

Software Requirements Specification

for

**Gaia-X Federation Services
Personal Credential Manager
IDM.PCM**

Published by

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

Copyright

© 2021 GAIA-X European Association for Data and Cloud AISBL

This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA



Table of Contents

List of Figures	vi
List of Tables	vi
1. Introduction	1
1.1. Document Purpose	1
1.2. Product Scope	1
1.3. Definitions, Acronyms and Abbreviations	2
1.4. References	2
1.5. Document Overview	4
2. Product Overview	4
2.1. Product Perspective	4
2.2. Product Functions	5
2.2.1. PCM Form Factors	8
2.2.1.1. Smartphone Application	8
2.2.1.2. Browser-based application/addon for stationary PCs and notebooks	8
2.2.1.3. Cloud-based User Agent/Wallet	8
2.3. Product Constraints	8
2.4. User Classes and Characteristics	9
2.5. Operating Environment	9
2.6. User Documentation	10
2.7. Assumptions and Dependencies	11
2.8. Apportioning of Requirements	11
3. Requirements	11
3.1. External Interfaces	12
3.1.1. User Interfaces	12
3.1.2. Hardware Interfaces	12
3.1.3. Software Interfaces	12
3.1.4. Communications Interfaces	12
3.2. Functional	14
3.2.1. Managing Connections	14
3.2.2. Managing Credentials	16
3.2.3. Wallet Backup	18
3.2.4. Credential Wallet Importing/Exporting	20
3.2.5. End User Authentication	21

3.2.6. DIDComm Login Support	23
3.2.7. NFC Scanning (DID Input)	24
3.2.8. QR Code Scanning (DID Input)	24
3.2.9. SIOP Login	25
3.2.10. App Settings Configuration (personalization)	25
3.2.11. Ledger Selection	25
3.3. Other Nonfunctional Requirements	26
3.3.1. HTTP Requirements	26
3.3.2. Logging Requirements	26
3.3.3. Performance Requirements	27
3.3.4. Safety Requirements	27
3.3.5. Security Requirements	27
3.3.5.1. General Security Requirements	27
3.3.5.2. Service Specific Security Requirements	27
3.3.6. Software Quality Attributes	30
3.4. Compliance	30
3.5. Design and Implementation	31
3.5.1. Installation	31
3.5.2. Distribution	31
3.5.3. Usability	31
3.5.4. Maintainability	31
3.5.5. Portability	31
3.5.6. Interoperability	32
4. System Features	32
4.1. Managing Connections	32
4.2. Managing Credentials	33
4.3. Wallet Backup	34
4.4. Credential Wallet Importing/Exporting	34
4.5. End User Authentication	34
4.6. NFC Scanning (DID Input)	35
4.7. QR Code Scanning (DID Input)	35
4.8. SIOP Login	36
4.9. Notification Support	36
4.10. Ledger Selection	36
4.11. App Settings Configuration (personalization)	36

4.12. Smartphone Application	37
4.13. Browser-based application/addon for stationary PCs and notebooks	37
4.14. Cloud based User Agent/Wallet	38
5. Verification	39
Appendix A: Glossary	40
Appendix B: Overview GXFS Work Packages	40

List of Figures

Figure 1: Personal Credentials Manager. Layering overview	5
Figure 2: Personal Credentials Manager: Application Cooperation View	7

List of Tables

Table 1: References	4
Table 2: User Classes and Characteristics	9
Table 3: Apportioning of Requirements	11
Table 4: Major Requirements on cryptographic algorithms and key length	28
Table 5: Functional Requirements Connection Management	33
Table 6: Functional Requirements Credential Management	33
Table 7: Functional Requirements Wallet Backup	34
Table 8: Functional Requirements Credential Wallet Importing/Exporting	34
Table 9: Functional Requirements End User Authentication	35
Table 10: Functional Requirements NFC Scanning (DID Input)	35
Table 11: Functional Requirements QR Code Scanning (DID Input)	35
Table 12: Functional Requirements SIOP Login	36
Table 13: Functional Requirements Ledger Selection	36
Table 14: Functional Requirements App Settings Configuration (personalization)	37

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 “Personal Credential Manager” 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 identity and document verification with the purpose to provide a credential manager application to be used by natural persons to participate in the Gaia-X trust structure.

1.2. Product Scope

The purpose of these products is to provide all necessary components for the self-sovereign administration of the digital identity of a principal in the Gaia-X context. The Personal Credential Manager enables a natural person to act as a *principal* of an organization within the SSI-based Gaia-X ecosystem in a privacy-preserving, trustful and secure way. This comprises the following main functionalities:

- Establishment of trustful connections to other parties
- Reception and management of verifiable credentials from other parties (e.g., a principal credential from a Gaia-X participant)
- Presenting Verifiable Presentations to other parties in a proved manner
- Secure storage and management of respective secrets

The Personal Credential Manager must be implemented as a full-featured smartphone-based application for Android and iPhone platforms. It may further be implemented as a browser-based application/addon for stationary PCs and notebooks as well as a cloud-based user agent/wallet, where the frontend layer resides at the end user side, for example browser-based, and the Core/Wallet Layer is implemented as a cloud agent.

Furthermore, the scope includes the provision of the developed software in a usable format for end users including the respective distribution channels (such as app distribution over Google Play and Apple AppStore). Also, the necessary tools to operate and maintain the created software components in an enterprise environment with focus on high-availability, security and monitoring and logging based on common standards. Documentation for developer, operator and user MUST be written in markdown format which is public consumable over a publicly accessible source repository without access limitations.

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 also includes 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

[Aries.RFC0004]	Daniel Hardman (2019), Aries RFC 0004: Agents
	https://github.com/hyperledger/aries-rfcs/blob/master/concepts/0004-agents/README.md (Status: 03-14-2021)
[Aries.RFC0005]	Daniel Hardman (2019), Aries RFC 0005: DID Communication
	https://github.com/hyperledger/aries-rfcs/tree/master/concepts/0005-didcomm (Status: 03-17-2021)
[Aries.RFC0036]	(Community) (2021), Aries RFC 0036: Issue Credential Protocol 1.0
	https://github.com/hyperledger/aries-rfcs/blob/master/features/0036-issue-credential/README.md (Status: 03-17-2021)
[Aries.RFC0037]	(Community) (2020), Aries RFC 0037: Present Proof Protocol 1.0 - Request Presentation
	https://github.com/hyperledger/aries-rfcs/tree/master/features/0037-present-proof#request-presentation (Status: 03-17-2021)
[Aries.RFC0023]	R.West, D.Bluhm, M.Hailstone, S.Curran, S.Curren, G.Aristy (2019), Aries RFC 0023: DID Exchange Protocol 1.0
	https://github.com/hyperledger/aries-rfcs/blob/master/features/0023-did-exchange/README.md (Status 03-18-2021)
[Aries.RFC0046]	Daniel Hardman (2019), Aries RFC 0046: Mediators and Relays
	https://github.com/hyperledger/aries-rfcs/blob/master/concepts/0046-mediators-and-relays/README.md (Status 03-18-2021)
[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]	Daniel Hardman (2021), DIDComm Messaging: Out Of Band Messages
	https://identity.foundation/didcomm-messaging/spec/#out-of-band-messages (Status 03-18-2021)
[DID.Peer]	Daniel Hardman, ... (2020), Peer DID Method Specification
	https://identity.foundation/peer-did-method-spec/ (Status 03-18-2021)
[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)
[IDM.AO]	Gaia-X WP1¹ (2021), Architecture Overview
	IDM.AO (Base of functional specification)
[ISO25000]	ISO 25000 Portal (n.d.), ISO/IEC 25010
	https://iso25000.com/index.php/en/iso-25000-standards/iso-25010 (Status: 03-17-2021)
[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”
[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)

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

[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 [RFC 2119], are written in capital letters (see also [IDM.AO] - Methodology).

2. Product Overview

2.1. Product Perspective

The personal credential manager is used by a natural person. Within the Gaia-X terminology, such a natural person is named *principal*. The principal utilizes the PCM in the respective form factor to store VCs issued to her/him as well as to prove the statements necessary to obtain a service.

The PCM must support the following overall Gaia-X processes:

- Principal Onboarding [IDM.AO], Section “Principal Onboarding”]
- DIDComm Authentication as the generic way to authenticate a principal to a participant acting as a provider of a service [IDM.AO], Section “Authentication”]
- SIOP DID as a method to implement backward compatibility with existing OIDC clients and OpenID Provider [DID SIOP]

Beyond that, further generic processes must be supported, namely:

- Establishment of secure and trusted connections with other parties
- Selection of supported Ledgers (DID) and Ledger-agnostic behavior
- Reception of verifiable credentials from attesting parties (e.g., a principal credential from a Gaia-X participant)
- Proving of verifiable presentations to other parties

- Multi-factor authentication for user access to the PCM
- Secure backup and restore
- Personalization
- Accessibility of the PCM

2.2. Product Functions

Personal Credentials Manager (PCM) enables end users to interact with the DID-based ecosystem in a privacy-preserving way. PCM acts as a user representative securely holding the acquired distributed identities and identity attributes and provides the technical means to selectively disclose the aforementioned attributes for authentication and service consumption.

The following represents the high-level functional architecture of the PCM.

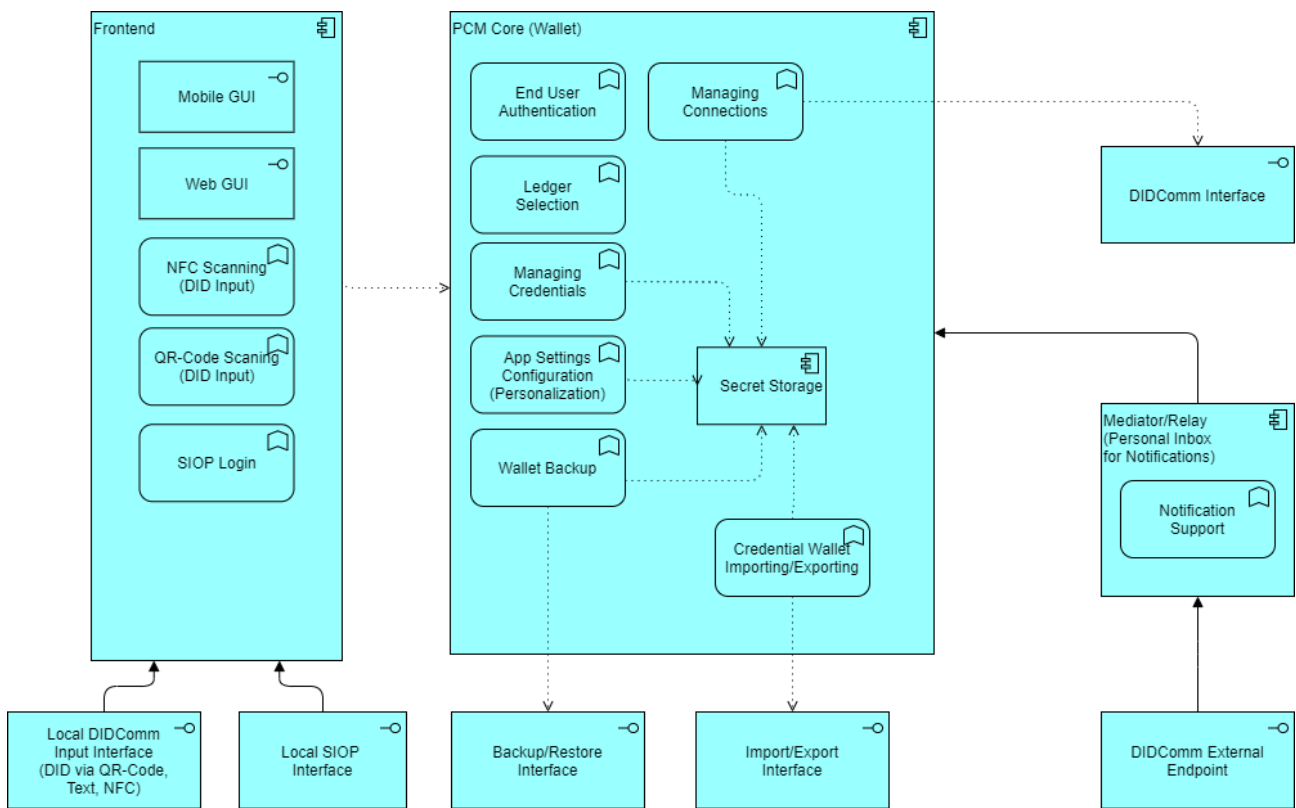


Figure 1: Personal Credentials Manager. Layering overview

As presented in the PCM big picture layering overview, PCM consists of different components which effectively comprise the following layers:

- The front-end layer
- PCM Core (Wallet) layer
- as well as the Mediator (Relay) layer

The front-end layer is comprised of the following features and components:

- End User Authentication

- Graphical User Interface (GUI)
- Local Input Interfaces

End User Authentication component provides for the implementation of secure user authentication policies which can include but are not limited to fingerprint authentication in the smart phone case, PIN, password, etc.

GUI component enables end users to interact with the PCM and use the PCM functions.

Local Input Interfaces comprise QR-code processing, bootstrapping over Near Field Communication (NFC), etc., and by that provides communication initiation, direct, or peer-to-peer (contact) exchange as well as credentials presentation ensuring the direct proximity requirements.

PCM Core (Wallet) layer consists of the following features and components:

- Managing Connections
- App Settings Configuration (Personalization)
- Managing Credentials
- Wallet Backup
- Credential Wallet Importing/Exporting
- Secret Storage

The *Managing Connections* feature establishes the connection with the communication entities outside PCM (such as a Service Provider or a Credentials Issuer) over DIDComm protocol, for example after receiving the bootstrapping request from the *Local Input Interfaces* in the *Frontend*.

App Settings Configuration abstracts the implementation of possible personalization properties that tailor the respective PCM instance to the needs and requirements of a specific end user.

The *Managing Credentials* feature provides the functionality for receiving credentials issued by other participants, enabling the user to view and inspect his/her credentials, and the basic functionality for proofing credential attributes to other participants according to the SSI paradigm. The credential manager ensures that the user is always in control, which attributes are provided to which participant. It further enables to create and check proofs according to the SSI paradigm to, for example, achieve DIDComm or SIOP Login, etc.

The *Wallet Backup* feature provides for the secure backup and restore capabilities of the obtained credentials and possibly the app settings.

The *Credential Wallet Importing/Exporting* feature allows to securely export the credentials to another PCM and to import the credentials into the current PCM. Through this, the so-called cross-wallet compatibility can be ensured in future (that is, the compatibility between different implementations of PCM that share the same standard for credentials definition and management).

Secret storage provides for secure persistent storage of the obtained credentials including the corresponding secret parts.

The **Mediator (Relay) layer** is represented by the Mediator/Relay component which effectively provides for notification management and therefore dispatches the externally received notifications for connection establishment to the concrete PCM.

PCM can be implemented according to the following form factors:

- A smartphone-based application (Frontend Layer and PCM Core Layer reside at the smartphone side)
- Browser-based application/addon for stationary PCs and notebooks (Frontend Layer and PCM Core Layer reside at the browser side)
- A cloud-based user agent/wallet (Frontend Layer reside at the end user side, for example browser-based, and the PCM Core Layer is implemented in the cloud)

Regardless of the chosen form factor, the privacy and security baseline applicable to the PCM scenario must be ensured.

The PCM functions are presented in a graphical form from the application cooperation viewpoint, see the respective figure below.

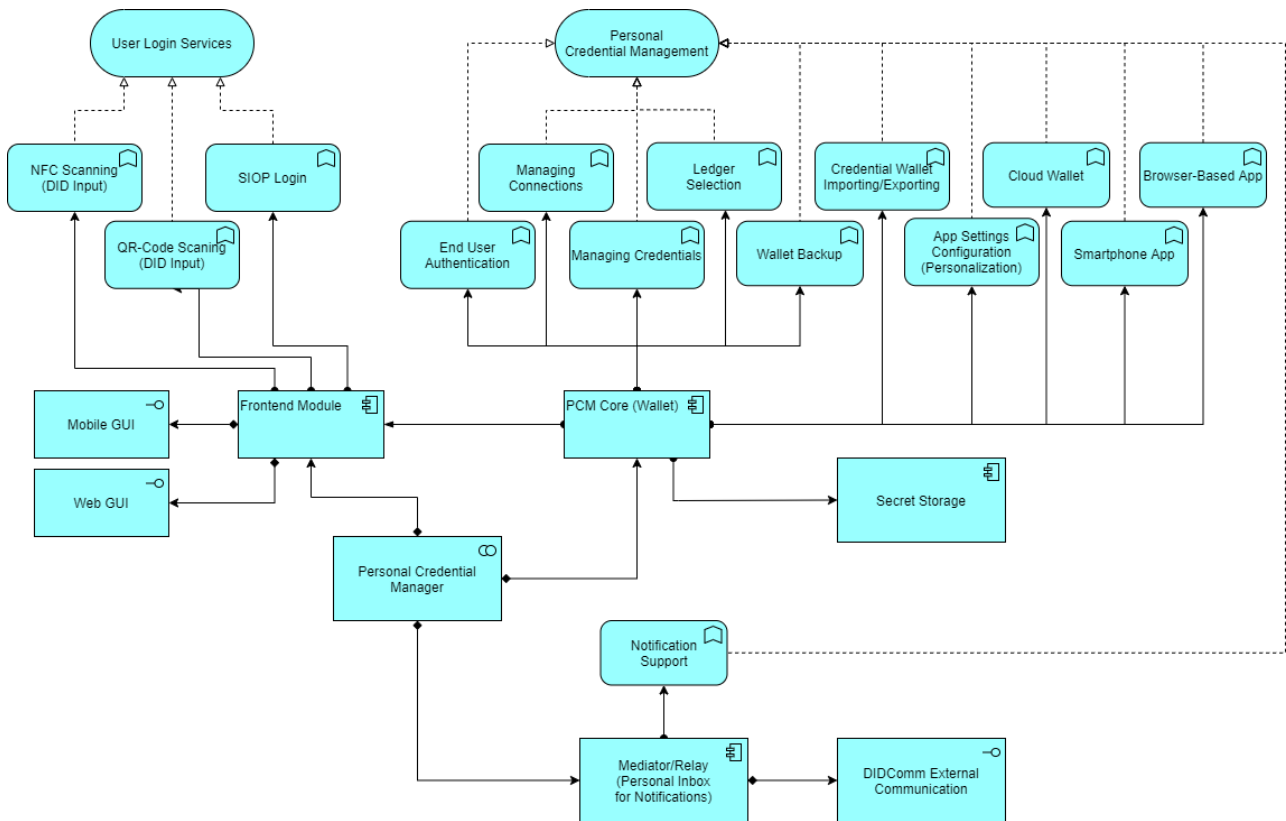


Figure 2: Personal Credentials Manager: Application Cooperation View

2.2.1. PCM Form Factors

As earlier mentioned, the PCM can be presented in different form factors that implement the necessary Gaia-X functionalities. The different form factors distribute the Gaia-X functionalities between a Smartphone App and a Cloud Environment. Depending on the maturity of the standards and the implementation grade of the components the objective is to increase the Gaia-X functionalities available in the cloud.

2.2.1.1. Smartphone Application

This form factor consists of a complex Smartphone App that implements locally the GUI functionalities, the connectivity functionalities and credential and personal wallet management. The backup/restore mechanisms and the configuration management are handled as well in the mobile Smartphone app. This alternative can benefit from all physical input and output interfaces present in a Smartphone, such as cameras for scanning QR-Codes for connection invitations or the NFC communication.

Because Smartphones do not usually have a fixed communication endpoint an SSI-Mediator needs to remain in the Cloud for PCM Notifications.

This approach poses some evident threats, such as the loss of the Smartphone (including private keys). To improve the availability and scalability of the solution and to improve the principal's flexibility to access the Gaia-X ecosystem it is planned in further project phases to move functionalities to a cloud environment.

2.2.1.2. Browser-based application/addon for stationary PCs and notebooks

This form factor implements the Frontend and Core functionalities in a browser-based application for stationary PCs and notebooks. GUI functionalities, the connectivity functionalities, credential and personal wallet management as well as backup/restore mechanisms and the configuration management are handled within the local Browser. For connection invitations, the provision of the invitation message over E-Mail or other means of communication should be supported.

Like the Smartphone App, the local user PC/notebook also does not usually have a fixed communication endpoint an SSI-Mediator needs to remain in the Cloud for PCM Notifications.

2.2.1.3. Cloud-based User Agent/Wallet

Simple Smartphone App with access to smartphone devices.

In this case, the principal connects and authenticates itself securely to an agent deployed in the cloud, from where the agent can access all the PCM functionalities. This approach offers the possibility for the users to keep their secrets protected against physical loss.

2.3. Product Constraints



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

The architecture document [\[IDM.AO\]](#) 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. ☐



IDM.PCM.00002 User restriction

The PCM MUST be designed in a way so that one PCM instance is to be used by one and only one personal user. One PCM instance MUST NOT be shared by multiple users. ☐



IDM.PCM.00003 Aries Agent compatibility

The PCM MUST comply to Aries RFC 0004: Agents [\[Aries.RFC0004\]](#). This means that the PCM acts as an Aries Agent according to this RFC. ☐

2.4. User Classes and Characteristics

<i>User Class</i>	<i>Description</i>	<i>Expertise</i>	<i>Privilege Level</i>	<i>Product Usage</i>
Personal User	The person in possession of the personal credential manager using all functionality of the product	Low	High	Frontend
Cloud Operator	In case the PCM core is implemented as a Cloud Wallet, the cloud operator naturally has some sort of access to the cloud wallet. The cloud wallet MUST be implemented in a way so that the cloud operator cannot access or use secrets stored in the wallet.	High	Operator	Backend/Cloud

Table 2: User Classes and Characteristics

2.5. Operating Environment

Please refer to [\[TDR\]](#) for further binding requirements regarding the operating environment.



IDM.PCM.00004 Browser Support

The product part for the browser extension, MUST be available for the common browsers: Firefox, Chrome, Microsoft Edge and Safari. ☐



IDM.PCM.00005 Operating Environments

The product needs to support different operating environments depending on the form factors.

Form factor “Smartphone App” (PCM Frontend and PCM Core/Wallet in one App):

- Android operating system versions which are still supported by Google (as of now > Android version 8) MUST be supported.
- iOS operating system versions which are still supported by Apple MUST be supported.
- Other platforms such as Smartwatches, other OS such as Samsung or Huawei or Windows Phone MAY be supported.

Form factor “Browser-based application/addon for stationary PCs and notebooks” (PCM Frontend and PCM Core):

- The common browsers Chrome, Firefox, Safari and Microsoft Edge MUST be supported.
- Other browsers compliant with W3C MAY be supported.

Form factor “Cloud-based agent” (Local Frontend, PCM core in the cloud):

- The Frontend MUST support the operating systems specified as mandatory for the “Smart Phone App” form factor.
- The Frontend MUST support the Browsers specified as mandatory for the “Browser-based application/addon for stationary PCs and notebooks” form factor.
- The PCM Core/Wallet MUST be runnable on the open Linux standard in the current LTS version. Additionally, it MAY be runnable within a container on any other platforms like Mac OS, Windows, BSD etc. ☐

2.6. User Documentation

Please refer to [\[TDR\]](#) for further requirements regarding documentation.



IDM.PCM.00006 **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 product from source code ☐



IDM.PCM.00007 **Participant Documentation**

The documentation MUST contain:

- Short Software Description/Usage
- Usage Guide
- GDPR Design Decisions
- Security Concept
- Operations Concept
- FAQ

- Keyword Directory 

2.7. Assumptions and Dependencies

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

2.8. Apportioning of Requirements

Feature	Priority
Managing Connections	1
Managing Credentials	1
Wallet Backup	1
End User Authentication	1
QR-Code scanning (DID Input)	1
Notification Support	1
App Settings Configuration	1
Ledger Selection	1
Smartphone Application	1
Browser-based application/addon for stationary PCs and notebooks	2
Cloud based User Agent/Wallet	2
Credential Wallet Importing/Exporting	2
NFC Scanning (DID Input)	2
SIOP Login	2

Table 3: Apportioning of Requirements

3. Requirements

Further binding requirements can be found in [\[TDR\]](#).

3.1. External Interfaces

3.1.1. User Interfaces

- ▶ IDM.PCM.00008 **Smartphone GUI**
The PCM MUST provide a GUI for the PCM user. All functions available to the PCM user MUST be available to the user via the GUI. ◀

- ▶ IDM.PCM.00009 **Web Interface for Browser-based Applications**
The PCM MUST provide a web interface for browser-based applications. All functions available to the PCM user MUST be available via this web interface. ◀

3.1.2. Hardware Interfaces

- ▶ IDM.PCM.00010 **Camera**
The PCM MUST be able to use the Smartphone's camera to scan QR-Codes. ◀

- ▶ IDM.PCM.00011 **NFC**
The PCM MAY be able to use the Smartphone's NFC communication to receive DIDComm invitation messages. ◀

- ▶ IDM.PCM.00012 **Fingerprint Scanner**
The PCM MUST be able to use the Smartphone's Fingerprint scanner to utilize the Fingerprint for user authentication. ◀

3.1.3. Software Interfaces

- ▶ IDM.PCM.00013 **Secure Storage**
The PCM implementation MUST be able to use secure storage (internal storage, encrypted storage, dedicated key storage) provided by the target platform/smartphone operating system for storing PCM data within the smartphone. ◀

3.1.4. Communications Interfaces

- ▶ IDM.PCM.00014 **DIDComm Interface**
All communication to other Gaia-X participants MUST utilize the DIDComm protocol. PCM communication to other participants via DIDComm is required for contact/connection management and for credential management.
The following protocols MUST be supported:
 1. Aries RFC 0005: DID Communication [[Aries.RFC0005](#)]: The PCM MUST implement the DIDComm protocol.

2. Aries RFC 0036: Issue Credential Protocol 1.0 [\[Aries.RFC0036\]](#): The PCM MUST support the Issue Credential Protocol in the role of the “holder”, e.g., the PCM must have the functionality of receiving a credential from other Gaia-X participants.
3. Aries RFC 0037: Present Proof Protocol 1.0 [\[Aries.RFC0037\]](#): The PCM MUST support the Present Proof Protocol implementing both roles, “verifier” and “prover”. The PCM MUST have the functionality of proofing verifiable presentations (VPs) to other Gaia-X participants, and the functionality of verifying VPs received from other Gaia-X participants. ☐

☐ **IDM.PCM.00015 Local DIDComm Interface**

The local DIDComm interface is provided for local applications (at the users Smartphone or local PC) to send DIDComm messages to be processed by the PCM. ☐

☐ **IDM.PCM.00016 External DIDComm Endpoint Interface**

The PCM MUST have an interface, where it can receive DIDComm messages also in the event the user device is offline. The PCM MUST implement a mediator/relay, which provides such an interface. The PCM MUST implement a method to process DIDComm messages received at this interface as soon as the user device is available. ☐

☐ **IDM.PCM.00017 Local SIOP Endpoint**

The PCM MAY provide an endpoint for SIOP requests [\[DID SIOP\]](#). Within the SIOP protocol, the PCM implements the role of the Self-Issued OpenID Provider (SIOP). Applications (PRs (Relying Parties) in SIOP terminology can send SIOP requests to the PCM. The PCM will process such requests and reply with a SIOP response. ☐

☐ **IDM.PCM.00018 Backup/restore interface for Personal Wallet**

The PCM MUST provide an interface for backup and restore of the PCM data (which is stored in the wallet). ☐

☐ **IDM.PCM.00019 Import/Export interface for Personal Wallet**

To provide flexibility and interoperability between Applications and PCM Form Factors, the PCM MUST provide interfaces for securely Exporting the personal wallet and other secrets preferably to a Cloud provider, as well as interfaces for securely Importing the personal wallet and other secrets and restoring and them in another Application or PCM Form Factor. ☐

☐ **IDM.PCM.00020 Personal Wallet and Secrets Synchronization**

Given that PCM users MAY for security reasons Import or Export their Personal Wallet and secrets, an interface MUST be provided so that Synchronization between Wallets can be established. ☐

3.2. Functional

3.2.1. Managing Connections



IDM.PCM.00021 Connection creation via invitation

DIDComm connection requests can be provided to the PCM user via QR-Code, Text input (URL), NFC (Out of band messages according to DIDComm Messaging specification [\[DIDComm\]](#), and by regular DIDComm messages received. The function can be activated by

1. the PCM user via the GUI. The user activates one of the different input methods for invitations (QR-Code scanning, Text input, NFC).
2. the Local DIDComm Input interface. An invitation message is received via this endpoint.
3. the DIDComm external endpoint interface. An invitation message is received via this endpoint.

Connection establishment MUST support Aries RFC 0023: DID Exchange Protocol 1.0 [\[Aries.RFC0023\]](#). The request is validated, and the information is shown to the user via the GUI. The user MUST get the possibility to accept or to reject the request. In case the request is accepted, the PCM MUST perform the DIDComm protocol required to establish the connection. The PCM MUST support PeerDID as the default DID method for the users DID, as specified in Peer DID Method Specification [\[DID.Peer\]](#) and Aries RFC 0023: DID Exchange Protocol 1.0 [\[Aries.RFC0023\]](#).

The PCM MUST request a user decision about the users DID to be used to create a connection to the provider DID. The user must be able to select from:

- a. Users DIDs used in connections already established to the provider DID (if available),
- b. other already available user DIDs (if available),
- c. a newly created DID (PeerDID).

The PCM MAY provide default settings for this decision to be pre-configured by the user.

The connection created MUST be stored in the PCM storage, so that it can be used later.

Constraints

The PCM user MUST be authenticated to use this function.

Interfaces

GUI, Local DIDComm Input interface, DIDComm interface, DIDComm external endpoint interface

Input

connection invitation.

Output

In case the user accepts the connection: The connection stored in the PCM storage.



**IDM.PCM.00022 List connections**

The PCM user can view a list of connections stored in the PCM.

Constraints

The PCM user **MUST** be authenticated to use this function.

Interfaces

GUI

**IDM.PCM.00023 Search connections**

The PCM user can search connections stored in the PCM. A full text search in all information available for the connection must be provided. The search result **MUST** be shown at the GUI.

Constraints

The PCM user **MUST** be authenticated to use this function.

Interfaces

GUI

**IDM.PCM.00024 Display connection details**

The PCM user can view detailed information about a connection. The information shown **MUST** include the DID document describing the connection contact.

Constraints

The PCM user **MUST** be authenticated to use this function.

Interfaces

GUI

**IDM.PCM.00025 Display connection communication history**

The PCM user can view detailed information about the DIDComm communication history of a given connection. The communication history **MUST** include the VC information (VC attributes) shown to the contact and the VC information (VC attributes) the contact has shown to the user, e.g., the *verifiable presentations* (VPs, [\[IDM.AO\]](#), section “Verifiable Presentation”) exchanged.

Constraints

The PCM user **MUST** be authenticated to use this function.

Interfaces

GUI

IDM.PCM.00026 **Delete connection**

The PCM user MAY delete a connection from his PCM.

Constraints

The PCM user MUST be authenticated to use this function.

Interfaces

GUI

**3.2.2. Managing Credentials**IDM.PCM.00027 **Receive a Verifiable Credential (VC)**

Within an established DIDComm connection, a VC can be issued to the user. VC issuing follows the Protocol defined in Aries RFC 0036: Issue Credential Protocol 1.0 [[Aries.RFC0036](#)]. The PCM is in the role of the “holder” within this protocol.

The function is started by a request for credential issuing (from PCM itself or other party).

The request is validated and the information is shown to the user via the GUI. The user MUST get the possibility to accept or to reject the request. The issued credential MUST be stored in the PCM storage.

Constraints

- The PCM user MUST be authenticated to use this function.
- There must already exist a connection to the issuer.

Interfaces

GUI, Local DIDComm Input interface, DIDComm interface, DIDComm external endpoint interface

Input

Credential Issuing request

Output

In case the user accepts the credential: The credential stored in the PCM storage.

IDM.PCM.00028 **Display/inspect a VC**

The PCM user can view detailed information about a VC. The information shown **MUST** include all VC data items. Showing a plaintext view of the VC **SHOULD** be supported.

Constraints

The PCM user **MUST** be authenticated to use this function.

Interfaces

GUI



IDM.PCM.00029 List VCs

The PCM user can view a list of VCs stored in the PCM.

Constraints

The PCM user **MUST** be authenticated to use this function.

Interfaces

GUI



IDM.PCM.00030 Search VCs

The PCM user can search VCs stored in the PCM. A full text search in all information available for the VCs must be provided.

Constraints

The PCM user **MUST** be authenticated to use this function.

Interfaces

GUI



IDM.PCM.00031 Answer Request for Identity Information (VP)

Within an established DIDComm connection, the PCM **MUST** implement the Aries RFC 0037: Present Proof Protocol 1.0 [[Aries.RFC0037](#)] in the role of the Prover.

The PCM **MUST** be able to receive and process Request Presentation messages received via any DIDComm interface.

The PCM **MUST** implement a method to create a Verifiable Presentation (VP) corresponding to the Request Presentation message, taking into account all VCs, which have been issued to the user (e.g., which are stored in the PCM). In case there are multiple options to fulfill the request, the PCM **MUST** require the user to select the preferred option. The PCM **MAY** implement preference settings for user selections, so that preselection's of preferred options are possible.

The user MUST be required by the PCM to give his consent for sending the presentation to the verifier.

In case the user does not agree to send the presentation, or if there is no VC data available to fulfill the request, the PCM MUST implement the option of proposing alternative presentations (preparing a Propose Presentation message). The PCM MUST implement a method to let the user select identity information to be presented from the VCs stored in the PCM.

The PCM must inform the user via the GUI of the result of the protocol run (e.g., success, or display of problem reports received.).

The event MUST be logged within the PCM, so that it can be inspected by the function “Display history of presenting identity information (VPs) to other participants”.

Constraints

- The PCM user MUST be authenticated to use this function.
- There must already exist a connection to the verifier.

Interfaces

GUI, Local DIDComm Input interface, DIDComm interface, DIDComm external endpoint interface

Acceptance Criteria

If the user has given his consent, a VP has been proved to the verifier. If the presentation has not been completed successfully, problems have been reported to the user and are logged in the PCM.



IDM.PCM.00032 Display history of presenting identity information (VPs) to other participants

The PCM user can view detailed information about the history of showing/proving identity information to other participants. For each run of the Present Proof protocol, the information shown MUST include all information contained in the VP shown to a verifier, to which verifier it has been shown, and transaction date/time.

Constraints

The PCM user MUST be authenticated to use this function.

Interfaces

GUI



3.2.3. Wallet Backup



IDM.PCM.00033 Create Backup

The PCM MUST provide a function to backup all information stored in the PCM. The backup file MUST be confidentiality and integrity protected. Protection MUST follow current standards regarding protocols and cryptographic artifacts (refer to section “Security Requirements”).

The PCM MUST provide a method for protecting access to the data within the backup with at least two authentication factors.

The Backup/Restore format MUST be compatible to guarantee PCM App interoperability between form factors and between different providers or Smartphone Applications.

Constraints

The PCM user MUST be authenticated to use this function.

Interfaces

GUI, Backup/Restore interface for Personal Wallet

Input

Security Tokens for protecting the backup

Output

protected backup data

Acceptance Criteria

Backup has been created. The backup can only be restored on provision of the correct authentication attributes.



IDM.PCM.00034 Restore Backup

The PCM MUST provide a function to restore a backup containing all information stored in the PCM. The Backup/Restore format MUST be compatible to guarantee PCM App interoperability between form factors and between different providers or Smartphone Applications.

Constraints

none.

Interfaces

GUI, Backup/Restore interface for Personal Wallet

Input

Security Tokens, Backup file

Output

success indication.

Acceptance Criteria

Backup has been restored. The backup can only be restored on provision of the correct authentication attributes.



3.2.4. Credential Wallet Importing/Exporting



IDM.PCM.00035 **Export Data**

The *Credential Wallet Importing/Exporting* feature allows to securely export the credentials to another PCM and to import the credentials into the current PCM. Through this, the so-called cross-wallet compatibility can be ensured in future (that is, the compatibility between different implementations of PCM that share the same standard for credentials definition and management). To ensure such interoperability, the PCM **MUST** implement a procedure to export personal wallet data and secret information. Exported data **MUST** be preferably stored in a Cloud environment although other types of secure local storage implementations **MAY** be also used, as well. Secure mechanisms for exporting the PCM Secrets **MUST** be provided.

Constraints

The PCM user **MUST** be authenticated to the PCM to use this function.

Interfaces

Import/Export interface for Personal Wallet

Acceptance Criteria

Exported data can be successfully imported again.



IDM.PCM.00036 **Import Data**

The *Credential Wallet Importing/Exporting* feature allows to securely export the credentials to another PCM and to import the credentials into the current PCM. Through this, the so-called cross-wallet compatibility can be ensured in future (that is, the compatibility between different implementations of PCM that share the same standard for credentials definition and management). To ensure such interoperability, the PCM **MUST** implement a procedure to import personal wallet data and secret information. Secure methods for Importing the PCM Secrets **MUST** be provided.

Constraints

none

Interfaces

Import/Export interface for Personal Wallet

Input

personal wallet data and secret information to be imported

Output

Success indication

Acceptance Criteria

Exported data can be successfully imported again.

**IDM.PCM.00037 Sync Wallets**

The PCM MAY provide a functionality to synchronize different personal wallets, e.g., synchronize a cloud wallet with a smartphone wallet. Secure methods for Importing the PCM Secrets MUST be provided.

Constraints

The user must authorize the synchronization at both wallets.

Interfaces

Personal Wallet and Secrets Synchronization

Input

sync data

Output

sync data

Acceptance Criteria

At least two wallets of a user can be synchronized successfully.

**3.2.5. End User Authentication****IDM.PCM.00038 Initial user creation**

The PCM MUST provide a function for the initial user creation, when the PCM is initialized on first use. Additionally, this function also initializes the wallet.

The PCM MUST provide a method for securing user login with at least two authentication factors.

Constraints

none.

Interfaces

GUI

Input

Login credentials

Output

success indication.

Acceptance Criteria

PCM user and wallet has been created and protected, so that only the allowed entity (user) can access the PCM.

**IDM.PCM.00039 User Authentication**

The PCM MUST provide a function which performs user authentication to open the PCM for the user.

Constraints

none.

Interfaces

GUI

Input

Login credentials

Output

success indication.

Acceptance Criteria

The protected PCM functions can be used by the user only if authentication was successful.

**IDM.PCM.00040 Configure login credentials**

The PCM MUST provide a function, where the user can modify his/her login credentials (e.g., change password).

Constraints

The PCM user MUST be authenticated to use this function.

Interfaces

GUI

Input

New login credentials.

Output

success indication.

Acceptance Criteria

Login credentials have been changed. The user can login with the new login credentials only.



IDM.PCM.00041 Secure Restore of authentication credentials

The PCM MUST provide a function for securely restoring the users' authentication credentials (e.g., forgotten password).

Constraints

The user must be authenticated in a sufficient way to restore authentication credentials according to a defined restoration process. The process must be documented in the security concept.

Interfaces

GUI

Input

user authentication

Output

restored authentication credentials

Acceptance Criteria

User can restore authentication credentials.



3.2.6. DIDComm Login Support



IDM.PCM.00042: DIDComm Login Support

Description

DIDComm login support follows the flow specification in [[IDM.AO](#)], Section 3.2.8 "Authentication" in the role of the Principal.

The PCM-related steps are as follows:

1. PCM receives a provider DID (e.g., of a service), which is treated as an implicit invitation according to the Aries RFC 0023: DID Exchange Protocol [[Aries.RFC0023](#)].
2. the PCM establishes a DIDComm connection to the provider DID using the PCM function "Connection creation via invitation". This function also provides privacy preserving connection establishment under user control.

3. The provider sends a decentralized login request to the user, which MUST be implemented as a Request Presentation message according to the Aries RFC 0037: Present Proof Protocol 1.0 [[Aries.RFC0037](#)].
4. The PCM executes the function “Answer Request for Identity Information (VP)” to verifiably present the information requested by the provider.

Further steps depicted in [[IDM.AO](#)], Section 3.2.8 “Authentication”] do not involve the PCM. An application using DIDComm login will receive the access secret from the provider in its own communication channel after the proof was presented by the PCM via DIDComm communication.

Constraints

The PCM user MUST be authenticated to use this function.

Interfaces

GUI, Local DIDComm Input interface, DIDComm interface, DIDComm external endpoint interface

Acceptance Criteria

DIDComm login can be demonstrated for a Gaia-X service.



3.2.7. NFC Scanning (DID Input)



IDM.PCM.00043 Scan NFC

The PCM MAY support scanning of DIDComm messages via NFC and processing these messages. (Out of band messages according to *DIDComm Messaging* specification [[DIDComm](#)])

Constraints

The PCM user MUST be authenticated to use this function.

Interfaces

NFC



3.2.8. QR Code Scanning (DID Input)



IDM.PCM.00044 Scan QR-Code

The PCM MUST support reading DIDComm messages via QR-Code and processing these messages. (Out of band messages according to *DIDComm Messaging* specification [[DIDComm](#)])

Constraints

The PCM user MUST be authenticated to use this function.

Interfaces

Camera

Input

QR-Code



3.2.9. SIOP Login



IDM.PCM.00045: **SIOP Login Support**

The PCM MAY implement a function to process SIOP requests [\[DID SIOP\]](#) in the role of the Self-Issued OpenID Provider (SIOP). Applications (RPs (Relying Parties) in SIOP terminology) can send SIOP requests to the PCM. The PCM will process such requests and reply with a SIOP response.

Constraints

The PCM user MUST be authenticated to use this function.

Interfaces

SIOP Endpoint

Input

SIOP request

Output

SIOP response



3.2.10. App Settings Configuration (personalization)



IDM.PCM.00046: **Configure Application Preferences**

The PCM MUST enable the user to configure his/her application preferences. Application preferences MUST include language settings.

Constraints

The PCM user MUST be authenticated to use this function.

Interfaces

GUI



3.2.11. Ledger Selection



IDM.PCM.00047: **Select Ledger**

The PCM MUST enable the user to select from a list of compatible Ledgers.

Constraints

The PCM user MUST be authenticated to use this function.

Interfaces

GUI




3.3. Other Nonfunctional Requirements

3.3.1. HTTP Requirements




IDM.PCM.00048 **HTTPS**

All HTTP communication MUST be protected by state-of-the-art transport security algorithms such as TLS 1.2 / TLS 1.3 (all protocol version numbers may be superseded by upcoming standards). Each endpoint of the product MUST support TLS certificates which are configurable by the administrator of the system. 



IDM.PCM.00049 **HTTP Protocol Definitions**

All HTTP Endpoints MUST follow RFC 7231² and RFC 5789³, but it MAY be chosen what of the protocols is necessary to realize the functionality. For problem reports the RFC7807⁴ MUST be used in combination with Standard HTTP Error Codes. 

3.3.2. Logging Requirements



IDM.PCM.00050 **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.



² <https://tools.ietf.org/html/rfc7231>

³ <https://tools.ietf.org/html/rfc5789>

⁴ <https://tools.ietf.org/html/rfc7807>

3.3.3. Performance Requirements

Not applicable.

3.3.4. Safety Requirements

- ▶ IDM.PCM.00051 **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.PCM.00052 **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.PCM.00053 **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. ◀

3.3.5. Security Requirements

3.3.5.1. 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.

3.3.5.2. Service Specific Security Requirements

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

- ▶ IDM.PCM.00054: **Secure user authentication**
To ensure that only allowed entities can access the PCM authentication methods **MUST** be implemented to grant access to the PCM. The PCM **MUST** provide a method for securing user login with at least two authentication factors. ◀
- ▶ IDM.PCM.00055: **Multimodal biometric authentication**
The PCM **MAY** provide (multimodal) biometric authentication methods to improve the usability of the PCM. ◀
- ▶ IDM.PCM.00056: **Protection of Secrets (Wallet) and Security for the Restore process**

The PCM secrets MUST be stored and processed securely. There MUST be additional security procedures in place to guarantee that the secret key can be recovered when the holder requires it, even in case the holder himself has lost access to his unlock key. State of the art methods that MAY be applied are for example Shamir's Secret Sharing. ☐

- ☐ IDM.PCM.00057: **Secure communication between frontend and cloud agent/wallet**
The communication interface in case of the cloud agent/Wallet form factor must be protected according to the latest security standards. ☐

- ☐ IDM.PCM.00058 **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 similar⁵ [[CryptoLen](#)]. The recommendations can be found in the technical guidelines⁶ TR 02102-1 [[TR02102-1](#)] and TR 02102-2 [[TR02102-2](#)] or SOG-IS Agreed Cryptographic Mechanisms⁷ [[SOG-IS](#)]. ☐

- ☐ IDM.PCM.00059 **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 \geq 256 Bit or better
RSA-PSS (recommended) 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 \geq 256 Bit or better
DSA	Min. 3000 Bit prime p 250 Bit key q	SHA-2 with an output length \geq 256 Bit or better

Table 4: Major Requirements on cryptographic algorithms and key length

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

- ☐ IDM.PCM.00060 **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.

⁵ See <https://www.keylength.com/en> for a comparison

⁶ See https://www.bsi.bund.de/EN/Service-Navi/Publications/TechnicalGuidelines/tr02102/tr02102_node.html

⁷ See <https://www.sogis.eu/documents/cc/crypto/SOGIS-Agreed-Cryptographic-Mechanisms-1.2.pdf>

Certificate owners MUST ensure that valid certificates are renewed and replaced before their expiration to prevent service outages. ☐

☐ IDM.PCM.00061 **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.PCM.00062 **Installation of Critical Security Updates**

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

☐ IDM.PCM.00063 **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.PCM.00064 **Pentesting**

All parts of the product have to be pentested, at least for the following criteria:

1. Unauthorized Access to the System MUST be tested
2. Unauthorized Actions MUST be triggered without a user action
3. All interfaces MUST be tested

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

☐ IDM.PCM.00065 **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.PCM.00066 **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.PCM.00067 **Support for Potential Requirements for Secret Storages**
Devices that hold cryptographic information and perform cryptographic functions MUST be compliant with standard PKCS #11 or other comparable cryptography standards. 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 (Source: https://en.wikipedia.org/wiki/FIPS_140-2).
- ▶ IDM.PCM.00068 **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.

3.3.6. Software Quality Attributes

- ▶ IDM.PCM.00069 **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 packages⁸
 - 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. ◀◀

3.4. Compliance

- ▶ IDM.PCM.00070 **GDPR Audit Logging**
All GDPR relevant access to personal relevant data MUST be logged for a later audit. ◀◀
- ▶ IDM.PCM.00071 **GDPR Data Processing**

⁸ Please refer to appendix B for an overview and explanation of the Work Packages (WP).

If it is 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.PCM.00072 **Wallet Installation**
PCM products or product parts for smartphones must be made available via the common smartphone app stores. ☐
- ☐ IDM.PCM.00073 **Software Updates**
For the PCM product regular as well as event-driven updates MUST be provided to fix security issues and to maintain platform compatibility. ☐

3.5.2. Distribution

There are no dedicated distribution requirements for the PCM.

3.5.3. Usability

- ☐ IDM.PCM.00074 **GUI usability**
GUI design MUST comply with common GUI recommendations for the target platforms. ☐
- ☐ IDM.PCM.00075 **PCM accessibility**
The product must comply with the accessibility requirements depending on the target platforms. ☐
- ☐ IDM.PCM.00076 **Internationalization Support**
The PCM MUST support internationalization. At least the following languages MUST be supported: English. ☐

3.5.4. Maintainability

- ☐ IDM.PCM.00077 **Continuous Integration**
All tests MUST be coded in a continuous tool to ensure the software quality in a further development. All the necessary scripts and setups MUST be provided on the public code repository to make it possible for everyone to compile and execute the product. ☐

3.5.5. Portability

- ▶▶ IDM.PCM.00078 **App Portability**
 The product **MUST** be portable to different devices, e.g., tablets. This includes as well lower end devices (Moto G3, Pixel 4a etc.) to support non-discriminatory all users without consideration of the purchasing power. ◀◀

3.5.6. Interoperability

- ▶▶ IDM.PCM.00079 **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. Managing Connections

Using this product, the user shall be enabled to manage his Gaia-X connections. Technically, connections are represented by DID-based connections to other Gaia-X participants. Connection data includes the contact DID, DID Document, DIDComm connection status data, and communication history (e.g., VPs exchanged).

Via the PCM, the user must be able to establish DIDComm connections based on invitations, which can be input to the PCM by scanning QR-Codes, Text input (URL), NFC, and by regular DIDComm Messages.

The following functions are required for connection management:

<i>Functional Requirement</i>	
<i>Functions</i>	
▶▶	IDM.PCM.00021 Connection creation via invitation
▶▶	IDM.PCM.00022 List connections
▶▶	IDM.PCM.00023 Search connections

▶▶ IDM.PCM.00024 Display connection details
▶▶ IDM.PCM.00025 Display connection communication history
▶▶ IDM.PCM.00026 Delete connection

Table 5: Functional Requirements Connection Management

4.2. Managing Credentials

The product shall enable the user to manage his verifiable credentials (VCs). Other Gaia-X participants can issue VCs to the user in possession of the personal credential manager. The user must be enabled to inspect his VCs and to show/proof VC information via verifiable presentations (VPs) to other Gaia-X participants.

Within the Gaia-X environment, persons in the role of Gaia-X principals need to be able to receive a VC onboarding them as a principal to an organization. Within the PCM, the function “receive a VC” can be used for this purpose.

The following functions are required for management of VCs:

<i>Functional Requirement</i>
<i>Functions</i>
▶▶ IDM.PCM.00027 Receive a Verifiable Credential (VC)
▶▶ IDM.PCM.00028 Display/inspect a VC
▶▶ IDM.PCM.00029 List VCs
▶▶ IDM.PCM.00030 Search VCs
▶▶ IDM.PCM.00031 Answer Request for Identity Information (VP)
▶▶ IDM.PCM.00032 Display history of presenting identity information (VPs) to other participants

Table 6: Functional Requirements Credential Management

4.3. Wallet Backup

The product must provide the functionality to create backups of the information stored within the PCM. Backups must be stored in a secure way, so that only the PCM user, who created the backup, can restore the backup. Backups must contain the full status of the PCM.

The following functions are required for backup:



<i>Functional Requirement</i>
<i>Functions</i>
 IDM.PCM.00033 Create Backup
 IDM.PCM.00034 Restore Backup

Table 7: Functional Requirements Wallet Backup

4.4. Credential Wallet Importing/Exporting

To ensure interoperability between providers, applications and form factors, the PCM must implement a procedure to export and import personal wallet data and secret information.

The following functions are required for this feature:

<i>Functional Requirement</i>
<i>Functions</i>
 IDM.PCM.00035 Export Data
 IDM.PCM.00036 Import Data
 IDM.PCM.00037 Sync Wallets

Table 8: Functional Requirements Credential Wallet Importing/Exporting

4.5. End User Authentication

The product must ensure that only the intended user can use his PCM. the product must require secure user authentication. The user must be enabled to configure authentication methods and artifacts.

The following functions are required for user management and authentication:





<i>Functional Requirement</i>
<i>Functions</i>
 <u>IDM.PCM.00038 Initial user creation</u>
 <u>IDM.PCM.00039 User Authentication</u>
 <u>IDM.PCM.00040 Configure login credentials</u>
 <u>IDM.PCM.00041 Secure Restore of authentication credentials</u>

Table 9: Functional Requirements End User Authentication

4.6. NFC Scanning (DID Input)

The product must be able to scan DIDComm messages via NFC, to support DIDComm login.



<i>Functional Requirement</i>
<i>Functions</i>
 <u>IDM.PCM.00043 Scan NFC</u>
 <u>IDM.PCM.00042: DIDComm Login Support</u>

Table 10: Functional Requirements NFC Scanning (DID Input)

4.7. QR Code Scanning (DID Input)

The product must be able to read DIDComm messages via QR-Code, to support DIDComm login.



<i>Functional Requirement</i>
<i>Functions</i>
 <u>IDM.PCM.00044 Scan QR-Code</u>
 <u>IDM.PCM.00042: DIDComm Login Support</u>

Table 11: Functional Requirements QR Code Scanning (DID Input)

4.8. SIOP Login

The product must provide support for applications to login into services via the SIOP protocol.

The following functions are required for this:




<i>Functional Requirement</i>
<i>Functions</i>
 IDM.PCM.00045: SIOP Login Support

Table 12: Functional Requirements SIOP Login

4.9. Notification Support

-  **IDM.PCM.00092 Notification Support**
 The product MUST implement a “Mediator/Relay (Personal Inbox for Notifications)” component which notifies the PCM Core/Wallet component in the shortest time possible upon events (successful credential verification, dispute) in a privacy preserving way. This is of special interest in case of the form factors “Smartphone Application” and “Browser-based application/addon for stationary PCs and notebooks” due to its mobile nature will lack a predefined fixed endpoint for receiving notifications from other Agents. The component MUST further support privacy preserving agent-to-agent messaging [\[Aries.RFC0046\]](#) 

4.10. Ledger Selection

-  **IDM.PCM.00093 Ledger Support (DID) and Ledger-agnostic behavior**
 The product MUST support multiple Ledgers according to the Architecture Overview [\[IDM.AO\]](#) , e.g., it MUST NOT be bound to a dedicated Ledger by design. 

<i>Functional Requirement</i>
<i>Functions</i>
 IDM.PCM.00047: Select Ledger

Table 13: Functional Requirements Ledger Selection

4.11. App Settings Configuration (personalization)

The product must provide means to the PCM user to configure and save PCM application preferences.

The following functions are required for this:


<i>Functional Requirement</i>
<i>Functions</i>
 IDM.PCM.00046: Configure Application Preferences

Table 14: Functional Requirements App Settings Configuration (personalization)

4.12. Smartphone Application

IDM.PCM.00094 **Smartphone Application**

The product **MUST** implement the form factor “Smartphone Application”, so that the PCM can be used as a full-featured app that implements the GUI functionalities, the connectivity functionalities and credential and personal wallet management locally on the smartphone. The backup/restore mechanisms and the configuration management are handled as well in the mobile Smartphone app. This alternative can benefit from all physical input and output interfaces present in a Smartphone, such as cameras for scanning QR-Codes for connection invitations or the NFC communication. Because Smartphones do not usually have a fixed communication endpoint an SSI-Mediator needs to remain in the Cloud for PCM Notifications.

The smartphone application **MUST** include the following system features:

- Managing Connections
- Managing Credentials
- Wallet Backup
- End User Authentication
- QR-Code scanning (DID Input)
- Notification Support
- App Settings Configuration
- Ledger Selection

The smartphone application **MAY** additionally include the following system features:

- Credential Wallet Importing/Exporting
- NFC Scanning (DID Input)
- SIOP Login



4.13. Browser-based application/addon for stationary PCs and notebooks

IDM.PCM.00095 **Browser-based application/addon for stationary PCs and notebooks**

The product **MUST** implement the form factor “Browser-based application/addon for stationary PCs and notebooks”, so that the PCM can be used as a full-featured browser-based application that implements the GUI functionalities, the connectivity functionalities and credential and personal

wallet management locally on the user's PC/notebook. The backup/restore mechanisms and the configuration management are handled as well locally on the user's PC/notebook.

Because PCs/Notebooks do not usually have a fixed communication endpoint an SSI-Mediator needs to remain in the Cloud for PCM Notifications.

The Browser-based application/addon for stationary PCs and notebooks MUST include the following system features:

- Managing Connections
- Managing Credentials
- Wallet Backup
- End User Authentication
- Notification Support
- App Settings Configuration
- Ledger Selection

The Browser-based application/addon for stationary PCs and notebooks MAY additionally include the following system features:

- Credential Wallet Importing/Exporting
- NFC Reading (DID Input)
- QR-Code scanning (DID Input)
- SLOP Login



4.14. Cloud based User Agent/Wallet

The product must provide an implementation of a cloud wallet.



IDM.PCM.00096 **Cloud based User Agent/Wallet**

The product MUST implement the form factor “cloud-based user agent/wallet”.

In this form factor, the PCM Core (Wallet) is implemented as a cloud application. This application implements connection, credential and personal wallet management, backup/restore mechanisms, user authentication and personal configuration mechanisms.

The Frontend is implemented as a Smartphone App and/or Web frontend which resides at the user's device (Smartphone or PC/Notebook). The Frontend implements the GUI and local interfaces for QR-Code scanning, NFC, etc., depending on the device/platform capabilities.

As the cloud application of the PCM Core/Wallet has a fixed communication endpoint the SSI-Mediator functionality for PCM Notifications may be included in that application.

The PCM in this form factor MUST include the following system features:

- Managing Connections
- Managing Credentials
- Wallet Backup
- End User Authentication

- Notification Support
- App Settings Configuration
- Ledger Selection
- QR-Code scanning (DID Input)

The PCM in this form factor MAY additionally include the following system features:


- Credential Wallet Importing/Exporting
- NFC Scanning (DID Input)
- SIOP Login



5. Verification



IDM.PCM.00097 **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 \(BDD\) \[BDD\]](#) using the “Gherkin Syntax”. 

Appendix A: Glossary

For the glossary refer to IDM.AO Glossary/Terminology [\[IDM.AO\]](#)

Appendix B: 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

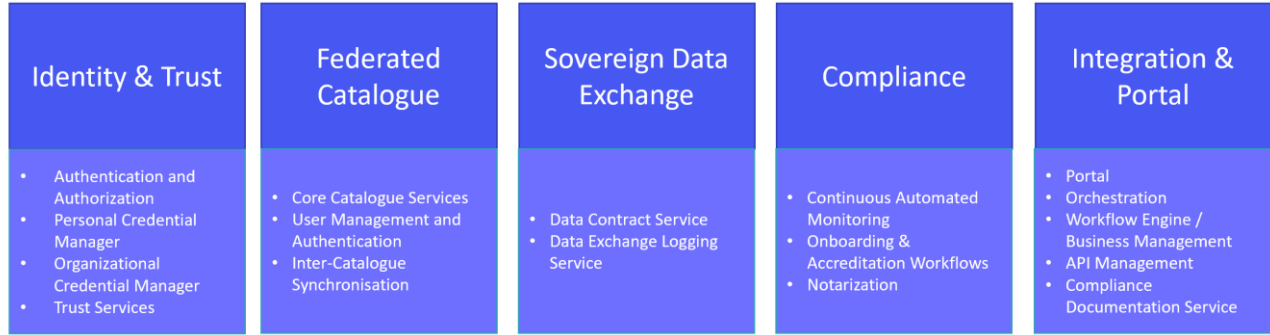
Work Package 4 (WP4): Compliance

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:



Further general information on the Federation Services can be found in [\[TAD\]](#).