RE4DY

MANUFACTURING DATA NETWORKS

RE4DY TOOLKIT

Name of the Tool	DIDI (Dataspace for Industrial Data Intelligence) Data Marketplace
Tool Owner	Industry Commons Foundation
Version	1.0
Date	Nov 2025
Version	V1.0

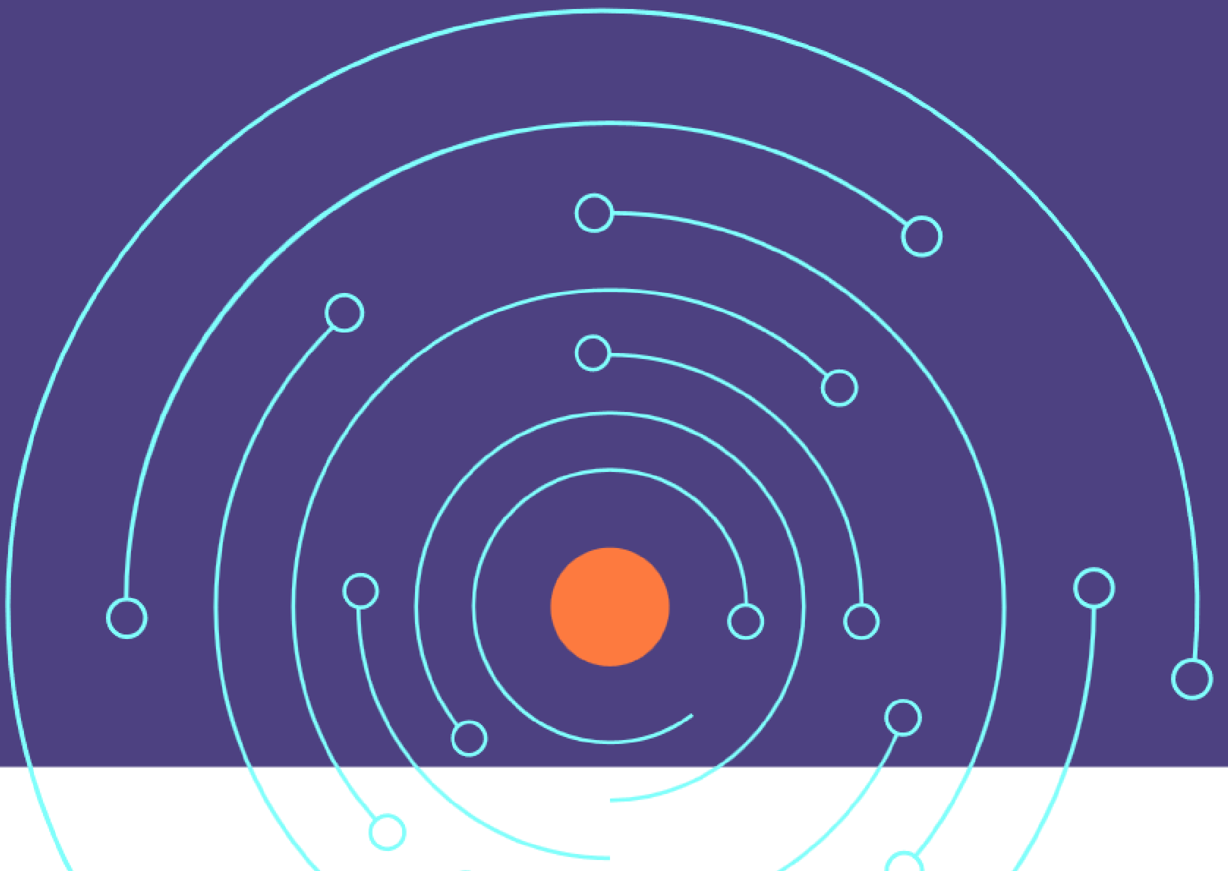


Table of contents

1. Component Description

Atos' DIDI Data Marketplace solution leverages on the power of Gaia-X Data Spaces architecture and open-source components, and provides a secure, interoperable, and highly efficient environment for organizations to seamlessly exchange, store, and manage data across a wide range of industries[1]. With advanced features such as data sovereignty, trustworthiness, and data sharing capabilities at its core, DIDI Data Marketplace solution empowers businesses to unlock new opportunities, drive innovation, and foster collaboration while maintaining strict data privacy and security standards. It serves as a versatile platform for companies seeking to harness the full potential of their data, enabling them to thrive in an increasingly data-driven world.

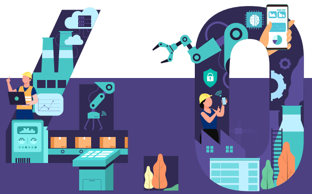
2. Input / Output

In the context of Gaia-X, dataspace **self-descriptions** refer to the metadata and information associated with a dataspace that provide essential details about its characteristics, data assets, and usage policies. These self-descriptions play a crucial role in facilitating data sharing and collaboration within the Gaia-X ecosystem, as they enable organizations to discover, evaluate, and interact with dataspaces in a standardized and trustworthy manner.

The following JSON snippets represent different examples of Self-Descriptions used in the interaction (via their APIs) with the GAIA-X components that conform DIDI Data Marketplace.

For detailed examples on how to use the DIDI Data Marketplace components (as well as their input and outputs), please refer to the OpenAPI links provided for each of them in the API section.

[1] **Note:** The 'Dataspace for Industrial Data Intelligence (DIDI) Data Marketplace' asset represents the evolution of former Atos' asset 'AGORA Data Marketplace'



Participant Self-Description example:

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1"
  ],
  "@id": "http://example.edu/verifiablePresentation/self-description1",
  "type": [
    "VerifiablePresentation"
  ],
  "verifiableCredential": [
    {
      "@context": [
        "https://www.w3.org/2018/credentials/v1"
      ],
      "@id": "https://www.example.org/legalPerson.json",
      "@type": [
        "VerifiableCredential"
      ],
      "issuer": "http://gaiax.de",
      "issuanceDate": "2022-10-19T18:48:09Z",
      "credentialSubject": {
        "@context": {
          "gax-core": "https://w3id.org/gaia-x/core#",
          "gax-trust-framework": "https://w3id.org/gaia-x/gax-trust-framework#",
          "xsd": "http://www.w3.org/2001/XMLSchema#",
          "vcard": "http://www.w3.org/2006/vcard/ns#"
        },
        "@id": "gax-core:Participant1",
        "@type": "gax-trust-framework:LegalPerson",
        "gax-trust-framework:registrationNumber": "1234",
        "gax-trust-framework:legalAddress": {
          "@type": "vcard:Address",
          "vcard:country-name": "Country",
          "vcard:locality": "Town Name",
          "vcard:postal-code": "1234",
          "vcard:street-address": "Street Name"
        },
        "gax-trust-framework:headquarterAddress": {
          "@type": "vcard:Address",
          "vcard:country-name": "Country",
          "vcard:locality": "Town Name",
          "vcard:postal-code": "1234",
          "vcard:street-address": "Street Name"
        },
        "gax-trust-framework:termsAndConditions": {
          "@type": "gax-trust-framework:TermsAndConditions",
          "gax-trust-framework:content": {
            "@type": "xsd:anyURI",
            "@value": "http://example.org/tac"
          },
          "gax-trust-framework:hash": "1234"
        },
        "gax-trust-framework:subOrganisation": [
          {
            "@id": "http://example.org/Provider1_1"
          },
          {
            "@id": "http://example.org/Provider1_2"
          }
```

```
          ],
          "gax-trust-framework:legalName": "Provider Name"
        },
        "proof": {
          "type": "JsonWebSignature2020",
          "created": "2022-12-02T16:05:37Z",
          "proofPurpose": "assertionMethod",
          "verificationMethod": "did:web:compliance.lab.gaia-x.eu",
          "jws":
"eyJiNjQiOmZhbHNlLCJjcml0IjpbImI2NCJdLCJhbGciOiJQUzI1NiJ9..efXJBVbLUieloVXmK11FEmM8ke_QYhf3ObjxKHz
zdujrwgToHcpVKNUYwlnUXk7-V6cuaUFwbWY1zl89u0U2Nu2UKA1kP6iBmUyg1JnWXsCtF1dpFyCZMhrYdbJxF-
USa9f4RbTDcymhbRx8ZI9R4qdhDGuxDrez_Nzl2DlJfSL7hE9JBG7R8cgAq1LIfWCTN0xjVr8QvVw3R_HIilDSLv-
Clf1WCSAl_7CWXDEGInBW6l7lrJ7efrjZ5GnEwbHZi0b2V0v9hjidqkMc1xl5pl9fIPk5wHsoVLdKgfJ8hUD-
EtyuFJGPwk77Mqf7BVcuKR4zqmNX3s0CPCGmAmxXfGyY5ARITjgmI4gXcp1I1ax1YTCjENR8nqgS8rouO6l3xAAFgBq
LdqfR1KdDiCYPFd0fTfX1T8A8yiT8o6Pg36Vewy-Jb79aWk9byxSS-
xKaQGkSMbpZeErz4FKTXLYwcD8bLXcndPMtF5UlBC_A_t-_BnG5cLeZddSow-
4sMs7g2qvnURIX_KRrPtuP5GRVG_cNPJagY5bzUc8lAlSNsRsivf5wJDwrRvYjn3U6FSV8sOBBGVv6UBlopl1JWjUL-9-
QLmZUD1jPv9mYUWWChm2YB8dvCjkrvwRCbHFAuab2rYNag61EcYI3lcGwS3Qez-P4AKIpRfTXEpnCNIzJr2E"
        }
      }
    ],
    "proof": {
      "type": "JsonWebSignature2020",
      "created": "2022-12-02T16:05:37Z",
      "proofPurpose": "assertionMethod",
      "verificationMethod": "did:web:compliance.lab.gaia-x.eu",
      "jws":
"eyJiNjQiOmZhbHNlLCJjcml0IjpbImI2NCJdLCJhbGciOiJQUzI1NiJ9..IYM1hcL55GNc115qwjdwAiHoxnx7DD4MwWcYN
UDgRO2Tj6vcEtKI5Ao1f_uwpTEJBImYrd4tZL9ojDNBOTmOnxFWorsUB-iq5PMvM11xS19tl-
hEhRVRY0mnFkT9er2xArWShcO6cNTnDAJuWGCtHxsU-bH3HMHCvT9u2WWKIIIJi9Axp-
CGwnNaF7vddEatiXRfuZCj8RCYxKa5goCxE4vueI-OMqIF-AWU86FjTNjDXS9DJI2yYt91SFdgxQqfuG0pJF7oJv9LI-
9bJjMRKBSfjPO1hqbxuPzxb4a7nywQedl1_2k2WttUQ3ZDsyju7ktkuGDDPL9p3xq6zyBsFN9shJqdS-9-
tw3Nptu_EsNj1vJdNgrOZt1VdfOEGoUsDtfNg2O5XtBt4-
TbE617_SgkWQhbzWAPWj47QJjcYJklzkJku7DwBicysvCiWGPOgCzFalTZPm6SL55Bz4a0UtEW8WzjtlxD_j3Mh7WW_
sA3vAO9oBLRaWJhQkt0f6EG7sElETizVDB9fq0Ur5QYuHhPmCqZVr5C0VSSBz4OSE_N7DFusNqLhox13f37myfs0RiT
pmwEh24EP67kyZSuUtOx004fqAWifIYiyBZj2OTdxnXy7NFiOZl80RM1cJ7WTL02-
159_7E88y1LiL39uWUatFYJBCKnaMRO5uQ8BGU"
    }
```

```
}
```

Provider’s Offering Self-Description

```
{
    "@id": "http://example.edu/verifiablePresentation/self-description1",
    "proof": {
        "created": "2023-03-01T11:05:08Z",
        "jws":
"eyJiNjQiOmZhbHNlLCJjcml0IjpbImI2NCJdLCJhbGciOiJQUzI1NiJ9..RMGGGRlviGNdoBBTygv9JohKwB88mUQirUOZn
OhxvNma-
kBvP1kXvnnMFVitJ26wsctJjk5hM4EDiagGaxNplBPtFgrrilp6UgGlQiQIx8arjAnwktQIx3HF9E7uZ5iucwlfrsUjro5svej4aqS
5n-tCNDW6wwkLgfd1frjIkWb01GA17qa3U1E762-1pnu5Z3EOSatufcnXgY3GdCOezP57y8-CLrpE9s8iTtRfNuJbs-
4ka09akuDq_Z-6K_buQ5IXo5_B9C69DAkH_uzj9CGEgZxRwQg-
ZykVXdDenfm9TkT2mPlqolt30cUyhDatMwHkOZpRk1vDdQMTmJwdPt_vpVpyXUGcN59ND5stuscXkU43e0lWoivtEs9e
HmcuFz-mRsx6VnI-
84qkH_TPdB2g5gfsH0HJCB9poK8wAPpNq3O7hf9rnctwt6doc_ZOKkZrVtJT9zaSR78vcER1FXInxx9Qe6idcIYR105YS9
Q_VNJSE0vRj6fddi3B05FjweLW3XIqpDmRsFkuAhZI6aI6auNRtLlpNg-
xgr_64Ek9EcaLc7PLw6iCvmdIYkl9NzUZgF0EDBPfOHIyjVprVO3bZ16WezbqjGnw6uR2MukDpretLN4ZbhqOiJM80BMz
6Kdyk99bWy1xtCseu2Z6eNjwo3QcPhaZfw0pFeWb0V8",
        "proofPurpose": "assertionMethod",
        "type": "JsonWebSignature2020",
        "verificationMethod": "did:web:compliance.lab.gaia-x.eu"
    },
    "type": ["VerifiablePresentation"],
    "@context": ["https://www.w3.org/2018/credentials/v1"],
    "verifiableCredential": {
        "issuanceDate": "2022-10-19T18:48:09Z",
        "credentialSubject": {
            "gax-trust-framework:policy": "list of policy",
            "gax-trust-framework:serviceTitle": "Software Title",
            "dcat:keyword": [
                "Keyword1_1",
                "Keyword1_2",
                "Keyword1_3",
            ],
            "@type": "gax-trust-framework:ServiceOffering",
            "gax-core:offeredBy": {"@id": "https://www.example.org/Provider1.json"},
            "@id": "https://www.example.org/mySoftwareOffering",
            "@context": {
                "gax-trust-framework": "https://w3id.org/gaia-x/gax-trust-framework#",
                "xsd": "http://www.w3.org/2001/XMLSchema#",
                "dcat": "http://www.w3.org/ns/dcat#",
                "gax-core": "https://w3id.org/gaia-x/core#"
            },
            "gax-trust-framework:dataAccountExport": {
                "gax-trust-framework:formatType": "application/gzip",
                "@type": "gax-trust-framework:DataAccountExport",
                "gax-trust-framework:accessType": "digital",
                "gax-trust-framework:requestType": "API"
            },
            "gax-trust-framework:termsAndConditions": {
                "gax-trust-framework:content": {
                    "@value": "http://example.org/tac",
                    "@type": "xsd:anyURI"
                },
                "@type": "gax-trust-framework:TermsAndConditions",
                "gax-trust-framework:hash": "1234"
            }
        },
        "@type": ["VerifiableCredential"],
        "@id": "https://www.example.org/SoftwareOffering.json",
        "proof": {
            "created": "2023-03-01T11:05:07Z",
            "jws":
"eyJiNjQiOmZhbHNlLCJjcml0IjpbImI2NCJdLCJhbGciOiJQUzI1NiJ9..Y9vkgfzscLnY82AUTJsevtve5pmLv28VP39SvjaeK
```

```
vUk9hUyzFIVrIUxfhIcMowRExLxzMVFKm3lz_1km6-6TU3HkKHgMIhUny6VEv_ozPQYYAWv-
TWahefjnpb1Ta6SI_K7mA4gpuKOFg7C-WQDz8JL-
L6OHCSeiPmBUxkhmKdcMFbBX4dMoj90Oqj05hCQqmSs_bYaipMXsa8lDlaZhNQ9mk10wqe3SEf-Cp9YS0d-
dvJvdw23HqB_DqH6UFoQ_NcANUKUKOxHDjIafCXhL8j59KezM0fV8rmVzRDxk5wsD0vjAwAOz4tCiiOxVPjQMdQub4I
SNEw0ZcLGOYYKsRkIHQHYQo1cbgWg_f8Z6hebI5Jx8rDdwz0vnJky1mw5I09CprPq03Kh6E9PXWPiIJFU4jZxZdDygin
m3bZfVaKZ07KRF3HEq00vOgjHBpNaNXkxloUi6hmVEGYTp04OQ-
jopQM87zvnoIOqB78jQpRSQK2CYckAph_R0Sna_qI5unb5t-
29DDNKuRxYOBJoK5O5i0OgbfecY82FyTvPOoYr2q7tN1WONlZqqBdlyqoo_NBsJ3xVsJQMXeTpF8XCOb9iCn-
R3qpCZFdyg-6C0MWMTft8dEZIqVOfyuDSKyiz24qiHufVt94v87LIYIkDnNIFitkLqh4hQ7R4uYIMtEo",

      "proofPurpose": "assertionMethod",

      "type": "JsonWebSignature2020",

      "verificationMethod": "did:web:compliance.lab.gaia-x.eu"

    },

    "@context": ["https://www.w3.org/2018/credentials/v1"],

    "issuer": "http://gaiax.de"

  }

}
```

Contract Negotiation Request example:

```
{

  "@context": {

    "edc": "https://w3id.org/edc/v0.0.1/ns/",

    "odrl": "http://www.w3.org/ns/odrl/2/"

  },

  "@type": "NegotiationInitiateRequestDto",

  "connectorId": "provider",

  "connectorAddress": "http://localhost:19194/protocol",

  "consumerId": "consumer",

  "providerId": "provider",

  "protocol": "dataspace-protocol-http",

  "offer": {

    "offerId": "MQ==:YXNzZXRJZA==:YTc4OGEwYjMtODRlZi00NWYwLTgwOWQtMGZjZTMwMGM3Y2Ey",

    "assetId": "assetId",

    "policy": {
      "@id": "MQ==:YXNzZXRJZA==:YTc4OGEwYjMtODRlZi00NWYwLTgwOWQtMGZjZTMwMGM3Y2Ey",
      "@type": "Set",
      "odrl:permission": [],
      "odrl:prohibition": [],
      "odrl:obligation": [],
      "odrl:target": "assetId"
    }
  }
}
```

3. Information Flow

This section outlines the basic interaction among various participants and the DIDI Data Marketplace components.

Participants, including Providers and Consumers within the ecosystem, are identified and comprehensively described through valid Self-Descriptions[2], created either before or during the onboarding process. Providers articulate their Service Offerings and make them available in the Federated Catalogue. In parallel, Consumers explore Service Offerings in Gaia-X Catalogues coordinated by Federators and the Gaia-X Registry. Once a Consumer identifies a suitable Service Offering in a Gaia-X Catalogue, Contract negotiation between Provider and Consumer establishes the specific conditions for providing the Service Instance.

The subsequent diagram (Figure 1) illustrates the general workflow for Gaia-X service provisioning and consumption processes.

[2] https://gaia-x.eu/wp-content/uploads/2022/08/SSI_Self_Description_EN_V3.pdf

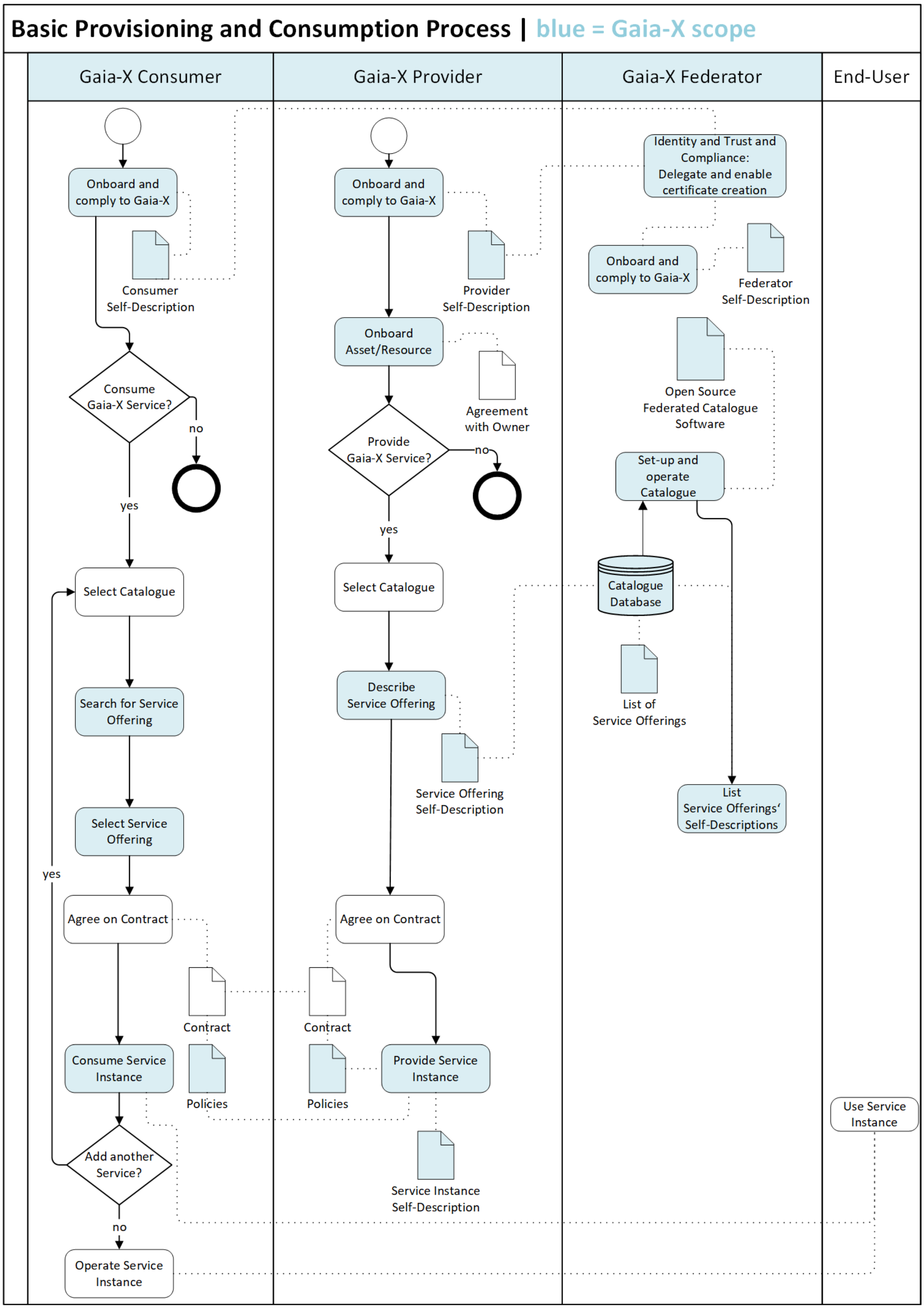

Basic Provisioning and Consumption Process | blue = Gaia-X scope
Gaia-X Consumer
Gaia-X Provider
Gaia-X Federator
End-User
Identity and Trust and Compliance: Delegate and enable certificate creation
Onboard and comply to Gaia-X
Consumer Self-Description
Onboard and comply to Gaia-X
Provider Self-Description
Onboard and comply to Gaia-X
Federator Self-Description
Onboard Asset/Resource
Agreement with Owner
Open Source Federated Catalogue Software
Consume Gaia-X Service?
no
yes
Provide Gaia-X Service?
no
yes
Set-up and operate Catalogue
Select Catalogue
Select Catalogue
Catalogue Database
Search for Service Offering
Describe Service Offering
List of Service Offerings
Service Offering Self-Description
List Service Offerings' Self-Descriptions
Select Service Offering
yes
Agree on Contract
Agree on Contract
Contract
Contract
Consume Service Instance
Policies
Policies
Provide Service Instance
Use Service Instance
Add another Service?
Service Instance Self-Description
no
Operate Service Instance

Figure 1 Basic Provisioning and Consumption Process (source GAIA-X)

For the sake of clarity, some of the flows in the above diagram are broken down and depicted in more detail in separate interaction diagrams illustrating the processes and the information flow among the DIDI Data Marketplace components:

Onboarding process

The onboarding process for registering Participants and Consumers involves several key steps to ensure a smooth and secure integration into the GAIA-X ecosystem. Participants, both Providers, and Consumers, initiate the onboarding process by creating a comprehensive Self-Description. This Self-Description document provides essential information about the entity, including identification details, capabilities, and data-related specifications, and it serves as a foundational element for interactions within the GAIA-X ecosystem.

The Self-Description undergoes a validation process to ensure its accuracy and compliance with GAIA-X standards. This step helps maintain the integrity of the information provided by the Participant or Consumer during the onboarding process.

Approved Participants and Consumers receive the necessary credentials and access rights to engage within the GAIA-X ecosystem. This may include authentication tokens, keys, or other secure means of access.

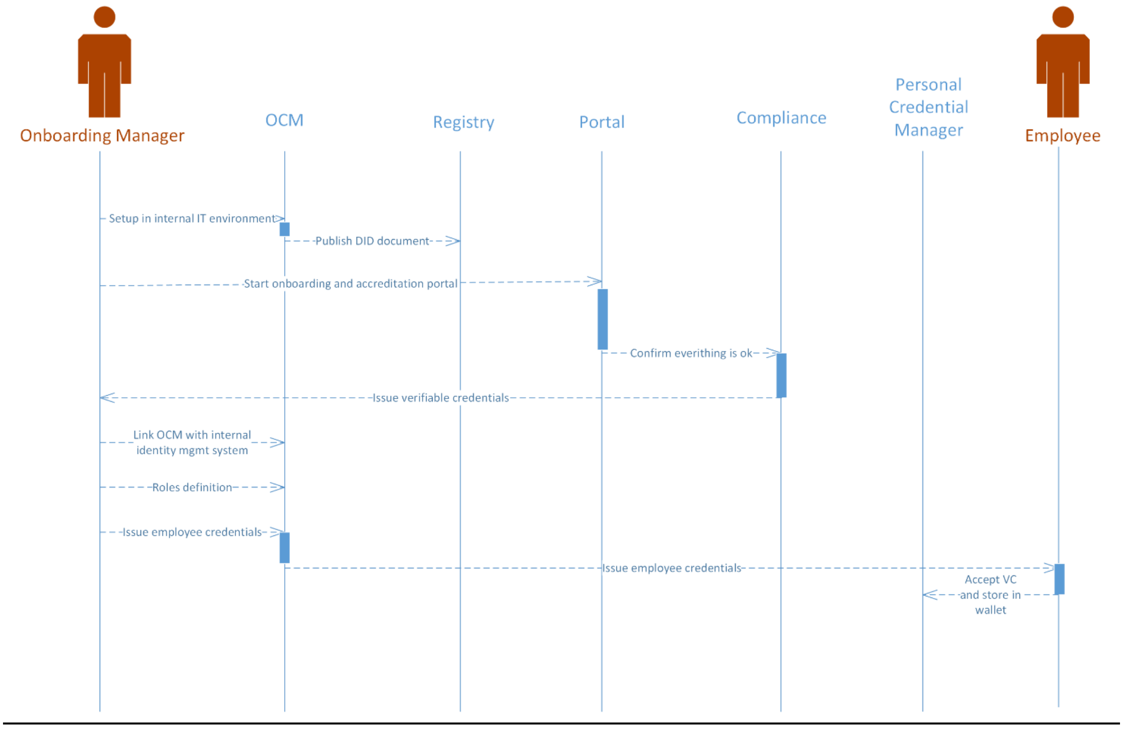

Onboarding Manager
OCM
Registry
Portal
Compliance
Personal Credential Manager
Employee
Setup in internal IT environment
Publish DID document
Start onboarding and accreditation portal
Confirm everithing is ok
Issue verifiable credentials
Link OCM with internal identity mgmt system
Roles definition
Issue employee credentials
Issue employee credentials
Accept VC and store in wallet

Figure 2 Onboarding Process

Federated Catalog

- Onboarding a new Participant in the Catalogue

This flow describes the onboarding of a new Participant to the catalogue.

Once the Approved Participants (in this case, the Providers) receive the necessary credentials and access rights to engage within the GAIA-X ecosystem they can define their Service Offerings. This involves specifying the types of services or data they intend to make available within the GAIA-X Catalogue.

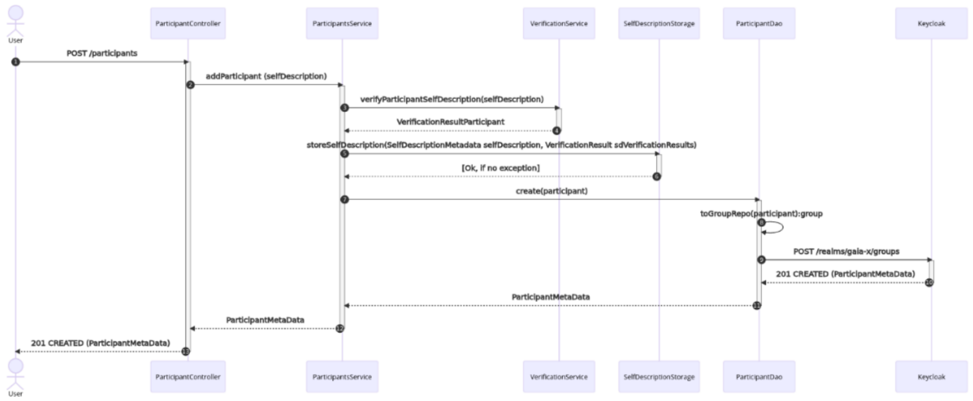

User
ParticipantController
ParticipantsService
VerificationService
SelfDescriptionStorage
ParticipantDao
Keycloak
POST /participants
addParticipant (selfDescription)
verifyParticipantSelfDescription(selfDescription)
VerificationResultParticipant
storeSelfDescription(SelfDescriptionMetadata selfDescription, VerificationResult sdVerificationResults)
[Ok, if no exception]
create(participant)
toGroupRepo(participant):group
POST /realms/gaia-x/groups
201 CREATED (ParticipantMetaData)
ParticipantMetaData
ParticipantMetaData
201 CREATED (ParticipantMetaData)
User
ParticipantController
ParticipantsService
VerificationService
SelfDescriptionStorage
ParticipantDao
Keycloak

Figure 3 Onboarding a new Participant (source: GAIA-X, https://gaia-x.gitlab.io/data-infrastructure-federation-services/cat/architecture-document/images/diag-a12195ee3006b3f0ab733e721e31291f.png)

- Adding a Self-Description for an Offering

The process begins with a Provider who intends to add a new offering to the GAIA-X Catalogue.

1. Authentication and Authorization: The Provider authenticates their identity using the appropriate credentials. The Catalogue service checks the Provider's authorization level to ensure they have the necessary permissions to add a new Self-Description for the offering.
2. Create or Update Self-Description: The Provider creates or updates a Self-Description for the offering. This Self-Description includes crucial information such as identification details, capabilities, data specifications, and any other relevant metadata.
3. Validation: The Catalogue service may perform validation checks on the submitted Self-Description to ensure it meets GAIA-X standards and is consistent with the required format.
4. Submission to Catalogue: Once the Self-Description is complete and validated, the Provider submits it to the GAIA-X Catalogue. The Catalogue service records the new or updated Self-Description in its database.

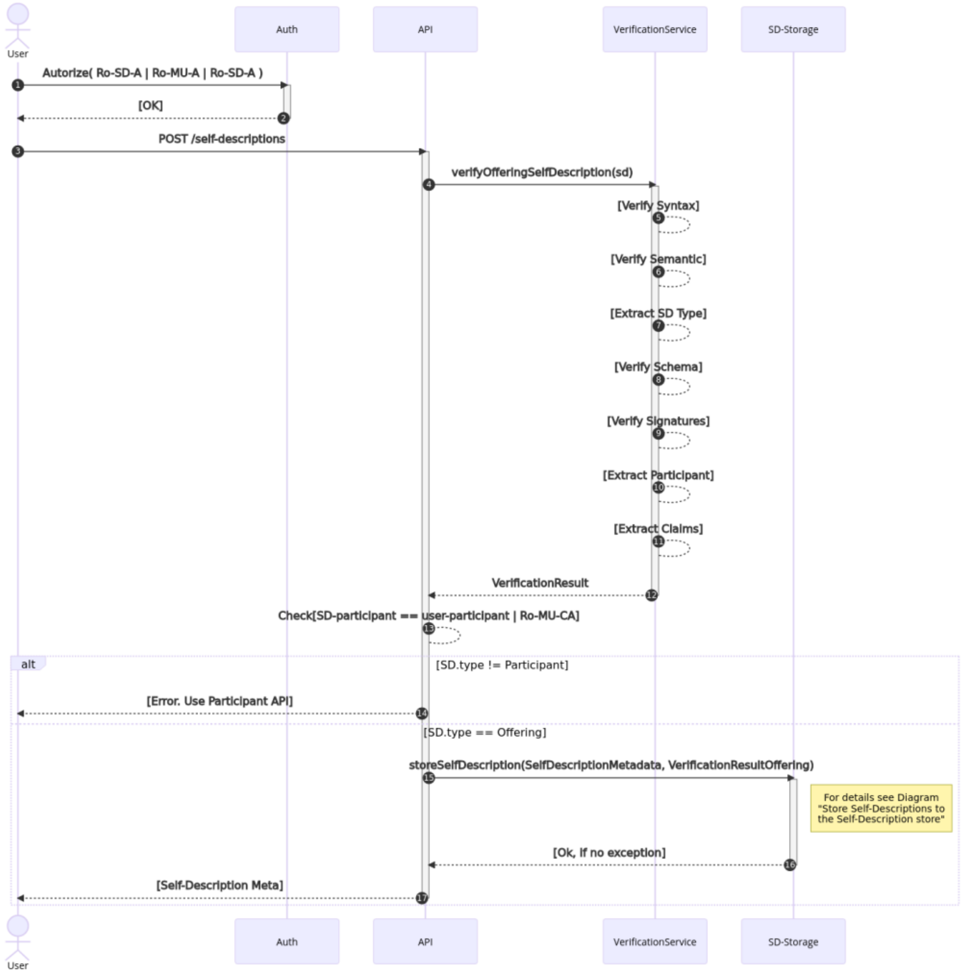


Figure 4 Adding a Self-Description for an Offering (source: GAIA-X, https://gaia-x.gitlab.io/data-infrastructure-federation-services/cat/architecture-document/images/diag-0b8bbfdd68c0886283d848ecbadcf283.png)

- Requesting all Self-Descriptions

  Once the Approved Participants (in this case, the Consumers) receive the necessary credentials and access rights to engage within the GAIA-X ecosystem they can interact with the GAIA-X Catalogue to explore available Service Offerings. The Catalogue provides a comprehensive view of the offerings, enabling Consumers to make informed decisions.

  The Consumer can narrow down the search within the Catalogue to find specific Self-Descriptions by using specific keywords and filters.

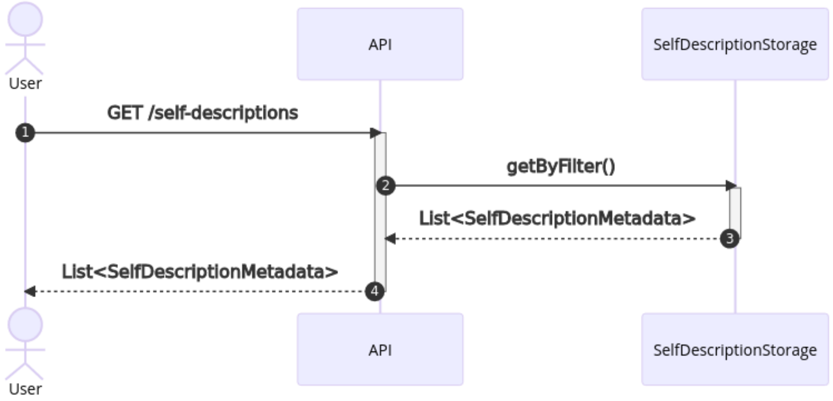


Figure 5: Requesting all Self-Descriptions (source: GAIA-X, https://gaia-x.gitlab.io/data-infrastructure-federation-services/cat/architecture-document/images/diag-6e1c2d3c7a88f8d142a97e259c67c314.png)

- Requesting a specific Self-Description

The Consumer can request access to the detailed Self-Description or additional information from the search results by specifying the concrete Self-Description hash in a new request to the Catalogue.

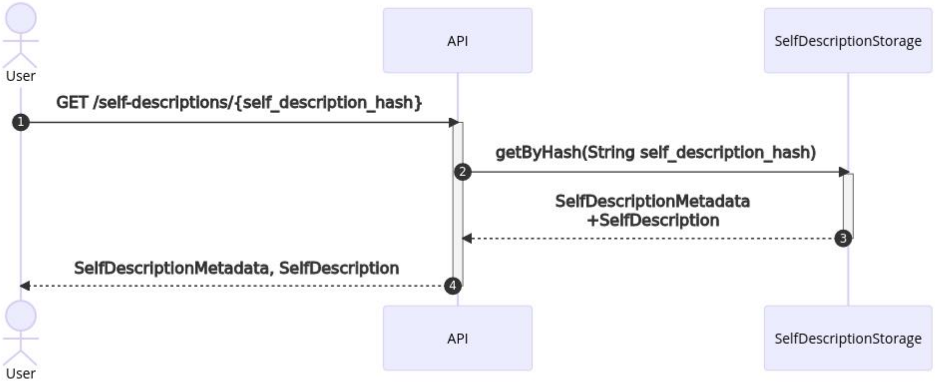

User
API
SelfDescriptionStorage
GET /self-descriptions/{self_description_hash}
getByHash(String self_description_hash)
SelfDescriptionMetadata
+SelfDescription
SelfDescriptionMetadata, SelfDescription
1
2
3
4

Figure 6: Requesting information about a specific Self-Description (source: GAIA-X, https://gaia-x.gitlab.io/data-infrastructure-federation-services/cat/architecture-document/images/diag-5832aadb62702f484c64f9335a6171ae.png)

Contract Negotiation

When a Consumer identifies a relevant Service Offering, they engage in Contract negotiations with the respective Provider. The negotiation process establishes the terms and conditions under which the Service Instance will be provided.

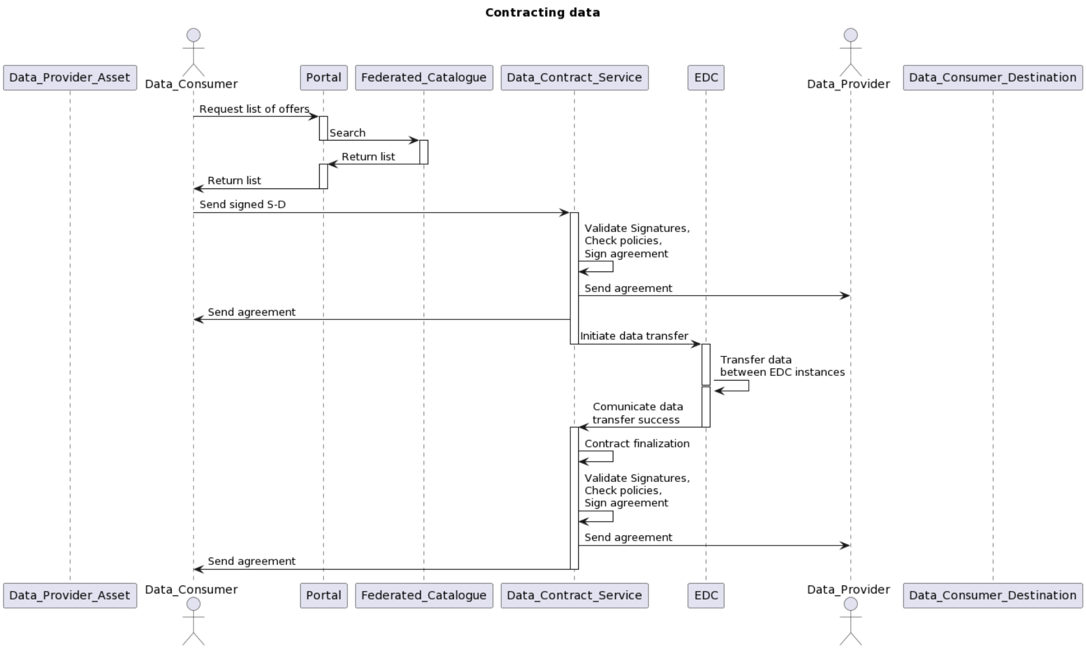


Figure 7: Contract Negotiation of an Offering

4. Internal Architecture

The current implementation of DIDI Data Marketplace is composed, as shown in Figure 8, of the following GAIA-X components:

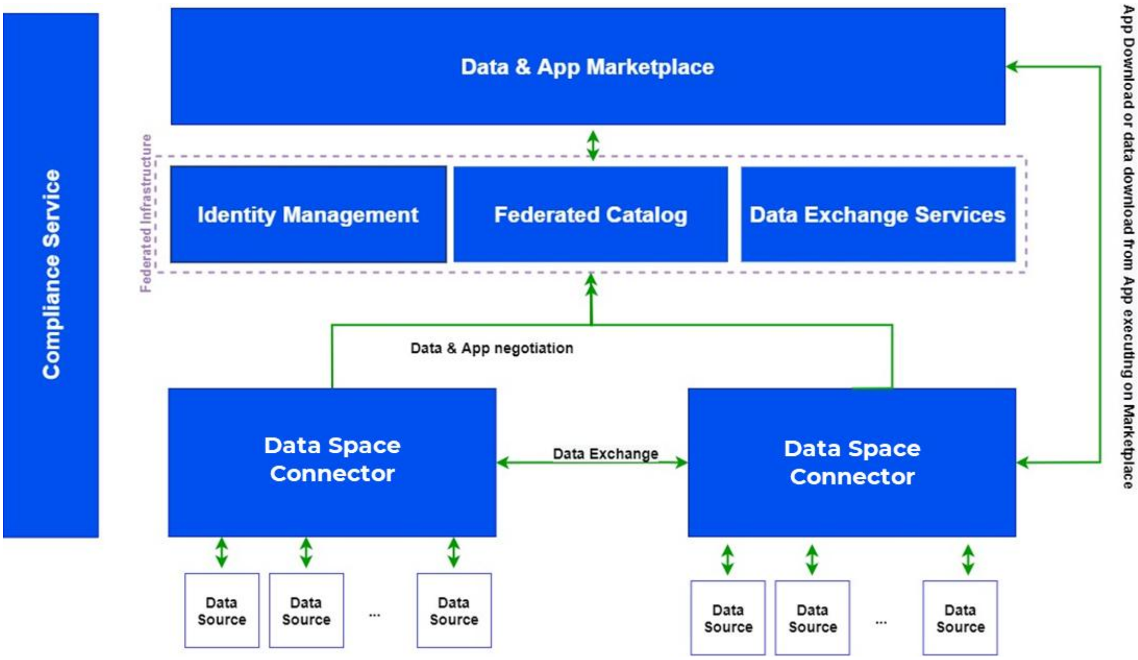


Figure 8: DIDI Data Marketplace High-level Architecture

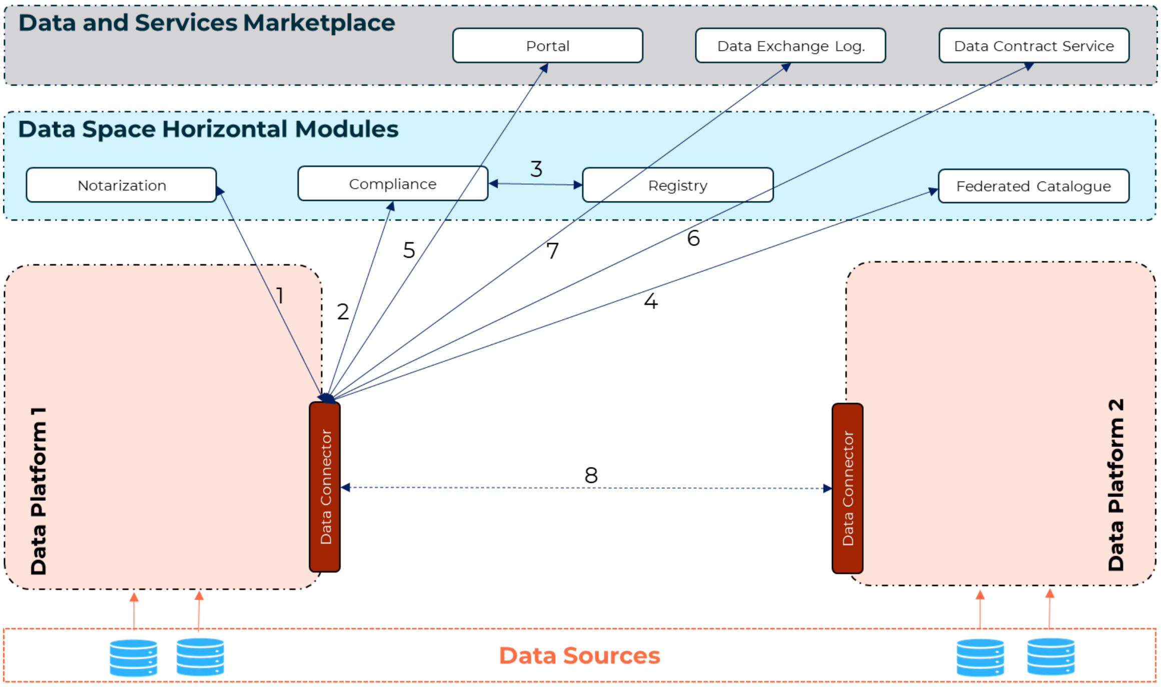


Figure 9: DIDI Data Marketplace High-level Architecture (sequence of interactions between the Data Connector and the rest of components)

Data Space Horizontal Modules:

- *Notarization Service*: it is a tool used to return a Verified Credential (VC) once the API successfully verifies the registration numbers provided by the participant.
- *Compliance Service*: It ensures that the required level of information for the users to take educated decisions is available and the information is verified or verifiable. This service is implemented via the Gaia-X Digital Clearing House, which runs services of the Gaia-X Framework necessary to achieve compliance. It integrates to external TA (Trust Anchors), Identity Verification (like eIDAS), and other TDS (Trusted Data Sources).
- *Registry Service*: it can verify X.509 certificates and public keys of a set of Trust Service Providers according to the Trust Framework document. The module offers the basis for the proof of compliance issued by the Compliance Service and the basis to validate the chain of trust for the Compliance Service.
- *Federated Catalogue:* it is a central component that facilitates the discovery, exchange, and usage of data and services within the GAIA-X ecosystem. The Catalogue Service essentially serves as a registry or index, providing metadata (self-descriptions) and information about available data, services, and resources. The Catalogue Service incorporates the concept of self-descriptions by allowing entities to register and publish metadata about themselves. Thus, entities within the GAIA-X ecosystem, such as data

sources or services, can use the Catalogue Service to provide detailed self-descriptions. This information helps users discover and understand the available data and services.

Data and Services Marketplace:

- *Data Contract Service:* it enables and governs data-sharing agreements between different participants in the GAIA-X ecosystem. This includes businesses, organizations, and other entities that contribute, share, or consume data and services.
- *Data Exchange Logging Service:* log and monitor activities related to the exchange of data within the GAIA-X ecosystem. It helps in tracking events and transactions associated with data sharing, providing an audit trail for accountability and compliance purposes. It contributes to regulatory compliance by providing a mechanism for organizations to demonstrate adherence to data protection, privacy, and other relevant standards.
- *Portal*: The frontend provides a user-friendly interface that allows participants to interact with the DIDI Data Marketplace. This includes data providers, data consumers, and any other stakeholders in the ecosystem. Among the features it provides we can find:
    - *Data Discovery*: Users can search and discover available datasets through the frontend. This may involve browsing categories, using search filters, and accessing detailed information about each dataset.
    - *Data Listings*: The frontend displays listings for various datasets, including metadata such as data source, format, size, and any other relevant information. This helps users make informed decisions about which datasets to explore or acquire.
    - *Authentication and Authorization*: The frontend includes features for user authentication and authorization. Participants may need to log in to access certain features, and permissions are enforced based on roles and agreements established in the GAIA-X ecosystem.
    - *Data Contracts and Agreements*: Participants can use the frontend to review and negotiate data-sharing contracts and agreements. This may include specifying terms such as data usage, access controls, and any other conditions agreed upon by the parties involved.
    - *Integration with Catalogue and Logging Services*: The frontend integrates with other GAIA-X services, such as the Catalogue Service and Data Exchange Logging Service, to ensure that data listings are accurate and up-to-date and that data exchange activities are appropriately logged.
    - *Transaction and Payment Management*: If the marketplace involves transactions or payments, the frontend may include features for managing these processes.

This could include payment gateways, transaction histories, and billing information.

Data Exchange Services: These services are designed to facilitate secure data exchange and interoperability between different data platforms and providers. They may include data integration, transformation, and messaging services:

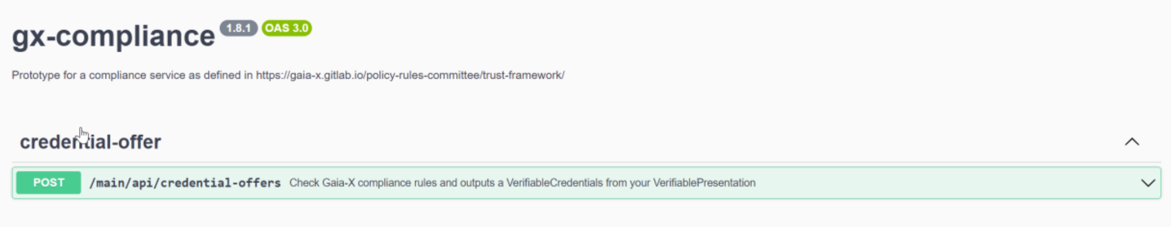
- *Data Connector*: it is designed to promote interoperability by providing a standardized mechanism for connecting and exchanging data between different participants in the GAIA-X ecosystem. Besides, it adheres to GAIA-X principles of data sovereignty and security, ensuring that data transfers and integrations comply with privacy regulations and security standards.

5. API

The following section provides a list of the main APIs of DIDI Data Marketplace components.

Compliance Service

Link to OpenAPI specification: https://compliance.lab.gaia-x.eu/main/docs



gx-compliance 1.8.1 OAS 3.0

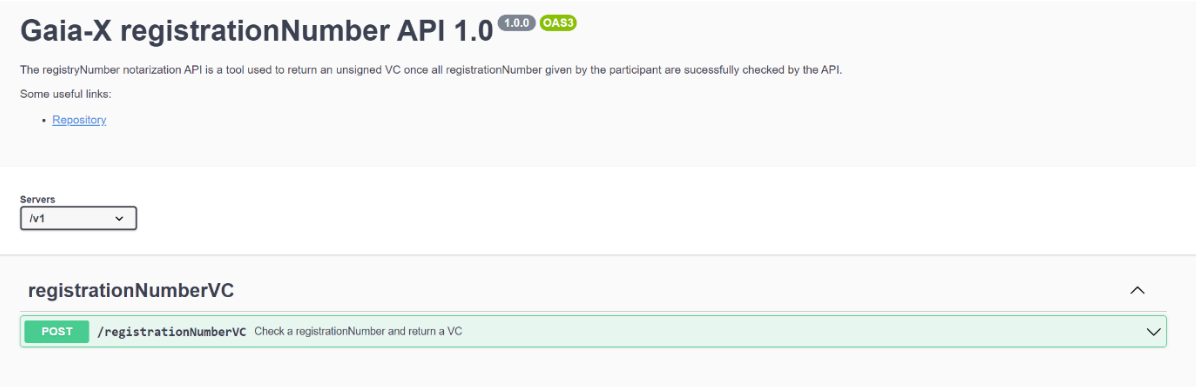
Prototype for a compliance service as defined in https://gaia-x.gitlab.io/policy-rules-committee/trust-framework/

credential-offer

POST /main/api/credential-offers Check Gaia-X compliance rules and outputs a VerifiableCredentials from your VerifiablePresentation

Notarization Service

Link to OpenAPI specification: https://registrationnumber.notary.gaia-x.eu/v1/docs/



Gaia-X registrationNumber API 1.0 1.0.0 OAS3

The registryNumber notarization API is a tool used to return an unsigned VC once all registrationNumber given by the participant are sucessfully checked by the API.

Some useful links:

- Repository

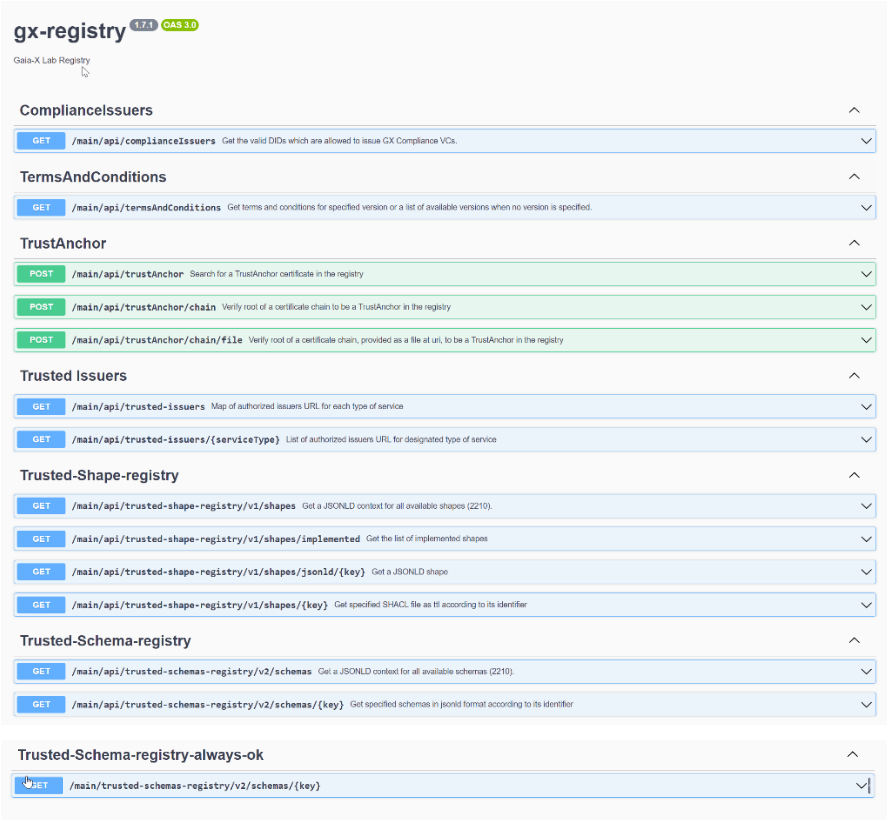
Servers: /v1

registrationNumberVC

POST /registrationNumberVC Check a registrationNumber and return a VC

Registry Service

Link to OpenAPI specification: https://registry.lab.gaia-x.eu/main/docs/



gx-registry 1.7.1 OAS 3.0

Gaia-X Lab Registry

ComplianceIssuers

GET /main/api/complianceIssuers Get the valid DIDs which are allowed to issue GX Compliance VCs.

TermsAndConditions

GET /main/api/termsAndConditions Get terms and conditions for specified version or a list of available versions when no version is specified.

TrustAnchor

POST /main/api/trustAnchor Search for a TrustAnchor certificate in the registry

POST /main/api/trustAnchor/chain Verify root of a certificate chain to be a TrustAnchor in the registry

POST /main/api/trustAnchor/chain/file Verify root of a certificate chain, provided as a file at uri, to be a TrustAnchor in the registry

Trusted Issuers

GET /main/api/trusted-issuers Map of authorized issuers URL for each type of service

GET /main/api/trusted-issuers/{serviceType} List of authorized issuers URL for designated type of service

Trusted-Shape-registry

GET /main/api/trusted-shape-registry/v1/shapes Get a JSONLD context for all available shapes (2210).

GET /main/api/trusted-shape-registry/v1/shapes/implemented Get the list of implemented shapes

GET /main/api/trusted-shape-registry/v1/shapes/jsonld/{key} Get a JSONLD shape

GET /main/api/trusted-shape-registry/v1/shapes/{key} Get specified SHACL file as ttl according to its identifier

Trusted-Schema-registry

GET /main/api/trusted-schemas-registry/v2/schemas Get a JSONLD context for all available schemas (2210).

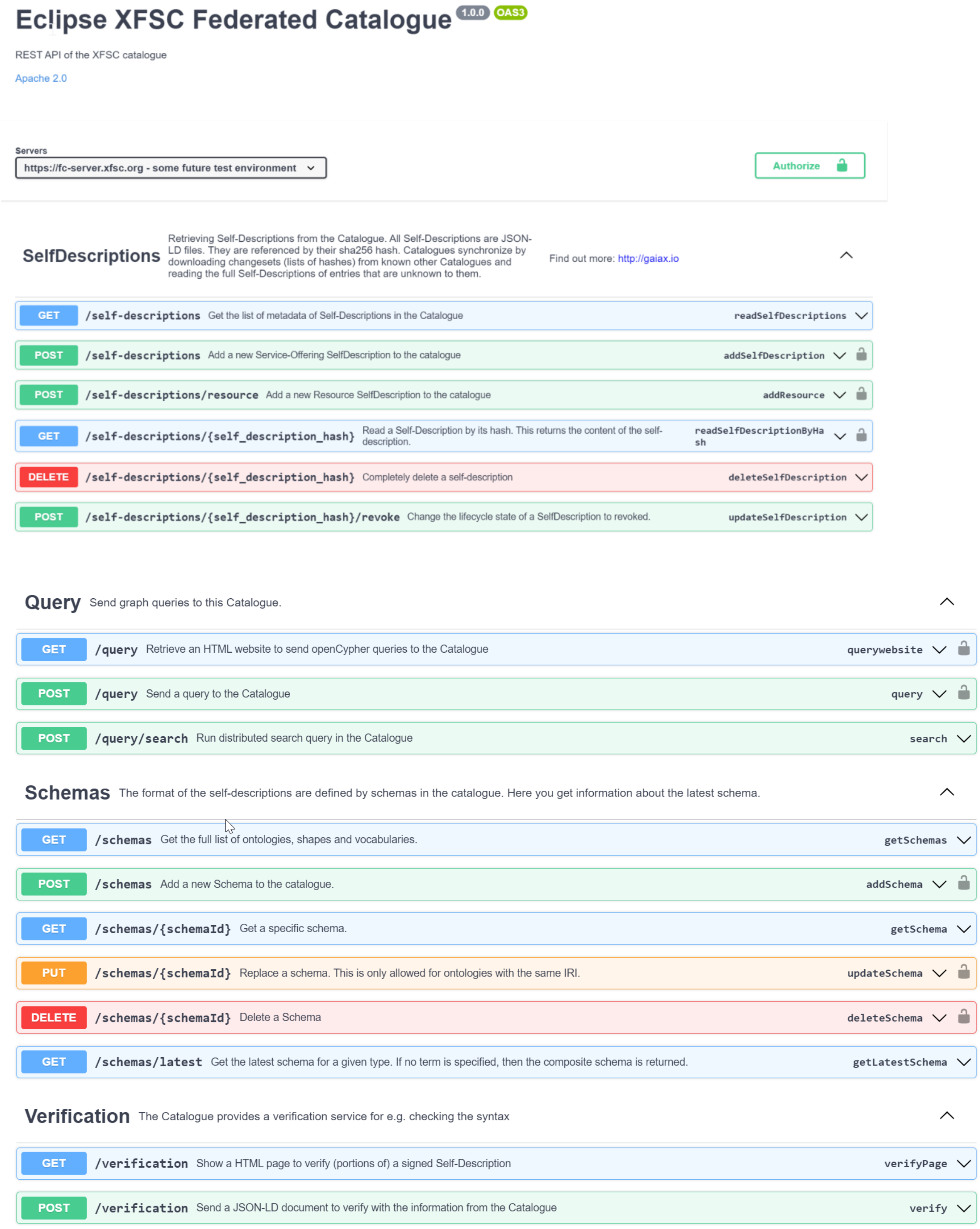
GET /main/api/trusted-schemas-registry/v2/schemas/{key} Get specified schemas in jsonld format according to its identifier

Trusted-Schema-registry-always-ok

GET /main/trusted-schemas-registry/v2/schemas/{key}

Federated Catalogue APIs

Link to OpenAPI specification: https://gitlab.eclipse.org/eclipse/xfsc/cat/fc-service/-/blob/main/openapi/fc_openapi.yaml



Eclipse XFSC Federated Catalogue 1.0.0 OAS3

REST API of the XFSC catalogue

Apache 2.0

Servers

https://fc-server.xfsc.org - some future test environment

SelfDescriptions

Retrieving Self-Descriptions from the Catalogue. All Self-Descriptions are JSON-LD files. They are referenced by their sha256 hash. Catalogues synchronize by downloading changesets (lists of hashes) from known other Catalogues and reading the full Self-Descriptions of entries that are unknown to them.

Find out more: http://gaiax.io

GET /self-descriptions Get the list of metadata of Self-Descriptions in the Catalogue readSelfDescriptions

POST /self-descriptions Add a new Service-Offering SelfDescription to the catalogue addSelfDescription

POST /self-descriptions/resource Add a new Resource SelfDescription to the catalogue addResource

GET /self-descriptions/{self_description_hash} Read a Self-Description by its hash. This returns the content of the self-description. readSelfDescriptionByHash

DELETE /self-descriptions/{self_description_hash} Completely delete a self-description deleteSelfDescription

POST /self-descriptions/{self_description_hash}/revoke Change the lifecycle state of a SelfDescription to revoked. updateSelfDescription

Query

Send graph queries to this Catalogue.

GET /query Retrieve an HTML website to send openCypher queries to the Catalogue querywebsite

POST /query Send a query to the Catalogue query

POST /query/search Run distributed search query in the Catalogue search

Schemas

The format of the self-descriptions are defined by schemas in the catalogue. Here you get information about the latest schema.

GET /schemas Get the full list of ontologies, shapes and vocabularies. getSchemas

POST /schemas Add a new Schema to the catalogue. addSchema

GET /schemas/{schemaId} Get a specific schema. getSchema

PUT /schemas/{schemaId} Replace a schema. This is only allowed for ontologies with the same IRI. updateSchema

DELETE /schemas/{schemaId} Delete a Schema deleteSchema

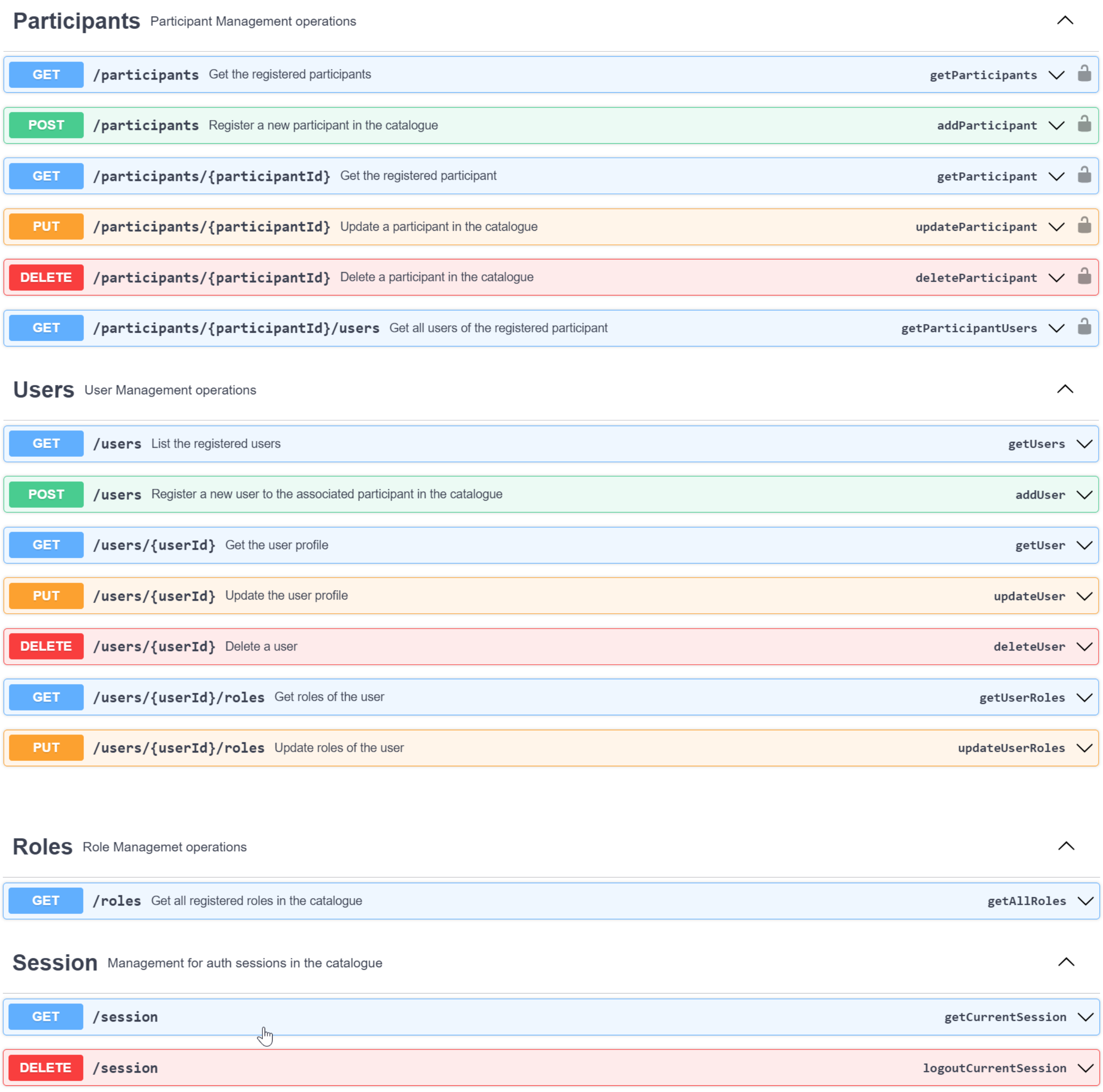
GET /schemas/latest Get the latest schema for a given type. If no term is specified, then the composite schema is returned. getLatestSchema

Verification

The Catalogue provides a verification service for e.g. checking the syntax

GET /verification Show a HTML page to verify (portions of) a signed Self-Description verifyPage

POST /verification Send a JSON-LD document to verify with the information from the Catalogue verify



Participants Participant Management operations

GET `/participants` Get the registered participants `getParticipants`

POST `/participants` Register a new participant in the catalogue `addParticipant`

GET `/participants/{participantId}` Get the registered participant `getParticipant`

PUT `/participants/{participantId}` Update a participant in the catalogue `updateParticipant`

DELETE `/participants/{participantId}` Delete a participant in the catalogue `deleteParticipant`

GET `/participants/{participantId}/users` Get all users of the registered participant `getParticipantUsers`

Users User Management operations

GET `/users` List the registered users `getUsers`

POST `/users` Register a new user to the associated participant in the catalogue `addUser`

GET `/users/{userId}` Get the user profile `getUser`

PUT `/users/{userId}` Update the user profile `updateUser`

DELETE `/users/{userId}` Delete a user `deleteUser`

GET `/users/{userId}/roles` Get roles of the user `getUserRoles`

PUT `/users/{userId}/roles` Update roles of the user `updateUserRoles`

Roles Role Managemet operations

GET `/roles` Get all registered roles in the catalogue `getAllRoles`

Session Management for auth sessions in the catalogue

GET `/session` `getCurrentSession`

DELETE `/session` `logoutCurrentSession`

6. Implementation Technology

Given the large diversity of modules and services that comprehend DIDI Data Marketplace a summary of the main technologies used in the implementation of the components depicted in the architecture diagram.

<u>**Data Space Horizontal Modules:**</u>

- *Notarization Service*: Node, *Docker*
- *Compliance Service*: Node, *Docker*
- *Registry Service*: Node, MongoDB, *Docker*

- *Federated Catalogue: Java 11, Spring Security and Spring Boot, Postgresql, Neo4j with neosemantics, Keycloak, Docker*

Data and Services Marketplace:

- *Data Contract Service:* Node, Redis, Docker
- *Data Exchange Logging Service:* Node, Docker
- *Portal*: React, Docker

Data Exchange Services:

- *Data Connector*:
  - Eclipse Data Connector (EDC)[3]: Java

7. Comments

The aim of this document is not to provide a comprehensive and fully detailed description of DIDI Data Marketplace and its components given the fact that its implementation leverages in many of the GAIA-X open-source components.

Please, refer to GAIA-X documentation for a complete and very detailed description of all concepts mentioned here as well as for the technical details of the software components:

- GAIA-X Architecture Document: https://docs.gaia-x.eu/technical-committee/architecture-document/22.10/
- GAIA-X Services (source repository): https://gitlab.com/gaia-x

[3] https://github.com/eclipse-edc/Connector