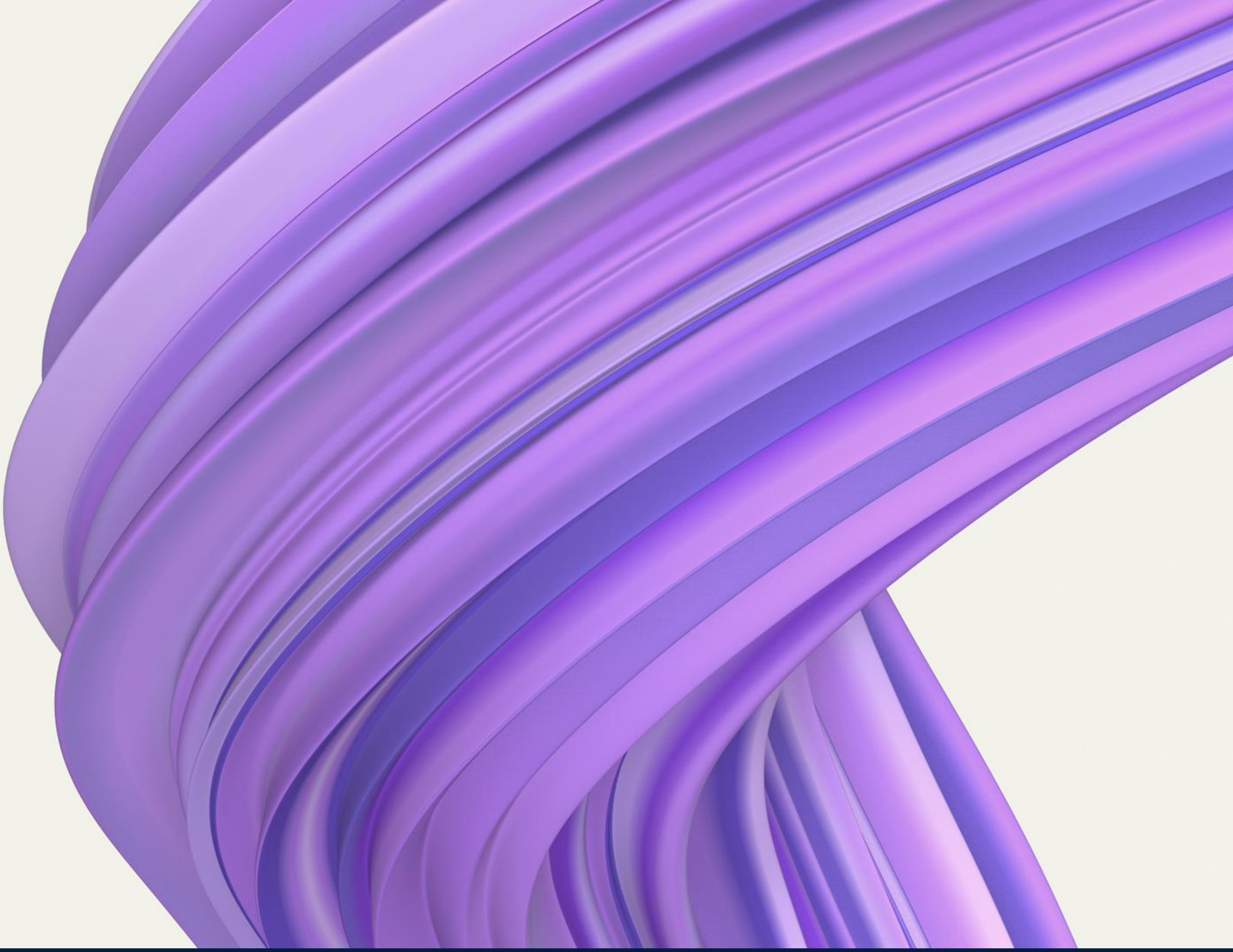


Gaia-X Technical Implementation Architecture

iECO Project

Author:	**Christoph F. Strnadl**
Date:	09 August 2023
Version	1.10

Source: iECO-Gaia-X Technical Implementation Architecture_v1.10.docx

Table of Contents

Version History

VERSION	DATE	REMARKS
1.00	14.07.23	Initial release of the document within the iECO work package 2 team. Selected communication to outside entities such as the Gaia-X Hub Austria, EuProGigant's DevOps team, to solicit feedback.
1.10	09.08.23	Included terminology-related feedback improving the differentiation between the (single and authoritative) «Gaia-X Architecture Document» and this document. Added a "Disclaimer" section.

DISCLAIMER

No warranty or representation, express or implied, is intended or deemed to arise in connection with this document. Neither Software AG, nor any of its affiliates, nor the iECO project or any of its project partners, nor the author assumes or accepts any warranties, representations, or liabilities whatsoever in connection to the contents of this documents and actions or decisions based on it.

List of Tables

List of Figures

1 Introduction

1.1 Purpose of this document

This document provides a description of (one possible) **generic technical implementation architecture for a Gaia-X compliant ecosystem as proposed in the iECO[1] project**. Here, we understand «architecture» as the fundamental organization of a system embodied in its components, their relationships to each other and to the environment[2].

Serving this cause, it will make use of the following mechanisms to organize and present its contents:

1. **Generalization**. The architectural description will “abstract away” or generalize use case-specific details of the actual architecture (e.g., it will not refer to any of iECO’s Advanced Smart Services or its novel operationalization of a distributed digital twin).
2. **Conceptualization**. The architecture will remain at a level of detail just above (or prior) to the actual IT-based implementation or deployment of any actual IT artifacts (such as software or hardware components). Technically, this means we concentrate on (i) (concrete) **software components** and (ii) (more abstract) **services** (always thought to be implemented by suitable software components).
   Hardware and other IT infrastructure-related services like networking will not be included in our conceptualization.
3. **Focus on Gaia-X**. The architecture of a single (albeit complex) system (here: iECO) typically needs several different architectural views each focusing on different stakeholders and their specific concerns. This document limits itself on Gaia-X-related aspects of the whole iECO technical architecture and, thus, represents a Gaia-X-specific view.
4. **Disregard for architectural governance**. We will not describe or explain architectural governance, that is the principles guiding the design and evolution of the system.

1.2 Disclaimers: What this (architecture) document is *not*

This is **not an official document endorsed or approved by the Gaia-X AISBL** or any of its committees or working groups. In particular, this is not any form of *official* implementation or technical architecture agreed upon on the Gaia-X Architecture Working Group. There is only one official Gaia-X architecture document[3] - and this document is it not!

However, the architecture and concepts authoritatively defined in the official Gaia-X Architecture Document **allow (many) different technical implementations**: As long as an implementation conforms to the official Gaia-X standards and procedures as specified in the Gaia-X Architecture document and others (e.g., Trust Framework, Policy & Rules), the participants using

[1] https://ieco-gaiax.de/
[2] cf. IEEE 1471 and ISO/IEC/IEEE 42010
[3] Version 22.10: https://docs.gaia-x.eu/technical-committee/architecture-document/22.10/

this particular implementation will be able to interact with other Gaia-X compliant services based on Gaia-X compatible implementations[4]. In that vein, this document presents only one possible implementation architecture out of other potential implementations[5].

We also stress the fact that at the time of releasing this document, the architecture described has not yet been actually implemented in the iECO project.

1.3 Audience

This architecture description is intended for the following stakeholders:

- technically inclined project managers and sub-project managers
- technical managers and technical task managers
- business analysts
- requirements engineers
- software/microservices architects
- lead software developer
- lead IaaS/infrastructure team
- IaaS/PaaS architects

The document focuses on the following concerns of the aforementioned stakeholders:

- services required from Gaia-X-related entities
- services the iECO project needs to establish or provide in order to evolve into a Gaia-X ecosystem or federation
- components needed to establish Gaia-X conformance or compatibility especially in the areas of identity, authentication, authorization, and trust in general

1.4 Non-Goals

This document, deliberately, does not include or consider the following aspects of the iECO project:

- use-case specific architecture elements (These may follow in future, extended versions of this document)
- actual software implementation
- project planning and scheduling
- aspects related to iECO’s novel distributed digital twin (dDTw) concept

[4] “It is crucial to differentiate compliance and compatibility. A service can be made Gaia-X compliant with Gaia-X Policy Rules. A software cannot. However, a software can be made Gaia-X compatible with Gaia-X specifications.” Gaia-X Architecture Document 23.10 section 7.

[5] Note, though, that the author is not aware of *any other* implementation architectures except the one based on the Eclipse Data Space Components.

1.5 Terminological Note

The Gaia-X Architecture Working Group is currently not only restructuring the Gaia-X architecture document (see https://docs.gaia-x.eu/technical-committee/technical-committee/architecture-document/22.10/ for the latest version from October 2022) but also refining and actually changing core terms the terminology such as self-description and Gaia-X federations. As this process has not yet been completed – the next 23.09 release of the architecture document is slated for September 2023 – the following table provides an unofficial mapping between old and new terms.

In its current version, this document uses the 22.10 and earlier nomenclature.

Table 1. Gaia-X Terminology change – Mapping

22.10 and before	23.09 and after
Gaia-X self-description	Gaia-X credential
Gaia-X ecosystem	It is currently unclear which term will survive. For the purpose of this document, a Gaia-X ecosystem and a Gaia-X federation are identical.
Gaia-X federation	
	Gaia-X Digital Clearing House (GXDCH) This term is new in 23.09 and refers to a special form of Gaia-X federator which is also officially endowed with the power to create Gaia-C credentials, i.e., to attest that Gaia-X «self-descriptions» conform to Gaia-X policies and rules.

1.6 Contributions

The author gratefully acknowledges discussions and contributions regarding the structure and contents of this document with the following iECO project team members: Dirk Mayer, Marco di Pasquale (even on very short notice).

The author appreciates feedback and concrete clarifying remarks received from Klaus Ottradovetz, Gaia-X Architecture Working Group Lead, to version 1.00 which have been included in v1.10.

2 Gaia-X Overview

2.1 Gaia-X Technical Characterization

On a high level, Gaia-X aims to create a federated open data infrastructure based on European values regarding data and cloud sovereignty. The mission of Gaia-X is to design and implement a data sharing architecture that consists of common standards for data sharing, best practices, tools, and governance mechanisms. From a decidedly technical point of view, Gaia-X may be characterized as a **technical and organizational standard for a virtualization layer realizing a self-sovereign hybrid multi-cloud service mesh**[6]. This single-sentence definition pivots around the following attributes:

- **technical standards** – These include the reference to external technical standards such as DIDs[7], VCs[8] or ODRL[9] as well as new technical standards developed by the Gaia-X working groups themselves, for instance, Gaia-X «credentials» (aka Gaia-X «self-descriptions»).
- **organizational standards** – These refer to policies and procedures carried out by designated organizations within the wider Gaia-X domain. Typically, these standards are used to reliably establish the identity of Gaia-X «participants» but also to create and maintain "trust" regarding Gaia-X or service-relevant claims of «participants».
- **service mesh** – Transcending the data exchange-focused view of data spaces, Gaia-X employs «service» as the fundamental concept of value exchange between «participants». As «services» may be related to each by aggregation or composition, the set of all «services» may be seen as forming a kind of service mesh.
- **virtualization layer** – Gaia-X cannot be implemented by simply installing a single piece of software component or using a single piece of IT-related artifact (e.g., a X.509 certificate). Ecosystems or federations become Gaia-X "compliant" by following and implementing certain technical and procedural rules (= standards) on top or within their software and other IT-artifacts. In this sense, Gaia-X standards turn into the virtual glue that pull together the individual organizations and services participating in a certain Gaia-X federation.
- **hybrid** – This means that Gaia-X standards are agnostic to the cloud *vs*. edge *vs*. on premise distinction and may be used anywhere in the edge-to-cloud (E2C) continuum.
- **multi-cloud** – Likewise to the agnosticism with regards to the edge-to-cloud continuum, Gaia-X equally addresses IT landscapes and implementations involving several different cloud providers at the same time.
- **self-sovereign** – Self-sovereignty of every Gaia-X «participant», that is the autonomy of every single actor within Gaia-X with regards to its decision making process, is at the

[6] This concise and technically correct definition is not unanimously accepted within the Gaia-X Architecture Working Group but has been repeatedly published and presented in peer-reviewed environments.
[7] decentralized identifiers, a W3C standard for identities
[8] verifiable credentials, a W3C standard for claims
[9] Open Digital Rights Language, a W3C standard to specify usage rights of digital assets

foreground of the overall Gaia-X initiative. The technical implementation, hence, has to demonstrate how an actor is able to actually exercise its power of independent decision making with regards to Gaia-X-relevant constructs, e.g., which service consumers to allow.

2.2 Gaia-X Technical Architecture

Gaia-X standards for trust mediation rely on DID and VCs contrary to typical identity and access management (IAM) mechanisms and standards used in corporate IT environments such as OAuth2 or Open ID Connect (OIDC). In order to bridge this incompatibility, organizations will have to implement suitable integration technologies and components to bidirectionally translate between these two IAM regimes. A minimal[10] architecture to achieve this task for a single «participant» and the relevant «federation» and Gaia-X services is depicted in Figure 1 below.

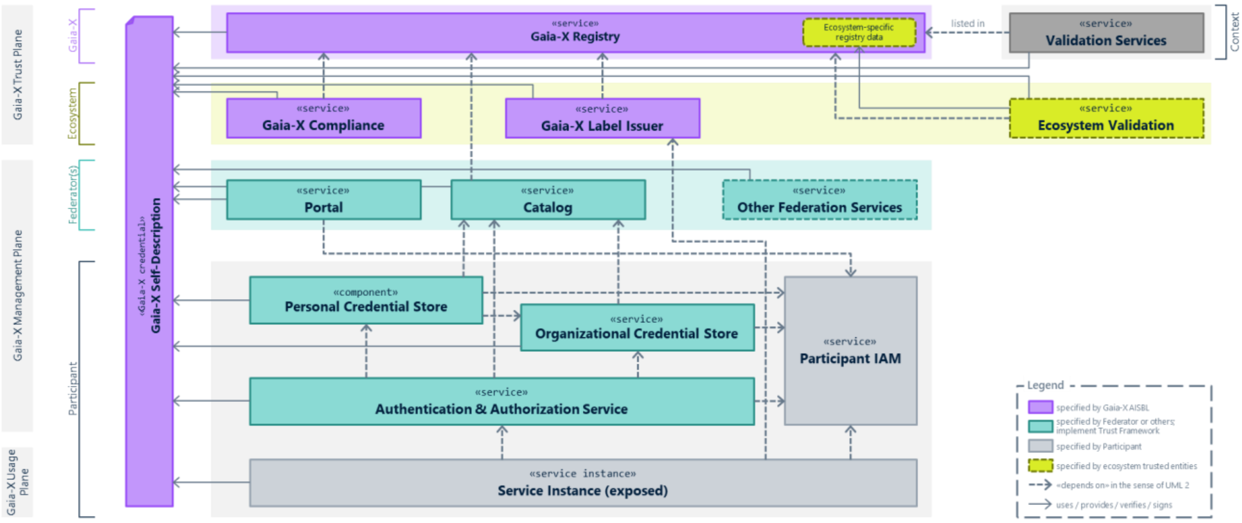


Figure 1. Gaia-X technical architecture

For potential Gaia-X «participants» this essentially means that they will have to install and operate suitable credential stores (think of "wallets" known from distributed ledger technologies) for legal persons in the form of an «Organizational Credential Store» and for natural persons (viz. employees) through «Personal Credential Stores». In principle, the translation of Gaia-X authentication and authorization flows to a «participant's» internal IAM should be accomplished by the «Authentication & Authorization Service». Obviously, services offered by a «participant», cf. the «service instance» artifact in Figure 1, will also need to interface with this service. Actual software packages implementing these services are being provided by the GXFS-DE project and are available – admittedly at a varying degree of maturity and documentation – at the time of the publication of this document. The current implementation of the «Authentication & Authorization Service» provides only an interface to «Participant IAM» systems supporting Open ID Connect (OIDC).

[10] see later in this section

For reasons of completeness, we also have depicted a minimal set of «federation services» for a given «federation» in the above architecture diagram such as the «Portal» and the «Catalog». These are complemented by the Gaia-X intrinsic services used to render conformity assessments. i.e., attesting "Gaia-X compliance" for the «self-descriptions» of «participants» or services. We discriminate two types of *validation*:

(i) validation services provided by suitable generic entities, so-called «Gaia-X trust anchors» as listed in the «Gaia-X Registry» – **«Validation Services»** in the diagram above.

(ii) ecosystem-specific conformity assessment bodies attesting that particular claims contained in a «self-description» fulfil the specified requirements -- **«Ecosystem Validation»** in the diagram above

The various artifacts in Figure 1 are described in more detail in the following Table 2.

Table 2. Gaia-X Technical architecture – Services and components

Gaia-X Service/Component	Description
Gaia-X self-description (= Gaia-X credential)	**Gaia-X self-descriptions** are structured, machine-readable, (automatically) verifiable documents used to describe and establish identity and authenticity of participants and of services offered by participants. **Identity** is established based on the mechanisms of decentralized identifiers (DID) which includes verification mechanisms. Properties and claims included in «**self-descriptions»** are based on the verifiable credentials (VC) standard.
Service Instance	Unfortunately, the Gaia-X Conceptual Model does not recognize *service* as separate (abstract) construct or term and only knows «service offerings» and «service instances». A **«service instance»** is a particular «service offering» from the Catalogue which is being performed and rendered (which is "executing", "running", "being enacted", "active"), at a particular moment in time.
Organizational Credential Store	Because of the sensitivity and criticality of an organization's «Participant Self-Description» (think of it like a passport), organizations will typically store their «Participant Self-Description» in a special facility which is called **«Organizational Credential Store»** in this document. Such a facility will typically limit access and usage rights to a set of especially privileged members of the organization (aka "administrators").
Personal Credential Store	In order to allow "normal" users and IT systems to act as service or data consumers (i.e., enable them to invoke another service) without having to use their organization's (sensitive) **«Participant Self-Description»**), they will typically receive personal credentials which can be linked in a provable way to their organization's self-description. This allows the invoked or called service to reliably authenticate the particular user.

Gaia-X Service/Component	Description
Participant IAM	The identity and access management (IAM) system an organization is using for internal purposes.
Authentication & Authorization Service	Due to the fact that the Gaia-X trust framework uses standards not yet widely implemented in participant's IAM systems, the **«Authentication & Authorization Service»** provides the necessary capabilities of translating Gaia-X trust standards such as DIDs and VCs to an organizations internal IAM standards such as OIDC or OAuth2.
Portal	The "home page" of a federation or ecosystem exposing useful information and, potentially, also workflows, related to the lifecycle of participants of this federation, e.g., explaining on-boarding requirements, capabilities to start the on-boarding workflow, links to or otherwise integrates important resources (e.g., the catalog[11]) and information regarding the inner workings of the federation.
(Federated) Catalog	A service storing a certain set of «service offerings» and exposing this information to external users (i.e., providing a GUI for people and an API for systems). Catalogs typically accept anonymous users as well as Gaia-X identified users (*viz.* users with a «Personal Self-Description» which has been suitable derived from a valid «Participant Self-Description»). The scope of «service offerings» actually included in a «Catalog» is not predefined by any Gaia-X standard and, consequently, may vary profoundly. Typically, it will be closely related to the ecosystem or federation within which the Catalog is operated. Sometimes, the «Catalog» is also called **«Federated Catalog»** in order to highlight the mesh-like, loosely coupled fabric of catalogs which is expected to emerge. In such a scenario, individual instances of a «Federated Catalog» communicate with other instances of «Federated Catalogs» in order to exchange and update information about «service offerings».
Other Federation Services	«Federators» (or GXDCHs) may autonomously decide to operate and provide services in addition to the **Portal** and the **Catalog** to the federation's participants, so called **other «Federation Services»**. Within the GXFS-DE project, the following other «Federation Services» have been developed: ■ Data Contract Service ■ Data Exchange Logging Service ■ Notarization API ■ Onboarding & Accreditation Workflow
Gaia-X Compliance	The «Gaia-X Compliance» service validates «Gaia-X Self-Descriptions», and checks and asserts whether any submitted «Gaia-X Self-

[11] The current (July 2023) GXFS-DE implementation of the «Portal» integrates the «Catalog» by acting as its (the «Catalog's» GUI.

Gaia-X Service/Component	Description
	Description» actually conforms to the relevant Gaia-X (and other applicable) standards. Currently, this service is provided by the Gaia-X AISBL itself. However, the Gaia-X strategy is to move this service to the «Federators» using suitable Gaia-X Trust Anchors as listed in the «Gaia-X Registry».
Gaia-X Label Issuer	Gaia-X foresees a standardized mechanism allowing «Gaia-X Self-Descriptions» of «participants» or «service offerings» to include certain properties or claims which can only be verified by dedicated entities (e.g., ISO certification bodies, security conformance attestation bodies, etc.). A «Gaia-X Label Issuer» simply is an *issuer* in the VC ecosystem, i.e., an entity asserting claims.
Ecosystem Validation	While «Gaia-X Label Issuers» have to follow certain Gaia-X standards for producing their assertions, some ecosystems or federations may extend this mechanism and define and standardize their own **«ecosystem validation»** services.
Gaia-X Registry	The **«Gaia-X Registry»** is a public distributed, non-repudiable, immutable, permissionless database with a decentralized infrastructure (and potentially additional capabilities not relevant for this document). It stores the core information necessary to create and operate Gaia-X ecosystems such as ■ list of the Trust Anchors providing «Validation Services» for Gaia-X ■ result of the Trust Anchors validation processes ■ revocations of Trust Anchors identities ■ URLs of Gaia-X credentials schemas defined by Gaia-X ■ URLs of Gaia-X Catalog’s credentials ■ other core Gaia-X governance-related data
Validation Service	A **«Validation Service»** is a «service» asserting certain claims contained in «Gaia-X Self-Descriptions». The entities providing these services need not be part of Gaia-X or a Gaia-X federation and may rather be thought of as highly trusted certification bodies or certificate issuers like quality conformance checking bodies, governmental units, and others.

3 Gaia-X Participants View

3.1 Ecosystem Overview

The following diagram presents an overview of the **four different roles** in any Gaia-X ecosystem and the major architectural artifacts, services, and components required.

The fundamental **service provider ⇔ service consumer** relationship (cf. the violet/dark purple elements at the bottom right in Figure 2) lies at the heart of any Gaia-X ecosystem. We note in passing that, of course, a single «Participant» may assume the role of «Service Provider» and «Service Consumer» at the same time.

The other two roles, (Gaia-X) federator and Gaia-X Core[12], "only" serve to facilitate this service-based value exchange by, e.g., providing trustworthy identities or other supportive services such as a service catalog.

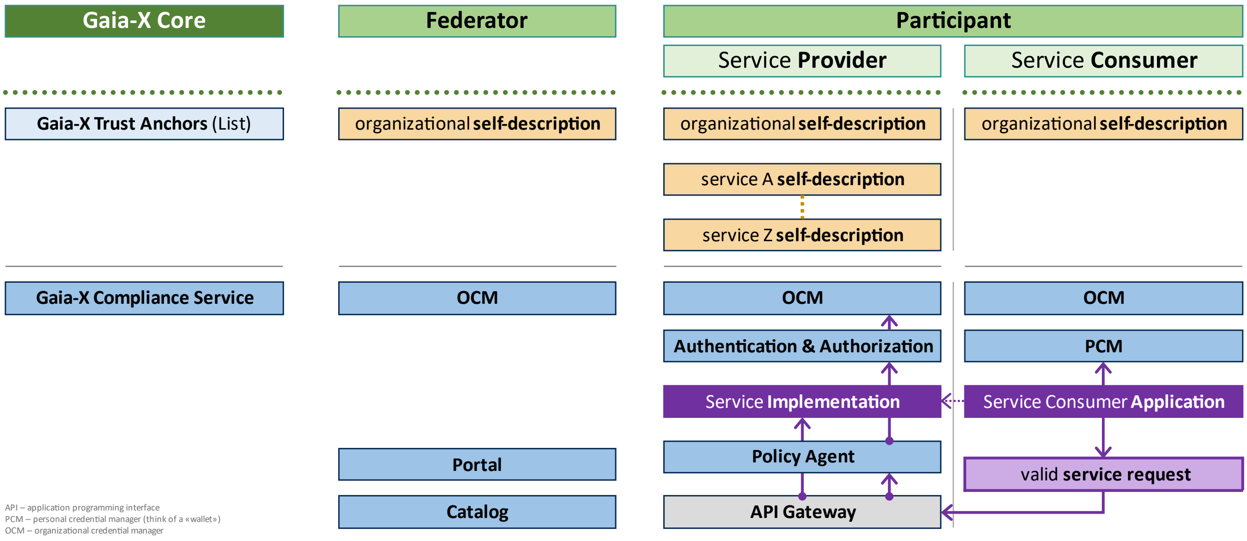

Gaia-X Core
Federator
Participant
Service Provider
Service Consumer
Gaia-X Trust Anchors (List)
organizational self-description
organizational self-description
organizational self-description
service A self-description
service Z self-description
Gaia-X Compliance Service
OCM
OCM
OCM
Authentication & Authorization
PCM
Service Implementation
Service Consumer Application
Policy Agent
valid service request
Portal
Catalog
API Gateway
API – application programming interface
PCM – personal credential manager (think of a «wallet»)
OCM – organizational credential manager

Figure 2. Gaia-X service consumption view.

The following sections describe the «service consumer» ⇔ «service provider» relationship and the roles of the «Federator» and the Gaia-X Core in more details

[12] The term "Gaia-X Core" is not part of the official Gaia-X Conceptual Model as the two services provided in this area shall be moved to either the Gaia-X federator itself (*viz.* the Gaia-X Compliance Service) or will be provided by completely other entities outside the Gaia-X AISBL or a Gaia-X federation, like issuing Gaia-X conformant identities, in the final version of Gaia-X. During the current design and implementation phase of Gaia-X ideas these services may be provided – interimistically – by other entities.

3.2 Service Provider-Service Consumer Relationship

At its core, Gaia-X realizes a virtualization layer allowing «service consumers» to invoke certain *services*[13] provided by certain «service providers».

While, in general, a ***service*** might be characterized as any provider-client interaction creating and capturing value, Gaia-X clearly restricts this generic (and highly appropriate) definition to *services* which are provided and consumed by IT systems. At the typical level of service-oriented architectures (SOA), microservices, and XaaS, a service then is any capability provided by some software components through a dedicated API. Gaia-X, though, does not limit its notion of «service offerings» and «service instances» to functions provided by software (*viz.*, programs or applications), but also includes hardware and IT infrastructure-related *services* such as providing access to a (fiber optic) STM-256[14] network or to an IoT device or anything else.

Using the terms and concepts from Figure 2, the table below traces in more detail the individual steps required for such a *service* interaction to take place. More specifically, it explains the steps

(i) the «service provider» needs to take in order to be able to create and execute a correct *service request* out of a software component[15] and

(ii) the «service consumer» needs to take in order to ensure whether it wants to allow this interaction to actually take place or not.

Table 3. Technical steps to call an exposed Gaia-X service

Step	Role / Component	Activity
1	Service Consumer	■ obtain an «Organizational Self-Description» for your own organization
2	Service Consumer OCM	■ securely store the received «Organizational Self-Description» in the «Organizational Credential Manager» OCM (think of this as something like a key store)
3	Service Consumer OCM	■ derive suitable personal «Gaia-X Self-Descriptions» for the users (natural persons and also IT systems as "technical users") from your own «Organizational Self-Description»
4	Service Consumer PCM	■ users store their individual personal «Gaia-X Self-Description» in their «Personal Credential Manager», PCM (think of this as a form of "wallet" on your mobile phone).
5	Service Consumer Catalog	■ Look up the «service self-description» of the *service* you want to invoke in the Catalog of your Federation. This *service* will be called *target service* in this sequence chart.

[13] Remember that Gaia-X does not identify the term «service» in its conceptual framework but only speaks of «service offerings» (typically listed in a «Catalog») and the actual execution of services, called «service instance».
[14] ITU-T Standards G.707, G.783, and G.803 for specifying the 39.813,120 Mbit/s level in the synchronous digital hierarchy (SDH).
[15] This trace does not include *services* other than those delivered by invoking certain software components.

Step	Role / Component	Activity
		■ Use this information to program a suitable service call into the application or software component which should, eventually, invoke the target service
6	Service Consumer Application	■ execute the service request, i.e., issue a suitable call against the specific API of the target service ■ This may be accomplished by any means available and ranges from the end user clicking on a suitable link via applications calling the HTTP/REST API endpoint exposing the target service to an IoT endpoint sending MQTT messages to a dedicated endpoint[16]
7	Service Provider API Gateway	■ receive the incoming service request at a suitable API Gateway ■ In addition to the typical Internet Firewall (not shown in Figure 2) organizations typically operate a dedicated application level firewall to guard against malicious API calls from the public internet. Within the API economy, this type of component is called API Gateway and acts as a first generic policy enforcement point (PEP).
8	Service Provider API Gateway	**Introduction.** Typically, an API Gateway's capabilities to define and process access and usage rights are limited (today) – especially compared to the highly sophisticated standards currently discussed in Gaia-X for formulating policies in machine-readable ways such as ODRL[17] or Rego[18]. In such cases, service consumers will want to couple their standard API Gateway to a suitable **Policy Agent** which is able to execute the individual policies attached to an invoked service. By doing so, the Policy Agent performs the role of a so-called **Policy Decision Point** (PDP). In such a case, the flow will continue as follows: ■ Using information from the service request just received (and suitable context information) the API Gateway asks the Policy Agent to decide whether the service consumer is allowed, at this point in time, to invoke the particular service or not. If the API Gateway itself is already capable of deciding such access and usage policies all by itself, the flow continues with step 12.
9	Service Provider Policy Agent	■ Acting as the PDP, the Policy Agent decides whether to allow the service invocation request to go through or not (= to deny) and sends its response (answer) back to the API Gateway. It does so by evaluating the relevant policies attached to the called target service. Currently, policies are directly (verbatim) included in the «self-

[16] Note for technical pundits: This endpoint does not even have to be a full-fledged MQTT broker.
[17] Open Digital Rights Language
[18] as specified by the CNCF for their Open Policy Agent (OPA)

Step	Role / Component	Activity
		description» of a «service offering», but we expect this to be changed to a more versatile link-based reference scheme. ■ In doing so, the Policy Agent will regularly have to authenticate the calling user and Participant. To the extent that the Policy Agent is not capable of doing that on its own (which is unlikely given the novelty of the DID- and VC-based standards used by Gaia-X), the Policy Agent then calls the **Authentication & Authorization Service** to verify the caller's identities.
10	Service Provider Authentication & Authorization Service	■ The Authentication & Authorization Service (AAS) will verify the identity of the user and the Participant trying to invoke the service. ■ This will often involve one's own **OCM** for checking foreign certificates. It may also involve using suitable information from external **«Trust Anchors»** (such as certificates or cryptographic proofs stored in the «Gaia-X Registry») to verify certain claims. ■ The AAS may also verify specific elements of the service request against other internal or application specific policies which have not yet been resolved by the API Gateway or the Policy Agent. ■ It returns an **`allow`** or **deny** answer to the calling Policy Agent.
11	Service Provider Policy Agent	■ The Policy Agent receives the evaluation result of the AAS ■ It completes its evaluation of the applicable access and usage policies (e.g., by executing the Rego code containing the policy[19]). ■ It returns an **`allow`** or **deny** answer to the calling API Gateway.
12	Service Provider API Gateway	If the policy evaluation (including a potential response from the Policy Agent) is **`allow`**: ■ Forward the service invocation request to the appropriate internal API endpoint at the software component implementing the called target service. ■ This creates a (potentially longer lasting network) connection between the service consumer application and the service provider application which is used for actually performing the service. If the policy evaluation (including a potential response from the Policy Agent) is **deny**: ■ Return a suitable error code to the service consumer application (such as an HTTP 403 Forbidden code).

[19] Unfortunately, no generic ODRL "execution engine" exists. This means that every «Participant» will need to program every individual ODRL rule by themselves.

3.3 Gaia-X Federator

The following diagram in Figure 3[20] provides an overview of the **functional architecture** for a **Gaia-X Federator** (or GXDCH) for the two most important use cases:

(i) A **(human) user** wants to obtain authenticated access the Gaia-X Portal provided by the Federator and

(ii) A **service provider** wants to have some of their «service offerings» included in the «Federated Catalog» of the Federator.

Architecture modeling convention notes:

- Rectangular "boxes" in the diagram may denote services, applications or other software components, apps on your mobile phone, documents, or any function connected to the aforementioned elements (e.g., management)
- The fact that boxes are arranged inside the shaded domain of one of the three actors ("User", "Federator", and "Provider") only means that the particular function is being executed, called, or invoked on behalf of the actor of this domain. It does not mean that the actor would in any way provide this function as part of its own responsibilities. This relates to the functions «GX Compliance Service» and the «Self-Description Wizard» in particular.

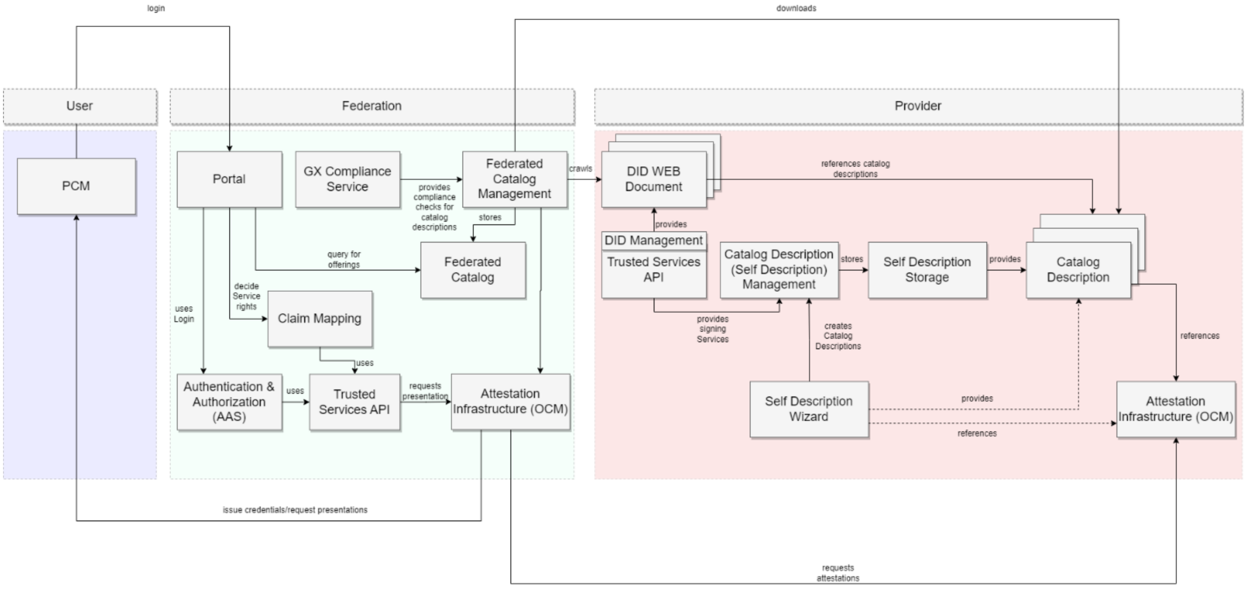


Figure 3. Gaia-X federator functional architecture

Notes:

[1] (Human) user interactions are limited to users obtaining authenticated access to the «Portal».

[2] (Human) user interaction with the «Federated Catalog» is provided only via the «Portal» and not directly.

[20] https://gitlab.eclipse.org/eclipse/xfsc/integration, last accessed on 13 July 2023

[3] «Service Offerings» of a «Service Provider» are found by having the «Federator» (the «Federated Catalog Management» function in the diagram) **crawl** suitable trusted sources for valid Gaia-X service self-descriptions (here: DID:WEB). The crawling algorithm then downloads the actual «self-description» of a «service offering» from a suitable endpoint reference in the «self-description» and typically being provided by the «service providers» themselves.
An alternative implementation, where «service providers» use a suitable API of the «Federated Catalog» for uploading the «self-descriptions» of their «service offerings» is possible in principle and very likely to be realized in practice, but not shown in the architecture diagram.

[4] Even though the catalog function is called «Federated Catalog», the federation mechanism itself is not in any way included in the architecture diagram.

[5] All implementation details including mandatory (and complex) dependencies to external blockchains/DLTs regarding identification and authentication are missing.

4 Gaia-X Implementation Remarks

4.1 General Remark & GXFS

It must be emphasized that **there exists no single way of "implementing" Gaia-X**. Any set of suitable software components implementing or realizing Gaia-X technical and procedural standards may be used by an organisation to actively participate in any Gaia-X federation or ecosystem.

Nevertheless, a set of open-source software components has been developed in 2022/23 under the name of **Gaia-X Federation Services** (GXFS-DE – because the project has been funded by the German government[21]) in order to accelerate the wide-spread adoption of Gaia-X. Because of the (time) lag between specification and implementation, the current version v1 of the GXFS implementation does not yet incorporate all features and functions of the latest Gaia-X architecture document.

4.2 Gaia-X Federator Implementation

4.2.1 Complex dependencies

Version v1 of the GXFS-DE heavily rely on some particular architectural choices which do not apply to many lighthouse projects including iECO. The currently known constraints are as follows.

Table 4. GXFS-DE constraints.

No	GXFS-DE Constraint	iECO Situation
01	**Sovereign Cloud Stack (SCS)** Heavy reliance on the sovereign cloud stack as underlying IaaS environment, e.g., Argo as orchestrator.	CNCF-certified Kubernetes environment. CI/CD scripts no longer work
02	**Hetzner-optimized** Deployment seems to be optimized for the German IaaS provider	A1 Digital/Exoscale uses different IaaS environments.
03	**DID:Indy for identity** Software components rely on DID:Indy and the corresponding Hyperledger Aries/Indy stack for anchoring identities as opposed, for instance, to the much wider accepted and accessible DID:WEB mechanism.	We would either use **DID:WEB** or another more commonly available format. Alternatively, we may want to use IOTA Identity for this.

[21] The French government has also funded the development of some GXFS which is known as GXFS-FR. However, this project seems to rather focus on developing (and filling) a service catalog than developing Gaia-X Federation Services. See https://cispe.cloud/first-gaia-x-federated-cloud-services-catalogue-demonstrated/

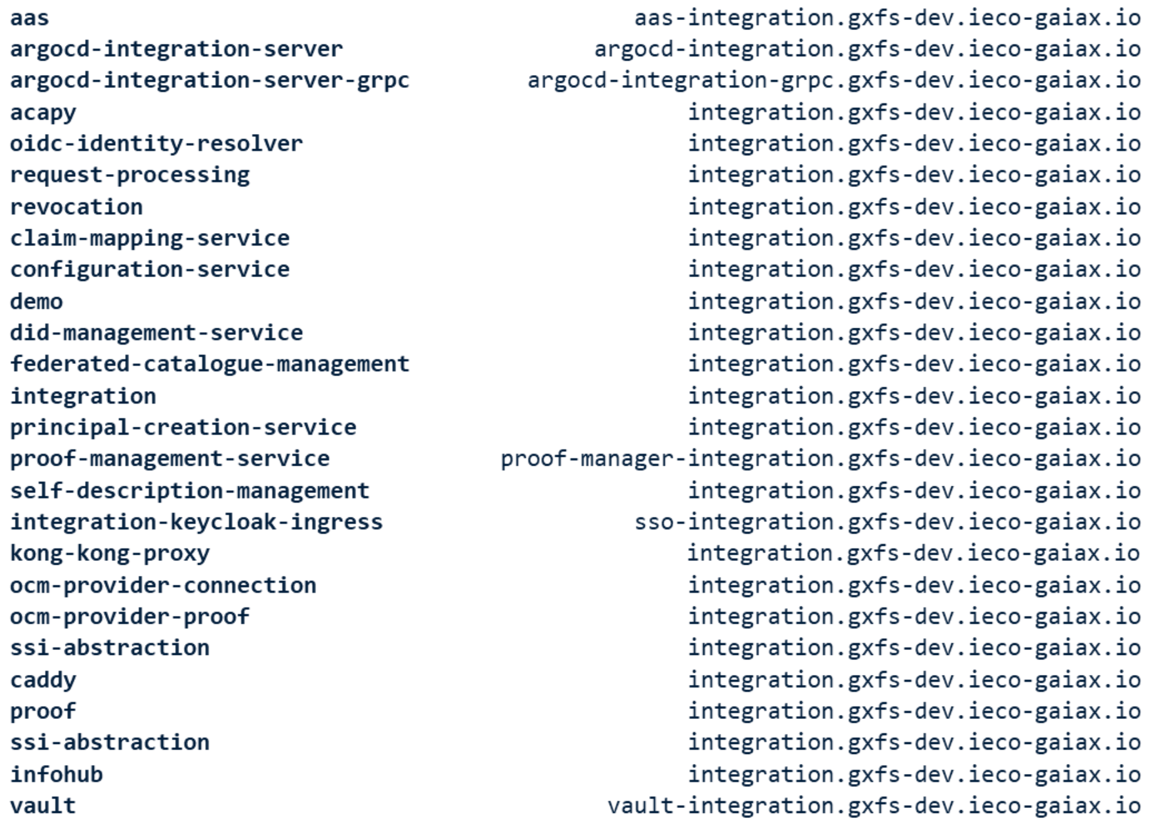
No	GXFS-DE Constraint	iECO Situation
	Moreover, the DID:Indy implementation seems to be directly implemented at the core of the relevant software components (such as OCM and PCM) without any abstraction layer.	
04	**Complex DID:Indy dependencies** In order to create identities as a Federator using GXFS-DE, one needs to have suitable access to a running Hyperledger Aries/Indy implementation such as ID Union.	We would either use DID:WEB or another more commonly available format. Alternatively, we may want to use IOTA Identity for this.

4.2.2 Complex deployment

The EuProGigant Gaia-X lighthouse project has already started implementing GXFS-DE on an A1 Digital/Exoscale IaaS infrastructure and was surprised at the extraordinary complexity of the implementation consisting of

- **7** host names required
- **26** endpoints (all in need of configuration)
- **75+** individual microservices (which all need to be deployed and up and running)

The following figure gives an impression of the complexity of the software required for a Gaia-X Federator. Note that node names and domains (e.g., `ieco-gaiax.io`) are illustrative only (at this point in time).



```
aas                                            aas-integration.gxfs-dev.ieco-gaiax.io
argocd-integration-server                   argocd-integration.gxfs-dev.ieco-gaiax.io
argocd-integration-server-grpc         argocd-integration-grpc.gxfs-dev.ieco-gaiax.io
acapy                                              integration.gxfs-dev.ieco-gaiax.io
oidc-identity-resolver                             integration.gxfs-dev.ieco-gaiax.io
request-processing                                 integration.gxfs-dev.ieco-gaiax.io
revocation                                         integration.gxfs-dev.ieco-gaiax.io
claim-mapping-service                              integration.gxfs-dev.ieco-gaiax.io
configuration-service                              integration.gxfs-dev.ieco-gaiax.io
demo                                               integration.gxfs-dev.ieco-gaiax.io
did-management-service                             integration.gxfs-dev.ieco-gaiax.io
federated-catalogue-management                     integration.gxfs-dev.ieco-gaiax.io
integration                                        integration.gxfs-dev.ieco-gaiax.io
principal-creation-service                         integration.gxfs-dev.ieco-gaiax.io
proof-management-service             proof-manager-integration.gxfs-dev.ieco-gaiax.io
self-description-management                        integration.gxfs-dev.ieco-gaiax.io
integration-keycloak-ingress                   sso-integration.gxfs-dev.ieco-gaiax.io
kong-kong-proxy                                    integration.gxfs-dev.ieco-gaiax.io
ocm-provider-connection                            integration.gxfs-dev.ieco-gaiax.io
ocm-provider-proof                                 integration.gxfs-dev.ieco-gaiax.io
ssi-abstraction                                    integration.gxfs-dev.ieco-gaiax.io
caddy                                              integration.gxfs-dev.ieco-gaiax.io
proof                                              integration.gxfs-dev.ieco-gaiax.io
ssi-abstraction                                    integration.gxfs-dev.ieco-gaiax.io
infohub                                            integration.gxfs-dev.ieco-gaiax.io
vault                                        vault-integration.gxfs-dev.ieco-gaiax.io
```

Figure 4. GXFS-DE – Complex federator software

Abbreviations

Even though the following table cannot (and does not want to) provide a full-fledged glossary of the, admittedly, sometimes arcane, abbreviations used in this document, the gentle reader may find the following spelling out of all used abbreviations convenient.

Table 5. Abbreviations

Abbreviation	Long Form
AAS	Authentication & Authorization Service
AISBL	Association Internationale Sans But Lucratif[22]
aka	also known as
API	application programming interface
CI/CD	continuous integration/continuous deployment
CNCF	Cloud-Native Computing Foundation
DE	Germany
DID	decentralized identifiers (W3C)
DLT	distributed ledger technology
E2C	edge-to-cloud (continuum)
FR	France
GXDCH	Gaia-X Digital Clearing House
GXFS	Gaia-X Federation Services
HTTP	Hypertext transfer protocol
IaaS	infrastructure as a service
IAM	identity and access management (system)
ID	identity
iECO	intelligent empowerment of construction industry (Gaia-X lighthouse project)
IoT	Internet of things
ISO	International Standardization Organisation (iso.ch)
IT	information technology
Mbit/s	million (1,000,000) bits per second
MQTT	Message Queuing Telemetry Transport (protocol)
OCM	Gaia-X organizational credential manager
ODRL	Open Digital Rights Language (W3C)
OPA	Open Policy Agent (CNCF)

[22] International non-profit association seated in Belgium

Abbreviation	Long Form
PaaS	platform as a service
PCM	Gaia-X personal credential manager
PDP	policy decision point
PEP	policy enforcement point
REST	representational state transfer
SCS	sovereign cloud stack
SDH	Synchronous Digital Hierarchy
STM	synchronous transmission mode (a level in SDH)
UML	unified modeling language
URL	uniform resource locator
VC	verifiable credentials (W3C)
W3C	World Wide Web Consortium
XaaS	anything as a service