# GAIA-X IRL: WHAT DO I HAVE TO DO TO SEND A CAT PICTURE?

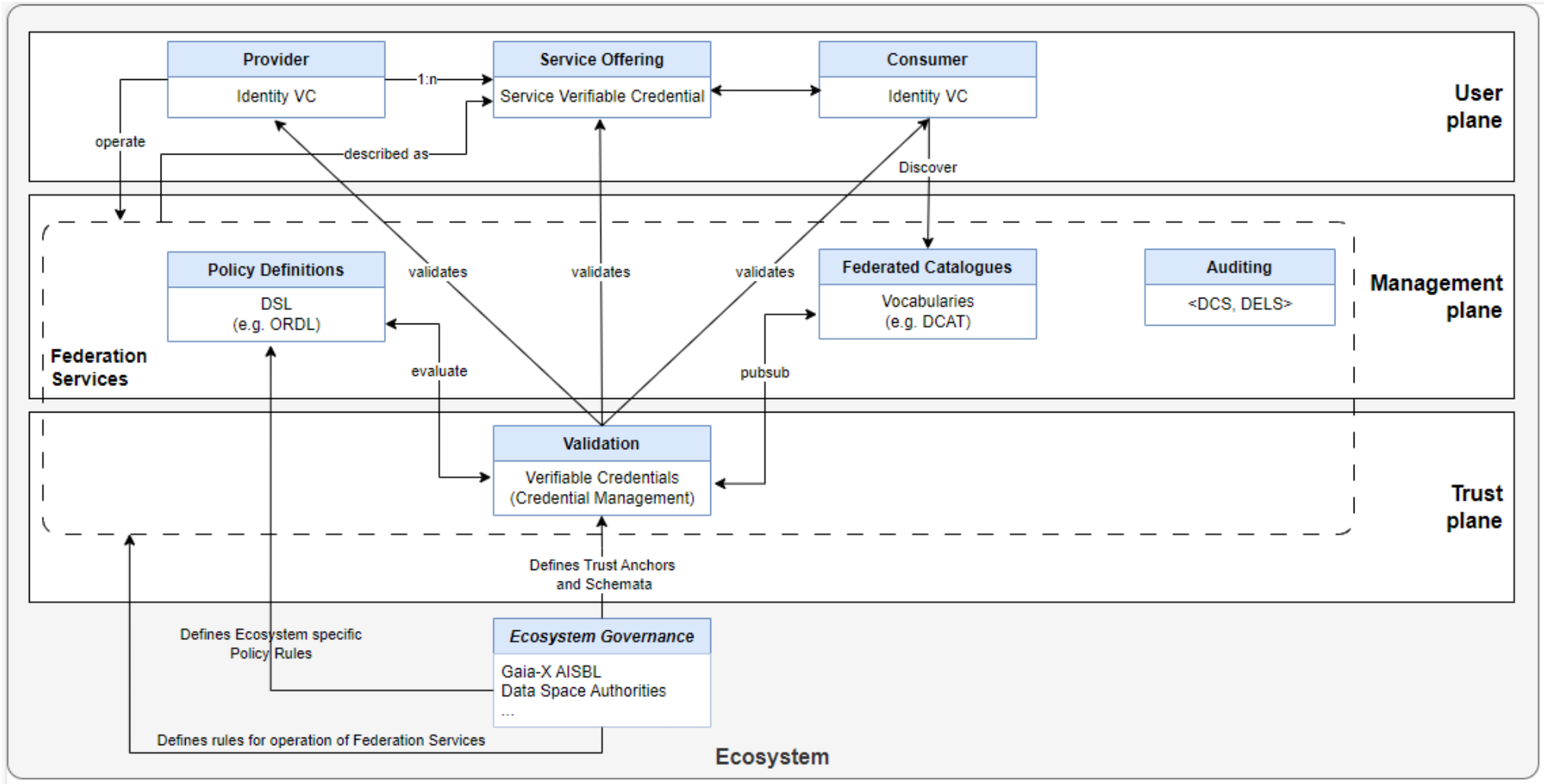
**Dataspace Connectors and Storage Bindings**

Tim Fritzsche, Institute of Data Science, 24 September 2024

# Goal

Establish data exchange which is Gaia-X conform as well as DLR conform

# Gaia-X Conceptual Model

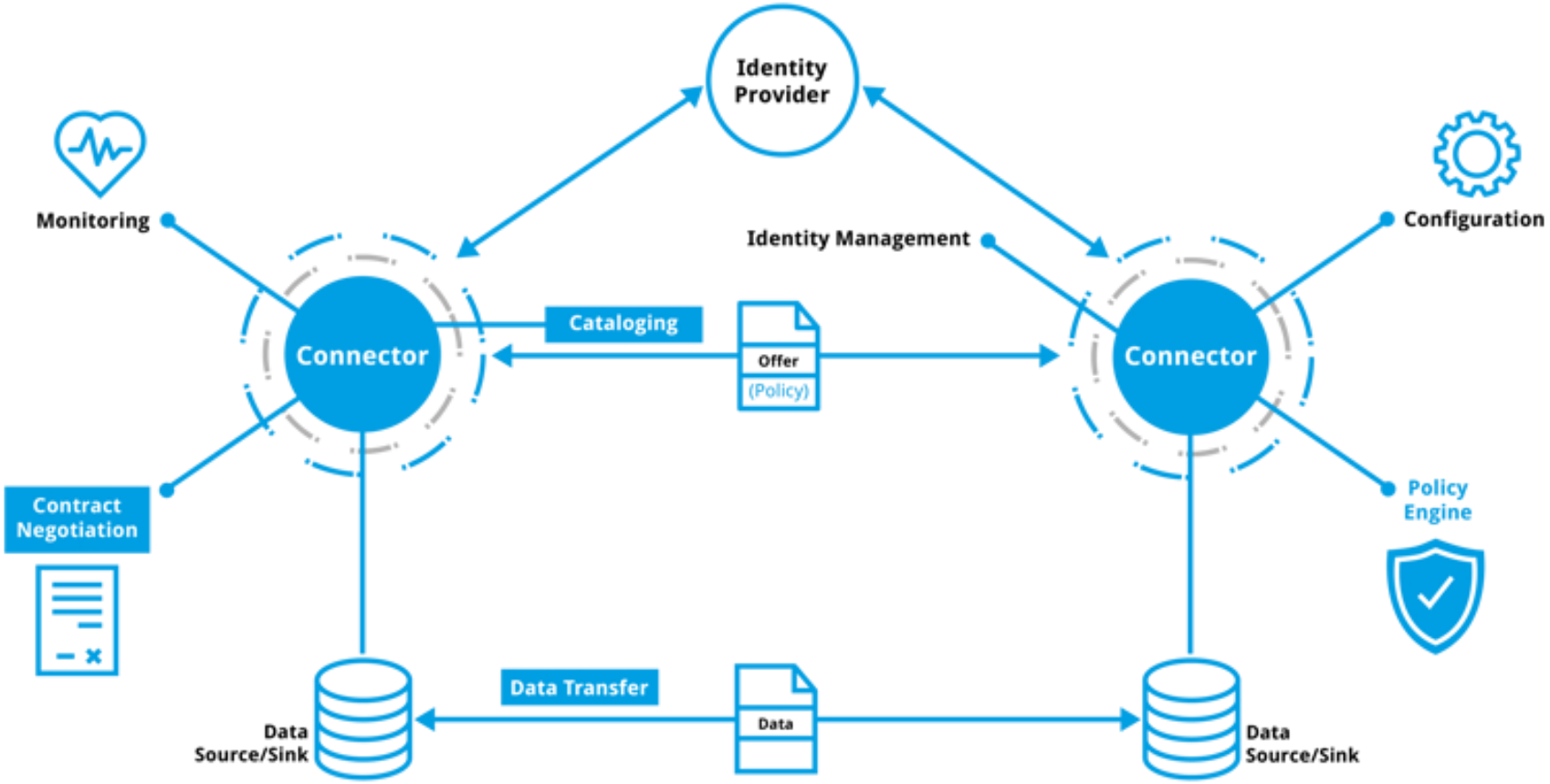



https://docs.gaia-x.eu/technical-committee/architecture-document/latest/gx_conceptual_model/

# International Data Spaces (IDS)

- Initiative with purpose to create secure and domain crossing dataspace (has been institutionalized to *International Data Spaces Association, IDSA*), Fraunhofer society is deeply involved
- Enterprises of different sectors and arbitrary size shall benefit
- Constructed reference architecture model for secure data exchange inside an ecosystem; try to establish reference architecture internationally (consistent with other existing models)
- Further development of IDS in several research projects, Gaia-X is one of them

# Dataspace Protocol (Overview)




https://docs.internationaldataspaces.org/ids-knowledgebase/v/dataspace-protocol

# Connector

- Software that provides technical access to IDS ecosystem
- Performs operations on behalf of a Dataspace member, e.g.:
  - Produce agreements
  - Manage dataset sharing
- Can take the role as *Provider* (offer a dataset) or as *Consumer* (request access to an offered dataset)

For our task to transfer data, this means:

- Gaia-X conform: Use an (open-source) connector that implements the IDS architecture
- DLR conform: Store data in an open-source storage

# Connector Implementations

- *Data Connector Report*[1] of IDSA lists 16 connectors;
  according to *A Survey of Dataspace Connector Implementations*[2] only 4 of them are open-source connectors
- No further development for at least 2 of the 4 open-source connectors; for a third open-source connector Gaia-X conformity is unclear
- Conclusion: Use Connector of Eclipse Dataspace Components (EDC)

[1]https://internationaldataspaces.org/wp-content/uploads/dlm_uploads/IDSA-Data-Connector-Report-November-2022.pdf
[2]https://arxiv.org/pdf/2309.11282.pdf

# EDC Connector

- Java project built with Gradle
- Very clean and modern code
- No major release (current version is 0.9.0)
- Until recently, bad state of documentation on multiple levels (set up project locally, REST endpoints, required artifacts, …; available documentation partially out-of-date; reverse engineering needed to get samples run); documentation is up to date and very detailed now
- Application expects REST requests to certain exposed ports
- Existing storage bindings: AWS S3, Azure Blob, Google Cloud (i.e. not DLR conform)

# MinIO Storage

- Open-source storage (license: GNU AGPL v3, or commercial license)
- S3 compatible
- Does not seem to be widely known (not even mentioned in German Wikipedia) – however, it is “the industry’s most widely deployed object store” (according to MinIO web page)
- Deployment uncomplicated and well documented
- Provides an API to access any S3 compatible object storage (the Java artifact is published under Apache 2.0 license)

# Extension of EDC Connector

- Create Java project with suitable Gradle build file (i.e. find out what dependencies are needed)
- Implement a *ServiceExtension* for MinIO storage binding as well as *DataSource* and *DataSink* for data transfer from/to MinIO storage (source code for AWS S3 binding provides good orientation)

# Transfer Data

- Store a cat picture in one of your MinIO buckets

https://www.freeimages.com/photo/lap-cat-1243719

# Transfer Data

- Run two instances of the extended EDC Connector
- Requests to provider:
  - Create asset
  - Create policy
  - Create contract
- Requests to consumer:
  - Fetch catalog
  - Negotiate contract
  - (Get contract agreement ID)
  - Start transfer process

# Summary – what do I have to do to send a cat picture?

- Install MinIO server if required
- Store a cat picture in a MinIO bucket
- Set up EDC Connector extension and implement MinIO storage binding
- Run extended EDC Connector instance(s)
- Transfer cat picture via extended EDC Connector

Some more details:
https://wiki.dlr.de/display/SoftwareEngineering/EDC+Connector+extension+with+MinIO+storage+binding