies
21 send presentation
22 presentation of did:ccc
Get IAT after policy is evaluated
23 request IAT for did:sss from did:iii
- Presentation Request Token
24 evaluate Policy for access did:sss from did:iii
Policy: getiatproof
- Presentation Request Token
25 get presentation
- Presentation Request Token
26 presentation data
27 Policy result
Access granted
28 get IAT
SSI IAT Provider is provisioned with rights at IAM to create IATs
29 send IAT
30 Request result
IAT
Autonomous Agent Client Registration - RFC 7591
31 Register Client with IAT
client_id/secret
jwt_auth
x.509
...
32 verify IAT
33 confirm successful client registration
Autonomous Agent access to Service
34 Get access token for a scope (Client Credential grant)
35 evaluate access
36 validate trust policies (PIP)
37 request additional attributes
38 request VP
39 check business rules
40 response VP
41 attributes
42 policies result and attributes
43 issue Access Token
44 access service (AT)
Trust Services
Organization Credential Manager did:ccc
Client
Trust Services
Organization Credential Manager did:iii
SSI IAT Provider
Trust Services
Organisation Credential Manager did:sss
Dynamic Client Registration API
Service IAM
Service API

***Figure 6***: *IAT Issuing*

### 4.2.3. Functional Requirements

| *Functional Requirement* | *Protocol* | *Parallelism* |
|---|---|---|
| *[IDM.AA.00014] Credential Based Access Control (CrBac...* | Trust Service API | Multi-client<br>Multi-session |
| *[IDM.AA.00025] Policy based authorization* | Trust Service API | Multi-client<br>Multi-session |
| *[IDM.AA.00026] Standard IAM Compatibility* | OAuth2 | Multi-client<br>Multi-session |
| *[IDM.AA.00027] Client Registration* | OAuth2 | Multi-client<br>Multi-session |

***Table 5****: Functional Requirements IAT Issuing*

# 5. Other Requirements

⏩ IDM.AA.00063 **Compatibility Report**

It SHOULD be tested and documented which IAM systems are compatible and support the offered solution. It MUST support some of the commonly used solutions (e.g., Microsoft ADS). ⏪

# 6. Verification

⏩ IDM.AA.00064 **Behavior Driven Design**

Verification of fulfillment of the requirements and characteristics MUST be done using automated tests which are part of the deliverables. They SHOULD be done by patterns of the Behavior Driven Development (BDD) using the “Gherkin Syntax”. ⏪

⏩ IDM.AA.00065 **Automated Test Environment**

All functionalities MUST be demonstrated in a complex test environment within a sandbox, with the following infrastructure components:

- Load Balancer, e.g., HAProxy
- API Gateway, e.g., Kong
- Service Mesh, e.g., Linkerd/Istio
- DNS
- Multiple Servers
- Firewalls

All security tests MUST be passed in this test environment automatically. ⏪

⏩ IDM.AA.00066 **Load Tests**

Scalability and Performance around the high workload scenarios MUST be demonstrated, by using any kind of Load Test Framework for HTTP APIs. e.g., Gatling[2]. ⏪

[2] https://gatling.io/

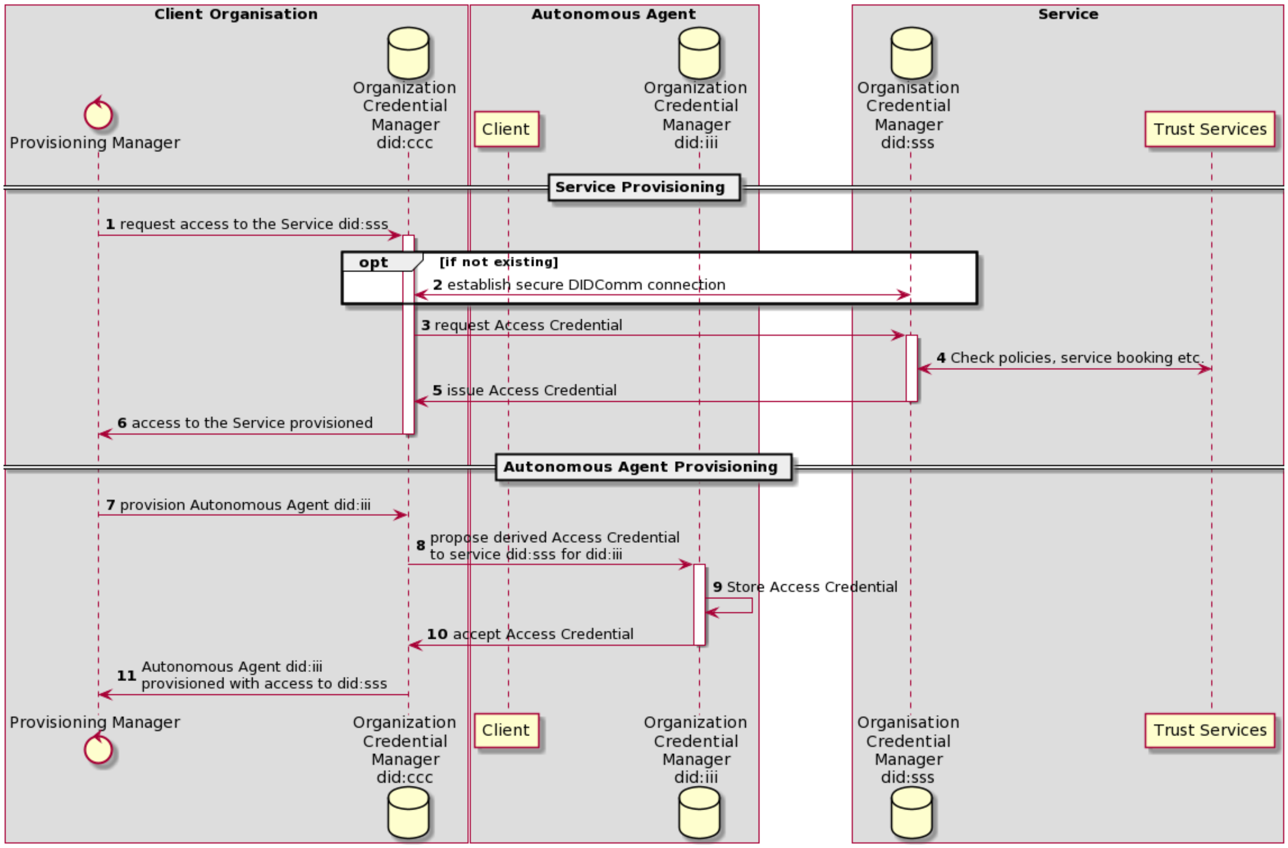
# Appendix A: Glossary

For the glossary refer to IDM.AO Glossary/Terminology [IDM.AO]

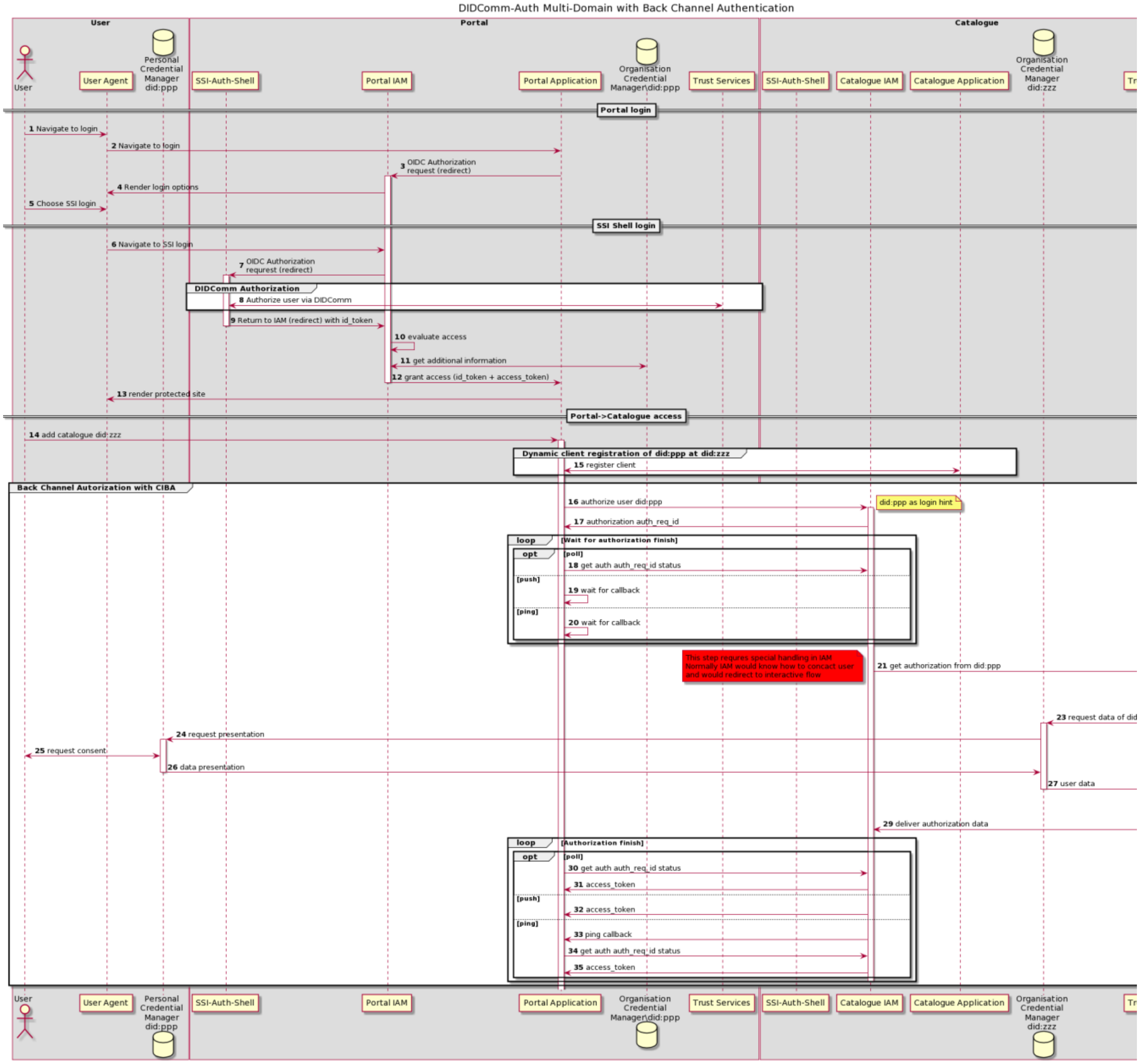
# Appendix B: Backup / Backlog

## IOT Access Credential pre-provisioning

IOT Preprovisioning with Access Credential


Client Organisation
Autonomous Agent
Service
Provisioning Manager
Organization Credential Manager did:ccc
Client
Organization Credential Manager did:iii
Organisation Credential Manager did:sss
Trust Services
Service Provisioning
1 request access to the Service did:sss
opt
[if not existing]
2 establish secure DIDComm connection
3 request Access Credential
4 Check policies, service booking etc.
5 issue Access Credential
6 access to the Service provisioned
Autonomous Agent Provisioning
7 provision Autonomous Agent did:iii
8 propose derived Access Credential to service did:sss for did:iii
9 Store Access Credential
10 accept Access Credential
11 Autonomous Agent did:iii provisioned with access to did:sss
Provisioning Manager
Organization Credential Manager did:ccc
Client
Organization Credential Manager did:iii
Organisation Credential Manager did:sss
Trust Services

## Multi-Domain User Authentication – Back Channel – Portal & Catalogue


DIDComm-Auth Multi-Domain with Back Channel Authentication
User
Portal
Catalogue
User
User Agent
Personal Credential Manager did:ppp
SSI-Auth-Shell
Portal IAM
Portal Application
Organisation Credential Manager\did:ppp
Trust Services
SSI-Auth-Shell
Catalogue IAM
Catalogue Application
Organisation Credential Manager did:zzz
Tr
Portal login
1 Navigate to login
2 Navigate to login
3 OIDC Authorization request (redirect)
4 Render login options
5 Choose SSI login
SSI Shell login
6 Navigate to SSI login
7 OIDC Authorization requrest (redirect)
DIDComm Authorization
8 Authorize user via DIDComm
9 Return to IAM (redirect) with id_token
10 evaluate access
11 get additional information
12 grant access (id_token + access_token)
13 render protected site
Portal->Catalogue access
14 add catalogue did:zzz
Dynamic client registration of did:ppp at did:zzz
15 register client
Back Channel Autorization with CIBA
16 authorize user did:ppp
did:ppp as login hint
17 authorization auth_req_id
loop
[Wait for authorization finish]
opt
[poll]
18 get auth auth_req_id status
[push]
19 wait for callback
[ping]
20 wait for callback
This step requres special handling in IAM
Normally IAM would know how to concact user
and would redirect to interactive flow
21 get authorization from did:ppp
23 request data of did
24 request presentation
25 request consent
26 data presentation
27 user data
29 deliver authorization data
loop
[Authorization finish]
opt
[poll]
30 get auth auth_req_id status
31 access_token
[push]
32 access_token
[ping]
33 ping callback
34 get auth auth_req_id status
35 access_token

# Appendix C: Overview GXFS Work Packages

The project "Gaia-X Federation Services" (GXFS) is an initiative funded by the German Federal Ministry of Economic Affairs and Energy (BMWi) to develop the first set of Gaia-X Federation Services, which form the technical basis for the operational implementation of Gaia-X.

The project is structured in five Working Groups, focusing on different functional areas as follows:

Work Package 1 (WP1): Identity & Trust

Identity &Trust covers authentication and authorization, credential management, decentral Identity management as well as the verification of analogue credentials.

Work Package 2 (WP2): Federated Catalogue

The Federated Catalogue constitutes the central repository for Gaia-X Self-Descriptions to enable the discovery and selection of Providers and their Service Offerings. The Self-Description as expression of properties and Claims of Participants and Assets represents a key element for transparency and trust in Gaia-X.

Work Package 3 (WP3): Sovereign Data Exchange

Data Sovereignty Services enable the sovereign data exchange of Participants by providing a Data Agreement Service and a Data Logging Service to enable the enforcement of Policies. Further, usage constraints for data exchange can be expressed by Provider Policies as part of the Self-Description

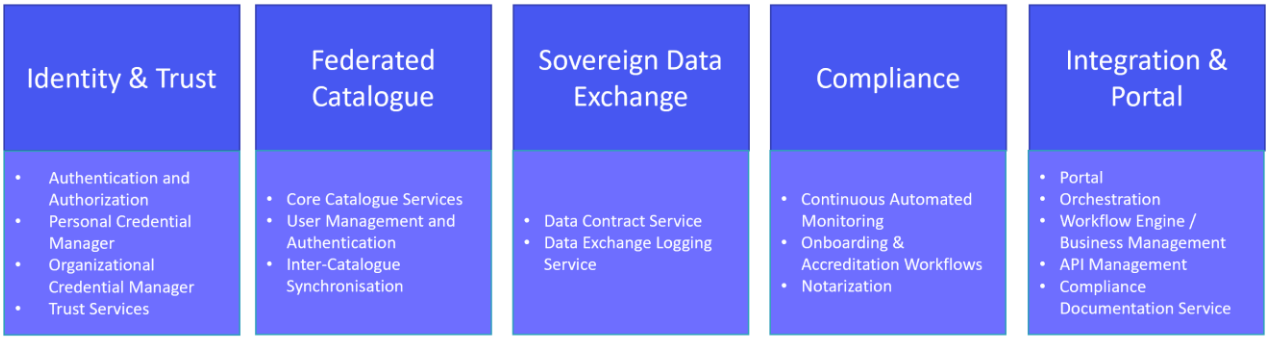
Work Package 4 (WP4): Compliance

Compliance includes mechanisms to ensure a Participant's adherence to the Policy Rules in areas such as security, privacy transparency and interoperability during onboarding and service delivery.

Work Package 5 (WP5): Portal & Integration

Gaia-X Portals and API will support onboarding and Accreditation of Participants, demonstrate service discovery, orchestration and provisioning of sample services.

All together the deliverables of the first GXFS project phase are specifications for 17 lots, that are being awarded in EU-wide tenders:



Identity & Trust

- Authentication and Authorization
- Personal Credential Manager
- Organizational Credential Manager
- Trust Services

Federated Catalogue

- Core Catalogue Services
- User Management and Authentication
- Inter-Catalogue Synchronisation

Sovereign Data Exchange

- Data Contract Service
- Data Exchange Logging Service

Compliance

- Continuous Automated Monitoring
- Onboarding & Accreditation Workflows
- Notarization

Integration & Portal

- Portal
- Orchestration
- Workflow Engine / Business Management
- API Management
- Compliance Documentation Service

Further general information on the Federation Services can be found in [TAD].