Software Requirements Specification

for

Gaia-X Federation Services

Sovereign Data Exchange Data Contract Service SDE.DC

Published by

Gaia-X AISBL
c/o Fraunhofer EU Office
Rue Royale 94
1000 Bruxelles

Table of Contents

List of Figures

List of Tables

1. Introduction

This document comprises a software specification of the Gaia-X Data Contract Service (GX-DCS), one of the Gaia-X Federation Services (GXFS) and the crystallization core of the future Gaia-X data ecosystem. GX-DCS acts as a broker of data delivery contracts between Data Providers and Data Consumers, much like a notary in the real world. There are three things about data transactions to make sense of GX-DCS:

1. Each data transaction from Provider to Customer, whether at a price or not, is a business transaction that has legal implications. Naturally, not each transaction needs an explicit lawful sales contract – you don't literally sign a contract when you buy a loaf of bread – but as soon as large sums of money, copyright laws, personal data, or other such complexities come into play, a legally waterproof contract is worthwhile for both parties.
2. The foundation of GX-DCS is the Data Asset Self-Description (SD). This SD is not just a haystack of technical metadata but, moreover, a robust *data contract template*. Depending on political decisions in the upper echelons of the Gaia-X multiverse, certain parts of the data delivery terms and conditions (compare the "general terms" outlined in section 2.3) might be part of the onboarding agreement that each Participant (including all Providers) has to sign. Until that is the case, however, and also for situations where particular Providers want to sell data under specifically defined rules and restrictions, the Data Asset SD – once signed by both parties – constitutes a valid, enforceable contract.
3. Data delivery contracts split the responsibilities between Data Provider and Data Consumer. While each instance of GX-DCS (there could be many, operated by various Federators) facilitates and enables agreements about data deliveries and, in the future, data sales and subscriptions, Data Providers remain responsible for data quality and legality of the data content itself, while Data Consumers remain responsible for lawful data usage according to the Data Contract. Gaia-X AISBL and the operators of GX-DCS ("Federators"), and that's the point of this elucidation, are not liable for the actual data transfer and data processing taking place.

1.1 Document Purpose

This document serves as a specification for the Gaia-X Data Contract Service, one of the Gaia-X Federation Services deemed essential to build a minimum viable Gaia-X ecosystem. The intended audience comprises potential contractors and IT companies, who are keen on implementing this Federation Service.

1.2 Product Scope

This specification deals with GX-DCS in its very first version, v1. Therefore, the scope has been deliberately limited to a handful of interfaces and functional requirements, which are outlined in section 2 and carefully spelled out in section 3.

The scope of GX-DCS v1 is limited to helping Data Providers and Data Consumers to arrive at a finalized Agreement about data delivery. Therefore, GX-DCS plays a crucial part in the Data Asset lifecycle right

from the start, assisting in Data Asset registration, Data Asset SD validation, contract negotiation, and log token renewal (see GX-DELS, the companion service of GX-DCS: https://www.gxfs.de/federation-services/sovereign-data-exchange/data-exchange-logging-service/).

Still, a few things are **out of scope** but might be added in a later version of GX-DCS:

- Data transfer: The actual processing and transfer of data is currently not a part of the GX-DCS; the Participants must take care of these themselves for the time being. The possible future use of dedicated Data Exchange Services or Connectors remains to be clarified for latter version of the GXFS.
- Authorization and access management regarding the data itself need to be covered by the Data Providers.
- Billing-as-a-Service is also not part of GX-DCS v1. While the Data Contract may contain pricing details, billing has to be realized by the Data Provider.

1.3 Definitions, Acronyms and Abbreviations

Term/Acronym	Meaning	References
Data Asset Self-Description (DASD)	The Data Asset's SD registered with an instance of GX-DCS	http://w3id.org/gaia-x/core#DataAsset See also section 4.1.
Data Contract or Agreement	Data Asset Self-Description, which was completed with Customer details and possibly filled placeholders. Both terms are used synonymously.	See section 3 and 4.1
Data Contract Offer	A DASD, which describes a fixed contract offer that can be finalized without further interaction with the Data Provider. This is the case if there are no negotiable terms (*false* assigned to all *gax:negotiable* attributes) and no Provider confirmation is required (*false* assigned to *gax:confirmationRequired*).	See section 4.1. Inspired by the concept of "offerta ad incertas personas" in the German Law

Invitation for Data Contract Offers	A DASD, which invites the Data Consumer to make an offer, but requires active confirmation from the Data Provider's side. This is the case if either Provider confirmation is required (*true* assigned to *gax:confirmationRequired*) or the DASD contains negotiable terms (*false* assigned to at least one *gax:negotiable* attribute).	See section 4.1. Inspired by the concept of "invitatio ad offerendum" in the German Law
Finalized Data Contract or Finalized Agreement	Data Contract/Agreement that has been signed by all involved parties (Data Consumer, Data Provider, GX-DCS)	See section 3 and 4.1
DID	Decentralized Identifier	see https://www.w3.org/TR/did-core/
GX	Gaia-X	Gaia-X Homepage of the Federal Ministry for Economic Affairs and Energy
GX-DELS	Gaia-X Data Exchange Logging Service	See https://www.gxfs.de/federation-services/sovereign-data-exchange/data-exchange-logging-service/
GX-DCS	Gaia-X Data Contract Service	The service specified in this document.
GX-FC	Gaia-X Federated Catalogue	See https://www.gxfs.de/federation-services/federated-catalogue/core-catalogue-features/
GXFS	Gaia-X Federation Service	See Ref-1 page 14
Proof	A concept relating to asymmetric signatures of a document or value (i.e., encryption with a private key)	https://www.w3.org/TR/vc-data-model/#proofs-signatures

SD	Self-Description	See Ref-1 page 17
URI	Uniform Resource Identifier	RFC 4151: https://tools.ietf.org/html/rfc4151 see also ADR-XXX: Identifiers used in Self-Descriptions in appendix C
VC	Verifiable Credential	https://www.w3.org/TR/vc-data-model/

Table 1: *Definition, Acronyms and Abbreviations*

1.4 References

GX-DCS is loosely coupled to at least five other GXFS from WP1, WP2, WP4, and WP5[1]. To get an overview please refer to www.gxfs.de. Furthermore, an understanding of the overall Gaia-X architecture [Ref-1] and philosophy is necessary to implement DCS.

ID	Description	Link
GXFS-1	WP1: Authentication/ Authorization Specification	refer to annex "SRS_GXFS_IDM_AA"
GXFS-2	WP1: Trust Services Specification	refer to annex "SRS_GXFS_IDM_TSA"
GXFS-3	WP3: Data Exchange Logging Service	refer to annex "SRS_GXFS_SDE_DELS"
GXFS-4	WP5: Non-Functional Security & Privacy by Design Requirements	refer to annex "GX-Nonfunctional_SPBD"
Ref-1	Gaia-X: Technical Architecture	Refer to annex "Gaia-X_Architecture_Document_2103"
Ref-2	Definition of the Data Asset's Self-Description	http://w3id.org/gaia-x/core#DataAsset
Ref-3	ENISA EUCS Draft (Version December 22, 2020)	https://www.enisa.europa.eu/publications/eucs-cloud-service-scheme
Ref-4	BSI Guidelines for Crypto	https://www.bsi.bund.de/EN/Service-Navi/Publications/TechnicalGuidelines/tr02102/tr02102_node.html

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

Ref-5	SOG-IS Guidelines for Crypto	https://www.sogis.eu/documents/cc/crypto/SOGIS-Agreed-Cryptographic-Mechanisms-1.2.pdf.
Ref-6	Verifiable Credentials Data Model 1.0	https://www.w3.org/TR/vc-data-model/
Ref-7	Linked Data Proofs 1.0	https://w3c-ccg.github.io/ld-proofs/
Ref-8	Gaia-X Federation Services Technical Development Requirements	Refer to annex "GXFS_Technical_Development_Requirements"

***Table 2**: References*

1.5 Document Overview

In the following, a general overview of the product in chapter 2 will be provided. Specific requirements for the implementation of the GX-DCS will be given in chapter 3 – divided into interface requirements (3.1.), functional requirements (3.2.) and non-functional requirements (3.3-3.5). The functionality of the GX-DCS with all the features it is offering can be viewed in chapter 4, whereas chapter 5 provides a short wrap-up with regards to the verification of an implementation based on this specification.

2. Product Overview

The promise of Gaia-X is twofold: First, often depicted as the lower half of the X, Gaia-X will be a federated meta-cloud of European platform and infrastructure service providers. This is a mouthful, but it means just that a mesh of existing and vastly diverse PaaS and IaaS solutions are being built, governed by the overarching Gaia-X principles, and connected by the Gaia-X Federation Services. Second, often depicted as the upper half of the X, Gaia-X promises a fully-fledged data ecosystem to enable flexible, secure, and sovereign data exchange. The Gaia-X Data Contract Service (GX-DCS), which is specified in this document, will be the crystallization core for the Gaia-X data ecosystem. The functionality of this version is limited to a handful of essentials, but this is a strength, not a weakness – the scope is clear and comprehensible, GX-DCS can be implemented as a neat stateless microservice within short order, and future improvements such as fully autonomous negotiations, data exchange connectors for technical policy enforcement, data apps gravitating to the data, and ad-hoc virtual enclaves can be mounted on top of GX-DCS. Naturally, that takes time.

Every data transaction within Gaia-X consists of the following parts, which are threaded together by GX-DCS:

1. A **Data Asset**: Without Data Assets no data ecosystem. It is assumed that not only existing datasets, databases, and sensor streams will be part of the Gaia-X data ecosystem, but also that new business models and future start-ups will pop up everywhere as soon as the opportunities and benefits become apparent. A Data Asset can be any static or dynamic data item that is a potential

"thing" to be bought or rented, e.g., a database of high-quality photos for ANN training, sensor streams from environmental measuring stations, models for 3-D printers, CCTV footage from a certain location and time interval, etc.

2. The **Self-Description** of the Data Asset: The Self-Description (SD) is the DNA of Gaia-X – every Service and Participant has one, and so do Data Assets. The SD of Data Assets has some rather tight restrictions and requirements, because the SD contains not only the usual metadata information like data type, size, content description, etc., but also legal statements that transform the SD from a set of nice key-value pairs into the template of a legally binding contract (see also *Data Asset Registration* in section 3.2). The underlying reason is this: Data Contract negotiation leads to the transmission of the Data Asset either manually or automatically triggered, but as soon as the Data Asset is in the hands of the Data Consumer, enforcement of the Data Asset's usage policies is impossible. Therefore, legal policy enforcement becomes the natural fallback option. But in order to get a trustworthy legal basis, a real contract must be made between Data Provider and Data Consumer (see below). Thus, the Self-Description is a *Ricardian contract*: A contract at law that is both human-readable and machine-readable, cryptographically signed and rendered tamper-proof, verifiable in a decentralized fashion, and electronically linked to the subject of the contract, i.e., the Data Asset. Note that a data contract is optionally a *smart contract* in the sense that the data transaction may execute automatically if and when the necessary conditions are fulfilled (see also *Negotiations* in section 3.2).
3. **Publication** of the Data Asset Self-Description: While the Data Asset resides with the Data Provider, the Data Asset SD has to be published in the Gaia-X Federated Catalogue (GX-FC) by the Data Provider.
4. **Search and Choice** of a Data Asset by the prospective Data Consumer: Data Consumers can either access GX-FC's API directly to search, filter, and otherwise navigate available Data Asset SDs, or they can make use of the Gaia-X Portal, which accesses GX-FC behind the scenes and provides a neat GUI for browsing Data Asset SDs. Naturally, GX-FC must be able to search for and filter by all relevant Data Asset SD parameters, beginning with data type, transmission type, keywords, and not quite ending with compensation (read: price), negotiatiability, and transmission details. Nevertheless, let's assume the Data Consumer has found a suitable Data Asset either in the Portal or directly in GX-FC. As a last step, the Data Consumer electronically signals their interest and is then forwarded to GX-DCS, where contract negotiation takes place (see below).
5. Contract **Negotiation and Signing**: In case of solid, low-price Data Assets like autonomously harvested picture or sound databases, contract negotiation (see also Making an Agreement in section 3.2) consists of a simple signature of the Data Contract, which is then forwarded to the Data Provider who then initiates the data transmission. In many cases, however, Data Provider may wish to choose their customers or, at any rate, determine the price and terms of usage depending on the specific Data Consumer in question. To this effect, GX-DCS supports automatic and semi-automatic contract negotiations, albeit to a limited extent. The general negotiation pattern works like this: First of all, the Data Provider defines general rules for the Data Asset (see 2 above), which

can include placeholders. If the Data Asset can be received using one standard Data Contract (automatic execution), a Data Consumer may order the asset as described by filling in Customer specific details and signing the Data Contract. The Data Provider has the possibility to make a confirmation of such a Data Contract obligatory before it takes effect.
If GX-DCS encounters contract details that preclude automatic execution due to placeholders or conditional policies, Data Consumers are required to fill the template with an offer from their side and sign this as a Data Contract proposal. The offer is then forwarded to the Data Provider who signs the Data Contract in case of agreement with the proposal. If a finalized Agreement is reached, the data transmission may begin. Note that delayed or timed data transfer could be indicated in future versions of the Data Asset SD.

6. Data **Transmission and Logging**: "transmission" as opposed to "exchange" since the Data Asset goes from Provider to Consumer; there's no exchange as such. However, a pre-defined or negotiated compensation in the shape of a fixed price or a subscription fee might come into play. Logging, then, is realized by the companion service of GX-DCS: The Data Exchange Logging Service (GX-DELS). Incidentally, GX-DCS issues and renews log authorization tokens needed for logging (see also *Get Log Token* in section 3.2). GX-DCS supports the following modes of data transmission:
    1. *Pull*: Direct download from an endpoint defined by the Data Provider
    2. *Stream*: Continuous download from an endpoint defined by the Data Provider; this is a special case of pull transmission
    3. *Push*: Data Provider submits the Data Asset to some endpoint defined by the Data Consumer
    4. *Publish/Subscribe*: Data Provider submits new versions of the Data Asset to some endpoint defined by the Data Consumer; this is a special case of push transmission for frequently updated Data Assets (e.g., weather forecasts, stock market alert)
7. **Billing**: Although it's clear that a fully functioning data ecosystem entails a marketplace where things can be bought and sold at a price, Gaia-X won't contain "Billing-as-a-Service" in its first incarnation. Nevertheless, the Data Asset SD surely contains pricing details; only the payment process itself must be realized by the Data Provider. In future versions, a Gaia-X billing service could look up the data transmission logs in the appropriate GX-DELS instance and initiate the money transfer.

It should have become clear that GX-DCS is *the* central part of the Gaia-X data ecosystem – a concise and compact service that is apt to become the foundation of much more complex services. Right now, GX-DCS is just a little factory for Ricardian smart contracts, but imagine advanced use cases like these:

- Service X, equipped with a Euro-budget, autonomously purchases Data Assets based on preliminary machine learning results gleaned from data samples… A human user, for instance, may have demanded the solution of a particular classification problem from service X, which is then solved without further human interaction. Result is there, money is gone.

- Data Assets from several sources are subscribed to, aggregated, and analyzed by service Y to produce error detection and data quality improvements that are then sold back to the original Data Providers in a loop which only terminates if quality gains become negligible.
- A clever service Z consumes Data Assets from different domains concerning a particular entity (e.g., weather data, traffic data, public event calendars, and crime statistics of a city), to infer higher-level patterns that can't be learned from one dataset alone. This gives rise to information or even knowledge that can be expressed as probabilistic logical rules, which are highly valuable for makers of self-driving cars, policing, and event management – data might be the new oil, but knowledge is what you put into your tank!

In most of the advanced use cases, the Data Consumer won't be a human being, but rather an autonomous Service instance. This fact makes automatic contract execution a strategic feature of Gaia-X Agreements.

2.1 Product Perspective

GX-DCS is a stateless microservice; think of it as a little blob of backend logic with a handful of interfaces and a thin layer of GUI, but without any kind of database, occasional performance-enhancing local caching, and a tiny vault for its private keys notwithstanding. But GX-DCS doesn't exist in a vacuum: As shown in Figure 1, many Gaia-X Federation Services (GXFS) are needed to accomplish even the most basic data ecosystem:

- The Gaia-X **Authentication and Authorization** is needed to provide an anchor of trust; from the onboarding of new Gaia-X Participants up to and beyond digitization of to-be-invented data quality labels, the involved Services provides an – but not *the* – interface between the Gaia-X and the rest of the world and anchors all other chains of trust within Gaia-X.
- The Gaia-X **Federated Catalogue**: This is the place where Data Providers publish Data Asset SDs. The fine details don't belong here, but in general, GX-FC expects Data Providers to operate an endpoint where GX-FC can subscribe to Data Asset SDs.
- The Gaia-X **Portal**: While some Participants will probably wire up their own systems with the various GXFS backends, a frontend for browsing and searching and comparing Gaia-X Assets of all kinds is an important service in itself. Naturally, behind the scenes, the Portal gets all its information from GX-FC – in fact, the Portal is largely a GUI of GX-FC.
- The Gaia-X **Trust Service**: The Trust Service is paramount for validating signatures. To this end, it provides functionality to resolve DIDs and retrieve public keys of any Participant. The Trust Service is only part of Gaia-X's Identity and Access Management framework, but for GX-DCS it's the only part that's needed.
- The Gaia-X **Data Exchange Logging Service**: GX-DELS complements GX-DCS – the latter is almost completely processing, the former is almost completely storage. In future manifestations of Gaia-X, data transaction logs will play a crucial part in billing, monitoring, and auditing. GX-DELS expects

valid Log Tokens from everyone attempting to log things, and those very log tokens are issued by GX-DCS.

GX-DCS is coupled to at least five other GXFS. However, this coupling is realized through precisely defined and sharply cut interfaces; the inside of GX-DCS is and must be a black box to the rest of Gaia-X.

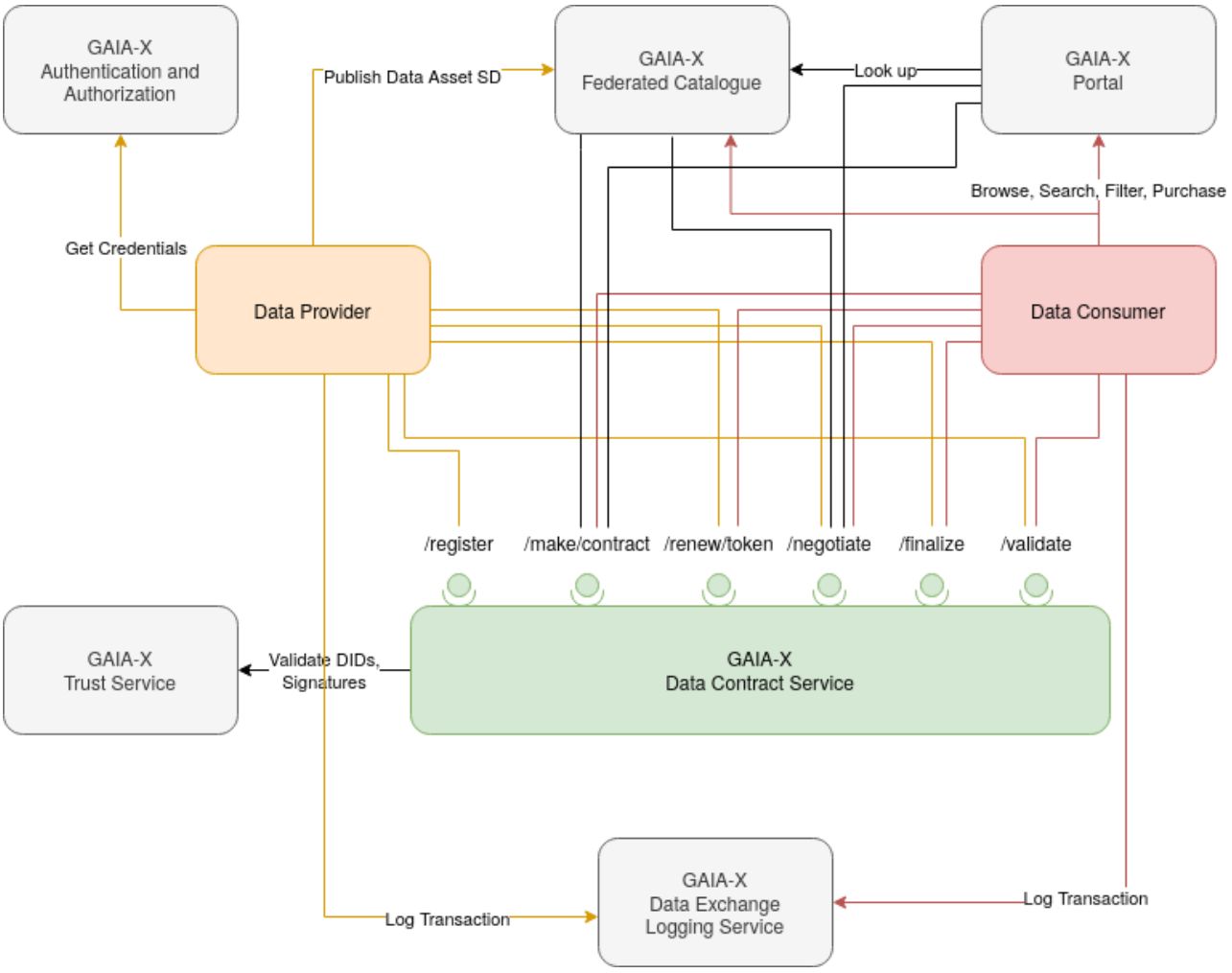


Figure 1: *Functional Overview Federation Services*

There is no arrow in Figure 1 between Data Provider and Data Consumer, concerning actual data transfer. That's for the simple reason that data transmission is out of band in v1; the Data Provider makes an end-point available for secure data transfer. In later stages of Gaia-X, of course, dedicated secure data transfer channels could be built on top of the foundation described in this document.

2.2 Product Functions

The main six functions, as indicated in Figure 1, are (see section 3.1 for the interface details and section 3.2 for a description of the functionality):

- **Data Asset Registration** (/register)
- **Making a Contract** (/make/contract)
- **Contract Negotiation** (/negotiate)
- **Finalization of a Contract** (/finalize)
- **Contract Validation** (/validate)
- **Get Log Token** (/log/token)

As discussed at the beginning of this chapter, these functions are relevant in different phases of the Data Asset lifecycle.

2.3 Product Constraints

Unless and until EU-wide legislation for data trading exists, Gaia-X provides general terms for data trading, or Data Providers and Data Consumers have made a deal outside Gaia-X, all the legal aspects of data delivery have to be contained in the Data Asset SD.

Since there's no EU contract law, Data Provider and Data Consumer have to agree on a *choice of law*, e.g., which EU member state's national law provides the legal basis of the contract. In practice, Data Providers will simply indicate the choice of law, which the Data Consumer can take or leave. In future versions, several options might be possible.

The largest part of the Data Asset SD probably consists of the usual metadata like timestamp, datatype, access URL, transaction type, etc., and formalized usage policies. Nevertheless, a number of points need to be covered that can't be easily quantified or categorized; these things will be bundled in the "general terms" of the Data Asset SD. And as much a general Gaia-X data delivery Agreement would simplify matters, the added flexibility of Data Asset-specific contracts has a charm of its own.

In order to simplify creating new contract templates, the essential parts of a data delivery contract are briefly mentioned:

1. **Subject matter** of the contract: A brief description referencing the other SD properties in a general way.
2. **Provisioning**: Mentions the rights and duties of Provider and Consumer in a general way, e.g., "the Data Provider makes the Data Asset available to the Data Consumer" and "the Data Consumer is required to perform necessary steps for data transaction and logging" etc.
3. **Usage rights**: Mentions qualitative usage rights like non-exclusivity, non-transferability, the right to store and process, prohibition to alter the Data Asset, etc., referencing the formal usage policies. Mentions also that ownership remains with the Data Provider.

4. **Warranty**: Guarantees the correctness of the statements in the Data Asset SD, in particular the right of the Data Provider to sell the Data Asset, legality of the Data Asset itself, and the promised data quality.
5. **Liability**: Details the circumstances under which the Data Provider is liable for damages resulting from the Data Asset, e.g., breach of warranty, product liability, fraud, willful intent, and gross negligence.
6. **Prohibition of Identification**: Insofar the Data Asset contains anonymized data that could theoretically be analyzed to infer identity features of persons, such analyses should be explicitly prohibited, defining, or at least indicating a contractual penalty.
7. **Confidentiality**: Mentions the confidentiality of the data contract, data transactions, and the Data Asset itself, including the duty of the Data Consumer for Data Asset access protection. Mentions also whether this paragraph outlasts the lifespan of the contract.
8. **Contract Lifespan**: Mentions when the contract begins, how it can end, and that the Data Consumer is required to delete the Data Asset if and when the formal usage policies dictate.
9. **Final Clause**: Mentions general terms of use (if relevant), immutability of the contract, and the famous severability clause.

> As long as the general terms outlined above aren't part of the general terms and conditions signed during Participant onboarding, **the Data Consumer must be a human being**. Use cases involving automated Data Contracts need Data Asset SDs to be completely free of ambiguities.

2.4 User Classes and Characteristics

Please note, that GX-DCS does not store anything persistently, least of all the Data Contracts themselves. In principle, every Gaia-X Participant may use any GX-DCS function. Whether the Participant acts as a Data Provider or a Data Consumer or an undefined entity just wanting to validate a finalized Data Contract is induced implicitly from interface usage, e.g., someone who calls "/register" will be treated as a Data Provider, etc.

2.5 Operating Environment

Please refer to Ref-8.

2.6 User Documentation

User documentation must be provided alongside the GX-DCS reference implementation. The documentation must contain sufficient information to deploy, operate, and use GX-DCS. This includes:

- **Deployment Manual:** The contractor must provide documentation describing deployment procedures for GX-DCS, including roll-out and roll-back.

- **Operations Manual:** The contractor must provide documentation describing all modes of operation including error handling, instructions for secure operation, and means of recovery.
- **Software Architecture:** The contractor must provide an overview of GX-DCS's software architecture, including a component view, a process view, and implementation considerations.
- **Security Concept:** The contractor must provide a security concept for the secure operation of GX-DCS.
- **Interface Usage:** The developer must provide descriptions of all GX-DCS interfaces and their usage for Gaia-X Participants, i.e., Data Providers and Data Consumers.

Further requirements regarding the documentation can be found in Ref-8.

2.7 Assumptions and Dependencies

Take special note that GX-DCS directly depends on other GXFS:

- During the Onboarding Process described in WP4 the GX-DCS registers a DID and a corresponding key pair. This is essential so that each GX-DCS instance can later sign Data Contracts – and others can validate these signatures in a decentralized fashion. This also includes compliance with the specifications for the Authentication and Authorization of Services from WP1 [GXFS-1].
- Moreover, DID resolving has to be aligned with the Core Functions of WP1's Trust Service [GXFS-2].
- While GX-DCS contains both an API and a GUI for Data Asset registration, the GX-DCS will not store Data Asset SDs since it is meant to be a stateless microservice. Instead, the Data Provider is expected to register Data Asset SDs in the GX-FC according to WP2's subscription requirements. The GX-DCS will then utilize the API of the GX-FC to retrieve the registered Data Asset SD whenever it is necessary for the making or negotiation of a contract.
- The Data Asset Self-Description will be implemented as a Verifiable Credential or Verifiable Presentation according to the W3C Recommendation [Ref-6] and the digital signatures for achieving a finalized Data Contract will be Linked Data Proofs [REF-7].

Additionally, other Federation Services will in turn be using interfaces specified in this document or need to interoperable work with this implementation. In particular, the Data Exchange Logging Service (GX-DELS) of WP3[2] relies on the issued log tokens from GX-DCS to determine whether a logging request is valid or not. Additionally, the Gaia-X Portal from WP5 might use these interfaces for making and negotiating contracts to directly offer those options to Data Consumers. .

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

3. Requirements

Requirements are prefixed “GX-DCS.IR” for interfaces, “GX-DCS.FR” for functionality, and “GX-DCS.NFR” for non-functional requirements.

3.1 Interfaces

3.1.1 User Interfaces

ID & Title	Description	Verification Method
GX-DCS.IR.001 **Data Asset Registration GUI**	The GX-DCS MUST offer a graphical interface to make Data Asset registration more accessible to small and first-time Data Providers. It MAY be a conversational, wizard-style UI based on GX-FC’s Data Asset SD model in its current version [Ref-2].	Review of Documentation Testing

***Table 3**: Requirements User Interfaces*

3.1.2 Hardware Interfaces

The GX-DCS holds one or more private keys for cryptographic signing and trustworthy signature validation.

ID & Title	Description	Verification Method
GX-DCS.IR.002 **Interface for use of Secrets**	Private keys of the GX-DCS MUST be secured using hardware mechanisms such as a HSM or TPM. The GX-DCS MUST utilize standardized interfaces to store and utilize the secrets in a non-manipulatable and protected way. For scalability, performance, and reliability reasons it SHOULD be realized in a virtualized environment.	Review of Documentation Testing

***Table 4:** Requirements Hardware Interfaces*

3.1.3 Communication Interfaces

The endpoint logic must not only care for the happy paths (as described in section 3.2), but also for the **unhappy paths**, which are indicated by the many possible error responses. Rigorous error-checking is not just a nicety but fundamental due diligence.

3.1.3.1. Data Asset Registration

ID & Title	Description	Verification Method
GX-DCS.IR.003 **Data Asset Registration Endpoint**	The GX-DCS MUST offer the communication interface for Data Asset Registration described below.	Review of Documentation Testing

***Table 5**: Requirements Communication Interfaces - Data Asset Registration*

The GX-DCS offers a Data Asset Registration endpoint, which on success returns a valid Data Asset Self-Description signed by the GX-DCS.

- **URL**

  /register

- **Method:**

  POST

- **URL Params**

  None

- **Request Body**
  Data Asset Self-Description as JSON-LD. The versioned Data Asset SD model is provided by GX-FC, where it can be freely downloaded [Ref-2]. The Data Asset Self-Description must be signed by the Data Provider beforehand, by adding the signature as a Proof.

- **Success Response:**
  - **Code:** 200
    Response Body Content: Error-checked, and validated Data Asset Self-Description (as JSON-LD) signed by GX-DCS.

- **Error Response:**
  - **Code:** 400
    Meaning: Bad request – missing or malformed request body (see also 415); response must contain a textual representation of the precise error
  - **Code:** 401
    Meaning: Unauthorized – invalid Data Provider signature
  - **Code:** 403
    Meaning: Forbidden – the requesting Participant is not a human being (this restriction might be removed in future versions)
  - **Code:** 404
    Meaning: Not found – Data Provider DID could not be resolved

- **Code:** 413
  Meaning: Payload too large – Request body exceeds 1 MB (this is an arbitrary value, but some such check is required to prevent accidental or malicious overloads)
- **Code:** 414
  Meaning: Request URI too long – occurs if the URI contains anything but "/register"
- **Code:** 415
  Meaning: Unsupported media type – occurs if request body isn't proper JSON-LD
- **Code:** 424
  Meaning: Failed dependency – could not validate Data Asset Self-Description model (via GX-FC)
- **Code:** 429
  Meaning: Too many requests – each Data Provider is allowed to use this endpoint at most once every two seconds (this value is arbitrary, but some fixed maximum access frequency is required). This error code is only applicable if DIDs are cached by the GX-DCS (see 3.2.6).
- **Code:** 451
  Meaning: Unavailable for legal reasons – Data Provider's Gaia-X Participant status has been revoked
- **Code:** 5xx
  Meaning: The usual server errors

- **Sample Call:**
  POST Body:


```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://w3id.org/gaia-x/core/context/DataAsset.jsonld"
  ],
  "type": "VerifiablePresentation",
  "verifiableCredential": [{
    "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://w3id.org/gaia-x/core/context/DataAsset.jsonld"
    ],
    "credentialSubject": {
      "@id": "http://example.org/data-asset-1",
      "@type":"gax:DataAsset",
      "gax:title":"Example title",
      "gax:description":"Example description",
      "gax:keyword":["key", "word"],
      "gax:category":["example"],
      "gax:publisher":"?publisherDID",
      "gax:creator":"?creatorDID",
      "gax:language":"http://id.loc.gov/vocabulary/iso639-1/en",
      "gax:distribution":{
        "gax:title":"Example distribution title",
        "gax:description":"Example distribution description",
        "gax:created":"2021-01-23T18:25:43.511Z",
        "gax:modified":"2021-01-25T12:20:34.007Z",
        "gax:mediaType":"text/csv",
        "gax:byteSize":"100000",
        "gax:accessURL":"www.example.com/data/example.csv"
      },
      "gax:created":"2021-01-23T12:21:23.876Z",
      "gax:modified":"2021-01-24T14:45:03.517Z",
      "gax:containsPersonalData":false,
      "gax:sampleAvailable":false,
      "gax:contractOffer":{
        "@type": "gax-GX-DCS:contractOffer",
        "gax:choiceOfLaw":"iso:Germany",
        "gax:generalTerms":"Example text for the general terms",
        "gax:confirmationRequired":false,
        "gax:loggingMode":"gax:LoggingMandatory",
        "gax:circulationDetails":"Example text for the circulation details",
        "gax:permission":{
          "@type": "gax:Permission",
          "gax:assigner": "?providerDID",
          "gax:target": "?AssetURI",
          "gax:action": "gax:USE",
          "gax:negotiable":false,
          "gax:postDuty": {
            "@type": "gax:Duty",
            "gax:action": { "@id": "gax:LOG" }
          }
        }
      }
    }}],
  "proof": [{
    "type": "Ed25519Signature2018",
    "proofPurpose": "assertionMethod",
    "created": "2019-08-23T20:21:34Z",
    "verificationMethod": "did:provider:123456#key1",
    "jws": "eyJ0eXAiOiJK...gFWFOEjXk"
  }]
}
```


- **Notes:**
  The Data Asset SD schema is bound to undergo evolutionary changes – minor corrections, additions, and structural updates are sure to happen. Therefore, the model of the Data Asset SD is centrally available via GX-FC [Ref-2] and newer versions of the DASD model might be published as Gaia-X evolves.

3.1.3.2. Making a Contract

ID & Title	Description	Verification Method
GX-DCS.IR.004 **Make Contract Endpoint**	The GX-DCS MUST offer the communication interface for making a contract that is described below.	Review of Documentation Testing

***Table 6**: Requirements Communication Interfaces - Making a Contract*

The GX-DCS offers an endpoint for directly making finalized Agreements based on an available Data Asset Self-Description. On success, it returns a finalized Agreement (i.e., a mutually signed Data Contract).

- **URL**

  /make/contract

- **Method:**

  POST

- **URL Params**

  None

- **Request Body**
  Agreement (Data Asset Self-Description with Consumer details added and placeholders filled) signed by the Data Consumer.

- **Success Response:**

  - **Code:** 200
    Content: Finalized Agreement (i.e., mutually signed data contract), which is also forwarded to the Data Provider if this request succeeds. This finalized Agreement, then, can be sent to the "/log/token" endpoint to retrieve a Log Token which authenticates access at GX-DELS.

- **Error Response:**

  - **Code:** 400
    Meaning: Bad request – missing or malformed request body (see also 415); response must contain a textual representation of the precise error

  - **Code:** 401
    Meaning: Unauthorized – invalid Data Provider signature or invalid Data Consumer signature

  - **Code:** 403
    Meaning: Forbidden – the "general terms" are not empty, but the requesting Participant is not a human being (this restriction might be removed in future versions)

- **Code:** 404
  Meaning: Not found – Data Provider DID or Data Consumer DID could not be resolved
- **Code:** 413
  Meaning: Payload too large – Request body exceeds 1 MB (this is an arbitrary value, but some such check is required to prevent accidental or malicious overloads)
- **Code:** 414
  Meaning: Request URI too long – occurs if the URI contains anything but "/make/contract"
- **Code:** 415
  Meaning: Unsupported media type – occurs if request body isn't proper JSON-LD
- **Code:** 424
  Meaning: Failed dependency – could not validate Data Asset Self-Description model (via GX-FC)
- **Code:** 426
  Meaning: Upgrade required – Data Asset SD model is no longer supported
- **Code:** 429
  Meaning: Too many requests – each Data Consumer is allowed to use this endpoint at most once every two seconds (this value is arbitrary, but some fixed maximum access frequency is required). This error code is only applicable, if DIDs are cached by the GX-DCS (see 3.2.6.).
- **Code:** 451
  Meaning: Unavailable for legal reasons – Data Provider's or Data Consumer's Gaia-X Participant status has been revoked (response must contain a textual explanation)
- **Code:** 5xx
  Meaning: The usual server errors

- **Sample Call:**
  POST Body:


```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://w3id.org/gaia-x/core/context/DataAsset.jsonld"
  ],
  "type": "VerifiablePresentation",
  "verifiableCredential": [{
    "@context": [
      "https://www.w3.org/2018/credentials/v1",
      "https://w3id.org/gaia-x/core/context/DataAsset.jsonld"
    ],
    "credentialSubject": {
        "@id":"?AssetURI",
        "@type":"gax:DataAsset",
        "gax:title":"Example title",
        "gax:description":"Example description",
        "gax:keyword":["key", "word"],
        "gax:category":["example"],
        "gax:publisher":"?publisherDID",
        "gax:consumer":"?consumerDID",
        "gax:creator":"?creatorDID",
        "gax:language":"http://id.loc.gov/vocabulary/iso639-1/en",
        "gax:distribution":{
          "gax:title":"Example distribution title",
          "gax:description":"Example distribution description",
          "gax:created":"2021-01-23T18:25:43.511Z",
          "gax:modified":"2021-01-25T12:20:34.007Z",
          "gax:mediaType":"text/csv",
          "gax:byteSize":"100000",
          "gax:accessURL":"www.example.com/data/example.csv"
        },
        "gax:created":"2021-01-23T12:21:23.876Z",
        "gax:modified":"2021-01-24T14:45:03.517Z",
        "gax:containsPersonalData":false,
        "gax:sampleAvailable":false,
        "gax:contractOffer":{
          "@type": "gax-GX-DCS:contractOffer",
          "gax:choiceOfLaw":"iso:Germany",
          "gax:generalTerms":"Example text for the general terms",
          "gax:loggingMode":"gax:LoggingMandatory",
          "gax:confirmationRequired":false,
          "gax:circulationDetails":"Example text for the circulation details",
          "gax:permission":{
            "@type": "gax:Permission",
            "gax:assigner": "?providerDID",
            "gax:assignee":"?consumerDID",
            "gax:target": "?AssetURI",
            "gax:action": "gax:USE",
            "gax:negotiable":false,
            "gax:postDuty": {
              "@type": "gax:Duty",
              "gax:action": { "@id": "gax:LOG" }
            }
          }
        }
      },
    "proof": [{
      "type": "Ed25519Signature2018",
      "proofPurpose": "assertionMethod",
      "created": "2019-08-23T20:21:34Z",
      "verificationMethod": "did:provider:123456#key1",
      "jws": "ac98eXAOOiJK...gHWFOEjXj"
    },
      {
        "type": "Ed25519Signature2018",
        "proofPurpose": "contractAgreement",
        "created": "2019-08-23T20:22:45Z",
        "verificationMethod": "did:consumer:123456#key1",
        "jws": "eyJ0eXAiOiJK...gFWFOEjXk"
      }]}],
  "proof": [{
    "type": "Ed25519Signature2018",
    "proofPurpose": "assertionMethod",
    "created": "2019-08-23T20:21:51Z",
    "verificationMethod": "did:GX-DCS:123456#key1",
    "jws": "wqiOUgf90iJK...j9I5V1jLo"
  }]
}
```

- **Notes:**
  "/make/contract" is a simple endpoint insofar no further interaction between GX-DCS and the contracting parties is required. This endpoint is specifically tailored for Data Contract Offers, i.e., those cases where the Data is offered "as is" and no properties are negotiable– every Gaia-X Participant gets the same thing for the same price. This case is comparable to a vending machine in the real world.

3.1.3.3. Contract Negotiation

ID & Title	Description	Verification Method
GX-DCS.IR.005 **Contract Negotiation Endpoint**	The GX-DCS MUST offer the communication interface for contract negotiation that is described below.	Review of Documentation Testing

***Table 7**: Requirements Communication Interfaces - Contract Negotiation*

The GX-DCS offers an endpoint for starting a contract negotiation. On success, it returns a confirmation that the Data Contract was valid and forwarded to the Data Provider.

- **URL**

  /negotiate

- **Method:**

  POST

- **URL Params**

  None

- **Request Body**
  Agreement (Data Asset Self-Description with Consumer details added and placeholders filled) signed by the Data Consumer

- **Success Response:**
  - **Code:** 202
    Meaning: Accepted – indicates that the Agreement has been forwarded to the Data Provider for confirmation

- **Error Response:**
  - **Code:** 400
    Meaning: Bad request – missing or malformed request body (see also 415); response must contain a textual representation of the precise error
  - **Code:** 401
    Meaning: Unauthorized – invalid Data Provider signature or invalid Data Consumer signature

- **Code:** 403
  Meaning: Forbidden – the "general terms" are not empty, but the requesting Participant is not a human being (this restriction might be removed in future versions)
- **Code:** 404
  Meaning: Not found – Data Provider DID or Data Consumer DID could not be resolved
- **Code:** 413
  Meaning: Payload too large – Request body exceeds 1 MB (this is an arbitrary value, but some such check is required to prevent accidental or malicious overloads)
- **Code:** 414
  Meaning: Request URI too long – occurs if the URI contains anything but "/negotiate"
- **Code:** 415
  Meaning: Unsupported media type – occurs if request body isn't proper JSON-LD
- **Code:** 424
  Meaning: Failed dependency – could not validate Data Asset Self-Description model (via GX-FC)
- **Code:** 426
  Meaning: Upgrade required – Data Asset SD model is no longer supported
- **Code:** 429
  Meaning: Too many requests – each Data Consumer is allowed to use this endpoint at most once every two seconds (this value is arbitrary, but some fixed maximum access frequency is required). This error code is only applicable, if DIDs are cached by the GX-DCS (see 3.2.6).
- **Code:** 451
  Meaning: Unavailable for legal reasons – Data Provider's or Data Consumer's Gaia-X Participant status has been revoked (response must contain a textual explanation)
- **Code:** 5xx
  Meaning: The usual server errors

- **Sample Call:**

  POST Body:


```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://w3id.org/gaia-x/core/context/DataAsset.jsonld"
  ],
  "type": "VerifiablePresentation",
  "verifiableCredential": [
    {
      "@context": [
        "https://www.w3.org/2018/credentials/v1",
        "https://w3id.org/gaia-x/core/context/DataAsset.jsonld"
      ],
      "credentialSubject": {
        "@id": "?AssetURI",
        "@type": "gax:DataAsset",
        "gax:title": "Example title",
        "gax:description": "Example description",
        "gax:keyword": [
          "key",
          "word"
        ],
        "gax:category": [
          "example"
        ],
        "gax:publisher": "?publisherDID",
        "gax:consumer": "?consumerDID",
        "gax:creator": "?creatorDID",
        "gax:language": "http://id.loc.gov/vocabulary/iso639-1/en",
        "gax:distribution": {
          "gax:title": "Example distribution title",
          "gax:description": "Example distribution description",
          "gax:created": "2021-01-23T18:25:43.511Z",
          "gax:modified": "2021-01-25T12:20:34.007Z",
          "gax:mediaType": "text/csv",
          "gax:byteSize": "100000",
          "gax:accessURL": "www.example.com/data/example.csv"
        },
        "gax:created": "2021-01-23T12:21:23.876Z",
        "gax:modified": "2021-01-24T14:45:03.517Z",
        "gax:containsPersonalData": false,
        "gax:sampleAvailable": false,
        "gax:contractOffer": {
          "@type": "gax:contractOffer",
          "gax:choiceOfLaw": "iso:Germany",
          "gax:generalTerms": "Example text for the general terms",
          "gax:confirmationRequired": true,
          "gax:loggingMode": "gax:LoggingMandatory",
          "gax:circulationDetails": "Example text for the circulation details",
          "gax:permission": [
            {
              "@type": "gax:Permission",
              "gax:assigner": "?providerDID",
              "gax:target": "?someDataAssetID",
              "gax:action": "gax:COMPENSATE",
              "gax:negotiable": true,
              "gax:constraint": {
                "@type": "gax:Constraint",
                "gax:leftOperand": "gax: PAY_AMOUNT",
                "gax:operator": "gax:EQ",
                "gax:rightOperand": {
                  "@value": "?price",
                  "@type": "http://www.w3.org/2001/XMLSchema#double"
                }
              }
            }
          ]
        }
      },
      "proof": [
        {
          "type": "Ed25519Signature2018",
          "proofPurpose": "contractAgreement",
          "created": "2019-08-23T20:21:34Z",
          "verificationMethod": "did:consumer:123456#key1",
          "jws": "eyJ0eXAiOiJK...gFWFOEjXk"
        }
      ]
    }
  ]
}
```

- **Notes:**
  The "/negotiate" and the "/finalize" endpoint describe below are used to implement the possibility for a Data Provider to merely provide an Invitation for Data Contract Offers in the GX-FC but keeping control over the contracts really being made then. For that purpose, the Data Provider can either set the property gax:confirmationRequired for an otherwise non-negotiable DASD or utilize placeholders (gax:negotiable to get more flexible offers from Data Consumer. The Providers need to offer an endpoint (that the GX-DCS can transfer contract offers to) utilizing the property gax:hasLegallyBindingAddress in their Self Descriptions. This endpoint could be used to automatically evaluate Agreements or to establish manual Confirmation flows on the Provider side.

The automated contract negotiation at the Data Provider endpoint has the potential of being utilized to discriminate certain Data Consumers. While economic competition dictates that not all Data Consumers can be treated the same and don't have to be, Data Providers must take special heed to **comply with national and EU Cartel Law** and other relevant anti-discrimination laws. The GX-DCS is not responsible for any rules utilized or applied by the Data Providers.

3.1.3.4. Contract Finalization

ID & Title	Description	Verification Method
GX-DCS.IR.006 **Finalize Endpoint**	The GX-DCS MUST offer the communication interface for finalizing the endpoint that is described below.	Review of Documentation Testing

***Table 8**: Requirements Communication Interfaces - Contract Finalization*

The GX-DCS offers an endpoint for finalizing a contract negotiation. On success, it finalizes the Agreement, signs it, returns it to the Data Provider and sends it to the Data Consumer.

- **URL**

  /finalize

- **Method:**

  POST

- **URL Params**

  None

- **Request Body**
  Agreement (Signed by the Data Consumer and Data Provider)

- **Success Response:**

  - **Code:** 200
    Response Body Content: Finalized Agreement (Now also signed by the GX-DCS), which is

also forwarded to the Data Consumer if this request succeeds. This finalized Agreement, then, can be sent to the "/log/token" endpoint to retrieve a Log Token which authenticates access at GX-DELS.

- **Error Response:**
  - **Code:** 400
    Meaning: Bad request – missing or malformed request body (see also 415); response must contain a textual representation of the precise error
  - **Code:** 401
    Meaning: Unauthorized – invalid Data Provider or Data Consumer signature
  - **Code:** 403
    Meaning: Forbidden – the requesting Participant is not a human being (this restriction might be removed in future versions)
  - **Code:** 404
    Meaning: Not found – Data Provider DID or Data Consumer DID could not be resolved
  - **Code:** 413
    Meaning: Payload too large – Request body exceeds 1 MB (this is an arbitrary value, but some such check is required to prevent accidental or malicious overloads)
  - **Code:** 414
    Meaning: Request URI too long – occurs if the URI contains anything but "/finalize"
  - **Code:** 415
    Meaning: Unsupported media type – occurs if request body isn't proper JSON-LD
  - **Code:** 424
    Meaning: Failed dependency – could not validate Data Asset Self-Description model (via GX-FC)
  - **Code:** 429
    Meaning: Too many requests – each Data Provider is allowed to use this endpoint at most once every two seconds (this value is arbitrary, but some fixed maximum access frequency is required). This error code is only applicable, if DIDs are cached by the GX-DCS (see 3.2.6).
  - **Code:** 451
    Meaning: Unavailable for legal reasons – Data Provider's or Data Consumer's Gaia-X Participant status has been revoked (response must contain a textual explanation)
  - **Code:** 5xx
    Meaning: The usual server errors

- **Sample Call:**

  POST Body:


```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://w3id.org/gaia-x/core/context/DataAsset.jsonld"
  ],
  "type": "VerifiablePresentation",
  "verifiableCredential": [
    {
      "@context": [
        "https://www.w3.org/2018/credentials/v1",
        "https://w3id.org/gaia-x/core/context/DataAsset.jsonld"
      ],
      "credentialSubject": {
        "@id":"?AssetURI",
        "@type":"gax:DataAsset",
        "gax:title":"Example title",
        "gax:description":"Example description",
        "gax:keyword":["key", "word"],
        "gax:category":["example"],
        "gax:publisher":"?publisherDID",
        "gax:consumer":"?consumerDID",
        "gax:creator":"?creatorDID",
        "gax:language":"http://id.loc.gov/vocabulary/iso639-1/en",
        "gax:distribution":{
          "gax:title":"Example distribution title",
          "gax:description":"Example distribution description",
          "gax:created":"2021-01-23T18:25:43.511Z",
          "gax:modified":"2021-01-25T12:20:34.007Z",
          "gax:mediaType":"text/csv",
          "gax:byteSize":"100000",
          "gax:accessURL":"www.example.com/data/example.csv"
        },
        "gax:created":"2021-01-23T12:21:23.876Z",
        "gax:modified":"2021-01-24T14:45:03.517Z",
        "gax:containsPersonalData":false,
        "gax:sampleAvailable":false,
        "gax:contractOffer":{
          "@type": "gax:contractOffer",
          "gax:choiceOfLaw":"iso:Germany",
          "gax:generalTerms":"Example text for the general terms",
          "gax:confirmationRequired":true,
          "gax:loggingMode":"gax:LoggingMandatory",
          "gax:circulationDetails":"Example text for the circulation details",
          "gax:permission": [ {
            "@type": "gax:Permission",
            "gax:assigner": "?providerDID",
            "gax:target": "?someDataAssetID",
            "gax:action": "gax:COMPENSATE",
            "gax:negotiable": true,
            "gax:constraint":{
              "@type": "gax:Constraint",
              "gax:leftOperand": "gax: PAY_AMOUNT",
              "gax:operator": "gax:EQ",
              "gax:rightOperand": {
                "@value": "?price",
                "@type": "http://www.w3.org/2001/XMLSchema#double"
              }
            }
        }]}},
        "proof": [{
          "type": "Ed25519Signature2018",
          "proofPurpose": "contractAgreement",
          "created": "2019-08-23T20:21:34Z",
          "verificationMethod": "did:consumer:123456#key1",
          "jws": "eyJ0eXAiOiJK...gFWFOEjXk"
        },
          {
          "type": "Ed25519Signature2018",
          "proofPurpose": "contractAgreement",
          "created": "2019-08-23T20:56:03Z",
          "verificationMethod": "did:provider:123456#key1",
          "jws": "iyJheXXiOiJK...gFWFOEoXk"
}]}]}
```

- **Notes:**
  This endpoint is intended to finalize the negotiation process. It makes sure, that both parties agree to the Terms of the Agreement and distributes the finalized Agreement to both parties. The Endpoints that are required are found in the Self Descriptions of the respective Participant in the property *gax:hasLegallyBindingAddress.*

3.1.3.5. Get Log Token

ID & Title	Description	Verification Method
GX-DCS.IR.007 **Log Token Endpoint**	The GX-DCS MUST offer the communication interface for getting a log token as described below.	Review of Documentation Testing

Table 9: *Requirements Communication Interfaces - Get Log Token*

The GX-DCS offers an endpoint for getting a Log Token. The prerequisite for that is a valid finalized Agreement which contains optional or mandatory logging. It returns a Log Token if the submitted finalized Agreement is valid.

- **URL**

  /log/token

- **Method:**

  POST

- **URL Params**

  None

- **Request Body**
  Finalized Agreement (i.e., mutually signed data contract)

- **Success Response:**

  - **Code:** 200
    Content: Log Token (must comply with the Log Token definition as described in [GXFS-3])

- **Error Response:**

  - **Code:** 400
    Meaning: Bad request – missing or malformed request body (see also 415); response must contain a textual representation of the precise error
  - **Code:** 401
    Meaning: Unauthorized – invalid Data Provider signature, invalid Data Consumer signature, or invalid GX-DCS signature

- **Code:** 403
  Meaning: Forbidden – Finalized Agreement indicates the logging is not allowed; of course, no Log Token is generated in this case
- **Code:** 404
  Meaning: Not found – Data Provider DID, Data Consumer DID, or GX-DCS DID could not be resolved
- **Code:** 413
  Meaning: Payload too large – Request body exceeds 1 MB (this is an arbitrary value, but some such check is required to prevent accidental or malicious overloads)
- **Code:** 414
  Meaning: Request URI too long – occurs if the URI contains anything but "/log/token"
- **Code:** 415
  Meaning: Unsupported media type – occurs if request body isn't proper JSON-LD
- **Code:** 424
  Meaning: Failed dependency – could not validate Data Asset Self-Description model (via GX-FC)
- **Code:** 429
  Meaning: Too many requests – each Data Provider and Data Consumer is allowed to use this endpoint at most once every sixty seconds (this value is arbitrary, but some fixed maximum access frequency is required). This error code is only applicable, if DIDs are cached by the GX-DCS (see 3.2.6).
- **Code:** 451
  Meaning: Unavailable for legal reasons – Data Provider's, Data Consumer's, or GX-DCS's Gaia-X Participant status has been revoked (response must contain a textual explanation)
- **Code:** 5xx
  Meaning: The usual server errors

- **Sample Call:**
  POST Body:


```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://w3id.org/gaia-x/core/context/DataAsset.jsonld"
  ],
  "type": "VerifiablePresentation",
  "verifiableCredential": [
    {
      "@context": [
        "https://www.w3.org/2018/credentials/v1",
        "https://w3id.org/gaia-x/core/context/DataAsset.jsonld"
      ],
      "credentialSubject": {
        "@id":"?AssetURI",
        "@type":"gax:DataAsset",
        "gax:title":"Example title",
        "gax:description":"Example description",
        "gax:keyword":["key", "word"],
        "gax:category":["example"],
        "gax:publisher":"?publisherDID",
        "gax:consumer":"?consumerDID",
        "gax:creator":"?creatorDID",
        "gax:language":"http://id.loc.gov/vocabulary/iso639-1/en",
        "gax:distribution":{
          "gax:title":"Example distribution title",
          "gax:description":"Example distribution description",
          "gax:created":"2021-01-23T18:25:43.511Z",
          "gax:modified":"2021-01-25T12:20:34.007Z",
          "gax:mediaType":"text/csv",
          "gax:byteSize":"100000",
          "gax:accessURL":"www.example.com/data/example.csv"
        },
        "gax:created":"2021-01-23T12:21:23.876Z",
        "gax:modified":"2021-01-24T14:45:03.517Z",
        "gax:containsPersonalData":false,
        "gax:sampleAvailable":false,
        "gax:contractOffer":{
          "@type": "gax:contractOffer",
          "gax:choiceOfLaw":"iso:Germany",
          "gax:generalTerms":"Example text for the general terms",
          "gax:confirmationRequired":true,
          "gax:loggingMode":"gax:LoggingMandatory",
          "gax:circulationDetails":"Example text for the circulation details",
          "gax:permission": [ {
            "@type": "gax:Permission",
            "gax:assigner": "?providerDID",
            "gax:target": "?someDataAssetID",
            "gax:action": "gax:COMPENSATE",
            "gax:negotiable": true,
            "gax:constraint":{
              "@type": "gax:Constraint",
              "gax:leftOperand": "gax: PAY_AMOUNT",
              "gax:operator": "gax:EQ",
              "gax:rightOperand": {
                "@value": "?price",
                "@type": "http://www.w3.org/2001/XMLSchema#double"
              }
            }
      }]}},
        "proof": [{
          "type": "Ed25519Signature2018",
          "proofPurpose": "contractAgreement",
          "created": "2019-08-23T20:21:34Z",
          "verificationMethod": "did:consumer:123456#key1",
          "jws": "eyJ0eXAiOiJK...gFWFOEjXk"
        },
          {
            "type": "Ed25519Signature2018",
            "proofPurpose": "contractAgreement",
            "created": "2019-08-23T20:56:03Z",
            "verificationMethod": "did:provider:123456#key1",
            "jws": "iyJheXXiOiJK...gFWFOEoXk"
          }]
        }],
  "proof": [{
    "type": "Ed25519Signature2018",
    "proofPurpose": "assertionMethod",
    "created": "2019-08-23T20:56:10Z",
    "verificationMethod": "did:GX-DCS:123456#key1",
    "jws": "eyJ0Ugf29iJK...j9ZFOEjXk"
  }]
}
```

- **Notes**
  The finalized Data Contract contains the agreement of data provider defining whether logging is forbidden, optional or mandatory. However, neither the "/make/contract" nor the "/finalize" endpoint return a Log Token directly – instead the "/log/token" is your endpoint of choice if you want to (or have to) deliver a log message to some GX-DELS instance. In case of streams and continuous data subscriptions, logging ideally occurs at regular intervals as indicated in the Data Asset SD. More details concerning the logging of the data exchange can be found in the specification of the GX-DELS [GXFS-3].

3.1.3.6. Contract Validation

ID & Title	Description	Verification Method
GX-DCS.IR.008 **Validation Endpoint**	The GX-DCS MUST offer the communication interface for Data Contract validation that is described below.	Review of Documentation Testing

***Table 10**: Requirements Communication Interfaces - Contract Validation*

The GX-DCS offers an endpoint for validation of a provided finalized Agreement. The GX-DCS evaluates the finalized Agreement and returns whether a finalized Agreement is valid or not, including a human-readable explanation.

- **URL**

  /validate

- **Method:**

  POST

- **URL Params**

  None

- **Request Body**
  Finalized Agreement (i.e., mutually signed data contract)

- **Success Response:**
  - **Code:** 200
    Meaning: OK – Finalized Agreement is valid

- **Error Response:**
  - **Code:** 400
    Meaning: Bad request – missing or malformed request body (see also 415); response must contain a textual representation of the precise error

- **Code:** 401
  Meaning: Unauthorized – invalid Data Provider signature, invalid Data Consumer signature, or invalid GX-DCS signature
- **Code:** 403
  Meaning: Forbidden – proof of identity failed (only the contracting parties – Data Provider and Data Consumer – plus GX-FC instances and GX Portal instances are allowed to use this request)
- **Code:** 404
  Meaning: Not found – Data Provider DID, Data Consumer DID, or GX-DCS DID could not be resolved
- **Code:** 413
  Meaning: Payload too large – Request body exceeds 1 MB (this is an arbitrary value, but some such check is required to prevent accidental or malicious overloads)
- **Code:** 414
  Meaning: Request URI too long – occurs if the URI contains anything but "/validate"
- **Code:** 415
  Meaning: Unsupported media type – occurs if request body isn't proper JSON-LD
- **Code:** 424
  Meaning: Failed dependency – could not validate Data Asset Self-Description model (via GX-FC)
- **Code:** 429
  Meaning: Too many requests – each client is allowed to use this endpoint at most once every two seconds (this value is arbitrary, but some fixed maximum access frequency is required). This error code is only applicable, if DIDs are cached by the GX-DCS (see 3.2.6).
- **Code:** 451
  Meaning: Unavailable for legal reasons – Data Provider's, Data Consumer's, or GX-DCS's Gaia-X Participant status has been revoked (response must contain a textual explanation)
- **Code:** 5xx
  Meaning: The usual server errors

- **Sample Call:**
  POST Body:


```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://w3id.org/gaia-x/core/context/DataAsset.jsonld"
  ],
  "type": "VerifiablePresentation",
  "verifiableCredential": [
    {
      "@context": [
        "https://www.w3.org/2018/credentials/v1",
        "https://w3id.org/gaia-x/core/context/DataAsset.jsonld"
      ],
      "credentialSubject": {
        "@id":"?AssetURI",
        "@type":"gax:DataAsset",
        "gax:title":"Example title",
        "gax:description":"Example description",
        "gax:keyword":["key", "word"],
        "gax:category":["example"],
        "gax:publisher":"?publisherDID",
        "gax:consumer":"?consumerDID",
        "gax:creator":"?creatorDID",
        "gax:language":"http://id.loc.gov/vocabulary/iso639-1/en",
        "gax:distribution":{
          "gax:title":"Example distribution title",
          "gax:description":"Example distribution description",
          "gax:created":"2021-01-23T18:25:43.511Z",
          "gax:modified":"2021-01-25T12:20:34.007Z",
          "gax:mediaType":"text/csv",
          "gax:byteSize":"100000",
          "gax:accessURL":"www.example.com/data/example.csv"
        },
        "gax:created":"2021-01-23T12:21:23.876Z",
        "gax:modified":"2021-01-24T14:45:03.517Z",
        "gax:containsPersonalData":false,
        "gax:sampleAvailable":false,
        "gax:contractOffer":{
          "@type": "gax:contractOffer",
          "gax:choiceOfLaw":"iso:Germany",
          "gax:generalTerms":"Example text for the general terms",
          "gax:confirmationRequired":true,
          "gax:loggingMode":"gax:LoggingMandatory",
          "gax:circulationDetails":"Example text for the circulation details",
          "gax:permission": [ {
            "@type": "gax:Permission",
            "gax:assigner": "?providerDID",
            "gax:target": "?someDataAssetID",
            "gax:action": "gax:COMPENSATE",
            "gax:negotiable": true,
            "gax:constraint":{
              "@type": "gax:Constraint",
              "gax:leftOperand": "gax: PAY_AMOUNT",
              "gax:operator": "gax:EQ",
              "gax:rightOperand": {
                "@value": "?price",
                "@type": "http://www.w3.org/2001/XMLSchema#double"
              }
            }
          }]}},
      "proof": [{
        "type": "Ed25519Signature2018",
        "proofPurpose": "contractAgreement",
        "created": "2019-08-23T20:21:34Z",
        "verificationMethod": "did:consumer:123456#key1",
        "jws": "eyJ0eXAiOiJK...gFWFOEjXk"
      },
        {
          "type": "Ed25519Signature2018",
          "proofPurpose": "contractAgreement",
          "created": "2019-08-23T20:56:03Z",
          "verificationMethod": "did:provider:123456#key1",
          "jws": "iyJheXXiOiJK...gFWFOEoXk"
        }]
    }],
  "proof": [{
    "type": "Ed25519Signature2018",
    "proofPurpose": "assertionMethod",
    "created": "2019-08-23T20:56:10Z",
    "verificationMethod": "did:GX-DCS:123456#key1",
    "jws": "eyJ0Ugf29iJK...j9ZFOEjXk"
  }]
}
```

3.2 Functional Requirements

3.2.1 Data Asset Registration

ID & Title	Description	Verification Method
GX-DCS.FR.001 **Registration Endpoint**	The GX-DCS MUST offer a Data Asset Registration Endpoint, namely GX-DCS.IR.003 (“/register”).	Testing
GX-DCS.FR.002 **Registration GUI**	The GX-DCS MUST offer a GUI for Data Asset Registration as defined in 3.1.1 (namely, GX-DCS.IR.003: “/register”)	Review of Documentation Testing
GX-DCS.FR.003 **Registration authorization**	For both interfaces, the GX-DCS MUST verify that users are GX Participants before allowing them to register a Data Asset.	Review of Documentation Testing
GX-DCS.FR.004 **Provider authorization**	For both interfaces, the GX-DCS MUST verify that users are GX Participants before allowing them to register a Data Asset.	Review of Documentation Testing
GX-DCS.FR.005 **Registration core**	For both interfaces, the GX-DCS MUST ensure that all mandatory elements of the Data Asset Self-Description are specified during the Data Asset registration process.	Review of Documentation Testing
GX-DCS.FR.006 **Registration completeness**	For both interfaces, the GX-DCS MUST allow all elements of the Data Asset Self-Description to be specified during the Data Asset registration process.	Review of Documentation Testing
GX-DCS.FR.007 **Registration placeholders**	For both interfaces, there MUST be the possibility to use placeholders to define elements that can be changed by the Consumer via the Contract Negotiation.	Review of Documentation Testing
GX-DCS.FR.008 **Indicate negotiability**	If placeholders are used, the modifiable nature MUST be indicated by setting the properties named gax:negotiable to *true* in each Rule that contains a placeholder.	Testing
GX-DCS.FR.009 **Formal correctness**	For both interfaces, the GX-DCS MUST check with the GX-FC if the Self-Description is formally correct.	Testing

GX-DCS.FR.010 **Registration Provider validation**	For both interfaces, the GX-DCS MUST ensure that the completed Data Asset Self-Description contains a valid signature from the Data Provider.	Testing
GX-DCS.FR.011 **Registration approval**	If the validation of the Data Asset Self-Description and the providers signature were correct, the GX-DCS MUST add its signature to the Data Asset Self-Description using a hash that includes the signature of the Data Provider and send the resulting signed SD back to the Data Provider (for both interfaces).	Review of Documentation Testing
GX-DCS.FR.012 **Registration error handling**	The GX-DCS MUST support at least the error messages outlined in GX-DCS.IR.003 ("/register") and send the correct error message if applicable.	Review of Documentation Testing

***Table 11**: Functional Requirements Data Asset Registration*

3.2.2 Making an Agreement

ID & Title	Description	Verification Method
GX-DCS.FR.013 **Making an Agreement Endpoint**	The GX-DCS MUST offer an Endpoint for making an Agreement, namely GX-DCS.IR.004 ("/make/contract").	Testing
GX-DCS.FR.014 **Make agreement Consumer authorization**	The GX-DCS MUST verify that users are GX Participants before allowing them to make an agreement.	Review of Documentation Testing
GX-DCS.FR.015 **Only non-negotiable**	In the case of making an Agreement the GX-DCS MUST accept only Agreements that have the value *false* assigned to all gax:negotiable properties inside the Rules.	Testing
GX-DCS.FR.016 **No confirmation requirement**	In the case of making an Agreement the GX-DCS MUST accept only Agreements that have the value *false* assigned to the gax:confirmationRequired property.	Testing
GX-DCS.FR.017 **Make Agreement signature check**	The GX-DCS MUST validate all signatures. To correctly validate the provider signature the DCS MUST remove the Consumer Details beforehand.	Testing

GX-DCS.FR.018 **Make Agreement key caching**	Public keys of already resolved DIDs MAY be cached for 24 hours.	Testing
GX-DCS.FR.019 **Make Agreement policy conformance**	The GX-DCS MUST check if the Consumer conforms to the policies, insofar that is technically feasible with fungible effort.	Review of Documentation Testing
GX-DCS.FR.020 **Make Agreement approval**	If the GX-DCS deemed the signatures to be valid and the Consumer to be conformant to the policies, he MUST sign the Agreement using a hash which includes the signatures of Data Provider and Data Consumer.	Review of Documentation Testing
GX-DCS.FR.021 **Make Agreement distribution**	After validation and signing of the Agreement, the GX-DCS MUST send the finalized Agreement to both Data Provider and Data Consumer.	Testing
GX-DCS.FR.022 **Make Agreement error handling**	The GX-DCS MUST support at least the error messages outlined in GX-DCS.IR.004 (“/make/contract”) and send the correct error message if applicable.	Review of Documentation Testing

***Table 12**: Functional Requirements Making an Agreement*

3.2.3 Agreement Negotiation

ID & Title	Description	Verification Method
GX-DCS.FR.023 **Negotiation Endpoint**	The GX-DCS MUST be offer an endpoint for triggering the negotiation of an Agreement over an API endpoint, namely GX-DCS.IR.005 (“/negotiate”) .	Testing
GX-DCS.FR.024 **Negotiation Consumer authorization**	The GX-DCS MUST verify that users are GX Participants before allowing them to negotiate an agreement.	Review of Documentation Testing
GX-DCS.FR.025 **Negotiation for invitations only**	In the case of Agreement negotiation, the GX-DCS MUST accept only Agreements that have the value *true* assigned to one or more gax:negotiable properties inside the Rules or *true* assigned to the gax:confirmationRequired property.	Testing

GX-DCS.FR.026 **Negotiation check original**	The GX-DCS MUST check with the Federated Catalogue if the original of the submitted Data Asset Self-Description is valid and MUST compare the original with the submitted one.	Testing
GX-DCS.FR.027 **Negotiation policy conformance**	GX-DCS MUST check if the Consumer conforms to the policies, insofar that is technically feasible with fungible effort.	Review of Documentation Testing
GX-DCS.FR.028 **Negotiation distribution**	If the GX-DCS deemed the signatures to be valid and the Consumer to be conformant to the policies, he MUST forward the Agreement to the Data Provider and MUST inform the Data Consumer about it.	Review of Documentation Testing
GX-DCS.FR.029 **Negotiation error handling**	The GX-DCS MUST support at least the error messages outlined in GX-DCS.IR.005 (“/negotiate”) and send the correct error message if applicable.	Review of Documentation Testing

***Table 13**: Functional Requirements Agreement Negotiation*

3.2.4 Finalization of a Contract

ID & Title	Description	Verification Method
GX-DCS.FR.030 **Finalize Endpoint**	The Data Contract Service MUST be able to finalize Agreements via an API, namely GX-DCS.IR.006 (“/finalize”).	Testing
GX-DCS.FR.031 **Finalization Provider authorization**	The GX-DCS MUST verify that the user is a GX Participant, and the Data Provider mentioned in the agreement before allowing the finalization of the agreement.	Review of Documentation Testing
GX-DCS.FR.032 **Finalize content validation**	To finalize an Agreement, the GX-DCS MUST first validate, that the content of the Data Contract forms a valid contract (by comparison with the ontology).	Review of Documentation Testing
GX-DCS.FR.033 **Finalize signature validation**	To finalize an Agreement, the GX-DCS MUST first validate, that the signatures of both the Data Provider and the Data Consumer are valid.	Review of Documentation Testing
GX-DCS.FR.034 **Finalization approval**	If the GX-DCS has confirmed the validity of both signatures it MUST sign the Agreement while including the signatures of Data Provider and Data Consumer in the hash.	Review of Documentation Testing

GX-DCS.FR.035 **Finalization distribution**	After signing the Agreement, the GX-DCS MUST send the finalized Agreement to Data Provider and Data Consumer	Testing
GX-DCS.FR.036 **Finalization error handling**	The GX-DCS MUST support at least the error messages outlined in GX-DCS.IR.006 (“/finalize”) and send the correct error message if applicable.	Review of Documentation Testing

***Table 14**: Functional Requirements Finalization of a Contract*

3.2.5 Issuing of Log Tokens

ID & Title	Description	Verification Method
GX-DCS.FR.037 **Log Token Endpoint**	If the data transfer can or must be logged, the GX-DCS MUST issue an authorization token – the Log Token – for the Gaia-X Data Exchange Logging Service (GX-DELS). The Log Token MUST be requestable and renewable via an API, namely GX-DCS.IR.007 (“/log/token”).	Testing
GX-DCS.FR.038 **Log Token signature validation**	The GX-DCS MUST validate *all* proofs and signatures of the finalized Agreement before returning the Log Token.	Testing
GX-DCS.FR.039 **Check loggingMode**	The GX-DCS MUST check if the finalized Agreement contains either the values gax:logging_mandatory or gax:logging_optional in the gax:loggingMode property.	Testing
GX-DCS.FR.040 **Log Token Agreement validation**	The finalized Agreement MUST also be validated like described in GX-DCS.FR.004 (“Agreement Validation”).	Review of Documentation Testing
GX-DCS.FR.041 **Log Token authorization**	The GX-DCS MUST validate that the requesting entity is allowed to receive the Log Token, i.e., that the requesting party is either the Data Provider or the Data Consumer participating in the Data Contract.	Review of Documentation Testing
GX-DCS.FR.042 **Log Token error handling**	The GX-DCS MUST support at least the error messages outlined in GX-DCS.IR.007 (“/log/token”) and send the correct error message if applicable.	Review of Documentation Testing

***Table 15**: Functional Requirements Issuing of Log Tokens*

3.2.6 Agreement Validation

ID & Title	Description	Verification Method
GX-DCS.FR.043 **Validation End-point**	The Data Contract Service MUST be able to validate finalized Agreements via an API, namely GX-DCS.IR.008 ("/validate").	Testing
GX-DCS.FR.044 **GX-DCS signature validation**	The GX-DCS MUST verify the GX-DCS signature of the finalized Agreement.	Review of Documentation Testing
GX-DCS.FR.045 **Consumer signature validation**	The GX-DCS MUST verify the Consumer signature of the Agreement.	Review of Documentation Testing
GX-DCS.FR.046 **Provider signature validation**	The GX-DCS MUST verify the Provider signature of the Agreement.	Review of Documentation Testing
GX-DCS.FR.047 **Provider signature validation for Offers**	If the finalized Agreement is based on a Data Contract Offer (*true* for gax:confirmationRequired or a gax:negotiable property), the GX-DCS MUST ensure the Provider signature correctly signs the Provider and Contract details (compare section 4.1).	Review of Documentation Testing
GX-DCS.FR.048 **Provider signature validation for Invitations**	If the finalized Agreement is based on an Invitation for Data Contract Offers (*false* for gax:confirmationRequired or all gax:negotiable properties), the GX-DCS MUST ensure the Provider signature correctly signs all details in the contract (compare section 4.1).	Review of Documentation Testing
GX-DCS.FR.049 **Validation error handling**	The GX-DCS MUST support at least the error messages outlined in GX-DCS.IR.007 ("/validate") and send the correct error message if applicable.	Review of Documentation Testing

***Table 16**: Functional Requirements Agreement Validation*

3.2.7 Caching

ID & Title	Description	Verification Method

GX-DCS.FR.050 **DID Cache**	GX-DCS MAY cache all resolved DIDs for 24 hours.	Review of Documentation Testing
GX-DCS.FR.051 **SD Model Cache**	Data Asset Self-Description models retrieved from GX-FC MAY be cached for 14 days.	Review of Documentation Testing
GX-DCS.FR.052 **Model Version Update**	If Data Asset Self-Descriptions are cached, a check for new available Data Asset Self-Description version SHOULD be performed at least every 24 hours. At the latest a new version MUST be retrieved when a Data Consumer or Provider refers to a newer version of the SD models in a request.	Review of Documentation Testing
GX-DCS.FR.053 **Cache Deletion**	If a cache is utilized, cache deletion MUST NOT have any effect on the output content of the API endpoints.	Review of Documentation Testing

***Table 17**: Functional Requirements Caching*

3.3 Non-functional Requirements

3.3.1 Performance Requirements

ID & Title	Description	Verification Method
GX-DCS.NFR.001 **Performance by Design**	The component SHOULD be designed and implemented with performance in mind. In particular, it MUST be implemented in a non-blocking way.	Review of Documentation Testing
GX-DCS.NFR.002 **Scalability**	The component MUST be scalable and able to handle multiple requests.	Review of Documentation Testing

***Table 18**: Non-functional Requirements Performance Requirements*

3.3.2 Safety Requirements

ID & Title	Description	Verification Method

GX-DCS.NFR.003 **Reset Possibility**	In case of errors, it MUST be possible to reset the component and continue execution as specified in this document. The component SHOULD be stateless and need no recovery in case of a reset.	Review of Documentation Testing

***Table 19**: Non-functional Requirements Safety Requirements*

3.3.3 Security Requirements

3.3.3.1 General Security Requirements

Each Gaia-X Federation Service must fulfil the requirements stated in [GXFS-4].

Federation Services specific requirements will be documented in the next chapter.

3.3.3.2 Service Specific Security Requirements

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

ID & Title	Description	Verification Method
GX-DCS.NFR.004 **Transport Layer Security**	Each communication with an interface of GX-DCS MUST utilize TLS of at least version 1.2. It SHALL use TLS in version 1.3.	Review of Documentation Testing
GX-DCS.NFR.005 **Remote Administration**	If the component can be remotely administrated by the Federator, the communication MUST utilize a secure communication channel such as SSH or VPN.	Review of Documentation Testing
GX-DCS.NFR.006 **State-of-the-art Cryptography**	Cryptographic algorithms and cipher suites MUST be state-of-the-art and chosen in accordance with official recommendations. Those recommendations MAY be those of the German Federal Office for Information Security (BSI) [Ref-4] or SOG-IS [Ref-5].	Review of Documentation Testing
GX-DCS.NFR.007 **Authentication and Authorization**	GX-DCS MUST grant access to its services only to authenticated and authorized Gaia-X Participants. The authentication MUST be based on a valid Gaia-X Identity as defined in [GXFS-1].	Review of Documentation Testing
GX-DCS.NFR.008 **Integrity Protection for Configuration**	Where the functionality of the GX-DCS is based on configuration files, those files MUST be authenticated, and integrity protected.	Review of Documentation Testing

GX-DCS.NFR.009 **Integrity Protection for the Service**	The Federator MUST utilize security measures to ensure the integrity of GX-DCS. It MAY support a proof of the integrity to remote parties using an additional interface (Remote Attestation).	Review of Documentation Testing
GX-DCS.NFR.010 **Storage of Secrets**	Secrets such as keys and other cryptography material MUST be stored in a secure and protected environment, e.g., a TPM, HSM, or TEE to ensure their confidentiality and integrity. See also GX-DCS.IR.002.	Review of Documentation Testing

***Table 20**: Non-functional Requirements Service Specific Security Requirements*

3.3.4 Software Quality Attributes

ID & Title	Description	Verification Method
GX-DCS.NFR.012 **Programming Style**	The implementation SHOULD follow best practices and a consistent style for coding, e.g., the source code shall be clearly structured and modularized; there should be no dead code; function and variables shall be clear and self-explaining. The code MUST be well documented to support adaptability, maintainability, and usability of the component.	Review of Documentation Source Code Review
GX-DCS.NFR.013 **Testing**	The development of GX-DCS MUST include functional and security testing, source code audits, and penetration testing. More details can be found in the [GXFS-4].	Review of Documentation

***Table 21**: Non-functional Requirements Software Quality Attributes*

3.3.5 Business Rules

In general, only Gaia-X Participants must be able to interact with the GX-DCS. Each participant shall be capable of registering Data Assets, negotiate, or make a Data Contracts for a Data Asset and get a validation confirmation for a finalized Agreement. A log authentication token (“Log Token”) must only be provided to the rightful Participants in this Data Contract (Data Provider and Consumer).

3.4 Compliance

ID & Title	Description	Verification Method
GX-DCS.NFR.015 **EUCS/ENISA Compliance**	The GX-DCS SHOULD fulfill the cybersecurity control set of the EUCS [Ref-3] Annex A according to its assigned Assurance Level as described in the [GXFS-4].	Review of Documentation

GX-DCS.NFR.016 **Legal Compliance**	The GX-DCS MUST ensure compliance with national and European laws.	Review of Documentation

***Table 22**: Requirements Compliance*

3.5 Design and Implementation

ID & Title	Description	Verification Method
GX-DCS.NFR.017 **Build Scripts**	The repository for the GX-DCS (GX-DCS.NFR.019) MUST include build scripts to build and run the Service from the repository.	Review of Documentation Testing

***Table 23**: Requirements Design and Implementation – Installation*

Further requirements are defined in Ref-8 and MUST be fulfilled.

4. System Feature

4.1 Data Asset Self-Description and Contract Lifecycle

The necessary basis for a Data Contract is the Data Asset Self-Description (DASD) which is either created utilizing the GX-DCS GUI for Data Asset Registration or prepared by the Data Provider to be sent to the Data Asset Registration Endpoint. For both the original DASD as well as the Data Contracts based on this DASD, the GX-DCS has two major tasks to help Data Provider and Data Consumer make a valid contract:

1. Validation of content: The GX-DCS ensures that the content of the DASD or Data Contract forms a correct Ricardian contract to be used as basis for the data transfer. For that purpose, the Gaia-X Ontology contains a description of the Data Asset and legal details for its transfer [DASD] such as the chosen law, general terms, metadata for the Data Asset and terms for the data transfer. The GX-DCS always validates that all mandatory attributes have been filled and the provided attribute values conform with the expected format or values.
2. Validation of the involved parties: Additionally, the GX-DCS ensures that both parties involved in the Data Contract (Data Provider and Data Consumer) are GX-Participants that may conduct such a data exchange and that both parties have reliably expressed their consent to the Data Contract utilizing digital signatures.

To summarize, the GX-DCS plays a role comparable to a notary in the “normal” world who ensures that the agreed contract is sound and both parties have signed the contract correctly. To achieve this goal, the GX-DCS supports and ensures the following signature scheme for the DASD:

A DASD and Data Contract abstractly contains the following 6 details:

1. Provider Details

2. Consumer Details
3. Contract Details
4. Provider Signature
5. Consumer Signature
6. GX-DCS Signature

After **Data Asset Registration**, the Provider Details (1) and Contract Details (3) are filled according to the Data Providers information. The Data Provider confirms the details with the Provider Signature (4) over those details (1,3). The GX-DCS confirms the validity of the contract and the provider signature with the GX-DCS Signature calculated over 1,3 and 4.

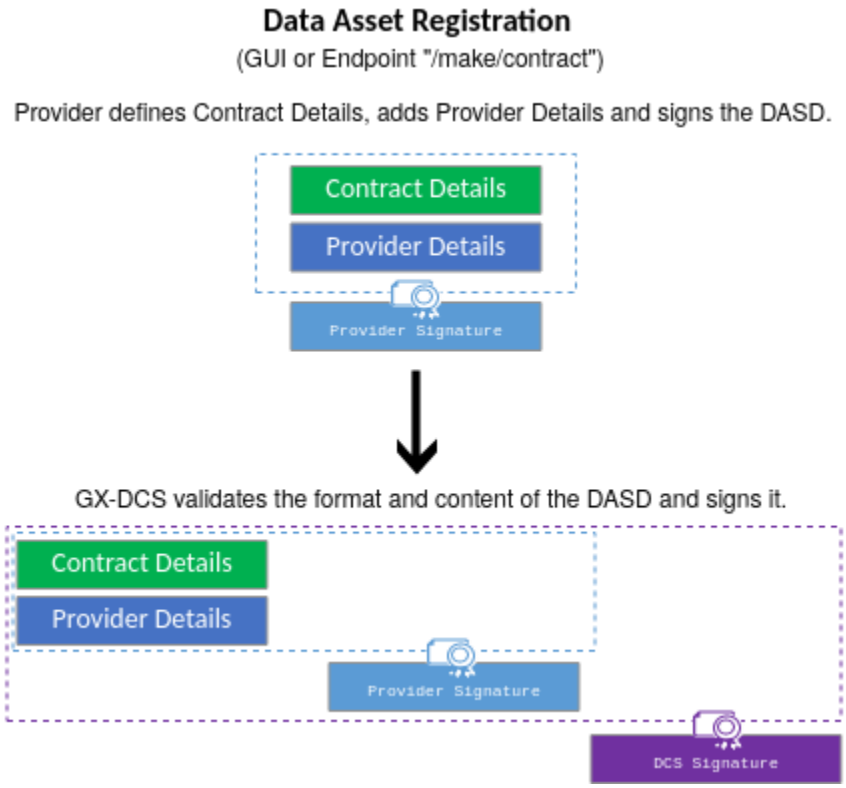


***Figure 2**: Data Asset Registration and Validation*

Afterwards, a contract can be established in two different ways:

1. **Data Contract Offer**: The Data Providers offer a Data Asset “as is” solely based on the DASD and allow each valid GX-DCS to finalize the Data Contract on their behalf. Thus, it is sufficient for Data Consumers to contact the GX-DCS, add the Consumer Details and confirm the Data Contract from their side with a Consumer Signature (over 1,2,3). The GX-DCS then confirms the Consumer’s identity and the provided details and finalizes the Data Contract by replacing the previous GX-DCS Signature with a new version now calculated over all details in the finale contract (1,2,3,4 and 5). The finalization of such a Data Contract Offer is achieved by using the “/make/contract” endpoint.

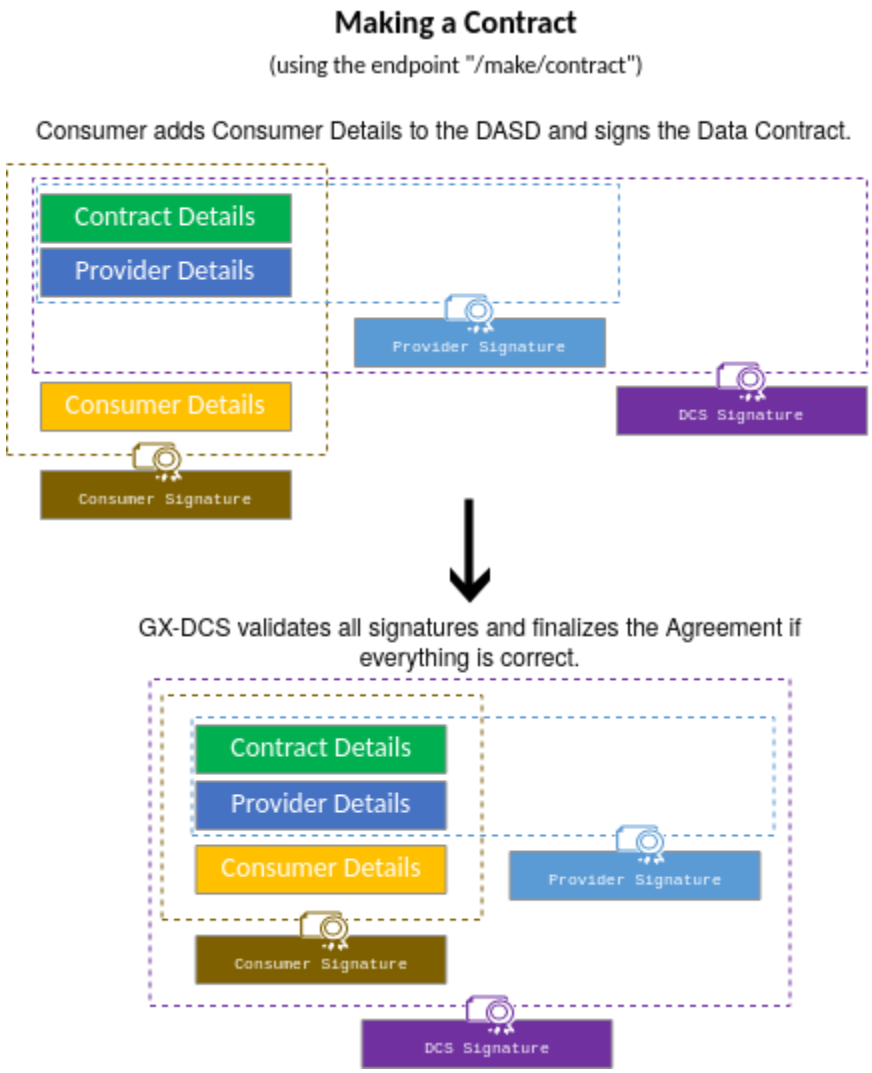


***Figure 3**: Making a Contract*

2. **Invitation for Data Contract Offers**: The Data Providers only provide the DASD as an invitation for Data Contract Offers from consumer side, but an active confirmation from the Data Provider is needed to finalize the contract. For this purpose, the Data Provider is allowed to specify all terms for the transfer of a Data Asset "as is" and reserve the right to accept or reject a contract offer made for that Data Asset by setting the gax:confirmationRequired property in the DASD to *true*. However, a Data Provider can also mark different properties in the contract as negotiable and decide then based on the Data Consumer's offer whether a contract will be made.

In both cases, Consumers adjust the DASD by adding their Consumer details and optionally also modify the contract details for the negotiable properties (3*). They sign their offer by adding a Consumer signature over all details in the contract (1,2 and $3^{(*)}$) and send it to the GX-DCS via the "/negotiate" endpoint. The GX-DCS validates the offer made by comparing it to the original DASD available in the Catalogue to ensure that only the negotiable properties were adjusted. The offer is then forwarded to the Data provider who can decide whether the offer is accepted. In case, a Data Provider wants to confirm and finalize the

contract, the "/finalize" endpoint of the GX-DCS is utilized and provided with the Data Contract signed by the Consumer as well as the Provider. The Provider Signature must include all contract details (1,2 and $3^{(*)}$). The GX-DCS then validates the entire contract and if the content is valid and the Participants have both successfully confirmed the contract, the GX-DCS adds its signature (containing (1,2, $3^{(*)}$,4,5) and distributes the Finalized Data Contract to all involved parties.

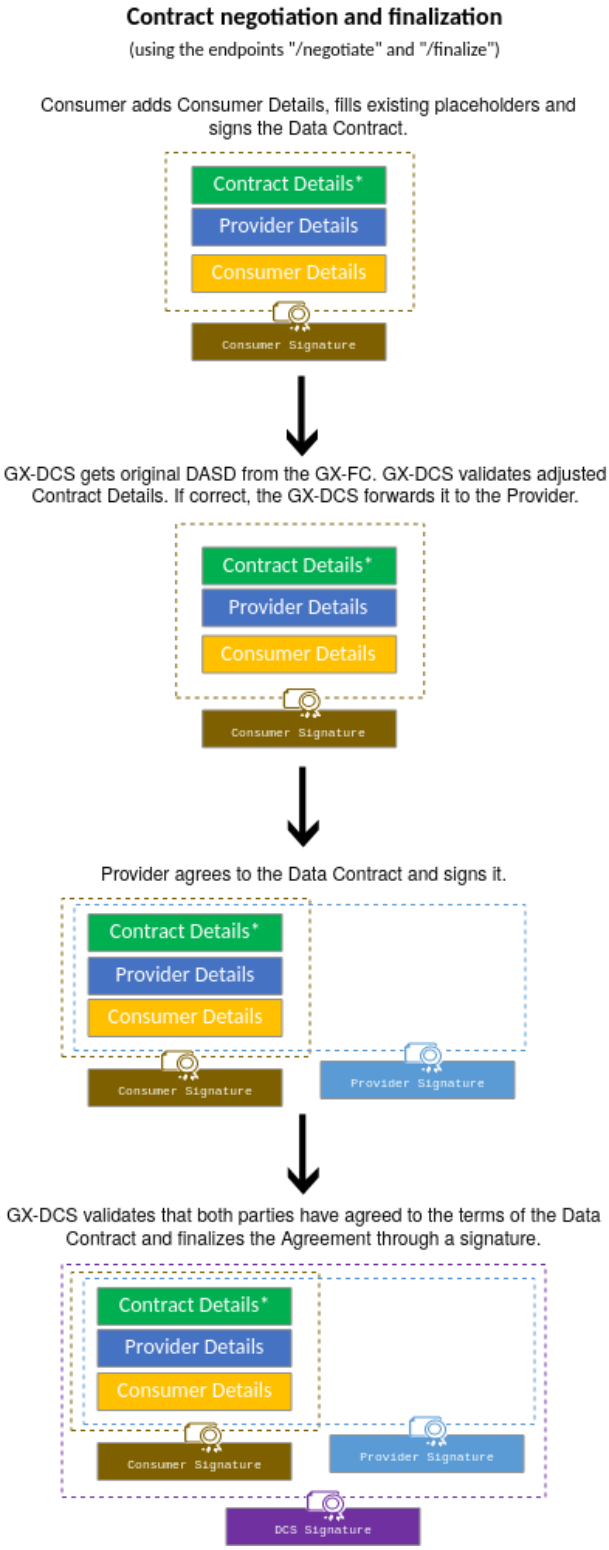


***Figure 4**: Contract negotiation and finalization*

4.2 Data Asset Registration

This section contains further details for the interaction with the interface for Data Asset Registration (GX-DCS.IR.003: "/register").

Description

Every Data Provider needs to be able to register a Data Asset. There MUST be an API for this registration (namely, GX-DCS.IR.003: "/register"). All elements of the Data Asset Self-Description MUST be specifiable via the Data Asset registration process. There MUST be the possibility to use placeholders to define elements that can be changed by the Consumer via the Contract Negotiation. If placeholders are used, the modifiable nature MUST be indicated by setting the properties named gax:negotiable to *true* in each Rule that contains a placeholder. The completed Data Asset Self-Description MUST be hashed and MUST be signed by the Data Provider before being submitted to GX-DCS. The GX-DCS MUST check with the GX-FC if the Self-Description is formally correct. The Data Asset Self-Description MUST then be signed by the Data Contract Service using a hash that includes the Data Provider signature. Then this signed Self-Description MUST be send back to the Data Provider as the result.

Input

Information about the Data Asset (See Data Asset Self-Description) and policies that apply to the Data Asset, signed by the Data Provider.

Output

Data Asset Self-Description signed by the Data Contract Service.

Acceptance Criteria

The Data Provider gets a valid Data Asset Self-Description that can be successfully registered with the Federated Catalogue (GX-FC). At least the unhappy paths outlined in GX-DCS.IR.003 ("/register") must be checked and covered by the GX-DCS.

4.3 Making a contract

This section contains further details for the interaction with the interface for making a Data Contract (GX-DCS.IR.004: "/make/contract"). In that context the GX-DCS also needs to interact with the Provider's endpoint for confirmation and reception of Agreements, the Data Consumer (requester) and the Federated Catalogue.

Description

It MUST be possible to trigger the making of an Agreement over an API. For this, the Data Consumer MUST send the respective Agreement. The Consumer MUST sign the filled-out (Consumer details added) Data Asset Self-Description before calling this request. The GX-DCS MUST accept only Agreements that have the value *false* assigned to all gax:negotiable properties inside the Rules. The GX-DCS MUST validate all signatures. To correctly validate the provider signature the DCS MUST remove the Consumer Details beforehand. Public keys of already resolved DIDs MAY be cached for 24 hours (note that DID resolving could

take several seconds). GX-DCS MUST check if the Consumer adheres to the policies. GX-DCS MUST sign the Agreement using a hash including the Data Provider signature and Data Consumer signature. Then it MUST be sent to both Data Provider and Data Consumer.

Input

Agreement signed by the Data Consumer

Output

Finalized Agreement signed by Data Consumer, Data Provider and GX-DCS

Acceptance Criteria

For each valid Data Provider, Data Consumer the instance of GX-DCS generates a finalized Agreement that can be positively validated by every GX-DCS instance. At least the unhappy paths outlined in GX-DCS.IR.004 ("/make/contract") must be checked and covered by the GX-DCS.

4.4 Contract Negotiation

This section contains further details for the interaction with the interface for negotiating a Data Contract (GX-DCS.IR.005: "/negotiate"). In that context the GX-DCS also needs to interact with the Provider's end-point for confirmation, the Data Consumer (requester) and the Federated Catalogue.

Description

It MUST be possible to trigger the negotiation of an Agreement over an API endpoint, namely GX-DCS.IR.005 ("/negotiate"). To this end, the Data Consumer MUST send the respective Agreement. The Consumer MUST sign the filled-out (Consumer details added and placeholders filled) Data Asset Self-Description before calling this request. GX-DCS MUST accept only Agreements that have the value *true* assigned to one or more of the gax:negotiable properties inside the Rules. The GX-DCS MUST check with the Federated Catalogue if the original Data Asset Self-Description of the submitted Agreement is valid and MUST compare the original with the submitted one. Public keys of already resolved DIDs MAY be cached for 24 hours (note that DID resolving could take several seconds). GX-DCS MUST check if the Consumer conforms to the policies, insofar that is technically feasible with fungible effort. If the request passes all tests, the GX-DCS MUST forward the Agreement to the Data Provider. The Data Consumer MUST be informed on if the Agreement was successfully forwarded or declined. The Data Provider then decides whether to accept or decline the order and proceeds with calling the /finalize Endpoint outlined in GX-DCS.IR.006 ("/finalize") in case of acceptance. If the Data Provider declines, he MUST inform the Data Consumer.

Input

Agreement signed by the Data Consumer.

Output

Confirmation that the Agreement was forwarded to the Data Provider.

Acceptance Criteria

Valid Agreements get forwarded to the Data Provider. Additionally, at least the unhappy paths outlined in GX-DCS.IR.005 (“/negotiate”) must be checked and covered by the GX-DCS.

4.5 Contract Finalization

This section contains further details for the interaction with the interface for getting an Agreement finalized (GX-DCS.IR.006: “/finalize”).

Description

The Data Contract Service MUST be able to finalize Agreements via an API, namely GX-DCS.IR.006 (“/finalize”). This is necessary for Agreements negotiated via the “/negotiate” endpoint. The Provider sends a request to the “/finalize“ endpoint which contains the Agreement signed by him and the Data Consumer. To finalize an Agreement, the GX-DCS MUST first validate, that the signatures of both the Data Provider and the Data Consumer are valid. Public keys of already resolved DIDs MAY be cached for 24 hours (note that DID resolving could take several seconds). If the GX-DCS has confirmed the validity of both signatures it MUST sign the Agreement while including the signatures of Data Provider and Data Consumer in the hash. After signing the Agreement, the GX-DCS MUST send the finalized Agreement to Data Provider and Data Consumer.

Input

Agreement (Signed by Data Provider and Data Consumer)

Output

Finalized Agreement (Signed by Data Provider, Data Consumer and GX-DCS)

Acceptance Criteria

Agreements that have been correctly signed by Data Provider and Data Consumer get identified as such and receive a signature of the GX-DCS. The finalized Agreement is sent to Data Provider and Data Consumer respectively.

4.6 Get Log Token

This section contains further details for the interaction with the interface for getting a new log auth token (GX-DCS.IR.007: “/log/token”).

Description

If the data transfer can or must be logged, GX-DCS MUST issue an authorization token – the Log Token – for the Gaia-X Data Exchange Logging Service (GX-DELS) [GXFS-3]. The Log Token MUST be requestable and renewable via an API, namely GX-DCS.IR.007 (“/log/token”). To request or renew a Log Token, the finalized Agreement has to be sent to a GX-DCS instance. GX-DCS MUST validate *all* proofs and signatures of the finalized Agreement before returning the Log Token. To be valid, the finalized Agreement MUST furthermore contain either the values gax:LoggingMandatory or gax:LoggingOptional in the gax:loggingMode property. See GX-DCS.FR.004 (“Agreement Validation”) for more details about when a finalized

Agreement is valid. GX-DCS MUST also validate that the requesting entity is allowed to receive the Log Token.

Input

Finalized Agreement

Output

Log Token (see Log Token definition [GXFS-3])

Acceptance Criteria

Any valid finalized Agreement gets a Log Token that works. Invalid finalized Agreements and invalid requesters don't get a Log Token (please study the error codes of GX-DCS.IR.007).

4.7 Contract Validation

This section contains further details for the interaction with the interface for getting a finalized Agreement validated (GX-DCS.IR.008: "/validate").

Description

The Data Contract Service MUST be able to validate finalized Agreements via an API, namely GX-DCS.IR.007 ("/validate"). To validate a finalized Agreement, a differentiation between negotiable Agreements and non-negotiable Agreements MUST be made. A negotiable Agreement has one or more of the gax:negotiable properties inside the Rules set to *true* or the property gax:confirmationRequired *set to true*. A non-negotiable Agreement has the value *false* assigned to all gax:negotiable properties inside the Rules. The GX-DCS MUST verify the GX-DCS signature of the finalized Agreement. The GX-DCS MUST verify the Consumer signature of the Agreement. The GX-DCS MUST verify the Provider signature of the Agreement. In case of a non-negotiable Agreement the validation of the Provider signature is achieved through removing the Consumer Information before validation. Public keys of already resolved DIDs MAY be cached for 24 hours (note that DID resolving could take several seconds). GX-DCS is only responsible for validating the finalized Agreement, the Provider is responsible for handling authorization and authentication of Consumers in the context of the data transfer.

Input

Finalized Agreement

Output

Validity of the finalized Agreement (HTTP status 200 if good, otherwise HTTP status 4xx)

Acceptance Criteria

Valid finalized Agreements are classified as valid and invalid finalized Agreements are classified as invalid with the correct error code. There MUST be no ambiguity: The same finalized Agreement MUST always be classified in the same way, and all existing GX-DCS instances must classify each finalized Agreement in the same way.

Appendix A: Glossary

The glossary is part of the Gaia-X Technical Architecture Document [Ref-1].

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.

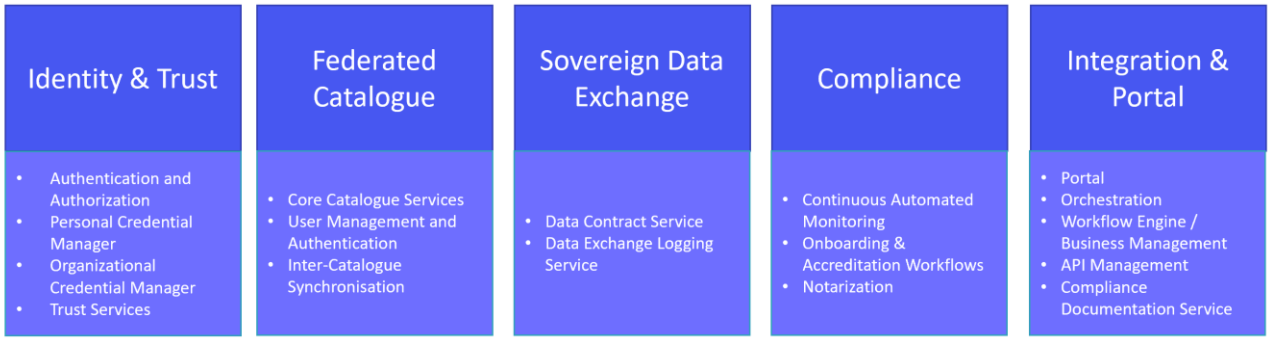
Work Package 2 (WP2): Federated Catalogue
The Federated Catalogue constitutes the central repository for Gaia-X Self-Descriptions to enable the discovery and selection of Providers and their Service Offerings. The Self-Description as expression of properties and Claims of Participants and Assets represents 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 [Ref-1].

Appendix C: ADR-XXX

ADR-XXX: GAIA-X Identifier Format
==========
:adr-id: XXX
:revnumber: 2.1
:revdate: 2021-03-19
:status: proposed
:author: GAIA-X Catalogue and IAM Community
:stakeholder: IAM WG, Self-Description WG, Catalogue WG
Summary

ADR-001 defines the use of JSON-LD for the Self-Descriptions. JSON-LD requires that Identifiers used for cross-referencing between self-descriptions are IRIs (Internationalized Resource Identifiers [RFC3987]). This ADR upholds this definition and further refines it.
Identifiers used in GAIA-X shall be URIs following the [RFC3986].
Identifier must re-use existing schemas or define their own private URI schema(s) eu.gaia-x. where necessary [BCP35]. The schema shall define additional semantics to indicate the underlying protocol.
Context

The generic structure of the identifier takes the form:
:
For protocols requiring a new URI schema a private schema should be defined following the pattern:
eu.gaia-x.

The following Identifier schemas have a defined mechanism to ensure uniqueness of the Identifier. More schemas may be added in the future.
Protocol Schema Protocol_Specific_Id

TAG [RFC4151] urn:tag ,:
OpenID Connect eu.gaia-x.openid ;
DID did :

Tag URI Schema
~~~~~~~~~~~~~~~
Identifiers used in Self-Descriptions may follow the conventions of RFC 4151 for the 'tag' URI scheme. Identifiers of this format contain the DNS domain name or an email of the issuing organization as well as a date at which the organization was in possession of the DNS domain. That way, the organization in possession of the DNS domain at that time is responsible to issue only unique Identifiers.

Some examples of Identifiers:
urn:tag:provider-name.com,2020:my-service:v1
urn:tag:subdomain.foobar.com,2020-01:org1/data-asset5/element20
urn:tag:foobar@acme.org,2020-01-29:e51a9f18273718445f0c016f23b2bc05919f7433

By the convention that only the organization owning the domain-name may use it for Identifiers, GAIA-X Participants can themselves issue new Identifiers and ensure that Identifiers are unique without the need for a central identifier registry for all GAIA-X Participants.

OpenID Connect URI Schema
~~~~~~~~~~~~~~~~~~~~~~~~~~
OpenID Connect eu.gaia-x.openid ;
Example:
eu.gaia-x.openid:https://example-idp.org/auth/realms/master;YWxpY2VAZm9vLmNvbQ
Companies have to host an endpoint that is part of the Identifier.
To ensure uniqueness, endpoints might need to change after a domain changes ownership and uniqueness of identifiers cannot be otherwise guaranteed.

DID URI Schema
~~~~~~~~~~~~~~~
DID identifiers according to W3C "Decentralized Identifiers", Candidate Recommendation: https://www.w3.org/TR/did-core refer to a "method".
1. The method refers to a "Verifiable Data Registry" where the DID can be resolved to a document.
2. The Verifiable Data Registry ensures uniqueness of the Identifiers.
Examples for such Verifiable Data registries include "distributed ledgers, decentralized file systems, databases of any kind, peer-to-peer networks, and other forms of trusted data storage" as described in https://www.w3.org/TR/did-core/#architecture-overview

Decision Statements

GAIA-X Identifiers uniquely identify an entity in GAIA-X.
Informational: Entities can refer to (non-exhaustive)
~~~~~~~~~~~~~~~

- Entities with a Self-Description (contains Participant)
- Principals (user accounts)
- Abstract Concepts (for example "European Economic Area" or "ISO 27001").

GAIA-X Identifiers are unique in the sense that an Identifier must never refer to more than one entity. There can be several GAIA-X Identifiers refering to the same entity.

Informational: As a policy, multiple Identifiers for the same entity should be avoided in the Catalogue.
All Identifiers used in GAIA-X are URIs following the [RFC3986] specification.

Informational: JSON-LD allows IRIs. URIs are a strict subset of that.
The lifetime of an Identifier is permanent. That is, the Identifier has to be unique forever, and may be used as a reference to an entity well beyond the lifetime of the entity it identifies or of any naming authority involved in the assignment of its name [RFC1737]. Reuse of an Identifier for a different entity, also at a later time, is forbidden.
There are multiple valid URI schemas defined, each associated with a technical mechanism to ensure uniquenes. The structure of an identifier has to ensure the uniqueness of the Identifier.

Informational:
GAIA-X Participants can self-issue Identifiers. It is solely the responsibility of a Participant to determine the conditions under which the Identifier will be issued. A self-issued Identifier can be used without publicly registering or announcing the Identifier first.
Not all URI schemas are usable for self-issuing.

Informational:
Identifiers shall be derived from the native identifiers of an Identity System without any separate attribute needed. The Identifier shall provide a clear reference to the Identity System technology used. OpenID Connect and DID shall be supported. Any scheme for Identifiers must permit future extensions to the scheme.

Informational:
The Identifier shall be comparable in the raw form. It shall not be needed to make any transformation to compare two identifiers and tell whether they are the same.

Informational:
Identifiers should not contain more information than necessary (including Personal Identifiable Information).
Consequences

GAIA-X Participants Identity Systems can self-issue valid Principal Identifiers.
Based on the identifier it is possible to determine the technology and the unique reference to the Identity.

ADR References
---
* ADR-001

External References
---
* [BCP35] Guidelines and Registration Procedures for URI Schemes. https://tools.ietf.org/html/bcp35
* [DID-Core] Decentralized Identifiers (DIDs) v1.0. https://www.w3.org/TR/did-core/
* [IAM Framework] GAIA-X IAM Framework v1.01. https://docs.google.com/document/d/1XCjIVRul_w_6runDn_Rh-8nVdMhSFmMxZTXoQAhtISA/
* [RFC1737] Functional Requirements for Uniform Resource Names. https://tools.ietf.org/html/rfc1737
* [RFC3986] Uniform Resource Identifier (URI): Generic Syntax. https://tools.ietf.org/html/rfc3986