Software Requirements Specification

for

Gaia-X Federation Services

Compliance

Notarization API

CP.NOTAR

Published by

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

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].

1.1. Document Purpose

The purpose of the document is to specify the requirements of the Compliance subcomponent “Notarization API” with the intention of an European wide public tender for implementing this software. Main audience for this document are 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 digital support for existing certification bodies within Gaia-X.

1.2. Product Scope

The purpose of this product is to provide an authorization officer a software component to attest given master data and transform it to a digital verifiable credential representation. These made tamper-proof digital claims about certain attributes are central to gain the desired trust in any provided self-descriptions of assets and participants in the distributed Gaia-X ecosystem. Examples of verification and digital attestation are:

- provided classic certificates of any 3rd party certifier
- Gaia-X participants and associated master data (e.g., address, name, tax identification number etc.)
- ownership of the given organization DID - relates it to the real verified organization
- the business owner (e.g., by eID)
- Organizations acting as trust anchor. E.g., Governments, Gaia-X AISBL, etc.

The product must include interfaces (API’s) to integrate the notarization component smoothly in external software for Non-IT operator usage (e.g., lawyers, notaries, governments, certifiers ...).

The scope also includes necessary tools (e.g., Command Line Scripts) 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.

1.3. Definitions, Acronyms and Abbreviations

Please refer to [IDM.AO] for Terminology/Glossary.

1.4. References

[Aries.RFC0036]	**Nikita Khateev (2019), Aries RFC 0036: Issue Credential Protocol 1.0**
	https://github.com/hyperledger/aries-rfcs/tree/master/features/0036-issue-credential (Status: 02-22-2021)
[Aries.RFC0037]	**Nikita Khateev (2019), Aries RFC 0037: Present Proof Protocol 1.0**
	https://github.com/hyperledger/aries-rfcs/tree/master/features/0037-present-proof (Status: 02-22-2021)
[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.RFC0116]	**Dan Gisolfi (2019), Aries RFC 0116: Evidence Exchange Protocol 0.9**
	https://github.com/hyperledger/aries-rfcs/tree/master/features/0116-evidence-exchange (Status: 03-24-2021)
[BDD]	**Specflow (n.D.), Getting Started with Behavior Driven Development**
	https://specflow.org/bdd/ (Status 03-18-2021)
[CEF.Notar]	**CEF Digital (n.d.), Notarisation**
	https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITALEBSI/Notarisation (Status: 02-22-2021)
[CryptoLen]	**Damien Giry, Prof. Jean-Jacques Quisquater (2020), Cryptographic Key Length Recommendation**
	https://www.keylength.com/en (Status 03-18-2021)
[CP.OAW]	**Gaia-X Federation Services Compliance – Onboarding & Accreditation Workflows**
	Please refer to annex “SRS_GXFS_CP_OAW”
[eIDAS.Bridge]	**(unknown) (n.D.), eIDAS Bridge Library**
	https://github.com/validatedid/eidas-bridge (Status 02-26-2021)
[ESSIF]	**European Commission (n.d.), European Self-Sovereign Identity Framework (ESSIF)**

	https://ec.europa.eu/cefdigital/wiki/pages/viewpage.action?pageId=262505734 (Status: 02-18-2021)
[ESSIF.DID]	**European Commission (n.d.), ESSIF DID Modelling**
	https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITALEBSI/Technical+Specification+%282%29+-+DID+Modelling (Status 02-26-2021)
[ESSIF.Bridge]	**European Commission (n.d.), ESSIF eIDAS bridge for VC-eSealing**
	https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITALEBSI/Technical+Specification+%2815%29+-+eIDAS+bridge+for+VC-eSealing (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)
[OpenContainer]	**The Linux Foundation® (2020), Open Container Initiative**
	https://opencontainers.org/ (Status: 02-22-2021)
[GXFS_SPBD]	**Gaia-X Federation Services Non-functional Requirements Security and Privacy by Design**
	Please refer to annex "GXFS_Nonfunctional_Requirements_SPBD"
[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: 03-17-2021)
[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"
[RFC7231]	**Internet Engineering Task Force (IETF) (2014), Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content**

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

	https://tools.ietf.org/html/rfc7231 (Status 02-25-2021)
[RFC5789]	**Internet Engineering Task Force (IETF) (2010), PATCH Method for HTTP**
	https://tools.ietf.org/html/rfc5789 (Status 02-25-2021)
[RFC7807]	**Internet Engineering Task Force (IETF) (2016), Problem Details for HTTP APIs**
	https://tools.ietf.org/html/rfc7807 (Status 02-25-2021)
[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)
[TAD]	**Gaia-X, European Association for Data and Cloud, AISBL (2021): Gaia-X Architecture Document.**
	Please refer to annex "Gaia-X_Architecture_Document_2103"
[TDR]	**Gaia-X Federation Services Technical Development Requirements**
	Please refer to annex "GXFS_Technical_Development_Requirements"
[TR02102-1]	**BSI (2020), Kryptographische Verfahren: Empfehlungen und Schlüssellängen BSI TR-02102-1**
	https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf (Status 03-18-2021)
[TR02102-2]	**BSI (2020), Technische Richtlinie TR-02102-2, Verwendung von Transport Layer Security (TLS)**
	https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102-2.pdf (Status 03-18-2021)
[VC.Evidence]	**Manu Sporny, Dave Longley, David Chadwick (2019), Verifiable Credentials Data Model 1.0 - Evidence**
	https://www.w3.org/TR/vc-data-model/#evidence (Status: 03-24-2021)

***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. **[CMP.NA.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 origin of the product is the requirement to establish digital trust in disclosed data (paper or electronic) of Gaia-X participants to use in the Gaia-X ecosystem. To reach this goal, a software component is necessary that outputs digital attestations in a standard format (Verifiable Credentials). With that component certification institutions (e.g., Government, Lawyers etc.) are enabled to prove the identity and provided data of any organization that wants to be a Gaia-X participant and deliver the desired verified attestation as a digital representation. Simply spoken, the process is necessary to "transform" physical and unstructured electronic documents into the digital W3C Verifiable Credential format to support the onboarding, accreditation and trustworthiness of interested organizations. Each attestation representation is named in the architecture concept testimonial and referenced from the self-description. In usage of the notarization component the compliance team can create, together with the provider, the self-description and issue that as the digital anchor of trust.

The concept of the notarization is introduced in this specification like eSSIF[2] suggests: backed by a notarization software component and used in the process by AISBL or any attested organizations to issue W3C standard based verifiable credentials in usage of cryptographic primitives. This service aims to bridge the gap of the "old" classic world based on paper trust to the "new" world of digital trust.

[2] https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITALEBSI/Notarisation

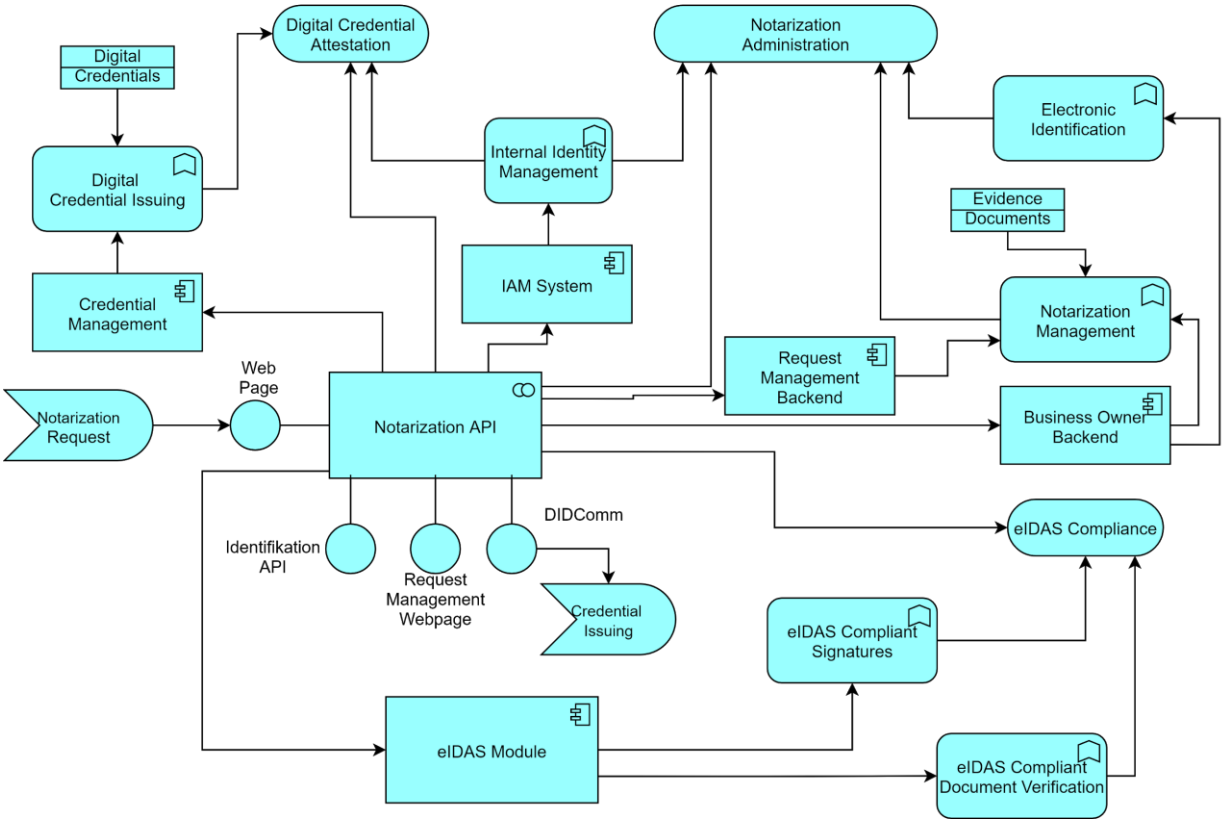


Figure 1: *Architecture*

2.2. Product Functions

The product itself follows the microservice component design principles. The functionality is exposed per REST Service and accessible over the Network per HTTPS protocol. The component has to be installed in multi locations and SHOULD NOT be planned as a central hosted system from ASIBL. Therefore, it must be possible to install it within the issuer's organization domain with multiuser access capabilities. There is no special requirement for high availability. The access to the offered functionality of the component MUST be protected for the usage in such an environment. This includes role concepts, data storage protection and access control. The overall functionality of the product MUST be auditable (GDPR conform), which means each action MUST be documented with all context specific information within the system.

The main functionality scope of this product is to provide some external interfaces to receive notarization requests and issue digital credentials with a legally guaranteed trust. To achieve that, the signatures of the verifiable credentials SHOULD be eIDAS compliant regarding the EU Regulation No 910/2014[3].

[3] https://eur-lex.europa.eu/eli/reg/2014/910/oj

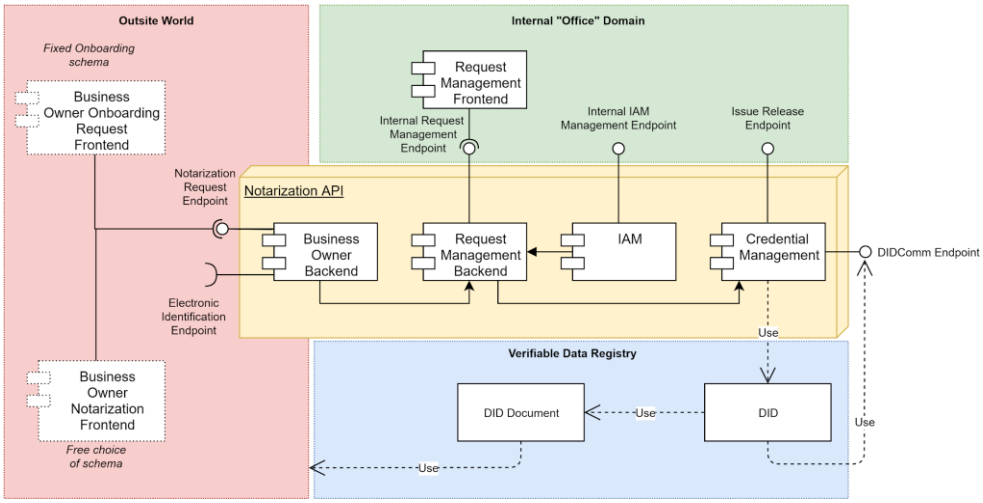


***Figure 2**: Sketch of the API structure*

The core functions of the product are:

- External APIs to fulfil notarization requests.
- Issue digital attestation credential of new participants (Organizations) to support the onboarding and accreditation workflows of Gaia-X
- Management and handling of attestation requests
- Produce digital attestation credentials of any given certificate (e.g., ISO27001, IOS9001, ...)
- Revocation of issued credentials
- Electronic identification (e.g., eId, Video-Ident) for verifying business owners
- Definition of credentials including Gaia-X schema definitions

All sketched parts of the product MAY be separated in different instances for micro service architecture. The product scope MUST include REST API's, SHOULD include command line client and MUST NOT include graphical user interfaces (optional).

2.3. Product Constraints

⏩ CMP.NA.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 taken into account during implementation. ⏪

CMP.NA.00002 **Internal IAM**

The product MUST contain an internal IAM for user accounts/login management. But all interfaces to the internal IAM interfaces SHOULD support already existing IAMs. Therefore, the internal IAM interfaces SHOULD be set up on standard OIDC implementations.

CMP.NA.00003 **GDPR Consideration**

The requests/proposals for attestation may contain GDPR relevant data, which has to be considered during the development. Issued verifiable credentials have to be deleted automatically after the expiration time.

CMP.NA.00004 **Integration of electronic identification**

The identification and authentication of the business owner of an organization SHOULD be done by an electronic identification system to minimize the effort for proving any identity of the requestor.

CMP.NA.00005 **Database System**

The product MUST have an open-source database system to store notarization requests and all other related data. The database system MUST have redundancy for real time backups, to avoid data loss. The database system MUST be focused on Consistency and Availability with the main target of a fully synchronized set of data. This goal MAY be reached over native mechanisms or over other mechanisms so far, the AC characteristics are fulfilled. The database content MUST be encrypted. This MAY be reached either by encryption at rest or data field encryption.

CMP.NA.00006 **Public Resolvable DID**

The product MUST use a public resolvable DID for credential issuing. The DID requirements and the process for obtaining a DID are described in the architecture overview document (see [IDM.AO]).

2.4. User Classes and Characteristics

User Class	Description	Frequency	Expertise	Privilege Level	Product Usage
Notarization Operator	Notarize given data and confirm the issuing of electronic	High	Low	High	Managing Frontend

	credentials to a given DID. An operator is an employee of the notarization office.				
Administrator	Setup the system and maintain operator identities.	Low	High	Low	Backend Maintenance
Organization Business Owner	Authorization Officer which represents a participant.	Low	Low	Low	Request Frontends

***Table 2**: User Classes and Characteristics*

2.5. Operating Environment

⏩ CMP.NA.00007 **Backend Operating Systems**

The product backend 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. ⏪

⏩ CMP.NA.00008 **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. ⏪

2.6. User Documentation

⏩ CMP.NA.00009 **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 ⏪

⏩ CMP.NA.00010 **Participant Documentation**

The documentation MUST contain:

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

Further requirements regarding the documentation can be found in [TDR].

2.7. Assumptions and Dependencies

Not applicable.

2.8. Apportioning of Requirements

Feature	**Priority**
Internal Identity Management	1
Notarization Request Management	1
Digital Credential Issuing	1
Electronic Identification	2
eIDAS Compliant Credential Signatures	3

***Table 3**: Apportioning of Requirements*

3. Requirements

Next to the requirements stated in this document, the requirements regarding the Technical Environment/ Development [TDR] must be also met.

3.1. External Interfaces

3.1.1. User Interfaces

User interfaces are not part of this specification. But it has to be demonstrated that the user interface endpoints are working. This MAY be done by any command line tool e.g., curl, but it MUST be done by unit tests.

3.1.2. Hardware Interfaces

Not applicable.

3.1.3. Software Interfaces

⏩ CMP.NA.00011 **OAuth2**

The product internal IAM MUST support OAuth2 to grant access to the API. ⏪

⏩ CMP.NA.00012 **Database Connection**

The connection of the product to its database MUST be TLS encrypted or a similar encryption of the transport level. ⏪

3.1.4. Communications Interfaces

⏩ CMP.NA.00013 **Notarization Request Endpoint**

The generic endpoint is required to send user specific JSON-LD data for notarization requests. The public resolvable JSON-LD context MUST be accepted by the request endpoint after successful validation. In addition, an upload functionality of notarization documents (e.g., PDF Files, Text Files) MUST be available.

Each endpoint action MUST be linked to the electronic identification endpoint. It MUST be avoided that any action in the notarization system is triggered, without any electronic identification before.

The uploaded attestation documents MUST create a hash using the keccak-256 function. The hash has to be embedded inside the issued verifiable credential for validation purposes.

Supported Actions: GET (view), POST (Notarization Request), PATCH (update existing Request), DELETE (Revoke existing Request) and POST for File Upload. ⏪

⏩ CMP.NA.00014 **Electronic Identification Endpoint**

The electronic identification endpoint structure and behavior depends on the chosen technology, but the result of the identification MUST be legally secure. A Notarization Operator MUST be enabled to decide legally secure that a given request was authorized by the business owner. It MUST be avoided that any third party can misuse this endpoint. ⏪

⏩ CMP.NA.00015 **Internal Request Management Endpoint**

The endpoint delivers all results from the notarization requests for authorized notarization operators. The endpoint MUST provide a bi-directional HTTP connection to inform the operator immediately for new requests. The notarization requests itself MUST be provided by GET actions. An operator MUST confirm with a POST action an valid request after the validation. Any rejection MUST be performed by a POST action. All actions SHOULD be protected by an internal IAM system or other security protection mechanisms MUST be established. Each decision MUST be documented by audit entries. ⏪

⏩ CMP.NA.00016 **Internal IAM Management Endpoint**

The internal IAM management endpoint provides the capability for the administrator to manage users and identities locally for the operations of this product. ⏪

⏩ CMP.NA.00017 **DIDComm Endpoint**

The DIDComm Endpoint provides an TLS connection to another DIDComm Endpoint.

This Protocol is specified in DIDComm Messaging specification[4] ⏪

⏩ CMP.NA.00018 **Issue Release Endpoint**

The issue release endpoint provides functionality to release credentials by a HTTP trigger from outside. This endpoint is an optional endpoint and MUST be configurable. (used or not) ⏪

3.2. Functional

⏩ CMP.NA.00019 **Creation of Notarization Operator Identities**

[4] https://identity.foundation/didcomm-messaging/spec/

Description

An administrator has to create an identity for the notarization operator within the internal IAM system. This identity gets the permissions and rights to manage notarization requests. The minimum permissions are confirm, delete, reject, issue and view. There MAY be other permissions if necessary. The user creation MAY be done by the user itself, if it's guaranteed that the user is an authorized operator for this process.

Constraints

None.

Interfaces

Internal IAM Management Endpoint.

Input

An operator username and password, combined with a Two Factor authentication. (Second Factor MUST be registered by the user)

Output

A notarization operator identity.

Acceptance Criterias

A created identity for a notarization operator, with a successful two factor authentication. ⏪

⏩ CMP.NA.00020 **Deletion of Notarization Operator Identities**

Description

The administrator MUST be able to delete an operator identity from the internal IAM.

Constraints

An existing identity.

Interfaces

Internal IAM Management Endpoint.

Input

Deletion request and a confirmation.

Output

A deletion response.

Acceptance Criterias

Successfully deleted identity. ◂◂

⏩ CMP.NA.00021 **Lock/Unlock of Notarization Operator Identities**

Description

The administrator MUST be able to lock/unlock an existing identity.

Constraints

An existing identity.

Interfaces

Internal IAM Management Endpoint.

Input

A lock/unlock request and a confirmation.

Output

A locked/unlocked identity.

Acceptance Criterias

If locked, the identity is not anymore able to login or to use the notarization API. If unlocked, the identity is able to access everything as before. ◂◂

⏩ CMP.NA.00022 **Creation of Notarization Requests**

Description

Stakeholders (e.g., Onboarding Authority, Conformity Assessment Bodies, Notarization Operators, business owner) need the capability to create notarization requests. It MUST be also possible to upload files for verification and documentation. This function MUST be on hold until a successful identification over the Electronic Identification Endpoint. If the identification is not successful, the creation of a request is not successful.

Constraints

Identification of the actor.

Database system.

DID of the organization.

Interfaces

Notarization Request Endpoint

Electronic Identification Endpoint

Input

JSON-LD Schema Reference, JSON-LD Content and optionally files and/or file links for download.

Output

An appropriate response to the HTTP action. The content must be stored in the database system.

Acceptance Criterias

1) A request has been stored successfully into the database. (201)
2) Context is not matching to content. (400)
3) Database contains the Request included the identification information.
4) Audit Entry created.

⏪

⏭ CMP.NA.00023 **Update of Notarization Requests**

Description

The business owner needs the capability to update notarization requests. This update MUST be possible, if the Request Record is not in progress by the operator. (State=Open, State=Rejected) This function MUST be on hold until a successful identification over the Electronic Identification Endpoint. If the identification is not successful, the update of a request is not successful.

Constraints

Identification of the actor.

Database system.

Interfaces

Notarization Request Endpoint

Electronic Identification Endpoint

Input

Patch Request with the updated values.

Output

An appropriate response to the HTTP action. The content must be stored in the database system.

Acceptance Criterias

1) A request update has been stored successfully into the database. (200)
2) A wrong context or missing data leads to an exception.(400)
3) Request not more available, if deleted. (410)
4) Audit Entry created.
5) Error, if record is in progress by the operator.

⏮

⏭ CMP.NA.00024 **Revocation of Notarization Requests**

Description

The business owner needs the capability to delete his records in the Notarization API. This delete MUST be possible, if the Request Record is not in progress by the operator. (State=Open, State=Rejected) This function MUST be on hold until a successful identification over the Electronic Identification Endpoint. If the identification is not successful, the revocation of a request is not successful.

Constraints

Identification of the actor.

Database system.

Interfaces

Notarization Request Endpoint

Interfaces

Notarization Request Endpoint

Electronic Identification Endpoint

Input

Delete Request with the updated values.

Output

An appropriate response to the HTTP action. The content must be deleted from the database system.

Acceptance Criterias

1) Request Successful deleted from database. (204)
2) Deletion not successful. (400)
3) Request not more available, if deleted (410)
4) Audit Entry created.
5) Error, if record is in progress by the operator.

⏩ CMP.NA.00025 **Automatically Revocation**

Description

If an identification is not done, or provided properly, all requests MUST be revoked automatically. Means unidentifiable Requests SHOULD be deleted and MAY archived after a time expiration. The core data of these deleted requests MUST be part of the monitoring and auditing. (e.g., IP Address, Provided Data, Timestamp etc.)

Constraints

Database system.

Interfaces

Database Interface.

Input

A database query for deletion for any record less than expiring date.

Output

Deleted record amount.

Acceptance Criterias

Expired Records are deleted.

⏪

⏩ CMP.NA.00026 **Storing/Update/Deletion of Requests**

Description

The internal API functions need the capability to store, delete and update request records in the database. All of the operations MUST be recorded for auditing. The operations of records SHOULD fire an event to support other functions with notifications.

Constraints

Database system.

Interfaces

Database Interface.

Input

A request id together with data values.

Output

A database response for success/fail.

Optionally a notification for the action.

Acceptance Criterias

All operations are supported and are working properly. (audit entries created, data stored, updated and deleted)

⏮

⏭ CMP.NA.00027 **Set Requests to Work in Progress**

The notarization operator needs the capability to pick any request to work on it. If the operator picks a record over the request management endpoint, the record MUST be locked in the database, so that no other actions can modify it anymore.

Constraints

Database system.

Interfaces

Request Management Endpoint.

Database interface.

Input

A work in progress operation of the operator for one request record id.

Output

A response for a successful work in progress operation.

Acceptance Criterias

1) Successful picked. (200)
2) Error Exception if not more available (410)

⏮

⏭ CMP.NA.00028 **Viewing of Requests**

The notarization operator needs the capability to view all requests. The endpoint MUST return all open requests. May the endpoint support filtering for the operator name and other criterias to filter the requests better.

Constraints

Database system.

Interfaces

Request Management Endpoint.

Database system.

Input

A view request operation. (GET)

Output

A response with a JSON-LD in the Body. MAY paging information can be inserted. The content of the requests MAY be separated in different routes or operations for a better UX support.

Acceptance Criterias

The request information stored in the database can be listed over the endpoint.

⏮

⏭ CMP.NA.00029 **Rejection of Requests**

The notarization operator needs the capability to reject requests, if there is any mismatch or wrong data. This rejection MUST be done over the id of the request record. The product MUST inform the requester about the rejection.

Constraints

Database system.

Interfaces

Request Management Endpoint.

Database system.

Input

A reject operation with a record id. (DELETE)

Output

A HTTP response with success or failure.

Acceptance Criterias

Request Record deleted.

⏮

⏭ CMP.NA.00030 **Electronic Identification of Business Owner**

The notarization operator needs information from the Business Owner who is registering some data to decide, if a credential can be issued or not. The endpoint MUST deliver this information for the request record from a third party legally secure, otherwise the request record cannot be accepted.

Constraints

Notarization Request Endpoint

Interfaces

Electronic Identification Endpoint

Input

Identification Request.

Output

Legally confirmed identity data of the business owner. (e.g., name, birthdate, address etc.)

Acceptance Criterias

1) Identity data is stored next to the request record.
2) Identity data is separately encrypted and just readable by an notarization operator.
3) Identity is deleted after a request confirmation (also deleted in backups)

⏮

⏭ CMP.NA.00031 **Confirmation of Requests**

Description

The notarization operator reviews the incoming requests and confirms the requests.

Constraints

Database system.

Interfaces

Request Management Endpoint.

Database system.

Input

A confirmation action with the record id. (POST)

Output

The record is set to status confirmed.

Acceptance Criterias

The record is set to status confirmed.

⏪

⏩ CMP.NA.00032 **Credential Issuing**

Description

The credential issuing is an asynchronous process which checks the database for confirmed request records in the database. All of the found confirmed records will be picked up. Within the record MUST be a linked DID of the participant, which is here used to establish a connection to the Credential Manager of the participant. In the case of a configured issue release endpoint, the linked DID is optional for the notarization request to the point of a release over this endpoint. Is the release of the credentials triggered for the notarization request, the credentials MUST be issued to the given DID. If the connection is accepted by the participant, the function MUST create a credential proposal, which can be accepted by the participant. This protocol must follow the Aries RFC 0036[5] based on indy technology. May the component can perform an Present Proof request before issuing defined in Aries RFC 0037[6]. The credentials MUST have a limited live time and the product DID as issuer. The schema SHOULD be chosen by the notarization operator and MUST be selectable from a selection list. This schema list and the supported DID Methods/Identity Networks MAY be statically configured. Optional the list can be loaded and updated automatically.

Constraints

Own DID for issuing.

DID of the organization.

Database system.

Aries RFC 0036

Aries RFC 005

DIDComm Protocol

[5] https://github.com/hyperledger/aries-rfcs/tree/master/features/0036-issue-credential

[6] https://github.com/hyperledger/aries-rfcs/tree/master/features/0037-present-proof

Interfaces

Database interface.

DIDComm Endpoint.

Issue Release Endpoint

Input

A confirmed request record.

Output

An issued credential to the participant DID.

Acceptance Criterias

1) Credential successfully issued to participant DID.
2) Credential has the exact type of the request context.
3) Deletion of the request record, after successful issuing.

⏮

⏭ CMP.NA.00033 **Proof of Credentials**

Description

To prove the trustworthiness of this product, it MUST support the ARIES RFC 0037 protocol and it MUST be configurable which proofs are fulfilled automatically (e.g., based on a configured schema).

Constraints

Control over the Own DID.

DIDComm Protocol

Interfaces

DIDComm Endpoint.

Input

Any presentation request described in ARIES 0037.

Output

A presentation request according to ARIES0037.

Acceptance Criterias

Product is able to prove the hold credentials, if a verifier requests a credential presentation.

⏮

⏭ CMP.NA.00034 **Revoke of Credentials**

Description

In some cases, the issued credentials have to be revoked. To reach this, the product MUST implement credential definitions to maintain revocation lists on the ledger. The revocation action itself is a manual action by the notarization operator and MAY be implemented in a self-chosen way. It SHOULD provide any UI HTTP endpoint for credential management.

Constraints

Control over the Own DID.

Interfaces

Ledger interface.

Input

The credential to revoke.

Output

Updated revocation list.

Acceptance Criterias

Credential is revoked.

⏮

⏩ CMP.NA.00035 **Accept Credentials**

Description

The DIDs (one or more) have to be configured during the installation of the product. This DID(s) will issue credentials to this product to establish the trust. The product MUST accept all credentials from these configured DIDs automatically. All other credentials from unknown DIDs MUST be rejected.

Constraints

DID configuration.

Interfaces

Input

Credential Offerings.

Output

Accept offerings.

Acceptance Criterias

1) Credentials are accepted automatically for configured DIDs.
2) DIDs must be configurable during Runtime

⏪

⏩ CMP.NA.00036 **eIDAS compliant Signature Creation/Validation**

Signatures must be generated/verified in compliance with eIDAS so that legally secure trust can be achieved. This should include the eIDAS signature types basic, advanced and qualified. The implementation variant must be selected individually in coordination with the used technology. ⏪

⏩ CMP.NA.00037 **Validation of Document Signatures**

Description

The validation of document signatures is necessary, when someone uploads a file with a digital file signature (e.g., QES). In this case the function MUST validate:

- is the signature certificate from a trusted CA
- was the signature valid on the time of signing

Depending on the result, the documents MUST be highlighted in the database with the error of the signing result, so that the operator can decide how to interact with the requestor.

Constraints

EU List of trusted certificate Authorities.

Creation of Notarization Requests

Interfaces

Notarization Request Endpoint

Input

Files e.g., PDFs, TXT etc.

Output

An Information in the database, whether the file has a valid signature or not.

Acceptance Criterias

1) A successful validation of the signature with a hint in the DB that the signature was valid proven
2) An unsuccessful validation of the signature with a hint in the DB that the signature was not valid included reasons why.

⏪

⏩ CMP.NA.00038 **Credential Schema/Evidence Link**

Description

Each credential schema for issuing MUST contain one or multiple hash values (keccak-256) and MAY HTTP links to the given evidence documents, to ensure that the attestation is legally compliant and done to the correct file. The statutory retention period has to be respected for all links and hashes within the credential creations. Retention periods depend on the notarization request and the respective document that should be verified.

Constraints

Credential Issuing

Compliance

Interfaces

Notarization Request Endpoint

DIDComm Protocol

Input

Files e.g., PDFs, TXT etc.

Output

A hash value within the issued credential and optional a HTTP Link to the evidence document.

Acceptance Criterias

1) Issued credential with the correct hash
2) Audit Path from Evidence Documents to Verifiable Credentials

3.3. Other Nonfunctional Requirements

3.3.1. HTTP Requirements

CMP.NA.00039 **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.

CMP.NA.00040 **HTTP Protocol Definitions**

All HTTP Endpoints MUST follow RFC 7231[7] and RFC 5789[8], but it MAY be chosen what of the protocols is necessary to realize the functionality. For problem reports the RFC7807[9] MUST be used in combination with Standard HTTP Error Codes. ◂◂

3.3.2. Configuration

▸▸ CMP.NA.00041 **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

▸▸ CMP.NA.00042 **Data Minimization**

From GDPR perspective the product MUST NOT log data which is related to personal information (e.g., Usernames, 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. ◂◂

▸▸ CMP.NA.00043 **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. ◂◂

[7] https://tools.ietf.org/html/rfc7231

[8] https://tools.ietf.org/html/rfc5789

[9] https://tools.ietf.org/html/rfc7807

3.3.4. Monitoring Requirements

⏩ CMP.NA.00044 **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. ⏪

⏩ CMP.NA.00045 **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 in order 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

⏩ CMP.NA.00046 **Electronic Identification Timeout**

If a request is open and the identification is missing, the timeout for this action MAY be a couple of hours until some days before automatic revocation and data deletion. The exact expiration time has to be configured by the administrator. ⏪

⏩ CMP.NA.00047 **Public Endpoints**

All public endpoints have an expected amount of some hundred users at the same time, the endpoints MUST therefore support parallel and non-blocking execution of parallel requests to provide an acceptable UX performance. This means a maximum waiting time for 5 seconds for a notarization request. Possible File uploads are not included in the time measurement. ⏪

⏩ CMP.NA.00048 **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 behaviour of minimum 50% more performance by each additional instance. ⏪

⏩ CMP.NA.00049 **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

▸▸ CMP.NA.00050 **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. ◂◂

▸▸ CMP.NA.00051 **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. ◂◂

▸▸ CMP.NA.00052 **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 MUST meet the requirements stated in the document “Specification of non-functional Requirements Security and Privacy by Design” [GXFS_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.

▸▸ CMP.NA.00053 **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[10]

[10] See https://www.keylength.com/en for a comparison

[CryptoLen]. The recommendations can be found in the technical guidelines[11] TR 02102-1 [TR02102-1] and TR 02102-2 [TR02102-2] or SOG-IS Agreed Cryptographic Mechanisms[12] [SOG-IS]. ◂◂

▸▸ CMP.NA.00054 **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) 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 250 Bit key q	SHA-2 with an output length ≥ 256 Bit or better

***Table 4**: Requirements cryptographic algorithms and key length*

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

▸▸ CMP.NA.00055 **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. ◂◂

▸▸ CMP.NA.00056 **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

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

[12] See https://www.sogis.eu/documents/cc/crypto/SOGIS-Agreed-Cryptographic-Mechanisms-1.2.pdf

different security risks of the product and how they are mitigated (e.g., by Threat Modeling Protocols)

CMP.NA.00057 **Installation of Critical Security Updates**

Node operators SHOULD deploy security critical updates without undue delay.

CMP.NA.00058 **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.

CMP.NA.00059 **HTTP Pentesting**

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

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 with a proper documentation for concerns of traceability.

CMP.NA.00060 **Storage of Secrets**

The storage of secret information such as private keys MUST take place in state-of-the-art secure environments in order 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.

CMP.NA.00061 **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. ◂◂

▸▸ CMP.NA.00062 **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. ◂◂

▸▸ CMP.NA.00063 **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

▸▸ CMP.NA.00064 **Quality Aspects**

The software MUST meet the following requirements:

- The quality standards MUST meet ISO 25010 [ISO25000] https://iso25000.com/index.php/en/iso-25000-standards/iso-25010
- 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 SHOULD be documented and mitigated. ◂◂

3.3.9. Business Rules

Issuing must be only possible with two factor authentication.

Each must have a QES SignatureCard like eSignature within the German ID-Card.

To perform the function CMP.NA.00037 Confirmation of Requests, the notarization officer has to perform certain (manual) verification tasks that are out of scope for this product. For the Gaia-X ecosystem, the most important verification tasks are performed during the Gaia-X Onboarding and Accreditation Workflow (OAW). In case a Gaia-X Provider wants to onboard itself or their Gaia-X Assets, they have to undergo the OAW, which is described in [CP.OAW] in detail. During the OAW, an Onboarding Authority (i.e., the AISBL), or Conformity Assessment Bodies that are approved by the AISBL, perform specified accreditation workflows to ensure that the Provider and its Assets are in compliance with the Gaia-X principles and policy rules [PRD]. In case the accreditation is successful, the Onboarding Authority, or approved Conformity Assessment Bodies, use or call this product to generate the verifiable credential proving Gaia-X compliance (i.e., the Gaia-X compliance attestation). Verifiable credentials proving Gaia-X compliance can only be issued if the Onboarding Authority or Conformity Assessment Bodies have approved the notarization request or perform the notarization request themselves. The same goes for revocation of verifiable credentials related to Gaia-X compliance.

Further verification checks during the function CMP.NA.00037 Confirmation of Requests are possible, depending on the notarization request, verifiable credential type, and document etc.

3.4. Compliance

⏩ CMP.NA.00065 **GDPR Audit Logging**

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

⏩ CMP.NA.00066 **GDPR Data Processing**

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. ⏪

⏩ CMP.NA.00067 **Legal compliance**

Depending on the function, compliance with national and European laws MUST always be ensured. ⏪

⏩ CMP.NA.00068 **Statutory Retention Period**

The verifiable credential, logs and respective data MUST be retained depending on the document and the notarization request. For example, creating a verifiable credential to prove the Gaia-X Compliance Attestation should be retained in accordance with requirements the Gaia-X Onboarding and Accreditation Workflows (i.e., 6 years). ◂◂

3.5. Design and Implementation

3.5.1. Distribution

▸▸ CMP.NA.00069 **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. ◂◂

3.5.2. Maintainability

▸▸ CMP.NA.00070 **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). ◂◂

▸▸ CMP.NA.00071 **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. ◂◂

3.5.3. Portability

▸▸ CMP.NA.00072 **Environment Portability**

The component MUST be run in different OS environments including windows and Linux machines. The deployment in well-known cloud native environments MUST be possible. Therefore, the product SHOULD be delivered in the portable container image format OCI[13] . ◂◂

[13] https://opencontainers.org/

3.5.4. Operability

⏩ CMP.NA.00073 **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

⏩ CMP.NA.00074 **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) ⏪

3.5.6. Availability

⏩ CMP.NA.00075 **Issue Request/Web Frontend Availability**

The product functions for issue requests MUST be available 24x7 to receive requests every time. ⏪

⏩ CMP.NA.00076 **Confirm Availability**

The product functions for confirmation MUST be 5x8 hours per week available, because in the background are manual validation operations necessary which are bound to office hours. ⏪

3.5.7. Scalability

⏩ CMP.NA.00077 **Backend Scalability**

The backend system SHOULD be scalable across multiple machines or with multiple processes on one machine (e.g., multiple instances of docker image or across a Kubernetes cluster). ⏪

4. System Features

4.1. Internal Identity Management

4.1.1. Description and Priority

The internal identity management is a feature to maintain notarization operators. This is necessary because not everyone is from a legal perspective permitted to notarize requests and issue digital credentials. It must be guaranteed that just authorized people have access to this highly critical trust infrastructure. Therefore, the identity management was also declared as “internal” to ensure in every case the availability of an appropriate identity management is given. This excludes not the usage of an already existing IAM within a company.

4.1.2. Stimulus/Response Sequences

All actions are manual actions by an administrator over the administration endpoint.

4.1.3. Functional Requirements

Functional Requirement	*Endpoint*	*Protocol*	*Actor*	*Parallelism*
CMP.NA.00020 Deletion of Notarization Operator Identities	Internal IAM Management	HTTP	Administrator	High
CMP.NA.00021 Lock/Unlock of Notarization Operator Identities	Internal IAM Management	HTTP	Administrator	High

***Table 5**: Internal Identity Management Functional Requirements*

4.2. Notarization Request Management

4.2.1. Description and Priority

The request management handles notarization requests from business owners for the notarization operator. The operator can view open/confirmed requests and which data is provided. After a (compliance)review process (not scope of this product, refer to the Gaia-X Onboarding and Accreditation Workflows [CP.OAW]), the operator decides, whether given data is acceptable or not. This action generates **Trust** in this given data by the participant, which is a very critical action. It must be ensured that no unauthorized third party can use this feature. It must be protected with state-of-the-art security mechanisms.

4.2.2. Stimulus/Response Sequences

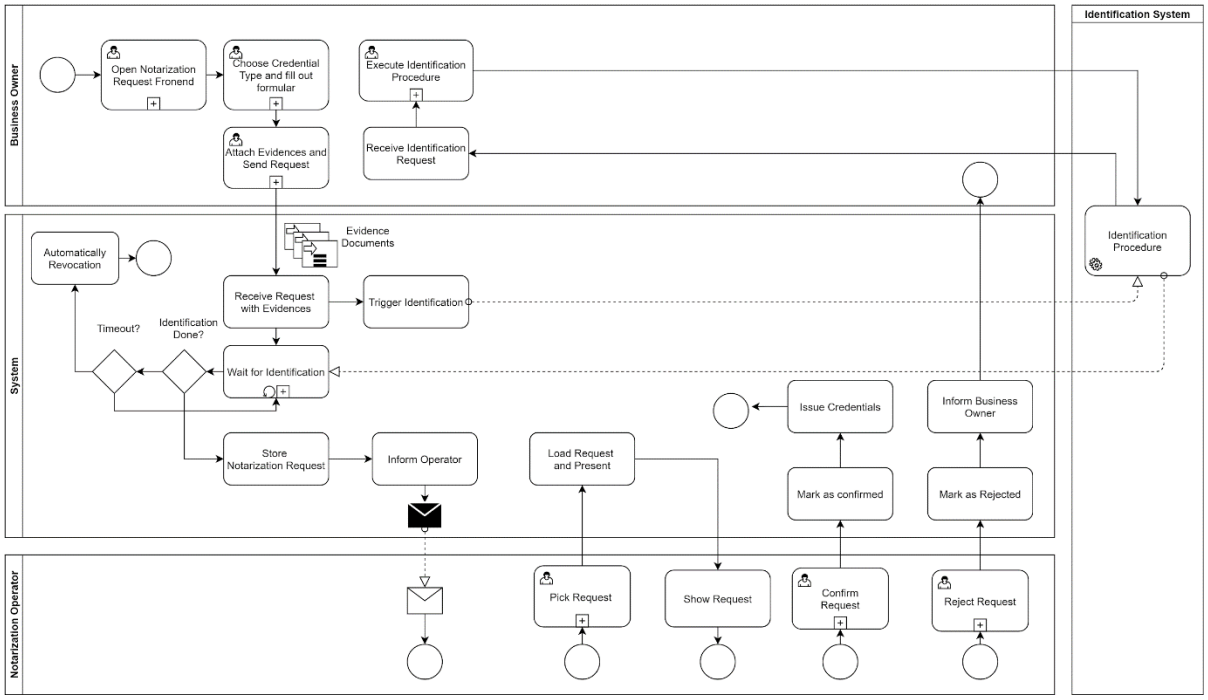


***Figure 3**: Notarization Request Management*

4.2.3. Functional Requirements

Functional Requirement	*Endpoint*	*Protocol*	*Actor*	*Parallelism*
CMP.NA.00022 Creation of Notarization Operator Identities	Notarization Request	HTTP	Business Owner	High
CMP.NA.00023 Update of Notarization Requests	Notarization Request	HTTP	Business Owner	High
CMP.NA.00024 Revocation of Notarization Requests	Notarization Request	HTTP	Business Owner	High
CMP.NA.00025 Automatically Revocation	-	-	System	High
CMP.NA.00026 Storing/Update/Deletion of Requests	-	-	System	High
CMP.NA.00027 Set Requests to Work in Progress	Request Management	HTTP	Notarization Operator	Medium
CMP.NA.00028 Viewing of Requests	Request Management	HTTP	Notarization Operator	Medium

CMP.NA.00029 Rejection of Requests	Request Management	HTTP	Notarization Operator	Low
CMP.NA.00031 Confirmation of Requests	Request Management	HTTP	Notarization Operator	Low

***Table 6**: Notarization Request Management Functional Requirements*

4.3. Digital Credential Issuing

4.3.1. Description and Priority

After the confirmation of any request, an automatic process in the background issues the credentials to the requesting parties. In a few words, this feature executes the decisions of the notarization operator. To fulfill this task, the feature provides functionalities based on the Hyperledger ARIES RFCs in combination with a unique DID of the notarization service to guarantee the cryptographically authenticity to the participant. The DID itself, represents the public identity of the notarization organization. (e.g., DID represents ”John Doe’s Notarization Office Ltd.”)

4.3.2. Stimulus/Response Sequences

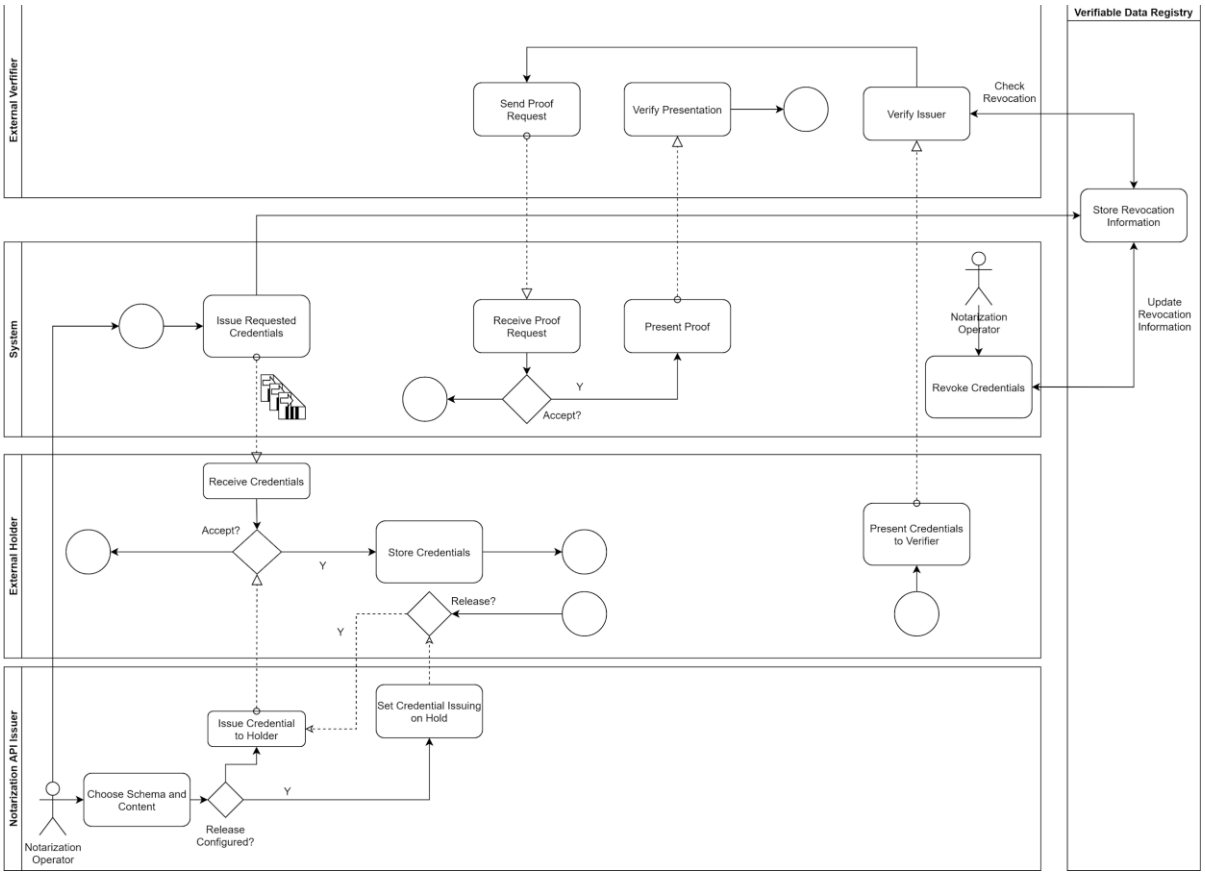


***Figure 4**: Digital Credential Issuing*

4.3.3. Functional Requirements

Functional Requirement	*Endpoint*	*Protocol*	*Actor*	*Parallelism*
CMP.NA.00032 Credential Issuing	*DIDComm*	*DIDComm*	System	High
CMP.NA.00033 Proof of Credentials	*DIDComm*	*DIDComm*	System	High
CMP.NA.00034 Revoke of Credentials	*DIDComm*	*DIDComm*	System	High
CMP.NA.00035 Accept Credentials	*DIDComm*	*DIDComm*	System	High

***Table 7**: Digital Credential Issuing Functional Requirements*

4.4. eIDAS Compliant Signatures

4.4.1. Description and Priority

In order to provide eIDAS compliant signatures the feature should be able to generate and validate eIDAS compliant signatures. In consideration of the different eIDAS types, legal signatures should be considered and a bridge functionality to sign the data should be implemented. A secure environment MUST be provided to store and execute the necessary functions (signature, validation) and SHOULD require at least two factor authentication.

4.4.2. Stimulus/Response Sequences

An approved credential issued by the notarization operator.

4.4.3. Functional Requirements

Functional Requirement	*Endpoint*	*Protocol*	*Actor*	*Parallelism*
CMP.NA.00036 eIDAS compliant Signature Creation/Validation Creation	-	-	Notarization Operator	Medium

***Table 8**: eIDAS Compliant Signatures Functional Requirements*

4.5. eIDAS Compliant Document Verification

4.5.1. Description and Priority

The signatures (if present) of the evidence documents must be checked for validity and the correct content. If the validation is valid, it will be highlighted to the notarization operator for checkup. If the signature is no more valid or present, the notarization operator needs that highlighted information as well, for this decision.

4.5.2. Stimulus/Response Sequences

An evidence document with a signature triggers the validation of the signature.

4.5.3. Functional Requirements

Functional Requirement	*Endpoint*	*Protocol*	*Actor*	*Parallelism*
CMP.NA.00037 Validation of Document Signatures	*Notarization Request Endpoint*	*HTTP*	System	High

***Table 9**: eIDAS Compliant Document Verification Functional Requirements*

4.6. Electronic Identification

4.6.1. Description and Priority

The electronic identification is an enhancement in a later version where the business owner is identified by an electronic system. All functionalities with electronic identification can keep this functionality out of scope until the version of the product where the feature is implemented.

4.6.2. Stimulus/Response Sequences

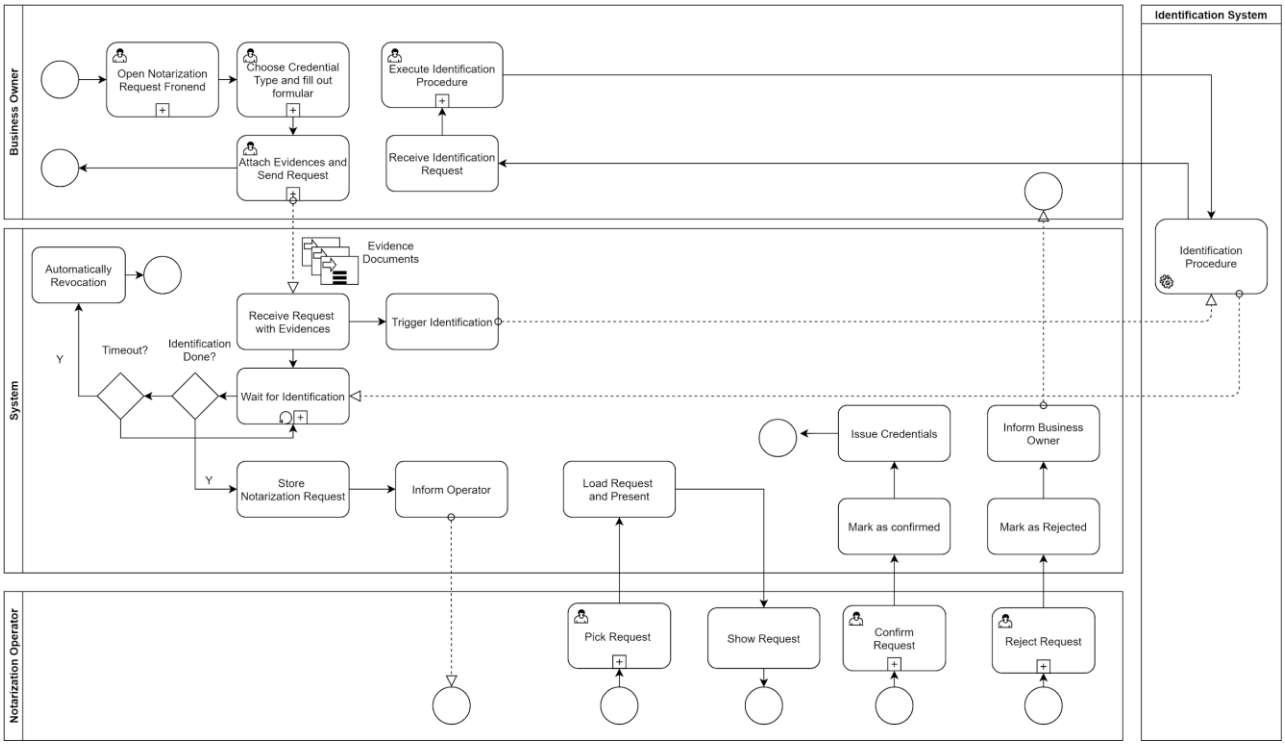


***Figure 5**: Electronic Identification*

4.6.3. Functional Requirements

Functional Requirement	*Endpoint*	*Protocol*	*Actor*	*Parallelism*
CMP.NA.00030 Electronic Identification of Business Owner	Electronic Identification	Open	System	High

***Table 10**: Electronic Identification Functional Requirements*

5. Other Requirements

Next to the requirements stated in this document, the requirements regarding the Technical Environment/ Development [TDR] must be also met.

6. Verification

CMP.NA.00078 **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” or similar.

⏩ CMP.NA.00079 **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. ⏪

⏩ CMP.NA.00080 **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[14]. ⏪

[14] https://gatling.io/

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

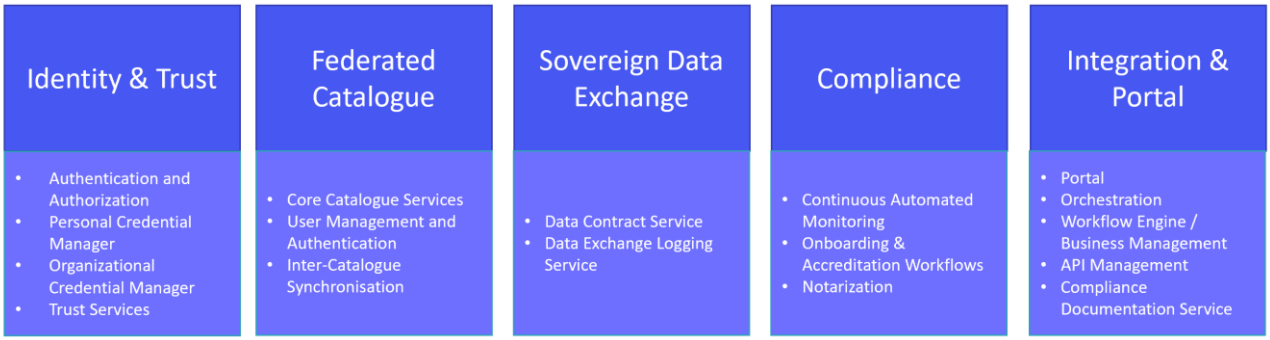
Work Package 4 (WP4): Compliance

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].