NextGen

Genome-Centric Multimodal
Data Integration in Personalised
Cardiovascular Medicine.

Deliverable D2.1
Platform design and architecture blueprint
Deliverable D2.2
Assessment/demonstration of MVP technology-1

Grant Agreement Number: 101136962

ESC European Society of Cardiology
hiro Edge Micro Data Centers
HL7 Europe
HUMAN COLOSSUS FOUNDATION
HUS
LiKe Healthcare Research
MyData Global
NEBS
WELLSPAN HEALTH
Earlham Institute Decoding Living Systems
EURECOM
GOETHE UNIVERSITÄT FRANKFURT AM MAIN
IDSIA SUPSI
Karolinska Institutet
MRI Klinikum rechts der Isar Technische Universität München
Queen Mary University of London
UMC Utrecht
UNIVERSITY of VIRGINIA
DATAPOWER CONSULTING
DIA
dpo ASSOCIATES

Funded by the European Union

UK Research and Innovation

The British associated partners of NextGen were funded by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant agreements No. 10098097, No. 10104323]

Project funded by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

NextGen	
Project full title	Next Generation Tools for Genome-Centric Multimodal Data Integration In Personalised Cardiovascular Medicine
Call identifier	HORIZON-HLTH-2023-TOOL-05-04
Type of action	RIA
Start date	01/ 01/ 2024
End date	31/12/2027
Grant agreement no	101136962

Funding of associated partners
The Swiss associated partners of the NextGen project were funded by the Swiss State Secretariat for Education, Research and Innovation (SERI Project No 23.00540). The British associated partners of NextGen were funded by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant agreements No 10098097, No 10104323]

D2.1 – Platform design and architecture blueprint	
Author(s)	Livia Carvalho, Catherine Chronaki, Vladimir Gladkii, Aaron Lee, Francesca Mangili, Evangelia-Anna Markatou, Rafy Mehany, Robert Mitwicki, Philippe Page, Luca Alessandro Remotti, Ali Shalbaf Zadeh, Geerte Slappendel, Christopher Wilson
Editor	Rafy Mehany, Vladimir Gladkii
Participating partners	HIRO, HL7, QMUL, HCF, UMCU, MYDTA, SUPSI, DataPower
Version	1.0
Status	Final
Deliverable date	M18
Dissemination Level	PU - Public
Official date	2025-06-31
Actual date	2025-06-27

Disclaimer

List of NextGen participants:

No	Partner organisation legal name	Short name	Country
1	UNIVERSITAIR MEDISCH CENTRUM UTRECHT	UMCU	NL
2	HIRO MICRODATACENTERS B.V.	HIRO	NL
3	EURECOM GIE	EURE	FR
4	JOHANN WOLFGANG GOETHE-UNIVERSITAET FRANKFURT AM MAIN	GUF	DE
5	KAROLINSKA INSTITUTET	KI	SE
6	HUS- YHTYMA	HUS	FI
7	UNIVERSITY OF VIRGINIA	UVA	US
8	KLINIKUM RECHTS DER ISAR DER TECHNISCHEN UNIVERSITAT MUNCHEN	TUM-Med	DE
9	HL7 EUROPE	HL7	BE
10	MYDATA GLOBAL RY	MYDTA	FI
11	DATAPOWER SRL	DPOW	IT
12	SOCIETE EUROPEENNE DE CARDIOLOGIE	ESC	FR
13	WELLSPAN HEALTH	WSPAN	US
14	LIKE HEALTHCARE RESEARCH GMBH	LIKE	DE
15	NEBS SRL	NEBS	BE
16	THE HUMAN COLOSSUS FOUNDATION	HCF	CH
17	SCUOLA UNIVERSITARIA PROFESSIONALE DELLA SVIZZERA ITALIANA	SUPSI	CH
18	DRUG INFORMATION ASSOCIATION	DIA	CH
19	DPO ASSOCIATES SARL	DPOA	CH
20	QUEEN MARY UNIVERSITY OF LONDON	QMUL	UK
21	EARLHAM INSTITUTE	ERLH	UK
22	ASSOCIAÇÃO DO INSTITUTO SUPERIOR TÉCNICO PARA A INVESTIGAÇÃO E DESENVOLVIMENTO	IST-ID	PT

Document Revision History

Date	Version	Description	Contribution
2024-12-31	0.5	Initiated document	HIRO
2025-06-27	1.0	Final document	HIRO

Authors/Editors

Author / Editor (alphabetical)	Organisation
Livia Carvalho	QMUL
Catherine Chronaki	HL7
Vladimir Gladkii	HIRO
Aaron Lee	QMUL
Francesca Mangili	SUPSI
Evangelia-Anna Markatou	HL7
Rafy Mehany	HIRO
Robert Mitwicki	HCF
Philippe Page	HCF
Luca Alessandro Remotti	DataPower
Ali Shalbaf Zadeh	HIRO
Geerte Slappendel	UMCU
Christopher Wilson	MYDTA

List abbreviations

Abbreviation	Description
AI	Artificial Intelligence
LLM	Large Language Models
DKMS	Decentralised Key Management System
EHDS	European Health Data Space
GDPR	General Data Protection Regulation
FL	Federated (machine) learning
GWAS	Genome-Wide Association Study
P2P	Peer-to-peer
ML	Machine Learning
MMIO	Multi-Modal Integration Object
OCA	Overlays Capture Architecture
API	Application Programming Interface
REST	Representational state transfer
UI	User Interface
WP	Work Package
DOA	Data Oriented Architecture
EDI	Equality, Diversity and Inclusion

Table of contents

List of Figures

List of Tables

1 Executive Summary

The NextGen Pathfinder represents a tangible implementation of core European Health Data Space (EHDS) specifications, serving as a federated, multi-site "mini-EHDS" network. It integrates advanced tools for genomic and multimodal data analysis while adhering to FAIR principles and cross-border governance frameworks. This document is a blueprint for the design of the Pathfinder platform architecture and core services implementation as part of MVP technology-1.

1.1 Key Components and Alignment with Objectives

1. The **Pathfinder** platform is grounded in a **Data-Oriented Architecture (DOA)** which prioritizes data as the main design element of the system. Rather than relying on tightly coupled workflows or centralized logic, the platform is structured around how data is accessed, transformed, and governed across distributed environments. This model allows for maximum flexibility, interoperability, and scalability in managing complex multimodal datasets across different institutions.

2. **Federated Infrastructure.** The Pathfinder enables secure, privacy-preserving analytics across five simulated geographically distributed biobanks without centralizing data using Data Oriented Architecture.

3. The **federated catalogue** introduces a secure flexible management of metadata that enables dataset discovery without requiring raw data to leave its source. Each participating site retains full control over its data, while exposing searchable metadata that adheres to shared ontologies and interoperability standards. This architecture aligns with the **FAIR data principles** and supports cross-institutional research without compromising data sovereignty.

4. **Multimodal Integration Objects (MMIOs)** integrate heterogeneous data formats with the aim to develop adaptor functionalities between standards and federated catalogs for enhanced data discoverability. The MMIO addresses gaps identified in the proposal's data management requirements.

5. Security and trust in this decentralized environment are ensured through the **Decentralised Key Management System (DKMS)** which enables distributed identity management and data provenance without relying on any single authority. It provides cryptographic guarantees for authentication for authorization, and traceability of all actions across the federated network.

6. Together, these components create an integrated proof-of-concept ecosystem that enables **federated analytics**, cross-border collaboration, and regulatory-compliant governance, while facilitating the development of **personalised medicine** based on multiomic data.

7. **Functionality demonstrated by pilots.** Pilots will demonstrate scalable genomic and Machine Learning workflows, aligning with Specific Objective 4 (SO4), using federated learning over distributed infrastructures (section 1.1.2.3) and federated catalogues (WP2) to remove technical barriers.

1.2 Expected Outcomes

- **Research Portability:** Federated computation and MMIOs enable cross-border analysis while preserving data sovereignty.
- **EHDS Synergy:** The NextGen specifically seeks to ensure project deliverables are synergistic with the EHDS (and other key initiatives) and the NextGen Pathfinder will demonstrate core EHDS functional specifications .
- **KPI Achievement:** By M42, the Pathfinder will comprise five simulated sites, demonstrate six pilots, and host one public demonstration, fulfilling proposal targets (KPIs), and organise events including workshops, webinars and hackathons.

SO 4: Integrate best practices through Pathfinder and pilots. Demonstrate advanced integration and workflow tools in piloted use cases showing removal of technical and operational barriers. Pilot integrated into the "NextGen Pathfinder": a multi-site “mini-EHDS” network showcasing NextGen innovations in data management, data governance, cataloguing, compute, advanced data integration, genomic and interoperability capacities. The Pathfinder will integrate best practices from evolving EU-wide initiatives such as the EHDS and 1+MG
Deliverables: ● Pilot implementations of project tools extending scope and quality of research outcomes ● Pathfinder network developed with five demonstration biobank sites demonstrating project tools
Measurable and verifiable (KPIs)
6 pilot demonstrations
5 sites included in Pathfinder
At least 1 successful public Pathfinder demonstration
Regulatory, governance and data tooling demonstrated for **7** countries (SE,UK,CH,FI,USA,DE,NL)

By using advanced technologies designed with information security in mind and aligning with major European health programs (EHDS, 1+MG, GA4GH), NextGen is building a **scalable, privacy-first ecosystem** to support the future of healthcare research.

2 Introduction

2.1 About NextGen

The **NextGen Pathfinder** represents core technical functionalities developed to support secure, scalable, and interoperable data sharing and analysis in personalised cardiovascular medicine. The technology readiness level target for the Pathfinder pilots is TRL 4/5 depending on the specific tool demonstrated. Acting as a "mini-EHDS" (European Health Data Space), the Pathfinder demonstrates how health data can be used for research and innovation across institutions and borders, while preserving privacy, governance autonomy, and legal compliance.

The overall **objectives of NextGen** are to:

- Develop **tools for personalised medicine** that enable prediction, prevention, diagnosis, and treatment of cardiovascular diseases using multiomic and multimodal data.
- Create a **scalable data analytics platform** for federated machine learning and cross-site genomic computation.
- **Overcome barriers** to integrating health data by building advanced data catalogues and workflow tools.
- Build trust by involving humans and addressing **key ethical and legal issues** like privacy, fairness, and accountability through clear guidelines..
- Build a sustainable, **privacy-first ecosystem** that supports ongoing innovation beyond the project's timeline.

The Pathfinder serves as the core technical foundation through which these objectives are demonstrated through a number of pilots.

The pilots mentioned in this document were introduced in Deliverable 5.2 and represent demonstrations of core functionalities of the Pathfinder.

2.2 Deliverables Structure

This document covers two deliverables—D2.1 and D2.2—and outlines the architectural blueprint of the Pathfinder platform, detailing how its design principles, technical components, and federated services support the broader objectives of the NextGen project. In section 3, the Pathfinder in its broader context is described, which includes: functional and non-functional requirements; relationship to initiatives such as the EHDS; incorporation of regulatory aspects; equality, diversity and inclusion; and the approach to risk management.

In section 4, the structural and architectural specifications are detailed. In Table 1, the relationship between data management issues (as specified in the proposal) and in this blueprint document is given.

Table 1: Mapping issues addressed to sections

Data management issues to address	Proposal Section(s)	Blueprint Section
Semantic ontologies	1.2.8.1 i.	Metadata Catalogue, Search Engine
Data standards and formats	1.2.8.1 ii.	Data Oriented Architecture, MMIO
Data quality	1.2.8.1 ii + 1.2.8.3	DAO, Data product, Quality Control
Data storage	Federation (1.2.6)	Data Oriented Architecture, Federated ML process ,Search for Data Products and Applications
Other data management	1.2.8.3	DKMS - MMIO
Enhanced findability through improved metadata standards/catalogues	1.2.8.1 iii	Search Engine, Metadata Catalogue, DKMS - MMIO
New techniques, support tools, mechanisms and modalities to enable	**Section(s)**	
GDPR compliant access to sensitive personal data + genomics	1.2.8.1.v	Applicable laws and Regulations, Governance, Gateway
Data re-use across borders	1.2.8.1.v	Node Architecture, EHDS
Integration of different data types	1.2.8	Multimodal Integration Object(MMIO)
Legal/ethical frameworks considering national/sectorial heterogeneity for access/re-use	1.2.8.1.v	EHDS, Applicable laws and regulations
Data management approaches for cross-border distributed data storage and processing	**Section(s)**	
Enable remote collaboration	Pathfinder (1.2.7.2)	Multimodal Integration Object(MMIO) Federated ML process Governance Monitoring and logging Interoperability
Electronic consent management	1.2.8.1 iv	Governance Multimodal Integration Object(MMIO)
Data provenance tracking	1.2.8.1	Governance MMIO
Scalability of data management resources	1.2.8.3	Data Oriented Architecture and Data Space Services
Ensure data privacy and security	1.2.8.1 iv	Multimodal Integration Object(MMIO)

		Governance Clearing house
Demonstrate robust support to advanced, innovative clinical workflows	1.2.2.1+2+3	Real-world Applications
Joint data governance piloted among several clinical centres across Europe	Pathfinder (1.2.7.2)	MMIO (data transformation) Decentralized catalogues (federated catalog) Governance layer Monitoring and logging Interoperability and ontologies
Data analytics platform to query and aggregate data from multiple sources securely	**Section(s)**	
Apply distributed learning	1.2.8.2+3	DOA, Federated ML process
Apply AI tools	1.2.8.2+3	DOA, Federated ML process
Monitor patient health status	1.2.8.2+3	DOA, Monitoring and logging
Analyse causal inference	1.2.8.1 iv	Governance layer, Clearing house, MMIO
Support health policymakers	1.2.8.1 v	EHDS, Applicable laws and regulations
Support diagnosis	1.2.2.1+2+3	Real-world Applications
Establish stakeholder recommendations	1.2.8.1 iii	Federated ML process, Decentralised Key Management System (DKMS), Recommendation system

This document also includes the services that were implemented by HIRO as a proof of concept as part of MVP technology-1 for deliverable 2.2.

3 Pathfinder Platform

3.1 Description

The NextGen Pathfinder establishes a federated network of five simulated geographically distributed sites, each contributing distinct yet semantically interrelated datasets under site-specific governance regulations.

The system supports:

- Secure **federated analytics**, including genomic computation
- **Scalable federated learning** across distributed infrastructures
- Advanced tools for genomic data analysis
- **Efficient variant prioritisation and genomic data curation**
- Improved **data discoverability** and management

Six pilots will demonstrate the core functionality of the Pathfinder.

The Pathfinder will demonstrate **FAIR-compliant** (findable, accessible, interoperable, and reusable) data sharing and governance, featuring:

- Joint governance models
- Decentralised authentication
- Semantic interoperability

Table 2: Pathfinder Architecture

Pathfinder component	Architectural components
User interface	- User Interface (UI)
Federated analysis	- Training builder - Escrow locker - ML Runner - Global Application Repository - Connector service - Governance - Contract Engine - Clearing House - Interfaces - Global OCA repository
Federated catalogue	- Gateway - Search Engine - Catalog Service - Knowledge graph

	- Global OCA repository - Node registry
Cross-Border Data Access and Interoperability	- Connector service - Governance - Contract Engine - Interfaces
AI/ML Model Management	- Training builder - ML Runner
Variant Prioritization	- Training builder - ML Runner
Optimised Genomic Processing	- Training builder - ML Runner
Genomic Data Curation and Analysis	- Training builder - ML Runner

3.2 Requirements

3.2.1 Functional Requirements

The Pathfinder must comply with relevant local and international regulations, ensuring privacy protection, security, and user-friendliness. The Pathfinder aims to create **a cohesive and secure environment for managing health research data**. This approach supports various stakeholders, including researchers, clinicians, and patients, while fostering innovation in healthcare research.

The functional requirements for the Pathfinder Platform have been identified and are:

- **Federated Catalogue**
  Enables the creation and management of distributed data catalogues, allowing users to locate and access datasets across a federated system and organisation to control the rules applying to the exposed data under their control.
- **Multimodal Data Integration**
  Supports the integration and processing of diverse data types (e.g., clinical, imaging, genomic) into unified analytical frameworks.
- **Federated Machine Learning**
  Enables machine learning across distributed datasets while maintaining privacy and regulatory compliance by avoiding centralized data pooling.
- **Federated Genomic Analysis**
  Facilitates the distributed analysis of genomic data while ensuring data remains securely stored at its original location.
- **Optimized Genomic Processing**
  Enhances the speed and scalability of genomic data analysis to deliver timely insights.

- **Variant Prioritization**
  Identifies and ranks genetic variants based on their clinical or research significance.
- **Genomic Data Curation and Analysis**
  Provides tools for annotating, curating, and analyzing genomic data to generate actionable insights.
- **AI/ML Model Management**
  Supports the entire lifecycle of AI and machine learning models, including development, deployment, and monitoring.
- **Data-Oriented Architecture**
  Implements a structured, data-centric approach to enhance scalability, interoperability, and usability.
- **Marketplace**
  Establishes a secure platform where users can discover, exchange, and utilize datasets, algorithms, and analytical tools.
- **Security Measures**
  Ensures strong security protocols, including encryption, access controls, and compliance with relevant regulations.
- **Governance Framework**
  Implements mechanisms to uphold ethical, legal, and policy standards across federated data ecosystems.
- **Cross-Border Data Access and Interoperability**
  Supports secure and compliant sharing of health data across regions for research, diagnosis, and policy development.

3.2.2 Non-Functional Requirements

Non-functional requirements define the quality attributes of a system, ensuring it operates securely, efficiently, and reliably.
For the Pathfinder, these requirements focus on **scalability, interoperability, privacy, and maintainability** while adhering to regulatory and ethical standards.

Scalability

- Handles **growing data and computation needs**; supports multi-site expansion

Interoperability

- Across NextGen participants operating in different systems
- Semantic agility. Supports widely used data formats and standards for both tabular (e.g., OMOP, FHIR, CDISC) and non-tabular data modalities to facilitate integration using MMIOs.
- Open design for connectivity, to the possible extent to other EU initiatives

Security & Privacy

- Complies with GDPR; uses decentralized access control and privacy-preserving methods (e.g., federated learning).

Reliability & Availability

- Support stable operations and minimize disruptions and where possible consider mechanisms for data integrity and fault tolerance.

Auditability & Transparency

- Privacy preserving mechanisms for traceability of data usage. The system must strike the right balance between transparency (logs) and confidentiality of transactions (anonymisation).
- Accountability of relevant actions for proof of lawful data usage

Maintainability & Sustainability

- Modular design for adaptability; where is it possible long-term sustainability planning.

Ethical & Legal Compliance

- Follows legal, ethical, and regulatory standards.

Cross-Border Data Access

- Secure and cross border data sharing with user-friendly tools for researchers.
- Fulfilling the EHDS regulatory requirements (as they will develop during the course of the project)
- Including assessment of connections with other EU initiatives or genomics (e.g. B1MG, EOSC.)

For more details about non-functional requirements the Reader can refer to Deliverable 5.2.

3.3 Equity, Diversity and Inclusion

This section considers how relevant aspects of **Equity, Diversity and Inclusion** (EDI) apply to the NextGen Pathfinder Platform.

3.3.1 Accessibility of Content / User Accessibility

As of 2019, the European Accessibility Act (EEA) has set the standards for digital accessibility within the European market. Services covered by the act, EN 301 549, which references the WCAG (Web Content Accessibility Guidelines) 2.1 AA criteria, define the minimum requirements for accessibility of services, going into effect as of 18 June, 2025. This aligns with the standard set by the Americans with Disabilities Act (ADA) in the USA.

As of December 2024, the W3C has updated the WCAG by releasing version 2.2, which includes improvements for users with cognitive or learning disabilities, users with low vision, and users with disabilities on mobile devices, and is fully backwards compatible with version 2.1. The

WCAG guidelines require content to be 1. Perceivable, 2. Operable 3. Understandable and 4. Robust against misinterpretation, with specific criteria for A, AA and AAA compliance levels. These criteria refer to practical issues such as color usage, ease of navigation and availability of text description of images, in order to ensure content can be accessed effectively by users with a wide range of disabilities. Full criteria can be found here[1]

While the Pathfinder Platform does not fall within the scope of the EEA, we recognize the best practice standard set by both the European Commission and the US government, and will aim for the Pathfinder to meet requirements for WCAG 2.2 AA conformity, with a minimum of A conformity where AA conformity is not feasible.

3.3.2 Guidance to Platform Users

Creating fair and equitable algorithms is complex, and requires active intervention at each stage of the process, from data collection to algorithm refinement. **Machine learning and artificial intelligence have brought new issues to the table when it comes to EDI**, and case studies showing how algorithms reinforce existing inequalities instead of correcting them are common. However, answers on how to prevent these issues are not yet easily available. Setting up a central system for federated machine learning provides an opportunity to nudge and guide the users towards more actively considering how they address these issues to support a structural shift in the field.
In order to support implementation of best practices, the Pathfinder platform will include guidance on best practices for building fair and equitable algorithms where available - these might include new texts, but also easily accessible references to existing tools.
Additionally, EDI advisers will be involved in the development of the Pathfinder, to help review which issues are likely to come up, and how we might be able to address them effectively in the way we set up the system. For example, consideration will be given to how to set up the recommendation module so it does not recommend common practices that are known to lead to EDI issues downstream in the algorithm development and implementation.
we will also encourage the deposit of diverse datasets, and develop tools that support ethical diversity and inclusion in the data analysis to prevent further widespread of AI biases that reinforce inequalities

[1] https://www.w3.org/TR/WCAG22/

3.4 Applicable Laws and Regulations

3.4.1 Overview and Context

As the European Union advances its **digital transformation initiatives**, numerous regulatory frameworks have been established to ensure that technology development aligns with core values like data protection, transparency, and ethical standards.

This regulatory landscape in the EU includes frameworks and policies designed to:

1. Safeguard personal data and privacy.
2. Ensure ethical AI practices.
3. Facilitate secure cross-border data sharing.
4. Promote fairness, transparency, and trust in digital health tools.

The core implications for the Pathfinder Platform will be to ensure:

- **Data Protection and Privacy**: Ensuring that all health data is managed securely and only accessible by authorized parties. Clear policies on data processing, consent, anonymization, and user rights are mandatory. This is achieved by both the Data Oriented underlying design of the infrastructure and the application of governance measures.
- **Transparency and Explainability**: Deployed AI models must be explainable, especially in clinical contexts where understanding predictions is critical. Transparent data governance processes that communicate how data is used, processed, and shared are essential for trust. This is achieved in NextGen, where applicable, through using established processes for the development of Trustworthy AI/ML.
- **Interoperability and Accessibility**: Systems within the Pathfinder must be interoperable with other EU health systems, following EU standards for data formats and accessibility to facilitate cross-border data sharing. This is implemented in Pathfinder through the guidance provided by the EU Health Data Space regulations and other requirements that surface in the WP6 regulatory review.
- **Ethics and Fairness**: Adhering to ethical AI principles, ensuring that the project's tools are free from unwanted biases, and implementing human oversight in critical decision-making areas.
- **Compliance Monitoring**: On a fully deployed Pathfinder (beyond TRL 4/5 expected project deliverable) regular audits and checks should be in place to ensure compliance with these regulations, especially as they evolve (this would be achieved by the risk management approach in place). The Pathfinder will be subject to a risk management methodology (see section "Risk Management Approach").

3.4.2 Key Implementational Aspects

As the EU drives forward its digital transformation, it has introduced various regulations to ensure technological progress respects fundamental values such as data privacy, openness, and ethics

Table 3: Key Implementational Aspects

Digital Strategies and Policies	Key Points	Pathfinder Implementation
EU Digital Strategy	Empowering of users. Innovation in health services. Infrastructure readiness.	Pathfinder will incorporate relevant EHDS functionality.
2030 Digital Decade	Contribute to the EU's Health Digital Transformation. Enable cross-border health data sharing. Oversight of ethics and skills.	
EU Strategy for Data	Creating a health dataspace. Data interoperability and security. Trust and data control.	
EU Declaration on Digital Rights and Principles	Upholding of digital rights and principles.	

Data Governance and Protection	Key Points	Pathfinder Implementation
Data Governance Act	Role as a data intermediary. Data access and reuse. Compliance with public and private data use.	Pathfinder demonstrates the impact of the tools developed (i.e. MMIO, privacy preserving ML) to facilitate compliance of data processing with the Data Governance Act and the General Data Protection Regulations.
General Data Protection Regulations	Lawful basis and consent for data processing. Data minimisation and purpose limitation. Security and anonymisation. Rights of data subjects. Cross-border data transfers.	
ePrivacy Regulation	Secure communications. User consent for tracking and profiling. Compliance with electronic health communication standards.	D2.1 develops a modular architecture isolating the nodes of the network from the actual data sources from the participants. This will enable us to identify the required communication standards required by a particular participant for a given Pathfinder functionality
Interoperable Europe Act and European Interoperability Framework	Technical and semantic standards. Cross-border data sharing. Interoperable infrastructure.	Considered through the lens of the EHDS regulation

Data Governance and Protection	Key Points	Pathfinder Implementation
Database Directive	Protection of proprietary databases. Data licensing agreements.	
Directive on Open Data and the Reuse of Public Sector Information	Reuse of public sector health data. Ethics in public data use.	The Pathfinder platform introduces functionality for FAIR data principles to be applied by users.

Ethical and Emerging Technologies	Key Points	Pathfinder Implications
Artificial Intelligence Act	Risk management. Transparency and explainability. Human oversight. Bias and fairness.	Risk management in NextGen will be based on NIST frameworks (see "Risk Management Approach"). Developers of ML models are expected to follow to the extent possible best practise in Trustworthy AI/ML and to document finalised models appropriately (e.g. using model cards).
Digital Fairness Act	Non-discrimination in AI models. Transparency in digital tools. Ethical use of health data.	
Ethics Guidelines for Trustworthy AI	Ethical design and deployment of AI models in healthcare.	
Web Machine Learning Ethics Guidelines	Ethical use of machine learning models in genomic data analysis.	

Digital Markets and Service	Key Points	Pathfinder Implications
Digital Services Act	Transparent content management. User safety and incident management. Accountability for digital content.	The Pathfinder platform is designed with a decentralized architecture that supports federated cataloging, node-level governance, transparent usage logging and interoperability using MMIO. While the Digital Services Act, Digital Markets Act, and Copyright Directive primarily target large-scale digital platforms, their principles—such as fair access, content accountability, and rights protection—are considered relevant for the future evolution of the platform. At this stage (TRL 4/5), the platform provides a technical foundation that can support alignment with these frameworks, and relevant regulatory developments will continue to be monitored by the project partners as the system matures.
Digital Markets Act	Ensuring access and fair use. Avoiding anti-competitive practices.	
Directive on Copyright in the Digital Single Market	Respecting third party content rights, Avoiding infringements.	

3.5 European Health Data Space (EHDS) and Other Initiatives

3.5.1 Introduction

The integration of innovative tools for multimodal data poses significant challenges, particularly when applied to real-world solutions that necessitate the collaboration of multiple stakeholders across diverse entities. The NextGen project is designed to function within an ecosystem of research organizations focused on personalized cardiovascular medicine, with operations spanning multiple jurisdictions.

Consequently, NextGen offers a **platform for researching technologies intended for implementation within a complex health data environment**. At the time of its submission, it was characterized as a "mini-EHDS" to emphasize the necessity for a data integration approach that is legally compliant across various jurisdictions and sensitive to differing ethical considerations.

The European Health Data Space (EHDS) Regulation, initially proposed in May 2022, underwent significant revisions before its formal adoption in March 2025. An initial assessment of proposed EHDS regulatory elements which were relevant/implementable in the NextGen Pathfinder was mapped to Deliverable 5.2 and to the Minimum Viable Products (see Deliverables 2.2 and 2.3). Following the submission of NextGen, the EHDS regulation has been approved by the EU Parliament on March 5, 2024, and is currently transitioning into its implementation phase. Therefore, the innovative tools developed by NextGen must be forward looking and aligned with the upcoming EHDS implementation. The EHDS introduces a governance system at both national and EU levels. Additionally, cross-border digital infrastructures will be established to facilitate data sharing for both primary and secondary uses of health data. With these upcoming EHDS requirements, NextGen integrates its output into the Pathfinder Platform. This enables NextGen tooling to be integrated into an experimental infrastructure accommodating EHDS requirements of integration with Member State governance and obligations. Many elements of the EHDS are still being defined for implementation, and NextGen is focused on demonstrating how its developed tools align with the requirements of the EHDS regulation's articles.

This section outlines the relevant EHDS articles and highlights how the Data Oriented Architecture (DAO) architecture will address them. In the following sections key areas of the EHDS are outlined where the Pathfinder will strive to demonstrate compliance in the sense of acting as a “mini-EHDS”.

3.5.2 EHDS regulation & Pathfinder design elements

The NextGen project develops tooling for better data integration (including genomics) with the ultimate goal of improving clinical outcomes. As a result, NextGen considers the processing of healthcare data which must occur within secure environments both from regulatory, ethical governance and technology perspectives, which are not comprehensively defined. Pathfinder,

provides experimental insights on how NextGen technologies can help design such secure environments.

The table below summarises NextGen Architecture Elements in regards to the relevant EHDS Regulation article. This is not a comprehensive assessment of NextGen activities versus the EHDS regulation. The table focuses on the items relevant from a technology architecture perspective.

Table 4: NextGen Architecture Design Elements & EHDS

EHDS Topic / Article	NextGen Design
Chapter 1 -General Provision	
Definitions *-Article 2*	NextGen uses EHDS definitions within the project. These includes GDPR definitions as well as the ones of related EU texts **Contributing NextGen Architecture Elements:** - WP1 T1.1 "Data Management Framework" - WP6 T6.2 "Legal Framework
Interoperability *-Article 2 Par.2 Let.(h)*	The EHDS defines '*interoperability*' as the ability of organisations, as well as of software applications or devices from the same manufacturer or different manufacturers, to interact through the processes they support, involving the exchange of information and knowledge, without changing the content of the data, between those organisations, software applications or devices. This includes the ability to transfer and receive personal electronic health data in a standardized, machine-readable format. NextGen's architecture is designed to provide an explicit solution via its decentralised and federated approach to data management and data processing. **Contributing NextGen Architecture Elements:** - Data Oriented Architecture to preserve data integrity throughout the data life-cycle - DKMS & MMIO to provide verifiable data accuracy (authenticity and integrity) across the NextGen ecosystem (e.g. Pathfinder pilots)
Chapter 2 -Primary use of electronic health data	
Section 1-*Articles 3 to 18* Rights of natural persons in relation to the primary use of their personal electronic health data, and related provisions	By design, NextGen deals with the secondary usage of data. The tooling for multimodal data integration developed by the participants supports a research ecosystem. The lawful use of health data rests with the organisation processing that data. NextGen governance federates independent legal entities, each one remaining accountable within their own jurisdiction.

EHDS Topic / Article	NextGen Design
	Contributing NextGen Architecture Elements: - Decentralised Data Oriented Architecture ("connected nodes" not platforms) - Federated computations - Open Source (access to critical algorithms) - Independent REG Board
European electronic health record exchange format. Article 15	By 26 March 2027, the Commission will have laid down the technical specifications for the priority categories of personal electronic health data referred to in Article 14(1), setting out the European electronic health record exchange format. Such format shall be commonly used, machine-readable and allow transmission of personal electronic health data between different software applications, devices and healthcare providers. Such format shall support transmission of structured and unstructured health data and shall include the following elements: - harmonised data, - code tables, - interoperability specification. NextGen research and experiment new data management mechanisms for the discovery and potential exchange of data. **Contributing NextGen Architecture Elements:** - Multi-format, multi-standards data management framework for multimodal data integration. - DAO, DKMS, MMIO
Section 2 Governance for primary use	Not applicable. Remains with the participating organisations **Contributing NextGen Architecture Elements:** - Regulatory, Ethics & Governance measures from participating organisations remain controlled in their respective nodes.
Chapter 3 -EHR Systems and wellness applications	
Out of Scope	
Chapter 4 -Secondary Use	
Section 1 General Conditions with regards to secondary use	EHDS defines general conditions for the secondary use of data. These precise the applicability or not of EHDS to specific data holders, the categories of data covered, intellectual properties rights, purpose of use and prohibited use.
Minimum categories of electronic health data for secondary use *-Article 52*	The NextGen project is scoped to a research community in personalized cardiovascular medicine with the objective to extend/scale the innovations and experience acquired outside the project scope. **Contributing NextGen Architecture Elements:** - Architecture for multimodal data integration - see Table 5 below: "EHDS minimum categories of electronic health data for secondary use"

EHDS Topic / Article	NextGen Design
Section 2 Governance and mechanisms for secondary use	NextGen Decentralised Data Oriented Architecture is designed to allow Member States and EU governance mechanisms to be implemented. **Contributing NextGen Architecture Elements:** - Decentralised Data Oriented Architecture ("connected nodes" not platforms) - NextGen REG for common governance mechanisms - Framework for social and economic engagement
Section 3 Access to electronic health data secondary use.	The EHDS establishes a framework that allows entities like researchers and policymakers to access health data for secondary purposes, including research and innovation. Applicants must justify the necessity for data access and demonstrate compliance with data protection regulations. Need to ensure accommodation of reversible opt-out mechanisms for individuals to be defined and implemented by Member States. **Contributing NextGen Architecture Elements:** - Decentralised Data Oriented Architecture ("connected nodes" not platforms) - Advanced decentralised authentication system (DKMS) for the cryptographic authentication for access across the ecosystem - MMIO privacy preserving container enabling dynamic data minimisation.
Secure Processing Environment - Article 73	Secondary data processing MUST occur within secure environments, which are not comprehensively defined. The new regulation also: - prohibits data downloads, - enforces strict anonymization or pseudonymization standards; - forbids explicitly re-identification of individuals, - requires to maintain detailed logs of data access, capturing information such as the identity of healthcare providers accessing the data, categories of data accessed, and timestamps of access. The NextGen architecture design enables transparency and accountability in data handling to align with secure processing environments as articulated in the new regulation. Member States may impose stricter measures for sensitive data categories, such as genetic information. **Contributing NextGen Architecture Elements:** - Independent node architecture separating semantics and governance from data records stored and accessible only through dedicated and locally controlled interfaces. - implemented in the pathfinder using "Clearing house" and "Monitoring & Logging"; - Federated architecture. Data and processing environment remains under the control of participating institution.
Section 4	**Contributing NextGen Architecture Elements:**

EHDS Topic / Article	NextGen Design
Cross-Border Infrastructure for Secondary use	- Decentralised Data Oriented Architecture ("connected nodes" not platforms) - WP6 NextGen REG for common governance mechanisms - WP7 Framework for social and economic engagement
Section 5 Health data quality and utility for secondary use	EHDS requires Health Data Access Bodies to maintain repositories (i.e. catalogues) making information about dataset availability and related information (i.e. meta-data) publicly available. This will require the EU to define by 2027 machine readable standards, vocabularies and ontologies enabling the Union to define data quality and utility label attached to the dataset and available.
Dataset description and dataset catalogue -Article 77	NextGen explores the concept of decentralised federated catalogue where the information exposed remains under the control of the participants while obeying commonly agreed governance mechanisms. Therefore, the data discovery mechanism explored provides a scalable solution for metadata management in EHDS. The project follows the developments of HealthDCAT in order to ensure FAIR data principles are met for datasets shared within the EHDS ecosystem. **Contributing NextGen Architecture Elements:** - Decentralised Federated Catalogue - DCAT-AP, DCAT v3 interoperability, HealthDCAT-AP integration - Federated Machine Learning framework
Data Quality and Utility Label -Article 78	NextGen introduces quality and utility labels for datasets, as part of its enhanced metadata management. The aim is to enable users, mainly researchers, to assess their relevance, accuracy for their research purposes and in compliance with EHDS standards requirements. **Contributing NextGen Architecture Elements:** - Enhanced metadata management through advanced semantic architecture (semantic repositories) - Multimodal Integration Objects for a cryptographically verifiable identification of dataset and content
Certification & Compliance	Manufacturers must ensure their products conform to essential requirements outlined in the EHDS, including interoperability, security, and logging capabilities. They are also obligated to provide technical documentation, accompany their systems with information sheets and user instructions, and affix a CE marking to indicate compliance.

Table 5: EHDS minimum categories of electronic health data for secondary use

	Minimum Data Categories *EHDS Article 52*
(a)	electronic health data from EHRs;
(b)	data on factors impacting on health, including socioeconomic, environmental and behavioural determinants of health;
(c)	aggregated data on healthcare needs, resources allocated to healthcare, the provision of and access to healthcare, healthcare expenditure and financing;
(d)	data on pathogens that impact human physical and mental health;

(e)	healthcare-related administrative data, including on dispensations, reimbursement claims and reimbursements;
(f)	human genetic, epigenomic and genomic data;
(g)	other human molecular data such as proteomic, transcriptomic, metabolomic, lipidomic and other omic data;
(h)	personal electronic health data automatically generated through medical devices;
(i)	data from wellness applications;
(j)	data on professional status, and on the specialisation and institution of health professionals involved in the treatment of a natural person;
(k)	data from population-based health data registries such as public health registries;
(l)	data from medical registries and mortality registries;
(m)	data from clinical trials, clinical studies, clinical investigations and performance studies subject to Regulation (EU) No 536/2014, Regulation (EU) 2024/1938 of the European Parliament and of the Council (35), Regulation (EU) 2017/745 and Regulation (EU) 2017/746;
(n)	other health data from medical devices;
(o)	data from registries for medicinal products and medical devices;
(p)	data from research cohorts, questionnaires and surveys related to health, after the first publication of the related results;
(q)	health data from biobanks and associated databases.

3.5.3 Connection with Other Initiatives (B1MG, EOSC)

NextGen introduces new tooling for the integration of multimodal data into analytical processes with clinical impact. Its federated architecture is designed to be open and dynamic while preserving a secure processing environment. As a result, the NextGen project has stated objectives to connect to other EU initiatives like the European Open SCience (EOSC), Beyond 1 Million Genome (B1MG).

Table 6: Connection with Other Initiatives

NextGen Project Objectives	NextGen Architecture Concept	NextGen Other contributing work
O1.1 Additive and Synergetic data management tools compatible with EHDS, EOSC.	Data Space	WP1 T1.1 T1.5 WP6 T6.4
O7.2 Ensure effective engagement with EU-wide initiatives and projects	Open Source, Open architecture	WP7 T7.1, T7.3,

The objectives listed above contribute to the project impact:

- Improved citizen trust through advanced technology, data privacy and security (GA Sec.2.1.1.7)
- Contributing to EU-wide research and innovation (GA Sec.2.1.2.1)
- International visibility and leadership (GA Sec.2.1.2.3)
- Communication with and outreach to target groups (GA Sec.2.2.2.1 par.3)
- Scientific, Social Science and Humanities Advisory Board (GA Sec.3.2.1.2)

3.6 Pathfinder platform Risk Management Approach

3.6.1 Introduction

A **Risk Management Framework** (RMF) is a systematic method for identifying, assessing, responding to, and monitoring risks which is applicable to both simple and complex systems.

This document **describes an architecture** supporting a diverse research ecosystem that federates participants, tools, and data sources of different modalities for the purpose of advancing personalized cardiovascular medicine. These tools require a specific risk management framework, applied to a distributed system of independent contributing participants. For example, in such an environment, risks introduced by NextGen's tools may be isolated to specific nodes or dispersed across the system.

The NextGen RMF provides the technical and organisational measures required to implement adequate risk mitigating measures in the Pathfinder's pilots. The RMF takes the perspective of a sustainable NextGen platform where pilots might evolve to production systems dealing with the complexity of participating node interactions that involve servers, containers, services, and users as listed in this document. As a result, the NextGen RMF relates to the **complete life cycle** of data management, AI/ML and genomic acceleration tools developed including their deployment.

NIST provides and maintains a set of risk frameworks suitable for the Pathfinder platform:

- NIST Cybersecurity Framework (CSF),
- NIST Privacy Framework (PF),
- NIST AI Risk Management Framework (AI RMF).

The official documentation on the CSF states[2]:

> *"The NIST Cybersecurity Framework (CSF) 2.0 provides guidance to industry, government agencies, and other organizations to manage cybersecurity risks. It offers a taxonomy of high level cybersecurity outcomes that can be used by any organization — regardless of its size, sector, or maturity — to better understand, assess, prioritize, and communicate its cybersecurity efforts. The CSF does not prescribe how outcomes should be achieved. [...]"*

These frameworks provide guidance, as opposed to direct instruction, and can be tailored or augmented as needed.

[2] https://doi.org/10.6028/NIST.CSWP.29

3.6.2 Pathfinder specific risk landscape

The Pathfinder architecture RMF must consider the following specific factors within a multi-layered architecture.

Pathfinder Specific Factors:

- **Heterogeneity:** Managing risks across diverse hardware, operating systems, software versions, and geographical locations,
- **Scale:** Managing risks from simple peer-to-peer interactions to complex AI/ML lifecycle across organisations in different countries within EU and outside EU,
- **Decentralization:** Handling distributed nodes and multiple data modalities,
- **Dynamism:** Accommodating multiple processes (e.g. Pathfinder use cases) with changing or unknown requirements (open architecture),
- **"Zero-Trust"**[3]: Everything must be verifiable across the network without introducing a high-risk single point of failure.

Multi-Layered Architecture in NextGen: NextGen employs a multi-layered security approach to ensure privacy, compliance, and control for Data Holders. This is achieved through four primary layers:

1. **Governance:** Allows Data Holders to maintain control over data access, usage, and sharing through policies, contracts, and secure authentication and authorization processes.
2. **Architecture:** Utilizes decentralized data storage, federated computing, and federated machine learning to keep data with its owner, ensuring it is never copied or moved and remains under their control.
3. **Infrastructure:** Based on Kubernetes and containerization, it offers secure computational zones, ensuring encrypted communication between nodes within private networks.
4. **Services:** Ensures secure, isolated communication between services, preventing direct data exposure and controlling data exchange with enforced access restrictions, ensuring only authorized interactions occur.

3.6.3 Cybersecurity and Privacy Frameworks

The CSF, PF and AI RMF are structured into a Core, Profiles and Tiers. The **Core** defines the sets of functions (e.g. Govern, Identify, Protect) that are applied in each framework. The **Profiles** are customisable and identify which Functions/Categories/Subcategories are considered applicable for each of user-defined Current and Target configurations. In the CSF and PF, **Tiers** are also present which indicate the level of risk treatment and are Partial, Risk Informed, Repeatable and Adaptive.

The application of the RMF within NextGen is through the process:

[3] as in NIST terminology.

1. Identify applicable Functions/Categories/Subcategories by creating Profiles
2. Using the appropriate methods action the Core requirements in the Profile.

The use of Tiers (likely Partial or Risk Informed at TRL 4/5) for the CSF and PF could be both an input (desired state) or output (attainable state) of the above process.

The Core functions in the CSF are Govern, Identify, Protect, Detect, Respond and Recover. For NextGen, the Govern, Identify and Protect functions are relevant together while the Detect, Respond and Recover functions pertain to the sustainable NextGen and fall outside the primary scope of the RMF.

Core functions include Categories/Subcategories. A first assessment of the one relevant for the Platform is listed in the following table. These will be revised and re-assessed as the overall project progresses. The "NIST CSF 2.0 Organisational Profile"[4] has been used to develop this profile.

Table 7: Cybersecurity and Privacy Frameworks CSF

CSF Outcome (Function, Category, or Subcategory)	CSF Outcome Description
GV	The organization's cybersecurity risk management strategy, expectations, and policy are established, communicated, and monitored
GV.OC	The circumstances - mission, stakeholder expectations, dependencies, and legal, regulatory, and contractual requirements - surrounding the organization's cybersecurity risk management decisions are understood
GV.RM	The organization's priorities, constraints, risk tolerance and appetite statements, and assumptions are established, communicated, and used to support operational risk decisions
GV.RR	Cybersecurity roles, responsibilities, and authorities to foster accountability, performance assessment, and continuous improvement are established and communicated
ID	The organization's current cybersecurity risks are understood
ID.AM	Assets (e.g., data, hardware, software, systems, facilities, services, people) that enable the organization to achieve business purposes are identified and managed consistent with their relative importance to organizational objectives and the organization's risk strategy
ID.RA	The cybersecurity risk to the organization, assets, and individuals is understood by the organization
PR	Safeguards to manage the organization's cybersecurity risks are used
PR.AA	Access to physical and logical assets is limited to authorized users, services, and hardware and managed commensurate with the assessed risk of unauthorized access
PR.DS	Data are managed consistent with the organization's risk strategy to protect the confidentiality, integrity, and availability of information
PR.PS	The hardware, software (e.g., firmware, operating systems, applications), and services

[4] https://www.nist.gov/document/csf-20-organizational-profile-template

CSF Outcome (Function, Category, or Subcategory)	CSF Outcome Description
	of physical and virtual platforms are managed consistent with the organization's risk strategy to protect their confidentiality, integrity, and availability
PR.IR	Security architectures are managed with the organization's risk strategy to protect asset confidentiality, integrity, and availability, and organizational resilience

At the time of writing, the development of the NIST Privacy Framework 1.1 was taking place. PF 1.1 will be aligned with CSF 2.0. As the additions relate to AI they are relevant to NextGen and we'll incorporate modifications as they come in. As a profile template for PF 1.1 did not exist, we have constructed an exemplar using the "PF 1.0 to PF 1.1 Core Mapping"[5] and the "NIST CSF 2.0 Organisational Profile"[6] for illustrative purposes. An illustrative NextGen PF Profile is given in the following table, to be revised and re-assessed over the course of the project.

Table 8: Cybersecurity and Privacy Frameworks PF

PF Outcome (Function, Category, or Subcategory)	PF Outcome Description
ID	Develop the organizational understanding to manage privacy risk for individuals arising from data processing.
ID.IM	Data processing by systems, products, or services is understood and informs the management of privacy risk.
GV.OV	Oversight (GV.OV-P): Results of organization-wide privacy risk management activities and performance are used to inform, improve, and adjust the risk management strategy.
CT	Develop and implement appropriate activities to enable organizations or individuals to manage data with sufficient granularity to manage privacy risks.
CT.DM	Data Processing Management (CT.DM-P): Data are managed consistent with the organization's risk strategy to protect individuals' privacy, increase manageability, and enable the implementation of privacy principles (e.g., individual participation, data quality, data minimization).
CM	Develop and implement appropriate activities to enable organizations and individuals to have a reliable understanding and engage in a dialogue about how data are processed and associated privacy risks.
CM.AW	Data Processing Awareness (CM.AW-P): Individuals and organizations have reliable knowledge about data processing practices and associated privacy risks, and effective mechanisms are used and maintained to increase predictability consistent with the organization's risk strategy to protect individuals' privacy.
PR	Develop and implement appropriate data processing safeguards.
PR.AA	Identity Management, Authentication, and Access Control (PR.AA-P): Access to data, devices, and systems is limited to authorized individuals, services, and hardware, and is managed commensurate with the assessed risk of unauthorized access.

[5] https://www.nist.gov/document/pf-11-10-core-mapping
[6] https://www.nist.gov/document/csf-20-organizational-profile-template

PF Outcome (Function, Category, or Subcategory)	PF Outcome Description
PR.DS	Data Security (PR.DS-P): Data are managed consistent with the organization's risk strategy to protect individuals' privacy and maintain data confidentiality, integrity, and availability.
PR.PS	Platform Security (PR.PS): The hardware, software (e.g., firmware, operating systems, applications), and services of physical and virtual platforms and associated data are managed consistent with the organization's risk strategy to protect individuals' privacy and maintain data confidentiality, integrity, and availability.
PR.IR	Technology Infrastructure Resilience (PR.IR-P): Security architectures are managed with the organization's risk strategy to protect individuals' privacy and maintain data confidentiality, integrity, and availability.

3.6.4 AI Risk Management Framework

The AI RMF "*is intended to improve the ability to incorporate trustworthiness considerations into the design, development, use, and evaluation of AI products, services, and systems.*"[7] The AI RMF Core comprises the four functions of Govern, Map, Measure and Manage and, similarly to the CSF and PF, incorporates the concept of a Profile (although standard templates are not provided) which allows the specification of the functions that are determined to be relevant for a given entity. The application of the AI RMF, within the context of the Pathfinder, will commence with the development of the NextGen AI RMG Profile, guided by the AI RMF Playbook[8] and will follow the process described at the beginning of this section.

An important note related to the application of the AI RMF (and applicable regulations) is that it applies to deployed AI infrastructure (technically, AI Systems as understood in the EU AI Act), while model building using the Pathfinder (for example in federated learning) is more properly characterised as the development of machine learning (ML) algorithms. AI is also a broader concept than ML as the former suggests a discriminatory component ('intelligence') while the latter comprise specific algorithms. This means that not all risk management concepts for AI Systems will apply to ML algorithms; however, there are many components in common, such as the need for fairness, non-discrimination, data privacy, accuracy and reliability. This is the context in which to understand how the AI RMF (and, in general, Trustworthy AI/ML) will be applied and interpreted in the context of NextGen. Notably, the "OECD Framework for the Classification of AI Systems"[9] distinguishes between AI systems "in the lab" and "in the field". "In the lab" is most applicable in the Pathfinder (TRL 4/5) context, which relates to developmental aspects, while "in the field" relates to deployment which would be relevant for the evolution of the Pathfinder to a production/deployed state.

[7] https://www.nist.gov/itl/ai-risk-management-framework
[8] https://airc.nist.gov/docs/AI_RMF_Playbook.pdf
[9] https://www.oecd.org/en/publications/oecd-framework-for-the-classification-of-ai-systems_cb6d9eca-en.html

In the following table, description of the functions (taken from the AI RMF Playbook) and *sample* implementation aspects are given (noting that the NextGen AI RMF Profile will cover this in the appropriate detail).

Table 9: AI Risk Management Framework

Function	Definition and selected implementational aspects
Govern	Policies, processes, procedures, and practices across the organization related to the mapping, measuring, and managing of AI risks are in place, transparent, and implemented effectively. *Govern is more appropriately associated with a deployment context. Relevant to NextGen is the provision of the appropriate guidance related to ML development, which should encompass application of Trustworthy AI/ML.*
Map	Context is established and understood. *While Map again has a deployment focus, developmental aspects include provision of adequate documentation of the ML algorithms being developed (for ultimate deployment). In NextGen, datasheets for datasets and model cards are employed to achieve this.*
Measure	Appropriate methods and metrics are identified and applied. *Key metrics are established that characterise the "usefulness, usability and efficiency"; "fairness and equity", and "safety and reliability" of the ML algorithms; these groupings taken from the CHAI applied Model Card[10] which is for an AI solution as part of an AI system.*
Manage	AI risks based on assessments and other analytical output from the Map and Measure functions are prioritized, responded to, and managed. *In the context of deployed AI Systems, risks (from Map and Measure) need to be appropriately addressed. In NextGen, where ML models are developed for Personalised Medicine, risks would arise from shortfalls or limitations in the models, which may represent an appropriate focal point for this function.*

[10] https://www.chai.org/workgroup/applied-model

4 Data Oriented Architecture and Data Space Services

4.1 State of the Art

This section provides an overview of the current state of the art in Data-Oriented Architecture and Data Space Services.

A **Data-Oriented Architecture (DOA)** is a design approach that focuses on organizing, accessing, and processing data. Instead of building systems around process control, DOA focuses on how **data flows and transforms within the system**. This approach aims to improve performance and scalability by optimizing data management.

Data Spaces are designed to improve data usage **across different organizations and platforms**. They allow users to keep full control over their data by storing it locally, while still providing access to the data for tasks like federated learning and analysis. **The data stays in the owner's space** and can only be accessed with their permission, ensuring full control over data access, usage, and management, while also complying with organizational policies and regulations.

Despite the progress made, **challenges remain in areas such as management, compatibility, scalability, and security**. Improvements are needed in governance, trust mechanisms, and the harmonization of data and metadata. Future innovations will focus on federated learning, federated catalogs, improving ontology alignment, and strengthening decentralized identification solutions. **The NextGen platform aims to integrate these advancements into its architecture**.

4.2 Key concepts

4.2.1 Data Space

Data Space in NextGen Platform provides a robust solution for managing and controlling data products within a decentralized, data-oriented architecture. It supports operations on various infrastructures, including on-premises setups, allowing organizations to securely manage data across diverse environments.

A data product is essentially a data asset that is refined and structured in such a way that it is ready for use by end users or applications. It has been processed, organized, and presented with a specific purpose or use case in mind. A more detailed definition of the term "data product" will be given in the next section.

The platform enables users to retain full control over their own data products by storing them locally while facilitating data access and requests through the platform. Data products themselves do not reside within the data space but remain securely on-site, with data space

requesting access only with the owner's permission. This approach ensures complete control over data access, usage, and governance, enabling organizations to comply with their own policies and regulatory requirements.

Key features of the platform include:

1. **Data Product Quality Validation**
   Users can validate the quality of their Data Products before sharing, helping maintain data standards and reliability.
2. **Customizable Application Sharing**
   Users can store and share custom applications with partners, providing flexible, secure collaboration across organizational boundaries.
3. **Metadata Sharing and Searching**
   The platform supports the sharing of data products metadata, allowing for seamless data discovery and collaboration. This includes the ability to search metadata to identify partners that possess the necessary data products.
4. **Machine Learning- and Federated Learning Pipeline Support**
   The platform facilitates the deployment of machine learning pipelines and federated learning models, empowering users to train models collaboratively while keeping data private and secure.
5. **Policy-Based Access Control**
   Users can enforce data access rules through a policy engine, allowing precise control over data sharing and usage based on specific policies and governance requirements.
6. **Immutable Data Usage History**
   All data usage history is encrypted and stored immutably, ensuring that usage records cannot be tampered with or altered, supporting full transparency and accountability.

At the heart of this platform lies a **governance layer** that provides authentication, authorization, and policy-based control over data and applications. This is particularly beneficial in **environments where multiple organizations or departments with limited trust must collaborate**. It also addresses legal and regulatory barriers by providing secure, policy-driven access control and data management within a decentralized framework.

The Data Space is ideal for scenarios involving complex data-sharing needs and regulatory constraints, such as in consortia or cross-organizational collaborations where data security, compliance, and autonomy are paramount, and this is what we need to achieve in NextGen.

4.2.2 Data Product

4.2.2.1 Definition

This quote from Zhamak Dehghani's original article[11] is key to understanding the definition of data as a product:
"Domain data teams must apply product thinking [...] to the datasets that they provide; considering their data assets as their products and the rest of the organization's data scientists, ML and data engineers as their customers."

High level definition
A data product is a curated and structured data asset designed to be directly usable by end users or systems. It is processed, organized, and tailored to serve a particular purpose or address a specific need.

Low level definition[12]
At its core, a data product is a tangible output resulting from the processing, analysis, and interpretation of data. Unlike traditional goods or services, data products are centred around information, taking raw data—like datasets—and using it to build valuable insights, predictions, or visualizations. A dataset can be transformed into a data product that can take various forms, ranging from simple reports and dashboards to more complex machine learning models and predictive analytics tools.
What they all have in common is their reliance on data as the primary input, emphasizing the extraction of meaningful patterns, trends, and knowledge to create value. Once defined, these end products are designed to be consumed either internally within an organization or externally by customers and stakeholders, providing actionable intelligence that drives decision-making.
Incorporating product development methodologies into data products can significantly boost their value and effectiveness.

4.2.2.2 Principals[13]

An important quality of any technical product, in this case, domain data products, is to help their consumers; in this case data engineers, ML engineers, or data scientists. To provide the best user experience for consumers, the domain data products need to have the following basic qualities[14]:

1. **Value**
   Data products should add value for the data teams, departments, and organizations that use them.

[11] https://martinfowler.com/articles/data-monolith-to-mesh.html
[12] https://www.getrightdata.com/blog/data-products-101-what-is-a-data-product
[13] https://martinfowler.com/articles/data-monolith-to-mesh.html#DataAndProductThinkingConvergence
[14] https://www.qlik.com/us/data-management/data-products

2. **Prepared**
   Cleaned, transformed, high-quality data ready for analysis.
3. **Findable**
   Data products should be searchable and easy to find.
4. **Understandable**
   Data products should be easy to understand and their use case clear.
5. **Interoperable**
   Consists of one or more datasets that work with each other to bring holistic, unbiased data insights.
6. **Shareable**
   Several datasets and data elements packed into a single trusted cohesive unit, making it easy to distribute.
7. **Accessible**
   Accessible to data consumers when needed in a standardized manner.
   Users can access data products in several different ways, including APIs and SQL.
8. **Reusable**
   A data product is reusable in that data consumers, including other systems, can use it in different ways within the scope of its functionality.
9. **Trustworthy and Quality**
   The data inside a data product should be from a trustworthy data source. Data quality must also remain reliable over time.
10. **Secure**
    Must meet access, confidentiality, and compliance requirements.

4.2.2.3 Methodology

Data products are complex, taking distributed raw data and turning it into valuable information, knowledge, or actionable insights for users. At a high level, that transformation entails six key steps:

1. **Data collection and storage**
   A data product wouldn't exist if we didn't have raw data that we wanted to use more effectively, so the foundation of any data product requires the collection and storage of relevant data. This involves identifying sources, ensuring data quality, and establishing robust storage infrastructure.
2. **Clean and preprocess data**
   Raw data is not very useful for building data products. This is why it's necessary to do data cleaning, also known as data scrubbing or data cleansing. Some of the things that happen during this stage include:
   - Filling out missing values in data fields
   - Deduplication
   - Format standardization
   - Correcting errors in data

- Handling outliers
- Normalization and scaling
- Data validation
- Addressing inconsistencies

Once the process is done, the data is formatted in such a way that it can further be used for exploration, visualization, and making important business decisions.

3. **Processing and analysis**
   Before the end user ever sees the outcome, the data product must be able to successfully process and analyse data, often through statistical techniques, machine learning algorithms, and other analytical methods to derive meaningful insights.
4. **Visualization and communication**
   Communicating information in a way that's digestible for users is crucial. Data products often incorporate visualizations, dashboards, and reports to make complex information easily understandable and actionable for end-users.
5. **Integration**
   Data products rarely exist on their own; instead, they typically must integrate with existing systems, applications, or workflows to ensure seamless adoption and utilization within an organization. The last step of the data product transformation process must then entail making sure the product can "talk" to its surrounding environment.
6. **Refining**
   Finally, refining the products through feedback.

4.2.2.4 Benefits

1. **Reusability**
   Can be extended and shared for further development and lower cost per use.
2. **Self-service**
   Enables independent access and analysis, decreasing time to value.
3. **Agility**
   Allows for quick iteration and deployment.
4. **Fusion teams and data sharing**
   Builds trust and breaks down silos between teams by facilitating collaboration.
5. **Improved decision-making and increased efficiency**
   Provides insights for better decision-making.
6. **Automation of data analysis and processing**
   Saves time and resources.
7. **New revenue streams**
   Can be monetized for additional revenue and to build a competitive advantage.

4.2.2.5 Advanced-Data Architectures

1. **Data meshes**
   Data products are the building blocks of a data mesh. They enable a decentralized architecture for managing distributed data assets. Each domain owns and operates its data products, which communicate through standardized protocols and APIs. This allows the organization to leverage diverse and distributed data sources without compromising on quality, governance, or scalability.
2. **Private data marketplaces**
   Data products are a key component of building private data marketplaces. This allows people inside a company to easily find and access the data they need.
3. **Multi-cloud data**
   Well-designed, logical data products don't create copies of data until the time of use. This allows for data products that can exist or move across clouds, permitting seamless multi-cloud data.
4. **Data fabrics**
   A metadata-augmented data fabric enables automated data product creation. These self-describing data products, in turn, enrich the data fabric with additional metadata, fostering the development of derived data products.

4.2.2.6 Real-World Applications

In modern, data-driven businesses, there is no shortage of opportunities for data products to be put to use. Rather than being used solely by more technical organizations, data products are versatile and able to deliver value in applications used across diverse industries.

- In **e-commerce**, recommendation engines, personalized marketing campaigns, and demand forecasting models are examples of data products that can enhance customer experiences and optimize operations.
- In **healthcare**, data products can exist as predictive analytics for patient outcomes, population health management tools, and diagnostic support systems, contributing to improved decision-making and patient care.
- In the **finance sector**, fraud detection algorithms, credit scoring models, and portfolio optimization tools are commonly used to aid in risk management and strategic investments.
- In **maintenance systems**, supply chain optimization tools, and quality control algorithms improve efficiency and avoid downtime.

4.2.2.7 Considerations

While data products offer immense potential, they come with their set of challenges[15]. Privacy concerns, data security, and ethical considerations are often the biggest obstacles, especially

[15] https://www.getrightdata.com/resources/getting-started-with-data-products-series-part-two

when dealing with sensitive information. Additionally, organizations need to address issues related to data quality, integration complexities, and continually evolving rules and regulations.

4.2.3 Data-Oriented Architecture (DOA)

The DOA is a design approach that prioritizes data organization, accessibility, and processing efficiency, structuring systems around data flow and transformations rather than the control flow of processes. This approach aims to optimize performance and scalability by focusing on how data is managed and utilized throughout the system, serving as the foundation of the NextGen project architecture.
In below diagram data holder and data provider are used interchangeably and they represent the data layer.

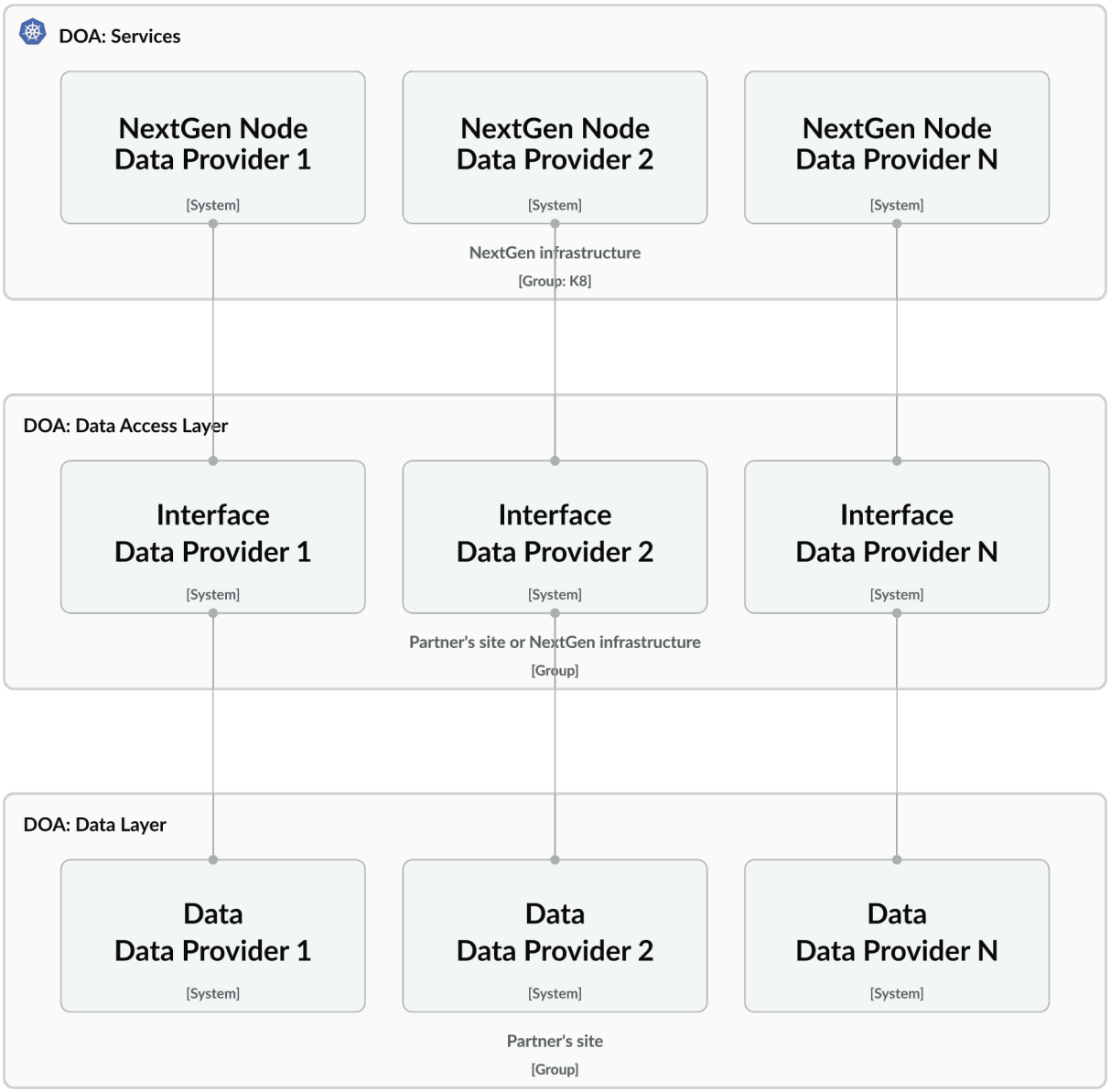

Figure 1: Data-Oriented Architecture

4.2.3.1 Principles

1. **Data-Centricity**
   Data is the core of the system, and everything is built around it. The main focus is on managing data as the most important resource.
2. **Components are stateless**
   Components don't keep any information between operations. This makes it easier to scale, run processes in parallel, and test the system.
3. **Separation of Data and Operations**
   Data and the operations that process it are kept separate. This makes the system more flexible and reliable.

4. **Component-to-component interaction is minimized**
   Components interact through the data layer rather than directly with each other. This reduces dependencies and makes scaling easier.
5. **Transparent Data Access**
   Access to data is standardized and simple, allowing applications to interact with data through common interfaces without needing to know implementation details.
6. **Flexible Data Architecture**
   The architecture can easily adapt to changes, using modular and extendable data schemas, allowing for quick responses to new requirements.
7. **Domain-Driven Design (DDD)**
   DOA heavily leverages DDD, which emphasizes understanding the core business domain and its data. The data model is designed to reflect real-world entities and their relationships, making it easier for developers to understand and work with the data.
8. **Event Sourcing**
   In DOA, changes to data are typically stored as a sequence of events. This allows for easier tracking of data history, auditing, and replaying past states if needed.
9. **CQRS (Command Query Responsibility Segregation)**
   This principle separates read and write operations. There might be a separate data store optimized for reads (queries) and another for writes (commands). This improves performance and scalability.
10. **Asynchronous systems**
    Components operate independently and communicate without synchronized timing. Communication is typically event-driven or message-based, allowing for non-blocking operations, concurrency, and fault isolation. This approach enhances flexibility, scalability, and resilience in distributed and real-time systems.

4.2.3.2 Benefits

1. **Improved Maintainability**
   Since data is central and well-defined, it's easier to understand how changes in one part of the system might impact others. This simplifies maintenance and reduces the risk of introducing bugs.
2. **Stronger Data Consistency**
   Having a single source of truth minimizes data inconsistencies that can plague distributed systems.
3. **Simplified Integration**
   New functionalities can be more easily integrated by interacting with the central data store, reducing the need for complex communication between services.

4.2.3.3 Challenges

1. **Potential Performance Bottleneck**
   If not properly designed, the central data store could become a bottleneck for read and write operations.
2. **Learning Curve**
   DOA concepts like event sourcing require a different mindset compared to traditional architectures. There's a learning curve for development teams.

4.2.4 Decentralised Key Management System (DKMS)

What is Key Management?
Cryptography plays a crucial role in securing communication over large-scale networks like the Internet by encrypting and decrypting messages using keys. As information flows across communication lines, encryption is essential to prevent unauthorized access to sensitive data.

In the domain of secure communication, asymmetric cryptography stands as a fundamental mechanism enabling safe interactions between users. This method relies on a key pair comprising two mathematically linked keys:

- **Private Key:** This key is used for encrypting outgoing messages, signing, and decrypting incoming messages intended for the key pair controller. The private key is strictly confidential to its owner.
- **Public Key:** Shared openly, the public key allows others to encrypt messages for the key pair controller or verify digital signatures.

Key Management ensures the secure handling of cryptographic keys throughout their lifecycle. This framework encompasses key aspects such as:

- **Key Generation**: Creating robust and secure key pairs.
- **Key Distribution:** Safely and accurately sharing public keys.
- **Key Storage:** Safeguarding keys from unauthorised access.
- **Key Rotation & Revocation:** Managing key updates and decommissioning compromised or expired keys.

A key focus of key management lies in the secure and reliable discovery of public keys to guarantee the accurate delivery of encrypted messages to the intended recipient and the verification of digital signatures. This process must maintain robustness and consistency across all interactions.

NextGen Decentralized Key Management
From a network standpoint, NextGen's Pathfinder represents a distributed system comprising research organizations located in various jurisdictions. The architecture of NextGen is structured around a *decentralized key management system*, facilitating a decentralized authentication mechanism that operates independently of a central server. This approach

guarantees the authenticity of data within a distributed environment, eliminating the necessity for a central location that may be subject to jurisdictional constraints in addition to creating a risk of single point of failure.

The implementation of a decentralised authentication framework is designed around a decentralized key management system. Combined with *self-certifying (SCID) and self-addressing (SAID) identifiers* the decentralised key management system also plays a key role in ensuring data integrity. SCID and SAID, as unique identifiers, ensure data integrity and security while being compatible across different platforms and networks. This robust framework enhances data security and integrity, offering a reliable mechanism to authenticate and track data across diverse research organizations seamlessly.
In summary, DKMS distributes the management of keys directly to the owners, eliminating single points of failure and reducing reliance on any one entity. This enhances security and resilience, as the compromise of one part does not jeopardize the entire system. DKMS offers the basis to build truly interoperable solutions, with mechanisms that ensure data integrity and authentication across various platforms/networks without needing centralized oversight.

DKMS -NextGen's authentication toolbox

DKMS[16] (Decentralized Key Management System) provides the tools needed to build secure authentication systems for NextGen's functional, non-functional and security requirements. DKMS is an advanced implementation of a decentralized authentication framework.

How Does DKMS Work?

DKMS binds an identifier to a log to track key pair changes. This ensures the identifier's provenance across infrastructures. The ordered log entries guarantee event authenticity, maintaining data integrity. System identifiers are network-agnostic for seamless interoperability.

DKMS prioritizes end-verifiability, allowing independent user verification for authenticity and integrity. This aligns with the zero-trust architecture "never trust, always verify" principle.

As a result DKMS supports identifier interoperability, data provenance, event streaming, and event-sourcing applications effectively. Specific DKMS impacts are:

- Scalable identity management (interoperability)
- Trackable data history without profiling (data provenance)
- Secured multi-party data processing (event streaming)
- Recording state changes to secure transaction (event sourcing)

What Technology Powers DKMS?

The core of DKMS is KERI[17] (Key Event Receipt Infrastructure), a protocol designed for secure, decentralized key management without depending on any specific network.

[16] https://dkms.colossi.network/
[17] https://keri.one/

KERI works by generating event logs that track every action taken with a cryptographic key—such as its creation, rotation, or revocation. These logs create an unbreakable, verifiable history, ensuring security and transparency. If a key is ever compromised, KERI can identify the exact moment and event without affecting the entire system.

Thanks to KERI, DKMS ensures secure, tamper-proof key management while allowing seamless interoperability across different data ecosystems.

To read more about DKMS you can check the official Deliverable D1.3 by HCF "Data Discovery Functionality (Part 1)" available on the NextGen website[18] or the dedicated DKMS website developed by HCF.

4.2.5 Multimodal Integration Object (MMIO)

The MMIO is an envelope concept designed to encapsulate any type of data (any modality) along with its semantic context and additional relevant information, such as purpose, consent, and access rules. To ensure a high level of portability and interoperability, all elements within MMIO are content-addressable, meaning each piece of content is represented by an identifier that is cryptographically derived from its data.

Within the NextGen Platform, MMIO is implemented as a library that enhances existing code with MMIO support. This library provides the following key functionalities:

- **Schema Inference**
  Extracts semantic meaning from raw data structures.
- **Semantic Interface**
  Enables fetching, querying, and framing of semantic objects.
- **Semantic Interoperability**
  Offers a mapping mechanism to represent data in specific formats or standards, allowing the linkage of multiple standards to create interoperability bridges.
- **Packaging Mechanism**
  Creates MMIO instances that encapsulate semantic information, data, and metadata, ensuring accurate data exchange with end-to-end verifiability through self-addressing identifiers.
- **Linking Mechanism**
  Establishes cryptographic links to external objects via self-addressing identifiers, ensuring secure and verifiable data connections.

The MMIO is the core component designed to enhance interoperability and data portability within a data ecosystem.

As part of **the information discovery mechanism**, the MMIO acts as an envelope that encapsulates any type of data (i.e., any modality) along with its semantic context and additional

[18] https://www.nextgentools.eu/deliverables/

relevant details such as purpose, consent, and access rules. Within the catalog, MMIO functions as a set of libraries that ingests data and its associated metadata, providing the necessary information for the Reasoning System to enable advanced search capabilities

An example of usage of MMIO for the qualification of data quality, the project is currently evaluating in WP6 (REG) the incorporation of a metric such as a "*diversity index*"[19]. For scientific or compliance purposes researchers using the Pathfinder architectures could be able to access such a composite diversity index compiled on an existing multimodal dataset. Such indices could be added as an attribute of the dataset or part of metadata files detailing the compliance of the dataset with ethics and regulatory requirements. For example, this will allow the verification of quality elements and potential existence of bias in alignment with EHDS Art. 78 "Data Quality and Utility Label" paragraph 3 (c) "for data quality management processes: the level of maturity of the data quality management processes, including review and audit processes, and bias examination;" and (d) "*for assessment of coverage: the period, population coverage and, where applicable, representativity of the population sampled, and the average timeframe in which a natural person appears in a dataset;".*

Deliverable D1.3 "Data Discovery Functionality (Part 1)" provides additional details regarding the MMIO.

4.3 Architecture

4.3.1 C4 Model

In the NextGen project, **the architecture** is structured using the **C4 model**. Both the system context diagram and the container diagram are developed, as these are the recommended diagrams and serve as the main focus of discussions with both technical and non-technical stakeholders.

The C4 model was created as a way to help software development teams describe and communicate software architecture, both during up-front design sessions and when retrospectively documenting an existing codebase. It's a way to create "maps of your code", at various levels of detail, in the same way you would use something like Google Maps to zoom in and out of an area you are interested in.

The goal of the C4 model is to raise the level of maturity associated with software architecture diagrams.

4.3.2 System Context Diagram (level 1)

A **System Context Diagram** serves as an effective starting point for visualizing and documenting a software system. It provides a high-level overview, presenting the broader

[19] https://www.sciencedirect.com/science/article/abs/pii/0022519366900130?via%3Dihub

system landscape. The diagram depicts the system as a central element, surrounded by users and other interacting systems.

The emphasis is placed on people (*actors, roles, personas*) and software systems rather than technical details such as protocols or technologies. This broader perspective illustrates how the system integrates into its environment and is suitable for both technical and non-technical audiences.

The intended audience includes individuals with both technical and non-technical backgrounds.

4.3.3 Container Diagram (level 2)

After establishing how the system fits into the broader IT environment, the next step is to **examine its internal structure using a Container Diagram**. A "container" refers to a separately deployable or executable unit, such as a server-side web application, single-page application, desktop application, mobile app, database schema, or file system. It represents any component that executes code or stores data within the system.

The Container Diagram offers a high-level representation of the software architecture, illustrating how responsibilities are distributed across various components. It emphasizes key technology choices and the interactions between containers. This technology-focused diagram serves as a valuable resource for software developers, support teams, and operations staff.

The intended audience consists of technical specialists both within and outside the software development team.

In our architecture containers are represented as services.

4.3.4 Component Diagram (level 3)

It is possible to examine each container in more detail to identify its key structural components and their interactions. The Component Diagram depicts the internal composition of a container, outlining each component's responsibilities, functionality, and the underlying technology or implementation details.

The intended audience includes software architects and developers.

Creating Component Diagrams is generally not recommended unless they provide clear value. In such cases, automating their generation is advisable to maintain accurate and up-to-date long-term documentation.

4.3.5 Code Diagram (level 4)

At the most detailed level, it is possible to focus on individual components to illustrate their implementation in code, typically using UML class diagrams, entity-relationship diagrams, or similar models. This level of detail is optional and is often accessible through development tools such as IDEs.

Ideally, these diagrams should be automatically generated using modelling tools (e.g., an IDE or UML tool). When creating them, it is advisable to include only the attributes and methods relevant to the specific context being conveyed. Such diagrams are generally reserved for the most critical or complex components.

The intended audience consists of software architects and developers.

Creating these diagrams is generally not recommended, particularly for long-lived documentation, as most IDEs can generate this level of detail on demand.

In the NextGen project, Levels 1 and 2 are developed. Level 3 is used only when needed. Level 4 is not used.

4.4 System Context Diagram

This section presents the system context diagram of the NextGen project.

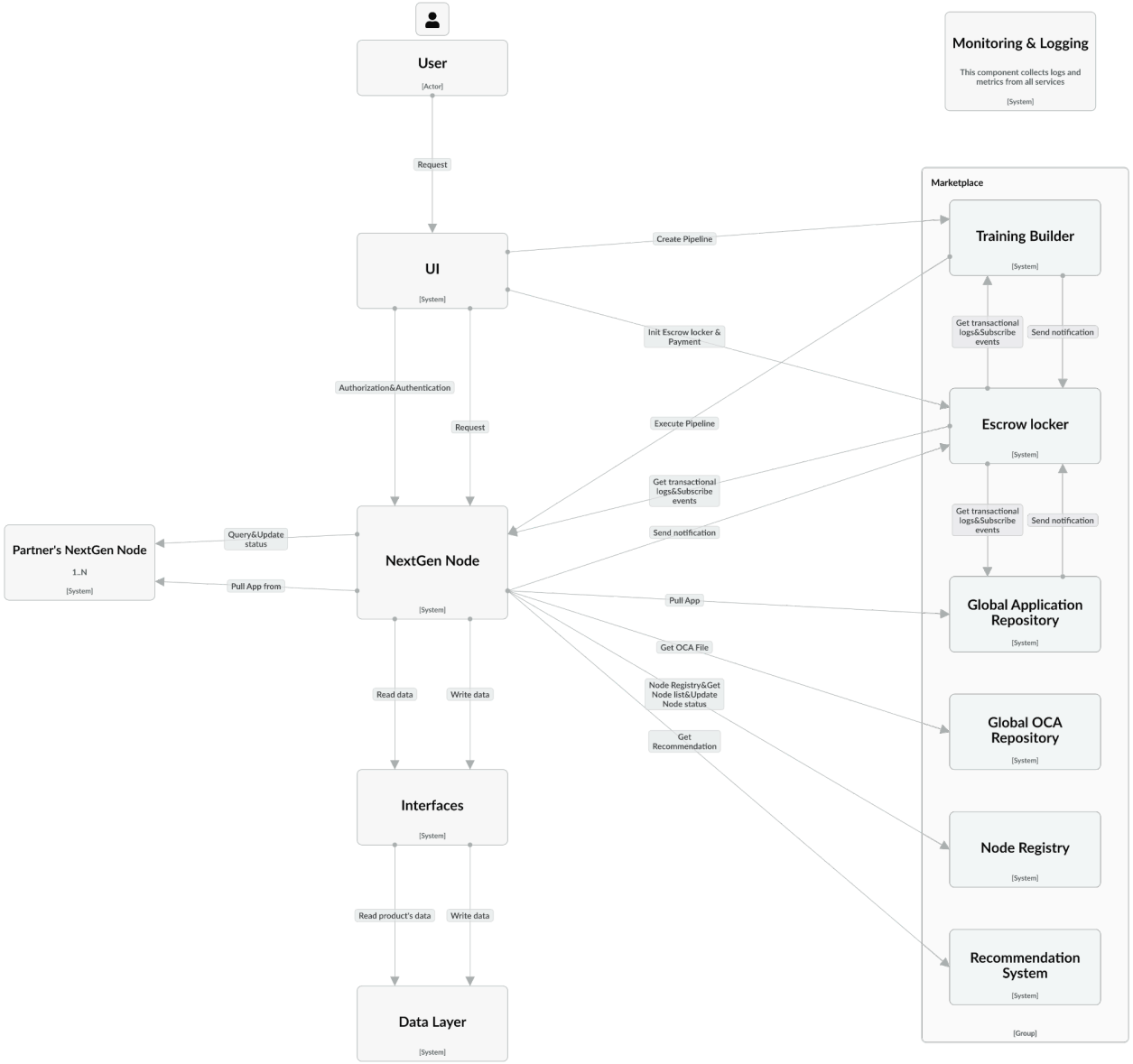

Figure 2: The system context diagram

NextGen Node is the core system, providing access to the Data Layer through Interfaces system that defines how data is managed.
The User interacts with their NextGen Node through the UI, which gives access to the node's features.
NextGen Node can also exchange data directly with other Partner's NextGen Nodes, forming a distributed network.
Additionally, the NextGen Node is connected to the Marketplace, which includes systems such as Training Builder, Escrow Locker, Global Application Repository, Global OCA Repository, Node Registry, and Recommendation System.

The Monitoring & Logging system collects, stores, and provides an interface for metrics and logs from all systems and infrastructure.
These systems and their architecture will be described in the following sections.

4.4.1 NextGen Node

NextGen Node is the system of the NextGen project, serving as a node in a decentralized network. It manages the partner's data Catalog, ensures secure access to Data Products and Applications for federated learning, and provides data format compatibility, data usage control, and data usage history tracking.

Each partner operates their own instance of the NextGen Node.

The system consists of:

- **Gateway**
  All requests to the NextGen Node are routed through the Gateway, which distributes them to internal services while ensuring authentication and authorization.
- **Governance** (Policy Engine, Authentication & Authorization, DKMS)
  Governance is a group of services designed to handle user authentication, authorization, and policy validation.
- **Catalog** (Catalog Service, Knowledge graph)
  A Catalog is a group of services running locally on the NextGen Node. The metadata of each Data Product or Application is stored in the Catalog as a Catalog Item. Catalog Items are stored in a knowledge graph following an ontology and are governed by a Policy that defines who can access the data, how it can be used, and when.
- **Connector Service**
  The Connector Service is a service that provides a unified interface for NextGen Node internal services to interact with the Data Access Layer that provides Data Products from the Data Layer.
- **Local Application Repository**
  The Local Application Repository stores and distributes Applications, which are algorithms packaged. These Applications are used to build machine learning (ML) or federated computing pipelines.
- **Search Engine** (Search Engine Service, Local Node Registry)
  The Search Engine is a group of services, designed to process search queries and aggregate results from decentralized Partner's Catalogs and the local Catalog.
- **Contract Engine** (Contract Generator, Contract Validator)
  The Contract Engine is a group of services that enables the creation and validation of Contracts for the use of Catalog Items in a machine learning Pipeline.
- **ML Runner**
  ML Runner is a service that executes the machine learning Pipelines.
  The Pipelines is a structured sequence of data processing and model training steps that automate the workflow to a predictive model.

- **Clearing House Service**
  The Clearing House is a distributed service, designed to collect and store transaction logs related to creating Contract, executing Contract, and accessing Data Products and Applications.
- **Quality Control** (Scoring Service, Data Product Validator)
  Quality Control is a group of services responsible for verifying that a Data Product complies with baseline requirements for quality, completeness, and documentation.
- **Local OCA Repository**
  Local OCA Repository is a service for storing and sharing data objects like OCA Bundles, Capture Bases, and Overlays.
- **Recommender**
  The Recommender service responsible for getting recommendations from the Recommendation System to the user, enabling personalized recommendation delivery.

A detailed description of each service will be described in the following sections.

Related systems

- **UI**
  Users (owners and authorized personnel) manage the NextGen Node via the UI.
- **Partner's NextGen Node**
  NextGen Node communicates with Partners' NextGen Nodes to exchange status updates, retrieve information about other nodes in the decentralized network, perform decentralized data catalog and application searches, and provide and pull applications.
- **Interfaces System**
  NextGen Nodes read and write data to the Data Layer through the Interfaces system only. The partner owning the NextGen Node implements Interfaces for secure integration with the Data Layer.
- **Training Builder**
  The Training Builder allows users to first create a Pipeline using contracts related to Data Products and Applications. After that, it runs the Pipeline in a federated way on the NextGen Nodes of the data providers.
- **Escrow locker**
  The Escrow Locker subscribes to events and receives transactional logs from the internal Clearing House Service of the NextGen Node to determine the execution status of Contracts for Data Product and Application usage.
- **Global Application Repository**
  The NextGen node uses common applications from the Global Application Repository that are required to run the Pipeline.
- **Global OCA Repository**
  The NextGen Node retrieves OCA files from the Global OCA Repository for processing MMIO.

- **Node Registry**
  The NextGen Node obtains information about network participants from the Node Registry during onboarding.
- **Recommendation System**
  The Recommendation System provides suggestions to NextGen Node for Data Products and Applications that align with user needs and interests.

Interfaces
it is the standard interface that is define by nextgen platform to be used by other systems.

4.4.2 User

The User in the NextGen architecture represents an active participant in the system who interacts with the platform through a user interface (UI). This component provides access to system features based on roles.

Main functions
- **Login & Access Control**
  Users securely log in to access their personalized data, services, and features.
- **Managing Catalogs**
  Users with the right permissions can create, edit, and delete Catalog Items.
- **Sharing Catalog Items**
  Users can control sharing Catalog Items with other participants.
- **Searching**
  Users can find Catalog Items listed in decentralized Catalogs.
- **Managing Contracts**
  Users can initiate contract agreements for the use of a Catalog Item.
- **Running ML & Analytics**
  Users can launch federated analytics and machine learning processes using distributed Catalog Items.

4.4.3 Roles

Roles define user responsibilities and access levels across various services. Security, access control, and data management are critical components of the system, ensuring that each user has the appropriate permissions while maintaining data integrity and privacy.

NextGen Architecture leverages a decentralised authentication mechanism based on DKMS (see sec. 4.2.4). This enables a dynamic management of the roles based on a system of credentials issued by legitimate participants and verifiable by all.

The following sections outline the key roles and their functions:

Catalog

- **Catalog Owner**
  A person who owns catalog items and can delegate permissions to control them.
- **Catalog Creator**
  A person who can create, read, update, and delete catalog items.
- **Catalog Provider**
  A person who can control and provide catalog Items for other partners.
- **Catalog Consumer**
  A person who can search and filter the catalog items.

Model

- **Model Owner**
  A person who owns the models after the model has been trained.
- **Model Consumer**
  A person who can deploy and use models.

Pipeline

- **Pipeline Creator**
  A person who creates the pipeline.
- **Pipeline Executor**
  A person who can execute the pipeline.

Application

- **Application Owner**
  A person who owns applications and can delegate permissions to control them.
- **Application Creator**
  A person who can create, read, update, and delete applications.
- **Application Provider**
  A person who can control and provide the applications for other partners.
- **Application Consumer**
  A person who can use applications.

Data Product

- **Data product Owner**
  A person who owns Data Products and can delegate permissions to control them.
- **Data product Creator (will be not part of NextGen as it should be part of our partners infrastructure)**
  A person who can create, read, update, and delete data products.
- **Data product Provider**
  A person who can control and provide the data products for other partners.
- **Data product Consumer**
  A person who can use the data products.

OCA

- **Vocabulary Owner**
  A person who owns ontologies, inferring and mapping rules and can delegate permissions to control them.
- **Vocabulary Creator**
  A person who can create, read, update, and delete ontologies, inferring and mapping rules.
- **Vocabulary Provider**
  A person who can control and provide the data ontologies, inferring and mapping rules for other partners.
- **Vocabulary Consumer**
  A person who can use ontologies, inferring and mapping rules.

Clearing House Transactions

- **Transaction Clearer**
  A person who can read transactions in the Clearing House.

Infrastructure

- **Infrastructure Operator**
  A person who can manage infrastructure and deploy services.

Data Space Operator

- A person who can manage marketplace systems.

Composite roles

- **Data Consumer**
  A person with multiple roles:
  - Catalog Consumer
  - Application Consumer
  - Data Product Consumer
  - Pipeline Creator
  - Pipeline Executor
- **Data Provider**
  A person with multiple roles:
  - Catalog Provider
  - Application Provider
  - Data Product Provider

These are the basic roles and they are subject to change if required

4.4.4 User Interface (UI)

User interface is the system that handles specific functions or displays certain content. It connects the front-end design with back-end services, allowing users to interact with the system efficiently.

The UI is the interface through which users interact with the NextGen platform.

- It enables users to manage the Catalog Items using the Catalog Service, execute search query for the Data Products and Applications using Search Engine Service, manage Local Application Repository and Local OCA Repository, list Data Products using the Connectors, read logs using Clearing House Service and create Contracts using Contract Generator Service via the Gateway as it is the entry point for all the traffic inside the node.
- It will allow the Data Consumer to initiate the Escrow Locker to validate the Contract registration.
- It will redirect the user to the Training Builder to build machine learning Pipelines and do the federated learning.

Related systems

- NextGen Node
- Training Builder
- Escrow Locker

Interfaces

The system provides APIs which support HTTP(S) requests following internal API specifications for accessing the Marketplace services and NextGen Node services.
It communicates with the node services using the Gateway, each request to the Gateway must include authorization credentials, verified against the Authentication & Authorization service.

4.4.5 Interfaces System

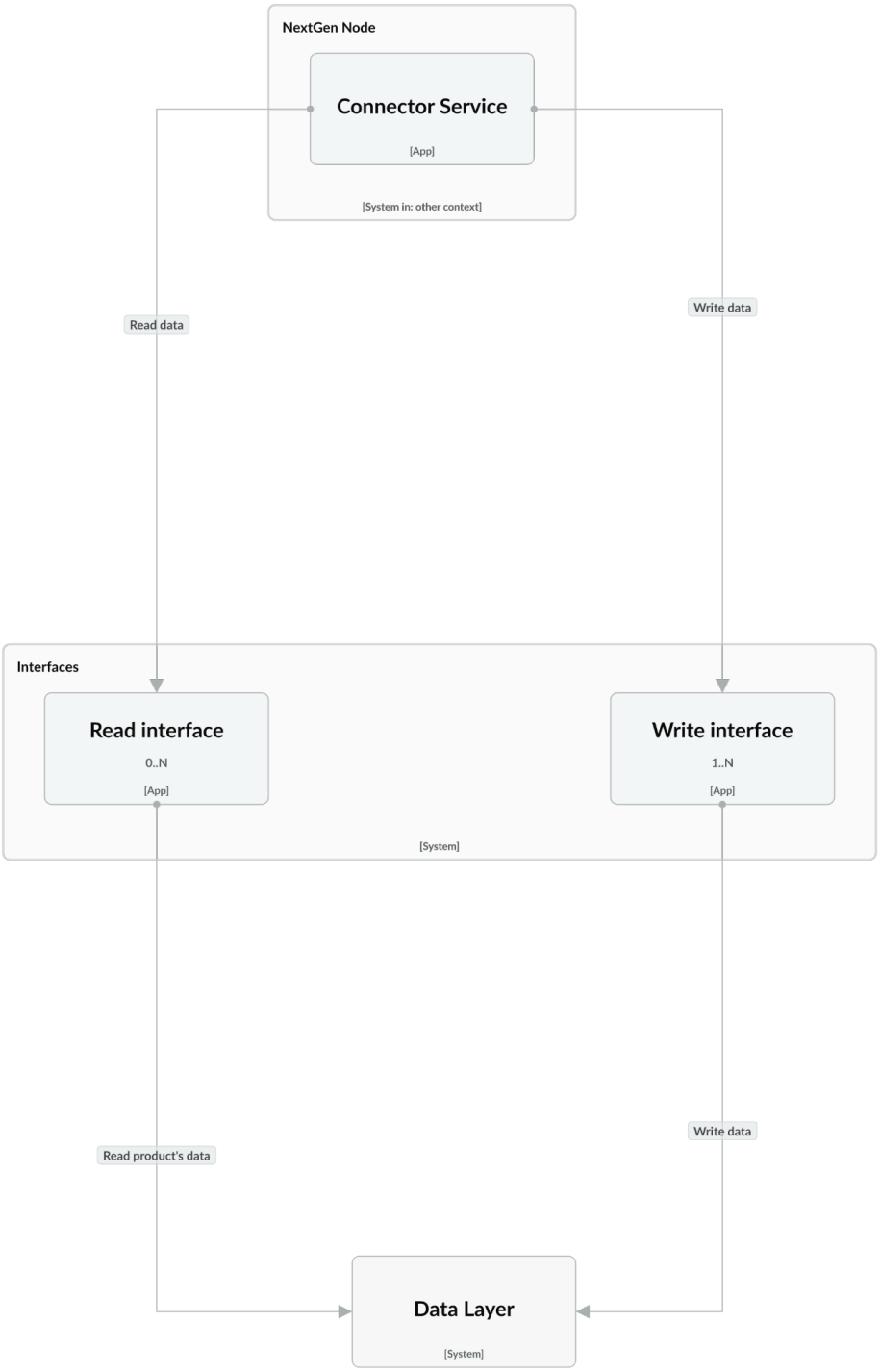

Figure 3: Interfaces

Interfaces is a system that acts as a bridge between the Connector Service and the Data Layer. It should be implemented and deployed by the Data Holder within their own infrastructure, providing interfaces for reading and writing data to the Data Layer. This ensures that the Data Holder retains full control over the access that the NextGen Node has to their data.

In abstract way interfaces standard way to communicate with NextGen services deployed in our partners infrastructure to allow managing the data.

The Interfaces system includes two types of interfaces:

- Read Interface, used for reading Data Product's data from the Data Layer.
- Write Interface, used for writing data to the Data Layer.

The Interfaces system can contain one or more Read Interfaces and Write Interfaces.

Related systems and services

- Connector Service
- Data Layer

Interfaces

- The Connector Service interacts with the Interfaces using an S3-compatible API.
- The interface and protocol between Interfaces and the Data Layer are defined by the Data Holder.

4.4.6 Data Layer

The Data Layer is a system that stores the actual data from a data provider.

The NextGen Node interacts with the Data Layer only through Interfaces, which must be implemented and deployed by the Data Holder.

Related systems and services

- Interfaces (Read Interface, Write Interface)

Interfaces

The interface and protocols are defined on the Data Holder's side during the Write Interface and Read Interface development.

4.4.7 Training Builder

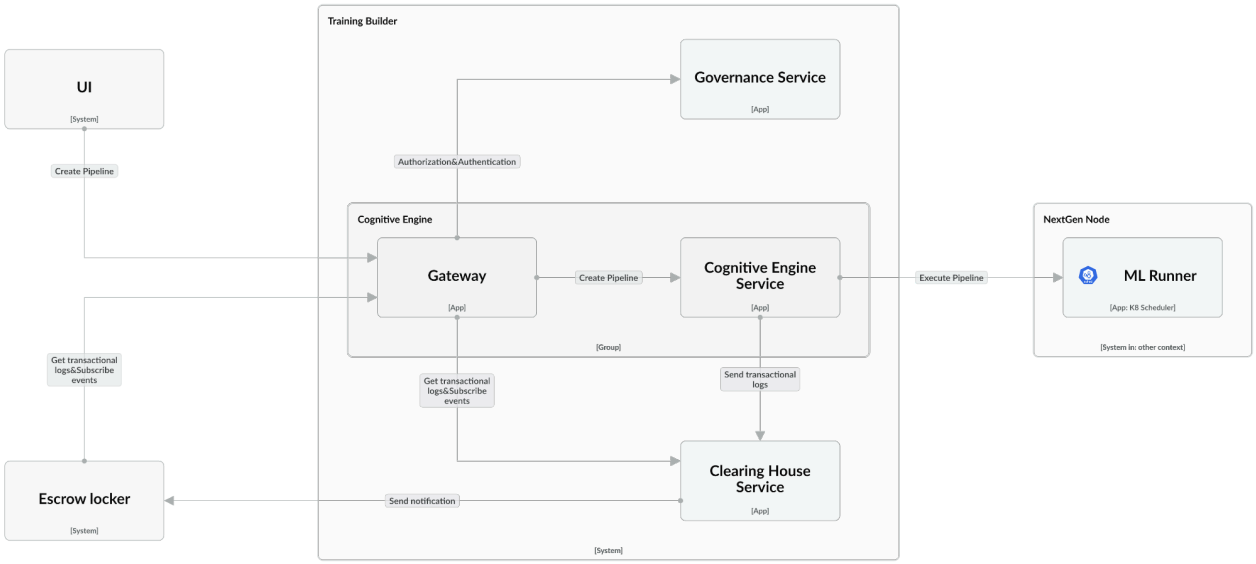

Figure 4: Training Builder

The Training Builder system is designed to construct both federated learning and machine learning Pipelines.

Training builder will include:

- **Sandbox**: which is the safe and secure environment where the model and sample dataset can be tested.
- **Model validator**: set of tools to validate the model against the data set by the researcher.
- **Model development**: set of supporting tools to allow the researcher to define and develop his own model.
- **Model Deployer**: component responsible for deploying the model into the existing infrastructure.

It requires the Contract, which provides the necessary information on how to locate and access Data Products and Applications.
The user uses this Contract to build a federated learning or machine learning Pipelines through training builder UI.

The Training Builder delegates the execution of the Pipeline to the ML Runner, utilizing a Kubernetes Job object; this Job runs within the node where the execution occurs.

While the Training Builder operates as a centralized system, the actual execution takes place in a decentralized manner via the ML Runner within the NextGen Node.

Cognitive Engine Service sends transactional logs to the Clearing House Service.
Escrow Locker receives transactional logs from the internal Clearing House Service.

Related systems and services

- UI
- ML Runner
- Escrow Locker

Interfaces

The system provides APIs which support HTTP(S) requests following internal API specifications for accessing the Training Builder.

4.4.8 Escrow Locker

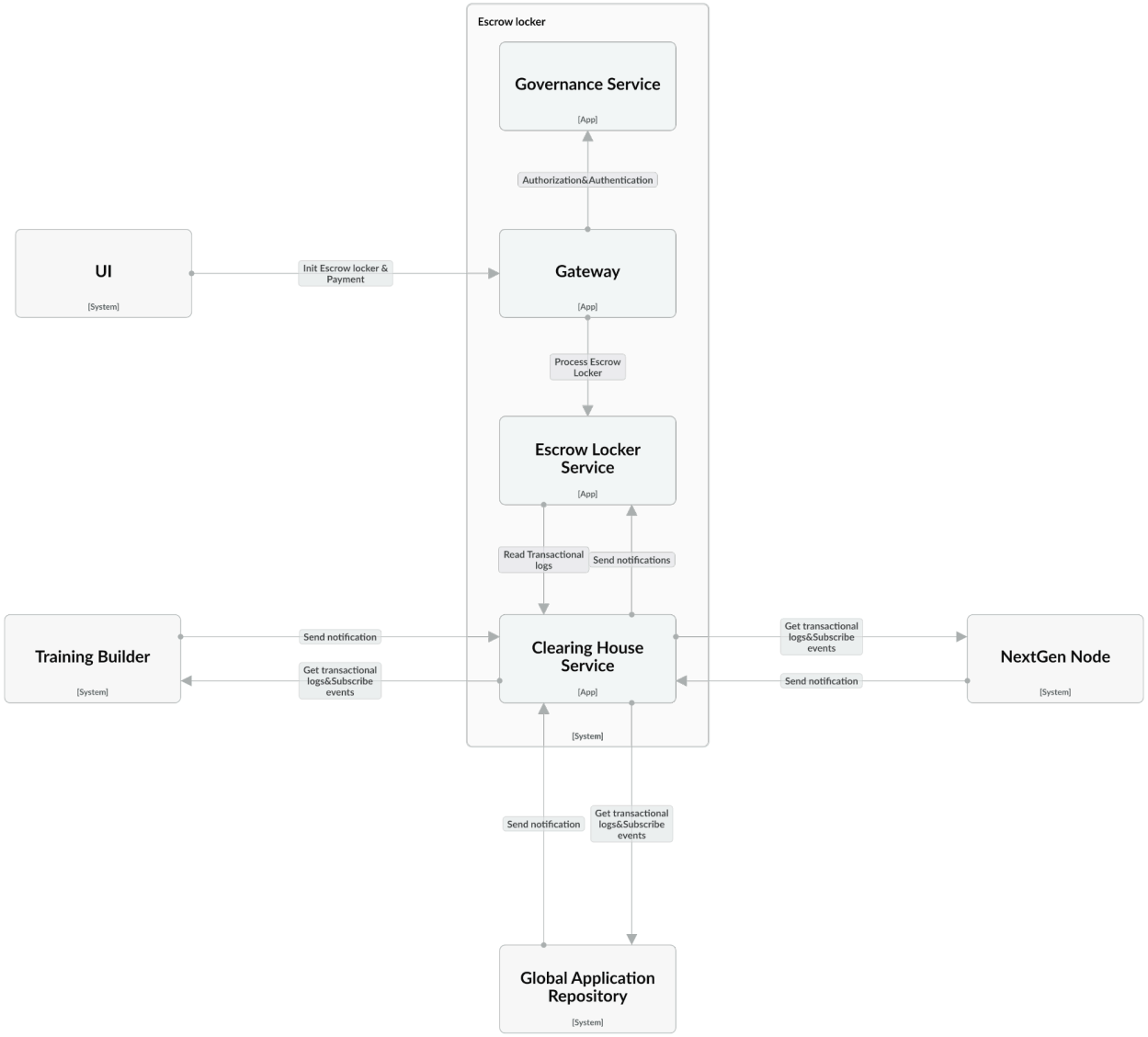

Figure 5: Escrow Locker

The Escrow Locker is the system responsible for locking and unlocking the money after ensuring that governance requirements for both the Data Provider and Data Consumer are in place. It receives the payment from the Data Consumer and holds it until the process is

completed. Once validation is confirmed, it transfers the payment to the Data Provider and notifies both parties that the process was successfully completed.

Currently, the NextGen project does not involve monetary transactions. Therefore, this system will primarily focus on verifying governance compliance, validating the Contracts and notifying the relevant parties that the process has been successfully completed.

The system consists of:

- Gateway
- Escrow Locker Service
- Governance Service
- Clearing House Service

Requests from the UI to the Escrow Locker go through the Gateway. The Gateway sends a request to the Governance Service to be authenticated and authorized. If the verification is successful, the request is forwarded to the Escrow Locker Service.

Process

The Data Consumer will initiate the Contract using the Escrow Locker and the Escrow Locker Service will validate that the Contracts is registered in the Clearing House, if so it will reply with success, sign the Contract to the Data Consumer, start a time and subscribe to the Contracts events from the participants' Clearing Houses so it will be notified about the Contracts updates and when it is done, if something went wrong or the process takes more than the expected running time it will stop the process and cancel the Contract.

Related systems and services

- UI
- Training Builder
- Global Application Repository
- NextGen Node

Interfaces

The system provides a single API entry point, which supports HTTP(S) requests following internal API specifications.

4.4.9 Global Application Repository

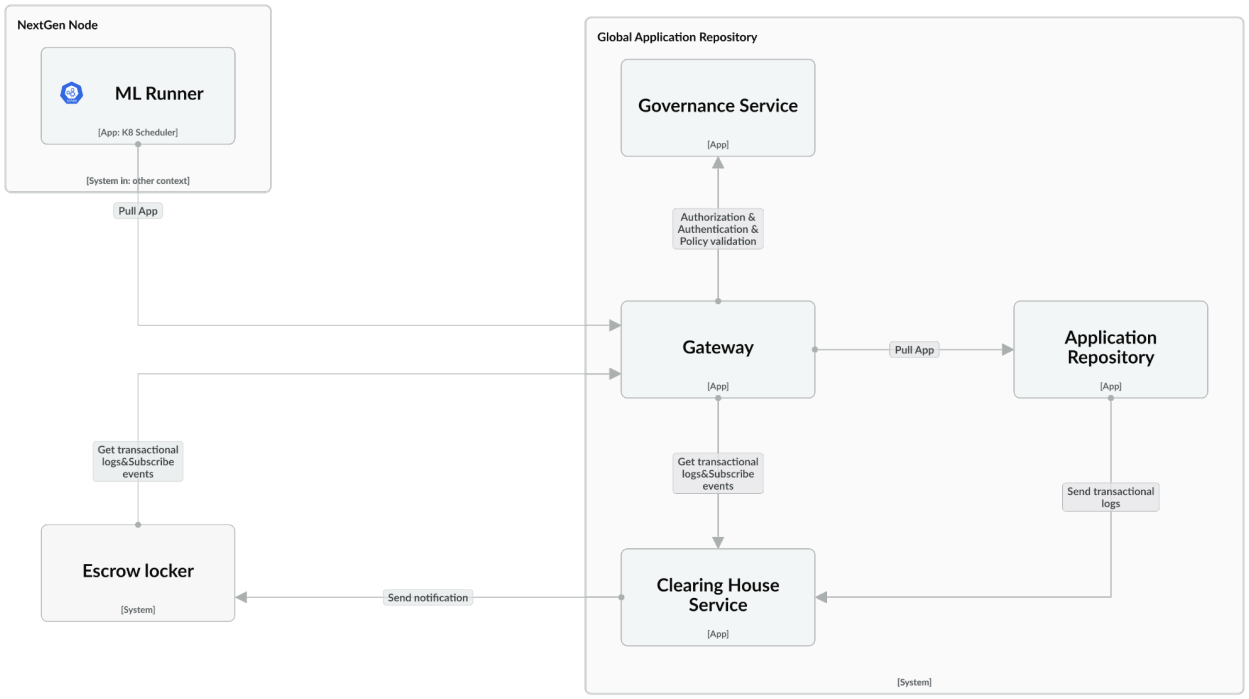

Figure 6: Global Application Repository

The Global Application Repository is a centralized storage for Applications, which are used by the Training Builder and shared across all nodes. It provides these Applications during Pipeline execution in the ML Runner.

The system consists of:

- Gateway
- Governance Service
- Application Repository
- Clearing House Service

All external pull/push requests go through the Gateway service. These requests include credentials. The Gateway requests credential validation from the Governance Service.

If the credentials are valid, the Gateway proxies the request to the Application Repository, which stores and serves. If the credentials are invalid, the component rejects the request.

Application Repository Service sends transactional logs about usage of Applications to the Clearing House Service.

Escrow Locker subscribes to notifications from the Clearing House Service. When a new transactional log is registered, the Clearing House Service sends a notification to its subscribers. Escrow Locker can also read transactional logs from the Clearing House Service through the Gateway.

Related systems and services

- ML Runner
- Escrow Locker

Interfaces

- The Global Application Repository receives and processes HTTP(S) API requests according to the documentation at https://distribution.github.io/distribution/spec/api/, which is the standard protocol for interacting with a Docker repository.
- The Global Application Repository also receives HTTP(S) requests to the Clearing House Service in accordance with internal API specifications.

Each HTTP(S) request can include an Authorization header with the credentials.

4.4.10 Global OCA Repository

The Global OCA Repository[20] is a key concept of the Overlays Capture Architecture used to ensure data integrity in an ecosystem. This component is at the heart of the OCA Ecosystem, a set of tools designed to facilitate the integration of OCA into software solutions. The OCA Repository enables the management, storage, and sharing of OCA Objects like OCA Bundles, Capture Bases, and Overlays. Furthermore, it comes with pre-baked support for OCAFiles.

Related systems and services

- NextGen Node

Interfaces

OCA-SDK Software Development Kit is available in a form of libraries and binary which allows to integrate that in the code or use it as standalone application serving functions to manage identifiers.

The Overlays Capture architecture (OCA) Software Development Kit is a Rust library with bindings to other languages providing programmable interfaces to interact with OCA artefacts like OCAFILE and OCA Bundles or specific overlays and capture base. The SDK is in development at this stage so for up to date details how to use the library please refer to the official documentation on the git repository[21].

[20] https://github.com/THCLab/oca-repository-rs
[21] https://github.com/THCLab/oca-sdk-rs

4.4.11 Node Registry

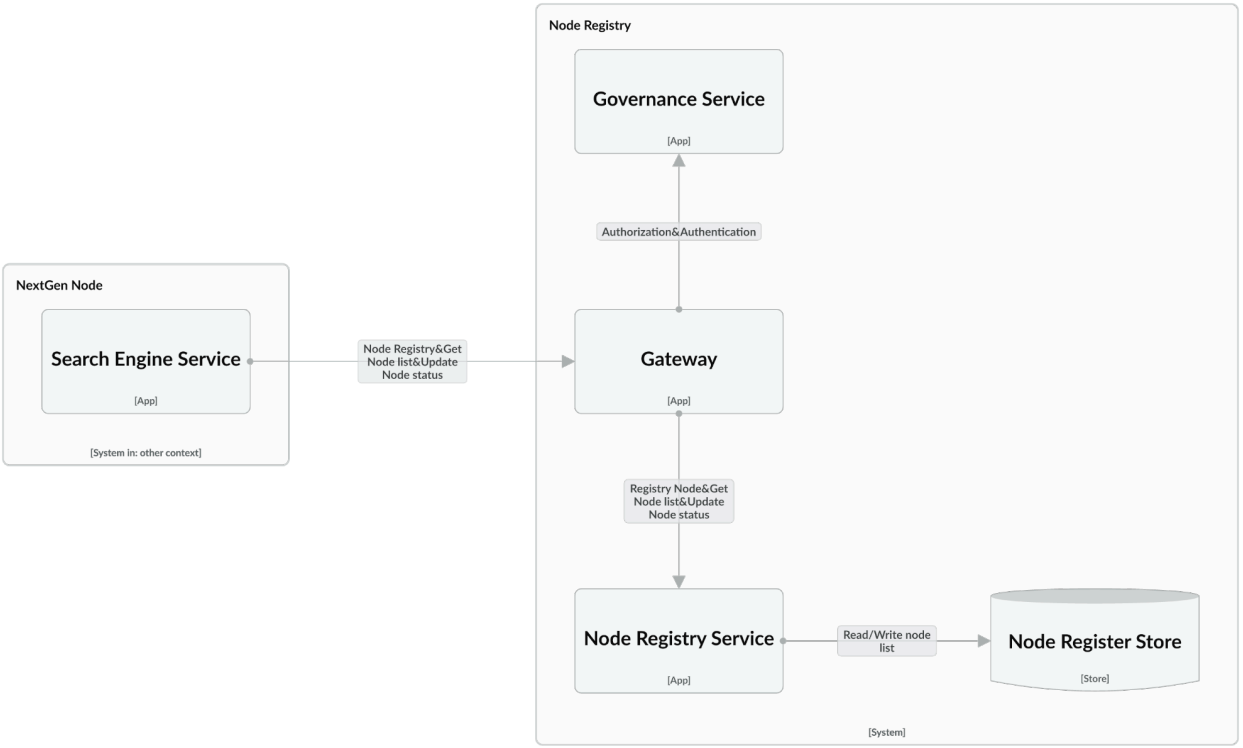

Figure 7: Node Registry

The Node Registry is a bootstrap node for new participants in NextGen.
During onboarding, a new participant receives the initial list of distributed NextGen nodes from the Node Registry. After that, the participant gets and sends updates to the list directly using a peer-to-peer (p2p) protocol.

The system consists of:

- Gateway
- Governance service
- Node Registry Service
- Node Register Store

All requests to the Node Registry go through the Gateway. The Gateway sends a request to the Governance Service for authentication and authorization. If the authentication and authorization are successful, the request is forwarded to the Node Registry Service.

During onboarding, the new participant's Search Engine Service sends a request to the Node Registry to register its Catalog, making it available to other participants. The Node Registry Service stores Catalog information in the Node Registry Store. Then, the Search Engine Service retrieves the list of NextGen participants.

The Node Registry Service also receives information about participant status changes directly from other participants using a peer-to-peer (p2p) protocol and updates the list of distributed Catalogs in the Node Registry Store.

Related systems and services

- Search Engine Service

Interfaces

Related systems and services interact with the Search Engine using HTTP(S) requests that follow internal API specifications.

4.4.12 Recommendation System

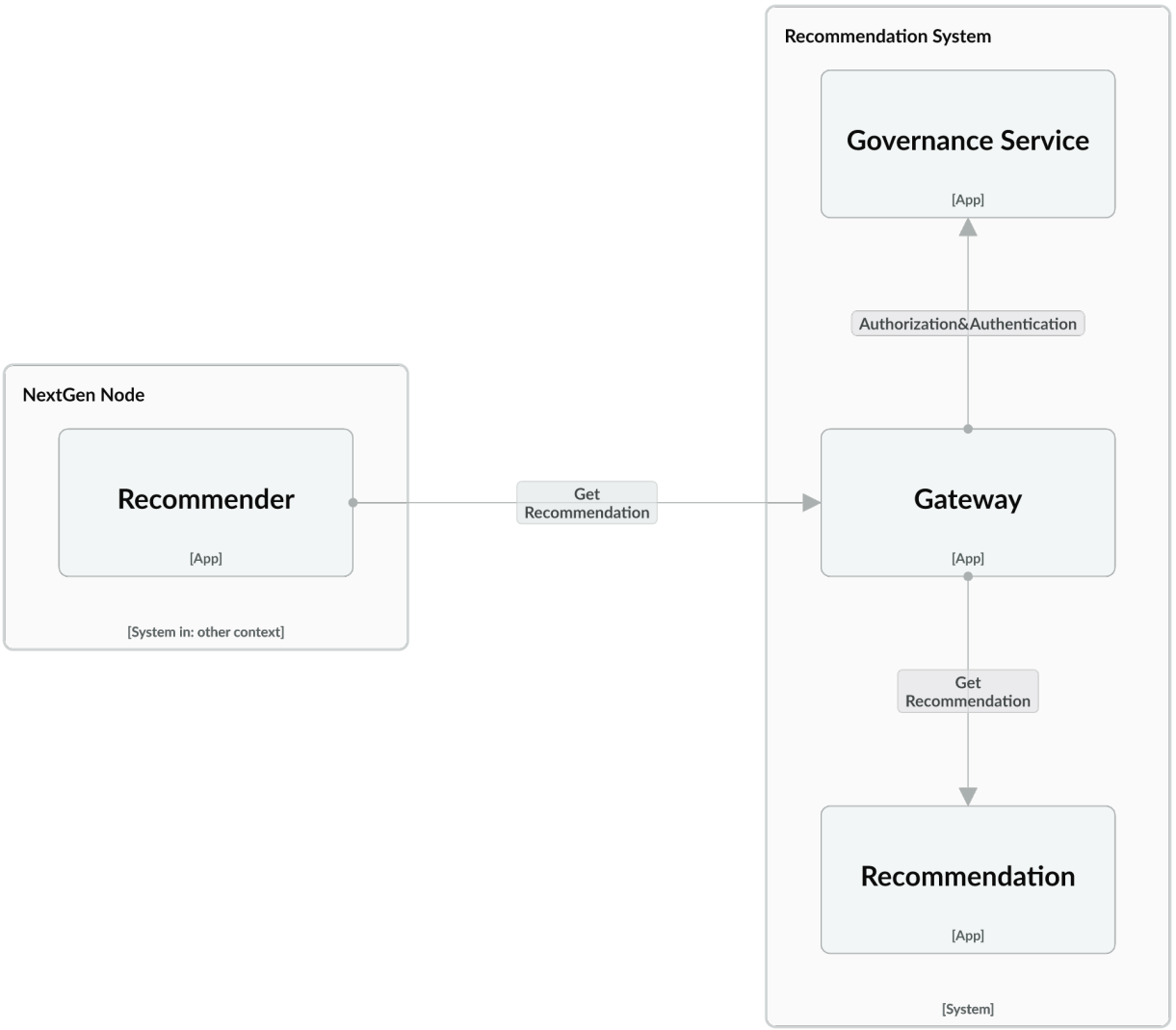

Figure 8: Recommendation System

The Recommendation System is a system designed to provide personalized suggestions to users by leveraging machine learning models. This system plays a key role in enhancing user

experience by offering recommendations for Applications and Data Products that align with user needs and interests.

The system consists of:

- Gateway
- Recommendation
- Governance service

All requests to the Recommendation System go through the Gateway. The Gateway sends a request to the Governance Service to be authenticated and authorized. If the verification is successful, the request is forwarded to the Recommendation.

Functionality

- **Data Aggregation and Model Training**
  Aggregates user interaction data over time.
  Trains machine learning models using this aggregated data to identify patterns and preferences.
- **Recommendation Generation**
  Based on the trained model, the system generates meaningful recommendations tailored to individual users.

Activation Criteria

The recommendation system will be activated only after collecting sufficient user interaction history. This ensures that the model is adequately trained and capable of delivering accurate and valuable recommendations.

Related systems and services

- Recommender

Interfaces

The system provides a single API entry point, which supports HTTP(S) requests following internal API specifications.

4.4.13 Monitoring & Logging

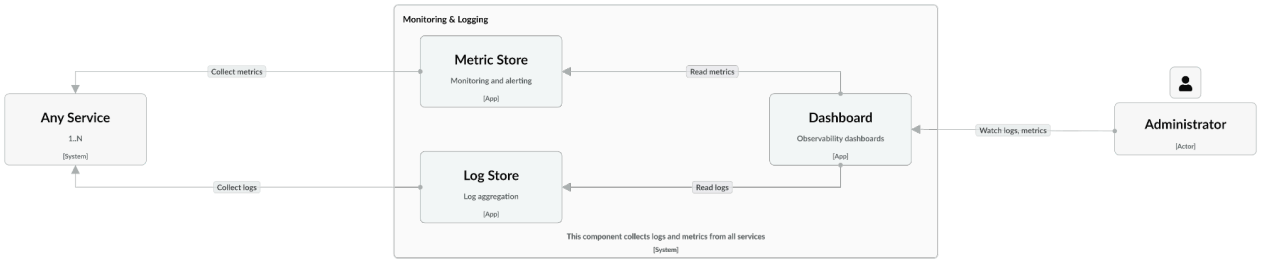

Figure 9: Monitoring & Logging

The Monitoring & Logging system collects telemetry metrics and logs, stores them, and visualizes data on dashboards. It gathers information from all NextGen systems, services, and the underlying infrastructure. Additionally, it manages alert configurations.

The system consists of:

- Metric Store
- Log Store
- Dashboard

The Metric Store collects and stores telemetry metrics from the underlying infrastructure and NextGen services.

The Log Store collects and stores logs from the underlying infrastructure and NextGen services.

The Dashboard gets metrics from the Metric Store and logs from the Log Store, providing administrators with an interface for monitoring and analysing the data.

Related systems and services
All NextGen systems, services, and the underlying infrastructure.

Interfaces
The system collects telemetry metrics in a Prometheus-compatible format.
The system includes the Dashboard for visualizing telemetry metrics and logs. Additionally, its HTTP(S) API endpoints enable programmatic interaction, allowing users to query and retrieve data, as well as manage alerts.

4.4.14 Marketplace

Marketplace is a centralized logical system that includes essential systems such as a Training Builder, Escrow Locker, Global Application Repository, Global OCA Repository, Node Registry, and Recommendation System.

It acts as a middleman between partners, helping them find other members of the distributed network during onboarding, create and run FL Pipelines, act as a trusted party, and more.
The functionality of the Marketplace is defined by the set of systems it includes.

Partners can choose which Marketplace they prefer to use.

Related systems and services

- UI
- NextGen Node

Interfaces
The interfaces of the Marketplace are defined by the set of systems it includes

4.5 NextGen Node Architecture

This section presents the architecture of the NextGen Node system at Level 2. It describes the internal services and interactions between them.

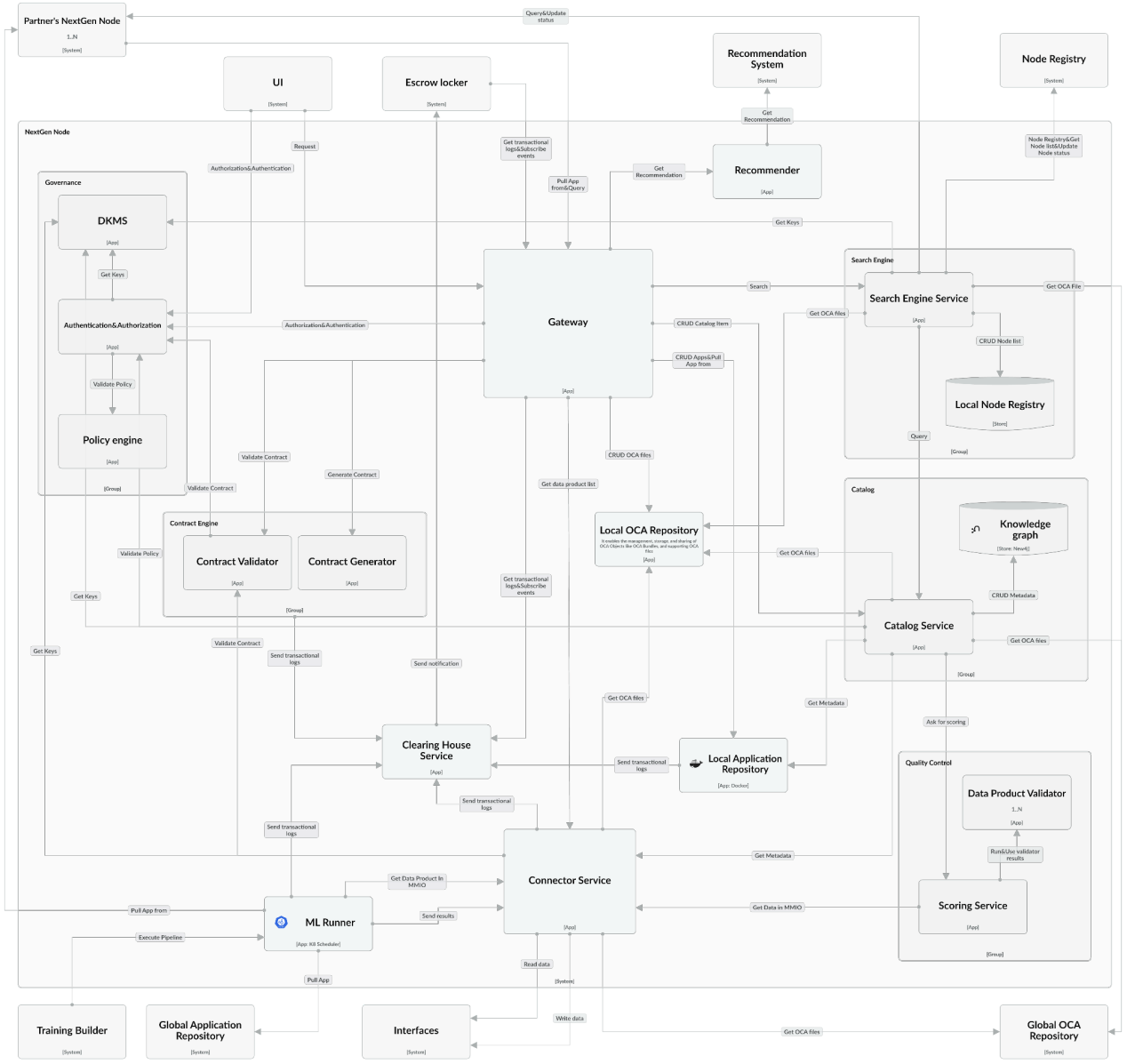

Figure 10: NextGen Node

4.5.1 Gateway

The Gateway serves as the central access point, handling requests from external services to the NextGen Node, verifying authorization and authentication through the Authentication & Authorization service, and proxying requests to downstream internal services.

This service can be deployed as a standalone service inside other systems, such as the Global Application Repository, the Node Registry, etc.

Related systems and services

External:

- UI
- Another NextGen Node
- Escrow locker

Internal:

- Authentication & Authorization
- Recommender
- Local Application Repository
- Search Engine Service
- Catalog Service
- Local OCA Repository
- Connector Service
- Clearing House Service
- Contract Generator
- Contract Validator

Interfaces

The Gateway provides a single API entry point, which supports HTTP(S) requests following internal API specifications for accessing the internal services. Each request to the Gateway must include authorization credentials, verified against the Authentication & Authorization service.

4.5.2 Governance

The Governance is a group of services designed to handle user authentication, authorization, and policy validation. It plays a critical role in ensuring secure and policy-compliant interactions across the system.

The group consists of:

- Policy Engine
- Authentication & Authorization
- DKMS

Policy Engine

Policy engine is a service that checks if users comply with the rules defined by Policies. It ensures that user actions align with the established policies.

Authentication & Authorization

Authentication & Authorization is a service that verifies user identity and determines their access rights to system resources. It integrates with DKMS to manage keys and ensure security, and integrates with Policy engine to validate Policies.

DKMS

DKMS (Decentralized Key Management System) DKMS is pivotal in the realm of digital security, offering a foundational component for constructing robust Digital Public Infrastructures (DPI). By leveraging DKMS, organizations can build DPIs that effectively address complex governance challenges within digital environments. This is crucial as DPIs rely on secure, scalable, and flexible key management systems to maintain the integrity and trustworthiness of any digital interactions in a given ecosystem.

Within NextGen, the DKMS infrastructure establishes a decentralized identity management system necessary for secured data provenance and its governance. With such a system, actors of NextGen are able to create cross-governance data exchange without losing control over their own realm.

DKMS consists of two high-level components:

1. Propagation Infrastructure: allows entities to propagate their key state into the network and is designated by the controller himself.
2. Duplicity Infrastructure: provides a mechanism for duplicity detection (fraudulent actor) which is designated by the governance of the ecosystem.

As a supporting infrastructure DKMS provides the means to manage the cryptographic keys in a decentralized ecosystem. WP1 implements in NextGen a new type of key management system through a set of agents:

1. Witnesses that propagate key state throughout the ecosystem
2. Watchers observe Witnesses and ensure consistency across different Witnesses.

For reference on how to bootstrap DKMS network please refer to official documentation[22]

Use cases

- Users rely on the Authentication & Authorization service for login and access control in the UI.

[22] https://dkms.colossi.network/

- All user requests from the UI to the NextGen Node via the Gateway include credentials, which are verified by the Authentication & Authorization service.
- During contract validation, the Contract Validator interacts with the Authentication & Authorization service to authenticate Contract participants and check the Policy.

This service can be deployed as a standalone service inside other systems, such as the Global Application Repository, Escrow Locker, Training Builder, etc.

Related components and services

- UI
- Gateway
- Contract Validator
- Catalog Service
- Connector Service

Interfaces

The Governance provides APIs which support HTTP(S) requests following internal API specifications.

DKMS exposes HTTP REST APIs. These APIs allow external applications to interact with DKMS services. For example, developers can use the DKMS API to retrieve the latest state of the public key of any digital identifier within the ecosystem. The interface is exposed through the REST API.
DKMS-SDK is a Software Development Kit which allows users to interact with that infrastructure without need to dive into the intricacies of the underlying protocol (KERI-Key Events Receipts Infrastructure) protocol which underpins those interactions. DKMS-SDK allows users to: 1. Create identifiers 2. Manage their key state (e.g. rotate key) for their safety 3. Discover other identifiers public key 4. Sign & Verify any data payload

4.5.3 Recommender

The Recommender service responsible for getting recommendations from the Recommendation System to the user, enabling personalized recommendation delivery. It is part of the NextGen Node and ensures seamless interaction with the Recommendation System.

Responsibilities

- Request Handling
  Processes user requests for recommendations.
  Validates and interprets the request to ensure proper interaction with the Recommendation System.
- Personalized Recommendation Delivery
  Fetches tailored suggestions from the Recommendation System based on the user's history, preferences, and the trained model.
  Ensures the recommendations are relevant and actionable for the user.

Activation Criteria
The Recommender component depends on the recommendation system which will be activated only after collecting sufficient user interaction history. This ensures that the model is adequately trained and capable of delivering accurate and valuable recommendations.

Related systems and services
- Recommendation system
- Gateway

Interfaces
The Recommender provides APIs which support HTTP(S) requests following internal API specifications.

4.5.4 ML Runner

The ML Runner is a service designed to execute both federated learning and machine learning Pipelines inside the NextGen Node.
The Training Builder delegates the execution of the Pipeline to the ML Runner, utilizing a Kubernetes Job object; this Job runs within the node where the execution occurs.
While the Training Builder operates as a centralized system, the actual execution takes place in a decentralized manner via the ML Runner within the node.

- ML Runner interacts with the Connector Service to retrieve and store the necessary data, utilizing MMIO as a wrapper to ensure data security.
- ML Runner communicates with the Global Application Repository and Partners' NextGen Nodes to pull the needed Applications.
- ML Runner communicates with the Clearing House Service to send transactional logs about pulling and using Applications.

Related systems and services
- Training builder
- Connector service
- Global Application Repository
- Partners' NextGen Node
- Clearing House Service

Interfaces
The ML Runner is deployed as a Kubernetes Job, triggered by job specifications, without direct API interfaces.

4.5.5 Catalog

The Catalog is a group of services that stores and provides Catalog Items (metadata of Data Products and Applications) from the Knowledge Graph.

The group consists of:

- Catalog service
- Knowledge Graph

Catalog Service

- It allows you to create, update, read, and delete Catalog Items through the Gateway. It communicates with the Knowledge Graph database to achieve this.
- During creation/updating the Catalog Item, the Catalog Service transforms the Catalog Items ontology into the catalog ontology using DKMS and Global/Local OCA Repository.
- It allows you to retrieve Catalog Items based on search queries from the Search Engine.
- It notifies the Quality Control Service to start scoring.

Knowledge Graph

It is a graph database designed to store Catalog Items according to a specific ontology.

Related systems and services

- Gateway
- Search Engine
- Quality control
- Connector Service
- Local Application Repository
- Local OCA Repository
- Global OCA Repository
- DKMS
- Authentication & Authorization

Interfaces

The Catalog provides APIs which support HTTP(S) requests following internal API specifications for accessing the Catalog, and it worked as a wrapper to deal with the knowledge graph database.

4.5.6 Quality Control

Quality Control is a group of services responsible for verifying that a Data Product complies with baseline requirements for quality, completeness, and documentation.

Its main functions include:

- **Schema Validation**
  Confirming that the data product's schema matches the expected structure, including data types, column names, and relationships between fields.
- **Metadata Verification**
  Ensuring that essential metadata (e.g., data source, timestamp, version) is present and correctly documented to provide traceability and context.
- **Documentation Check**
  Assessing the completeness and clarity of the documentation, ensuring it covers data sources, transformation steps, limitations, and intended usage.

The group consists of:

- Scoring Service
- Data Product Validator

The Scoring Service receives metadata from the Catalog Service and a Data Product from the Connector Service. The obtained Data Product is passed to one or more Data Product Validator services, which return scores for the Data Product. Then, the Scoring Service aggregates the scores from different Data Product Validator services and sends the result back to the Catalog Service. The Catalog Service stores the score for the Data Product in the knowledge graph.

Related systems and services

- Catalog Service
- Connector Service

Interfaces

Internal services (Scoring Service, Data Product Validator) and external services (Catalog Service, Connector Service) communicate with each other using HTTP(S) requests following internal API specifications.

4.5.7 Connector Service

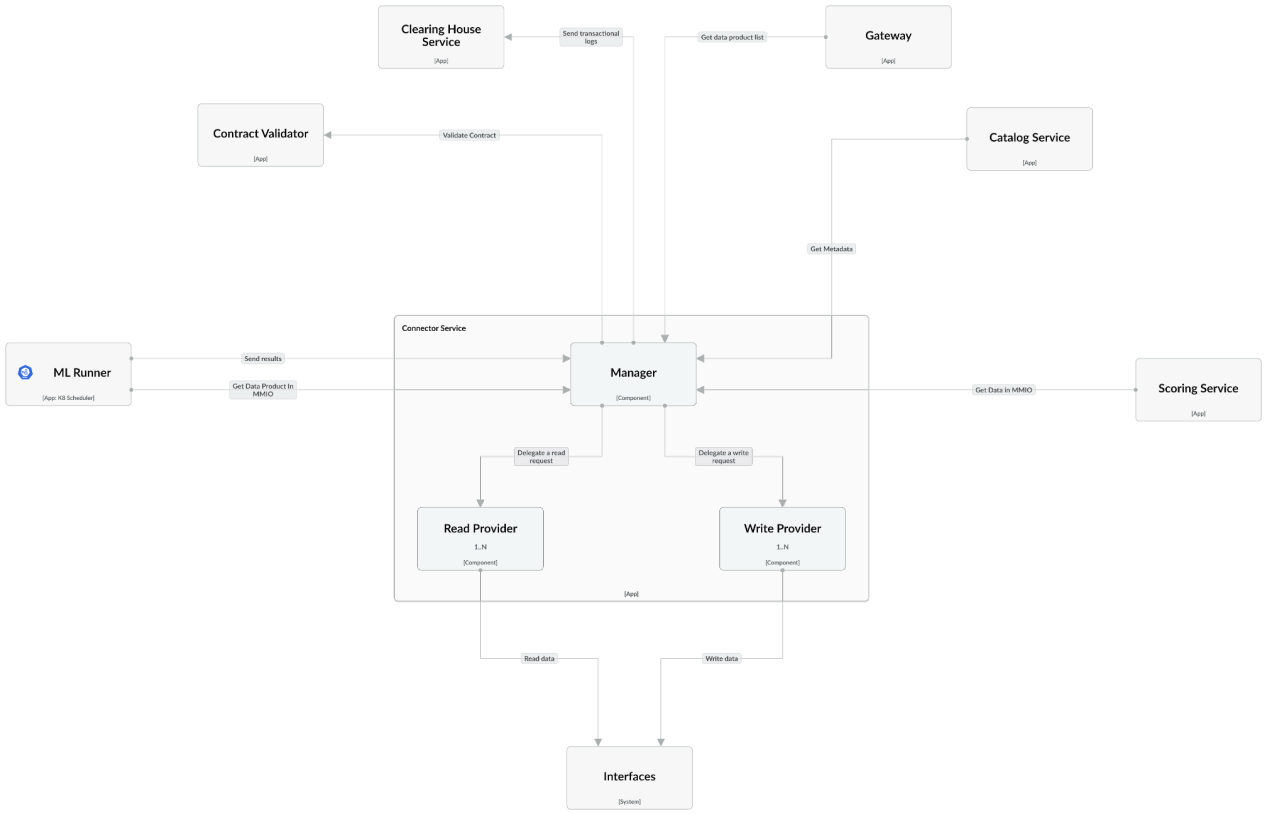

Figure 11: Connector Service

The Connector Service is a service that provides a unified interface for NextGen Node internal services to interact with the Data Access Layer.
The Connector Service acts as a bridge between the Catalog Service and ML Runner on one side, and Interfaces on the other. It manages data fetching and sending results, verifies Contracts with the Contract Validator service, and logs Transactions in the Clearing House Service.

The Connector Service allows retrieving a list of available Data Products through the Gateway, provides Data Products as MMIO, and delivers MMIO to the Scoring Service for quality assessment and to the ML Runner for federated learning. To create MMIO, it uses the Local OCA Repository, Global OCA Repository, and DKMS.

The Connector Service consists of a Manager and Providers. The Manager receives requests and decides which Provider should handle them. Providers process requests by interacting with Interfaces to read and write data.

There are two Providers in the NextGen project:

- The Read Provider, which reads data from the Data Layer through Interfaces.
- The Write Provider, which writes data to the Data Layer through Interfaces.

Related systems and services

- Gateway
- Contract Validator
- Clearing House Service
- Catalog Service
- Scoring Service
- ML Runner
- Interfaces
- Local OCA Repository
- Global OCA Repository
- DKMS

Interfaces

- Connector Service provides a single point for reading and writing data to the Data Access Layer, which supports HTTP(S) requests following internal API specifications. Each request to the Connector Service must include authorization credentials, verified against the Authentication & Authorization service.
  The Connector Service ensures that data is transmitted wrapped in MMIO.
- Connector Service interacts with Interfaces using HTTP(S) requests following S3 API specifications.

4.5.8 Local Application Repository

The Local Application Repository stores and distributes Applications, which can be used in a Pipeline.

The Local Application Repository provides information about stored Applications. This allows the Catalog to create, store, and share Application metadata according to the Policy. Application Consumers can find Applications in the Catalog and, if they meet the Policy conditions, use them in their Pipelines.

During execution of the Pipeline on a Data Provider's node, the ML Runner pulls Applications from the appropriate application repository.

All requests to the Local Application Repository go through the Gateway service. The request includes a Contract as credentials. The Gateway requests the Contract validation from the Contract Validator. If the Contract is valid, the Gateway proxies the request to the Local Application Repository, which stores and serves. If the Contract is invalid, the component rejects the request.

The Local Application Repository sends transactional logs about application usage to the Clearing House Service.

Related systems and services

- Gateway
- Clearing House Service
- Catalog Service

Interfaces

The Local Application Repository receives and processes HTTP(S) API requests according to the documentation at https://distribution.github.io/distribution/spec/api/, which is the standard protocol for interacting with a Docker repository.
Each HTTP(S) request must include an Authorization header with the credentials.

4.5.9 Contract Engine

The Contract Engine is a group of services that enables the creation and validation of Contracts for the use of Catalog Items (metadata of Data Products and Applications).

The group consists of:

- Contract Generator
- Contract Validator

Contract Generator

The Contract Generator is a service that creates Contracts for selected Catalog Items. Its main task is to generate a Contract that describes the terms and rules for transactions between participants.
The Contract is machine-readable and can be validated using cryptographic methods.

In the baseline scenario, the user selects Catalog Items and creates Contracts (each Catalog Item requires a separate Contract). To do this, the user sends a request from the UI through the Gateway to the Contract Generator. The Contract Generator creates the Contract(s), registers them with the Clearing House Service, and returns them to the user. The user can then use the Contracts to create a Pipeline in the Training Builder to have access to Data Products or Applications.

Contract Validator

The Contract Validator is a service responsible for validating the integrity, authenticity, and compliance of Contracts. It ensures that Contracts are cryptographically signed, have not been tampered with, and adhere to the policies and rules.

The validation process involves checking the digital signatures of the Contract to verify the identity of the parties involved and confirming that the Contract's terms are consistent with the selected Catalog Items and the system's policies.

The Contract Validator communicates with the Policy Engine through the Authentication & Authorization service to verify compliance with policies and rules, and communicates with the Clearing House Service to verify the registration of the Contract.

Validation is triggered whenever a Data Consumer or Application Consumer accesses Data Products or Applications through the Connector Service, ensuring that only valid and authorized Contracts are used for these operations. If any discrepancies are found during validation, the Contract Validator returns an error, specifying the issue for correction.

Related systems and services

- Authentication & Authorization
- Gateway
- Connector Service
- Clearing House Service

Interfaces

The service provides a REST API that follows internal specification.

4.5.10 Search Engine

The Search Engine is a group of services, designed to process search queries and aggregate results from decentralized Catalogs.

The group consists of:

- Search Engine Service
- Local Node Registry

Local Node Registry

The Search Engine stores a list of distributed Catalogs in the Local Node Registry. During onboarding, a new participant receives the initial list of distributed Catalogs from a centralized bootstrap service (Node Registry). After that, the participant gets updates to the list directly using a peer-to-peer (p2p) protocol.

Search Engine Service

A user sends a search query from the UI through the Gateway to the Search Engine Service, which executes the query across the local Catalog and distributed Catalogs from the Local Node Registry. The Search Engine Service transforms the query ontology into the catalog ontology using DKMS and Global/Local OCA Repository, then processes the search. Finally, it aggregates the search results and converts them into the required ontology and format.

Related systems and services

- Gateway
- Catalog Service
- Node Registry

- DKMS
- Local OCA Repository
- Global OCA Repository
- Partner NextGen Nodes

Interfaces
The Search Engine provides a REST API that follows internal specification.

4.5.11 Clearing House

The Clearing House is a distributed service in NextGen Node designed to collect and store transaction logs related to creating Contract, executing Contract, and accessing Data Products and Applications. It ensures transparency and enables future audits.

Transaction logs in the Clearing House cannot be changed, and their integrity is guaranteed using cryptographic methods.

During the Contract process, the ML Runner, Connector Service, Local Application Repository, and Contract Engine send their transaction logs to the Clearing House.

Escrow Locker subscribes to notifications from the Clearing House Service through the Gateway. When a new transactional log is registered, the Clearing House Service sends a notification to its subscribers. Escrow Locker can also read transactional logs from the Clearing House Service through the Gateway.

This service can be deployed as a standalone service inside other systems, such as the Global Application Repository, Escrow Locker, Training Builder, etc.

Related systems and services
- Gateway
- Escrow Locker
- Local Application Repository
- Connector service
- Contract Engine
- ML Runner

Interfaces
The Clearing House provides a REST API that follows internal specification.

4.5.12 Local OCA Repository

The OCA Repository[23] is a key concept of the Overlays Capture Architecture used to ensure data integrity in an ecosystem. This component is at the heart of the OCA Ecosystem, a set of tools designed to facilitate the integration of OCA into software solutions. The OCA Repository enables the management, storage, and sharing of OCA Objects like OCA Bundles , Capture Bases, and Overlays. Furthermore, it comes with pre-baked support for OCAFiles.
The Local OCA Repository is a service that plays a key role in the Overlays Capture Architecture used to ensure data integrity in an ecosystem. This component is at the heart of the OCA Ecosystem, a set of tools designed to facilitate the integration of OCA into software solutions. The OCA Repository enables the management, storage, and sharing of OCA Objects like OCA Bundles, Capture Bases, and Overlays. Furthermore, it comes with pre-baked support for OCAFiles.

Related systems and services

- Gateway
- Search Engine Service
- Catalog Service
- Connector Service

Interfaces

OCA-SDK Software Development Kit is available in a form of libraries and binary which can be integrated in the code or used as standalone application serving functions to manage identifiers.

The Overlays Capture architecture (OCA) Software Development Kit is a Rust library with bindings to other languages providing programmable interfaces to interact with OCA artefacts like OCAFILE and OCA Bundles or specific overlays and capture base. The SDK is in development at this stage so for up to date details how to use the library please refer to the official documentation on the git repository[24].

[23] https://github.com/THCLab/oca-repository-rs
[24] https://github.com/THCLab/oca-sdk-rs

4.6 Scenarios

4.6.1 Onboarding a New Partner to NextGen Process

NextGen Node installing
A new Partner needs to set up their instance of a NextGen Node. The NextGen Node serves as the entry point to the distributed NextGen data space.
The Partner implements the necessary Interfaces so the NextGen Node can read and write data to the Data Layer. The Partner deploys the interface's services in his own infrastructure.

Generating cryptographic keys
The Partner generates cryptographic keys independently using a standard algorithm supported by DKMS, leveraging a Cryptographic Provider for secure key management and operations. These keys are necessary for cryptographic operations such as creating digital signatures and encryption.
The private key is stored securely by the Partner within the cryptographic provider, ensuring enhanced protection, and all cryptographic operations are performed locally by the Partner.

Generating AID
Using the cryptographic keys, the Partner generates an AID (Autonomous Identifier) through the DKMS. This AID serves as a unique decentralized identifier used to identify and interact with other NextGen Nodes. The DKMS is provided by the Governance component within the Node, ensuring secure and decentralized management of identifiers.

Setting Up Access for Staff
The Partner creates an access Policy for the NextGen Node for the staff. This Policy is based on the Roles defined in the NextGen project.

Retrieving the Initial List of NextGen Nodes (from Node Registry)
To interact with other NextGen Nodes, the partner's NextGen Node retrieves an initial list of available NextGen Nodes from the Node Registry service. This list is used for decentralized search.
To get the node list, the partner's NextGen Node communicates to the Node Registry service via its REST API, retrieves the list, and saves it in the Local Node Registry.
After obtaining the initial NextGen Nodes list, updates are handled automatically through a decentralized P2P protocol.

Registering a Catalog for distributed search (in Node Registry)
To make the partner's NextGen Node discoverable in the initial list of available nodes, the NextGen Node must be registered in the Node Registry. Once registered, the node will appear in the list provided to new Partners by the Node Registry.
To register, the partner's NextGen Node sends a request using the Node Registry API, including the AID, node address (host, port, protocol), etc. The request must be signed with the Partner's private key to confirm authenticity.

Next steps
Next, the Partner can create Catalog Items, perform decentralized searches through nodes of other participants, start federated learning processes, and more.

4.6.2 Creating, Updating, and Deleting a Catalog Item Processes

The User (Data Creator, Application Creator) can manage their local Catalog by adding, updating, or deleting Catalog Items for both Data Products and Applications.
Interactions with the NextGen Node are handled through a UI, which communicates via a Gateway, sending HTTP(S) requests to the Catalog Service, Connector Service, and Local Application Repository. These operations adhere to an internal API specification.
Authentication and authorization are mandatory.

4.6.2.1 Creating a Catalog Item

Data Product

1. The user uploads a MMIO file with Domain-Specific metadata. The MMIO file is stored inside the NextGen Node.
2. The User requests a list of Data Products. The Connector Service retrieves the Data Product list from the Data Layer. Only metadata of the Data Products is returned to the User.
3. The User selects the MMIO file, and fills out a Catalog Item form by selecting a Data Product and providing Catalog Item Metadata, such as name, version, access rules, licence, etc.
4. The User submits the form data to the Catalog Service.
5. The Catalog Service creates a new subgraph in the Knowledge Graph.

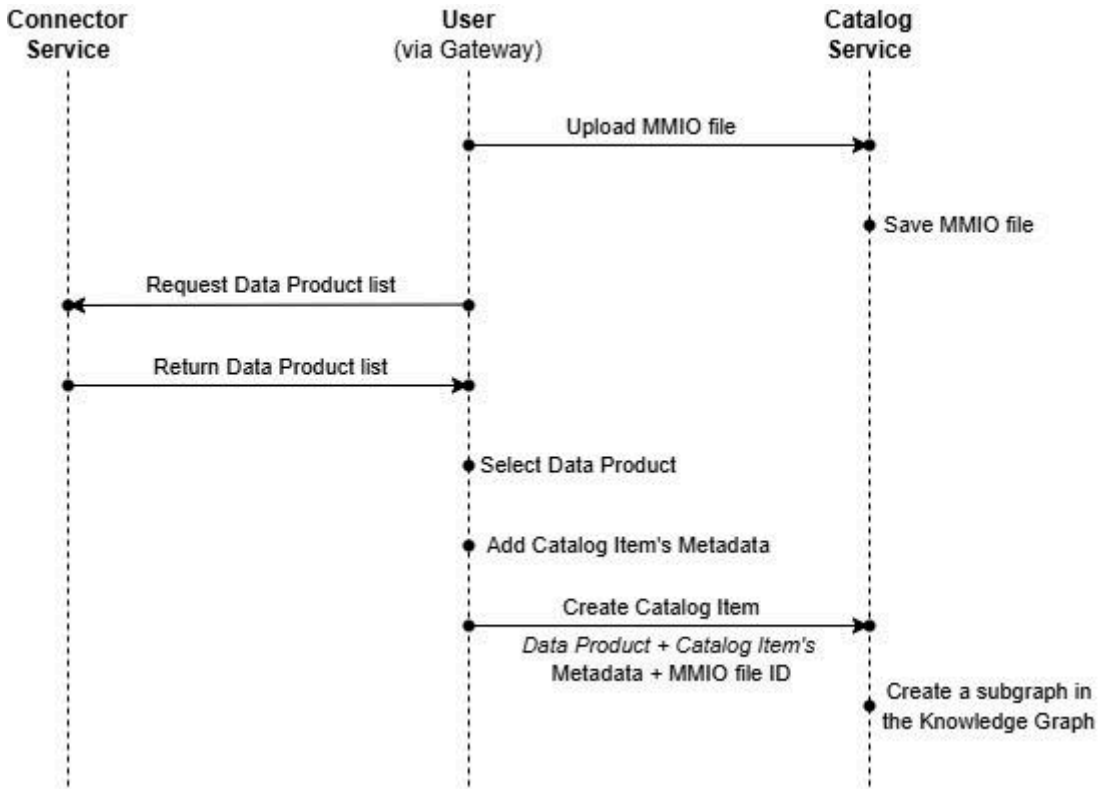

Figure 12: Creating a Data Product

Application

1. The User requests a list of Applications. The Application Repository retrieves the list of Applications. Only metadata of the Applications is returned to the User.
2. The User fills out a Catalog Item form by selecting an Application and providing Catalog Item Metadata, such as name, version, access rules, license, etc.
3. The User submits the form data to the Catalog Service.
4. The Catalog Service creates a new subgraph in the Knowledge Graph.

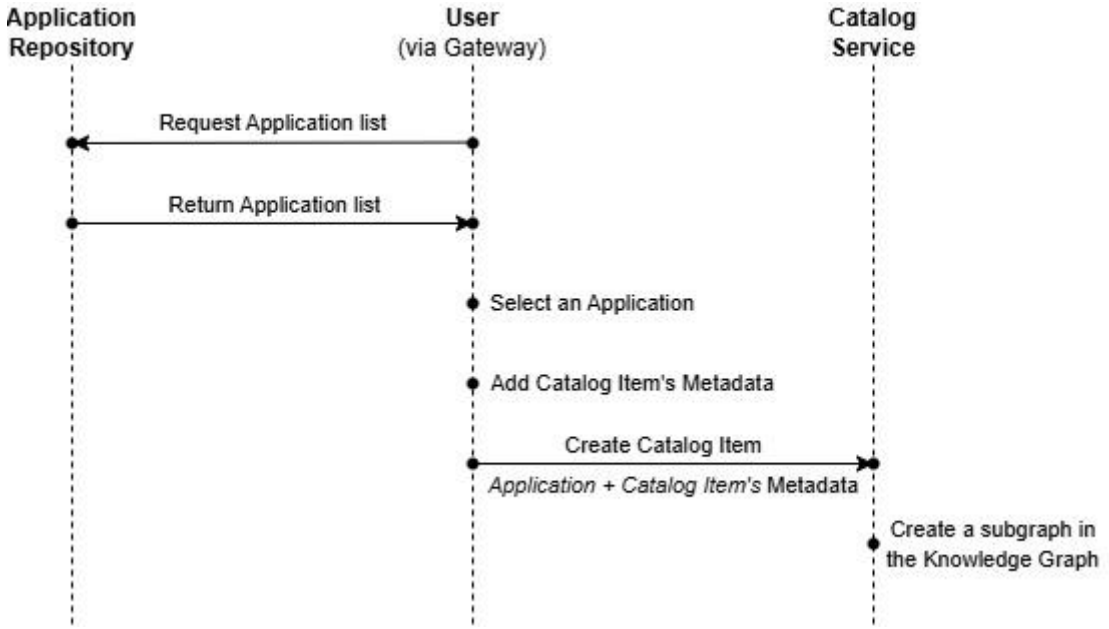

Figure 13: Creating an Application

4.6.2.2 Updating a Catalog Item

To update a Catalog Item, the User requests the Catalog Item's metadata from the Catalog Service. After making the necessary changes, the updated data is sent back to the Catalog Service. The Catalog Service then updates the corresponding subgraph of the Catalog Item in the Knowledge Graph.

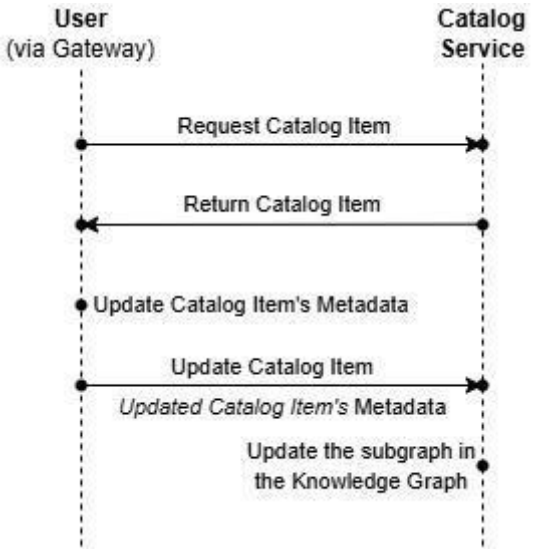

Figure 14: Updating a Catalog Item

4.6.2.3 Deleting a Catalog Item

To delete a Catalog Item, the User first retrieves the item's metadata from the Catalog Service. Then, the User sends a DELETE request to the Catalog Service, providing the ID of the Catalog Item to be removed.

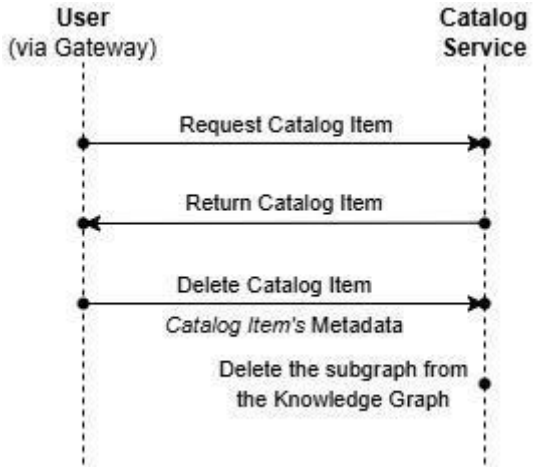

Figure 15: Deleting a Catalog Item

4.6.3 Search for Data Products and Applications Processes

The process for searching Data Products and Applications is the same because they are both represented by Catalog Items in the Catalog.

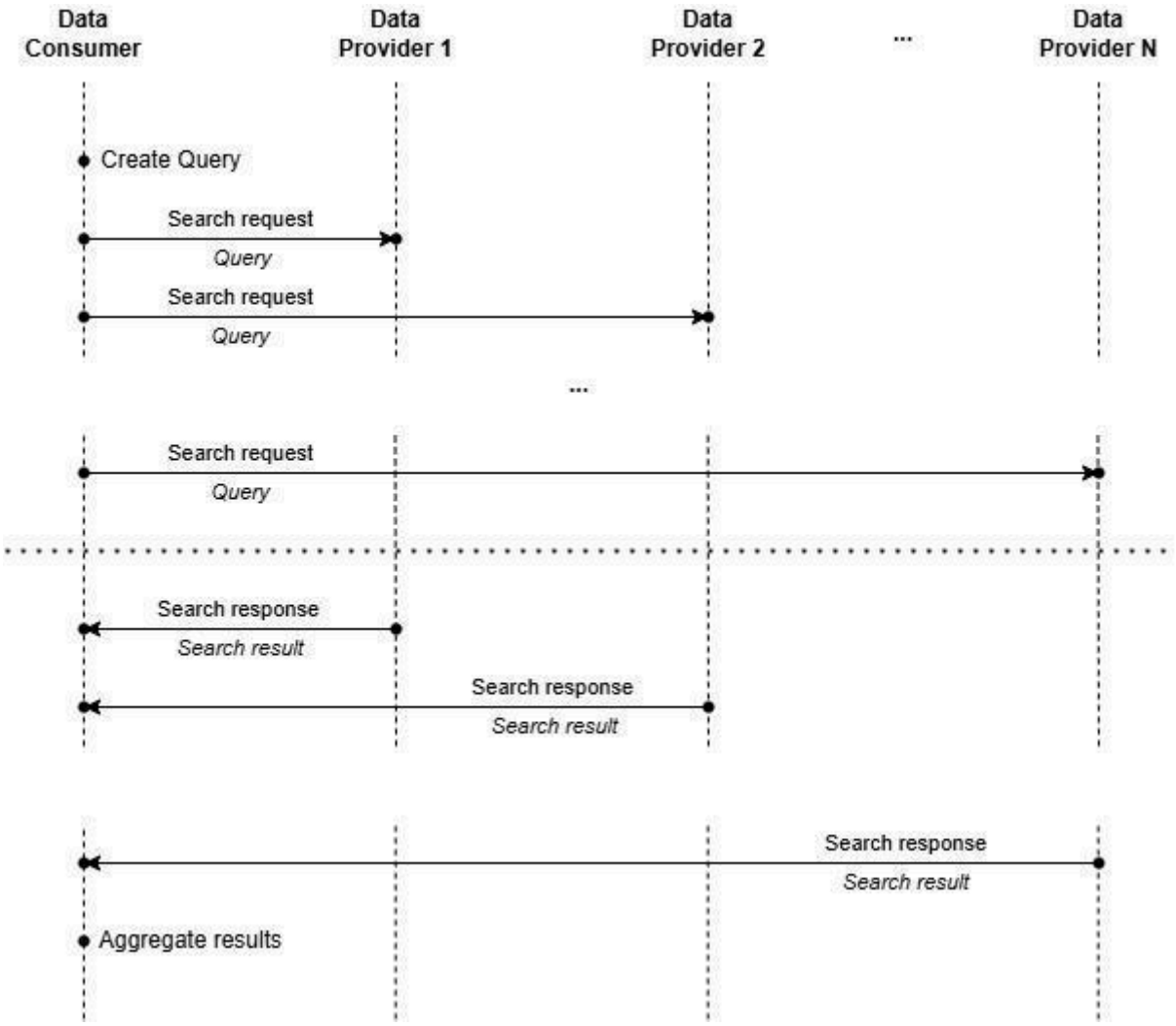

Figure 16: Search for Data Products and Applications

1. A Data Consumer creates a search query.
2. The Data Consumer sends the search query to each Data Provider node.
   The list of nodes is stored locally on each NextGen Node in the Local Node Registry.
3. Data Providers process the queries and return Catalog Items.
   Each Data Provider only returns Catalog Items that the Data Consumer is allowed to access based on the Catalog Item Policies.
4. The Data Consumer collects search results from the Data Providers and combines them into a single search result.

4.6.4 Federated ML Process

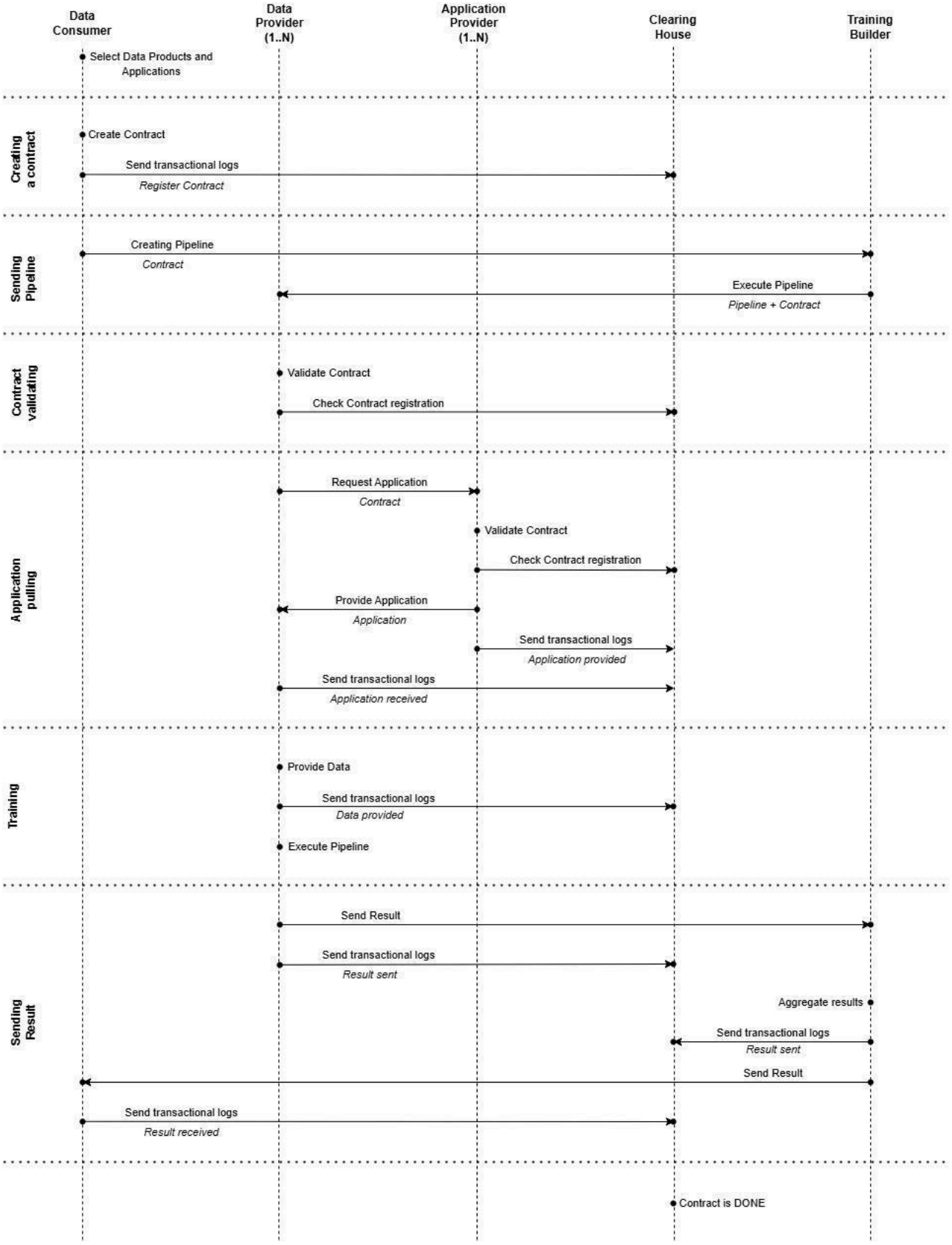

Figure 17: Federated ML process

1. The Data Consumer selects one or more Data Products and Applications after searching Data Products and Applications.
   Metadata for Data Products and Applications include Policies that specify who can use them, when, and under what conditions.
2. The Data Consumer creates a Contract and registers it in the Clearing House.
   A separate Contract is created for each Data Product or Application, which includes the metadata and Policy and all of these Contracts will be aggregated in a single Pipeline Contract.
3. The Data Consumer builds a Pipeline in the Training Builder using the selected Data Products and Applications following the Pipeline Contract.
4. The Training Builder sends the Pipeline to all Data Provider nodes offering the chosen Data Products with the Pipeline Contract
5. The Data Provider checks the Contract for compliance with the Data Consumer's attributes and confirms its registration in the Clearing House.
   If the Contract fails validation, the Data Provider notifies the Clearing House and cancels the Contract.
6. If the Contract is valid and all checks are passed, the Data Provider executes the Pipeline by delivering the Data Product, pulling the required Applications from the Application Provider, and training the model.
   The Data Provider registers the events of data delivery and application usage in the Clearing House.
7. The Data Provider sends the Pipeline results to the Training Builder and registers this event in the Clearing House.
8. The Training Builder collects the data from the Data Providers, registers this event in the Clearing House, and aggregates the results. Depending on the federated learning (FL) algorithm, steps 4-8 may repeat.
9. The Training Builder delivers the final result to the Data Consumer and registers this event in the Clearing House.
10. The FL process is complete. The Contract is fulfilled.

4.6.5 Escrow Locker Process

Escrow Locker is a general-purpose service that, within NextGen, will not be integrated with a specific billing system.

This sequence diagram is an extension of the Federated ML process. Blocks highlighted in yellow indicate additional steps to support the Escrow Locker process.

The specific steps highlighted in red within NextGen will be for demonstration purposes only and will not involve real money.

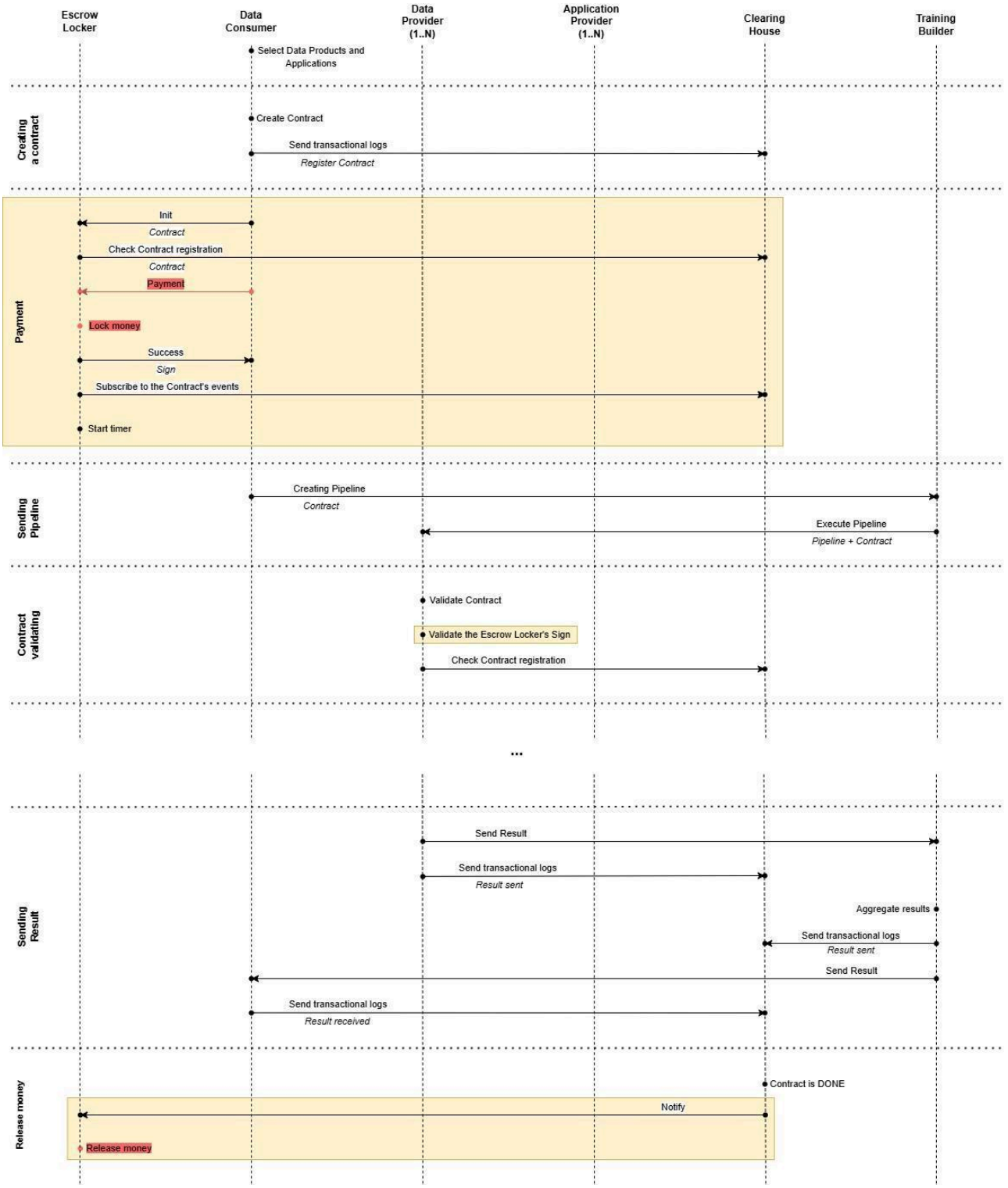

Figure 18: Escrow Locker process

1. After a Contract is created, The data consumer will initiate the Contract with the Escrow Locker and the Escrow Locker will store it in the Clearing House (In this step if the Data Product requires payments, the Data Consumer will initiate the Contract with the Escrow Locker).

2. The Escrow Locker checks if the Contract is registered in the Clearing House. If the Contract is not registered, the Escrow Locker rejects the Contract and stops the process.
3. If the Contract is registered in the Clearing House the process will continue (In this step if the Data Consumer initiates the payment process. The money is locked in the Escrow Locker account).
4. The Escrow Locker provides the Data Consumer with a Contract that has all the needed information signed.
5. The Escrow Locker subscribes to Clearing House events related to the Contract and starts a timer.
6. During the Contract validation phase, the Data Provider or Application Provider verifies the digital signature on the Contract. If the signature is invalid, they reject the Contract.
7. If the signature is valid and all other checks are passed, the Pipeline process begins.
8. When the Contract is completed, the Clearing House sends notifications to its subscribers. The Escrow Locker receives the notification (In this step escrow locker releases money to the accounts of the Data Providers or Application Providers).
9. If the Escrow Locker does not receive the notification before the timer expires the Contract is cancelled (In this step the locked money is returned to the Data Consumer's account).

4.6.6 MMIO Processes[25]

4.6.6.1 Overview

A Data Catalog stores metadata about Data Products and Applications. Each Data Product or Application is a Catalog Item in the Catalog. A Catalog Item contains Metadata structured using Catalog Ontology, which includes information from the Connector Service or Local Application Repository, and additional metadata added during its creation. Additionally, the Metadata of a Catalog Item may include information following any Domain-Specific Ontology.

Data Products

Data Products are stored in the Data Layer. They are accessed via a Connector Service, which retrieves them from the Data Layer through Interfaces.

The data of Data Products in the Data Layer can follow any ontology and format. Rules for inferring and transforming (mapping) their ontology and format into other ontologies must be stored in a local or global OCA Repository.

Applications

Applications are stored in the Local or Global Application Repository. Application Repository provides information about Applications following the Docker Registry HTTP API V2 specification[26].

Application data are transferred as-is without any ontology transformation.

[25] See also D1.3 "Data Discovery Functionality (Part 1)" December 2024

[26] https://distribution.github.io/distribution/spec/api/

4.6.6.2 Creating a Catalog Item

4.6.6.2.1 Data Product

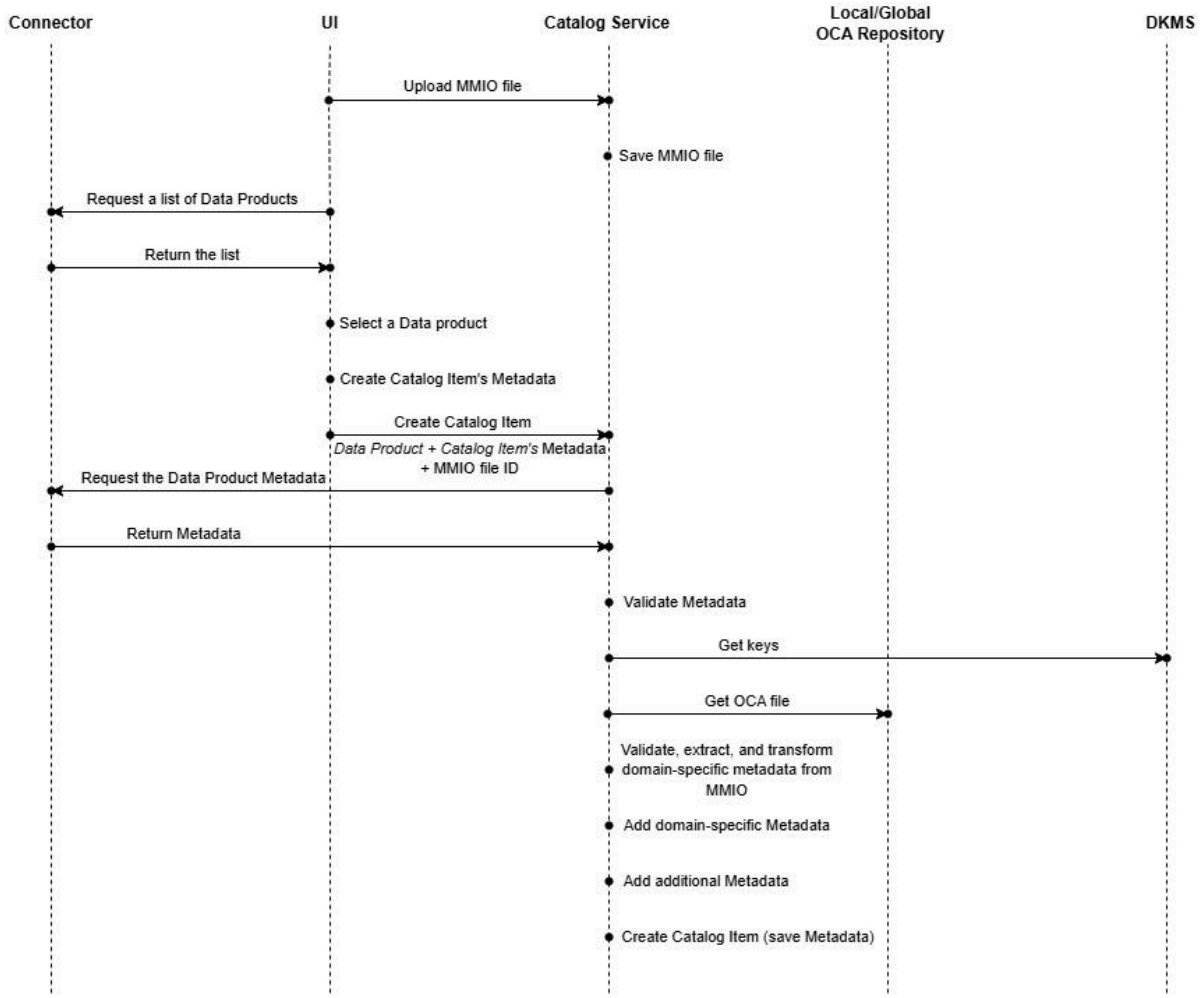

Figure 19: Creating a Catalog Item for a Data Product

1. The user, via the UI, uploads a MMIO file with Domain-Specific metadata. The MMIO file stores inside the NextGen Node.
2. The user, via the UI, requests a list of Data Products from the Connector Service.
3. The user selects a Data Product for which he wants to create a Catalog Item.
4. The User selects the MMIO file, and creates a Metadata of Catalog Item following the Catalog Ontology. Some of this metadata is obtained from the Connector Service in step 1. Other metadata is added by the user through the UI. The user can also add additional metadata in the UI using the Domain-Specific Ontology.
5. The user sends the Metadata to the Catalog Service.
6. The Catalog Service validates Metadata and matches it with the Metadata from the Connector Service.
7. The Catalog Service requires necessary keys from the DKMS, and OCA files from Local/Global OCA Repository. Then, the Catalog Service uses this data to validate,

extract, and transform domain-specific metadata from the MMIO file to the catalog format and ontology.
8. The Catalog Service adds domain-specific metadata.
9. The Catalog Service adds additional metadata (e.g., timestamps).
10. The Catalog Service creates a Catalog Item (saves the Metadata in the Knowledge Graph).

4.6.6.2.2 Application

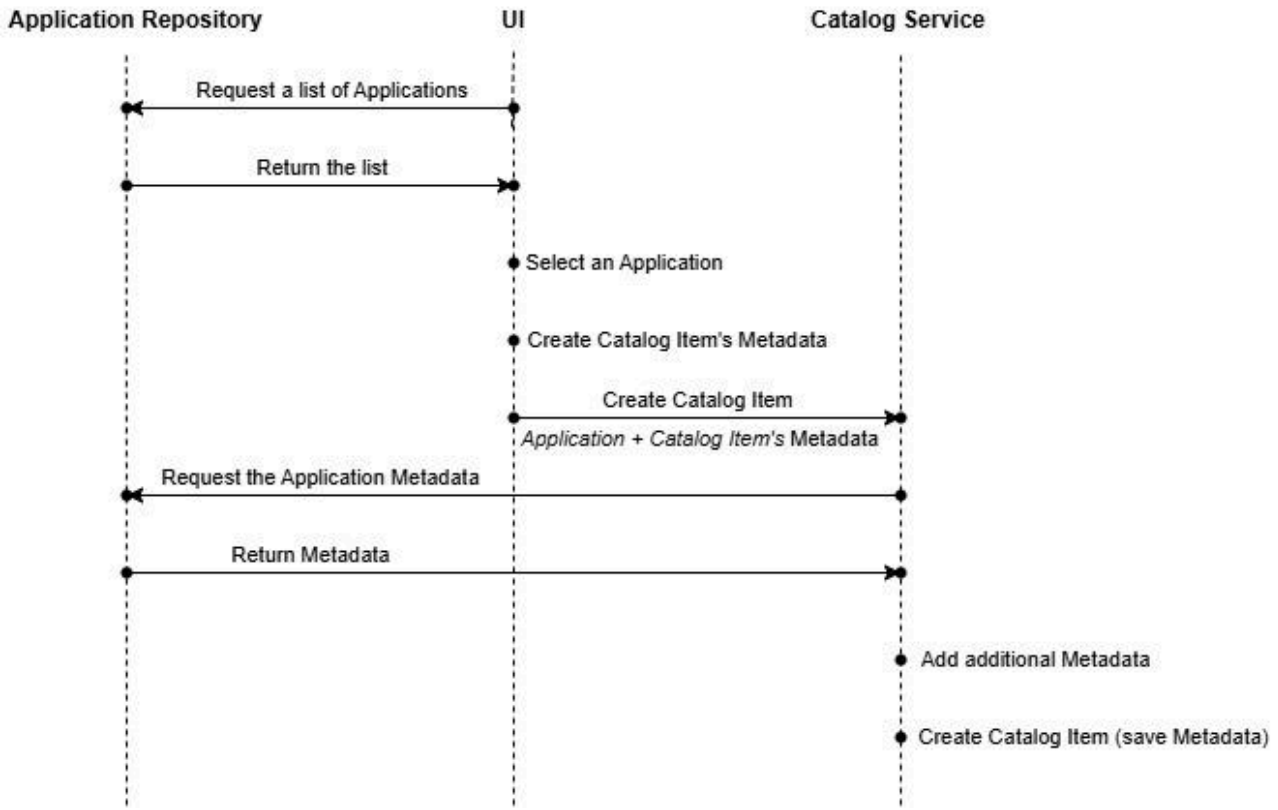

Figure 20: Creating a Catalog Item for an Application

1. The user, via the UI, requests a list of Applications from the Application Repository.
2. The user selects an Application for which he wants to create a Catalog Item.
3. The User creates a Metadata of Catalog Item following the Catalog Ontology. Some of this metadata is obtained from the Application Repository in step 1. Other metadata is added by the user through the UI. The user can also add additional metadata in the UI using the Domain-Specific Ontology.
4. The user sends the Metadata to the Catalog Service.
5. The Catalog Service validates Metadata and matches it with the Metadata from the Application Repository.
6. The Catalog Service adds additional metadata (e.g., timestamps).
7. The Catalog Service creates a Catalog Item (saves the Metadata in the Knowledge Graph).

4.6.6.3 Searching Process

4.6.6.3.1 Local Searching

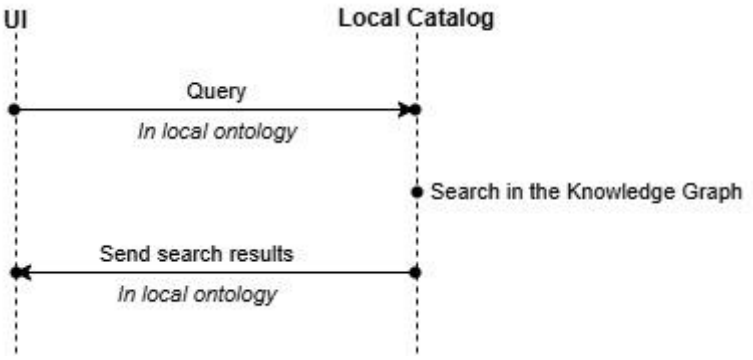

Figure 21: Local searching

4.6.6.3.2 Distributed Searching

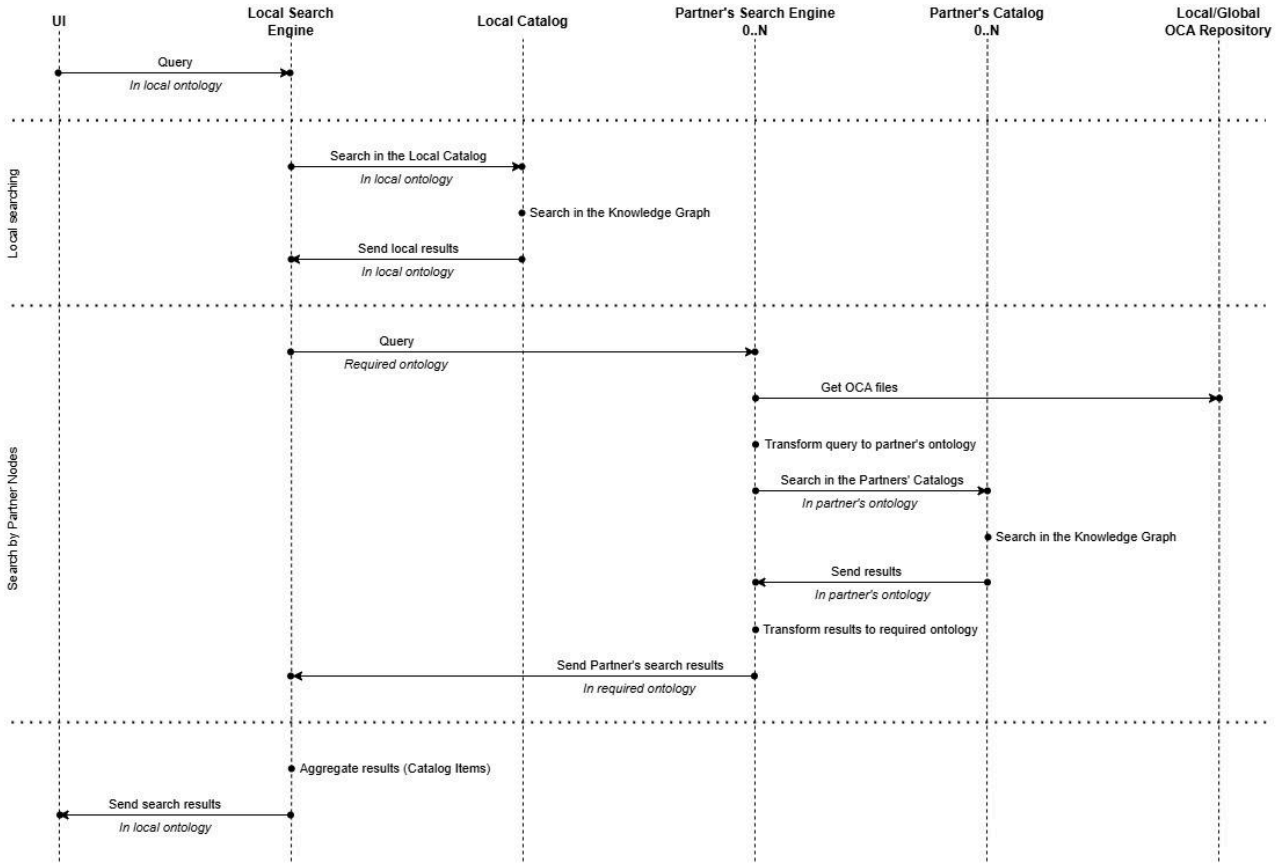

Figure 22: Distributed searching

1. The Local Search Engine searches in the Local Catalog.
    a. The UI and NextGen Nodes make queries using the local ontology.
2. The Local Search Engine searches across the Partner's Catalogs.

   a. The Local Search Engine sends a query to the Partner's Search Engines. Query includes a query and a result ontology.
   b. Partner's Search Engine translates the query into their catalog ontology using inferring and mapping rules from the OCA Repository.
   c. Partner search engine queries the Knowledge Graphs through its own Catalog Service.
   d. Partner search engine transforms the search result to the required ontology.
3. The Local Search Engine collects results from the Local Catalog and Partner's Search Engines, aggregates them, and returns them to the requester.

For domain-specific queries (e.g., using Domain-Specific Ontology attributes), only Catalog Items using the same ontology will be found.

How External Data Spaces Can Search for Data in NextGen (Interoperability)

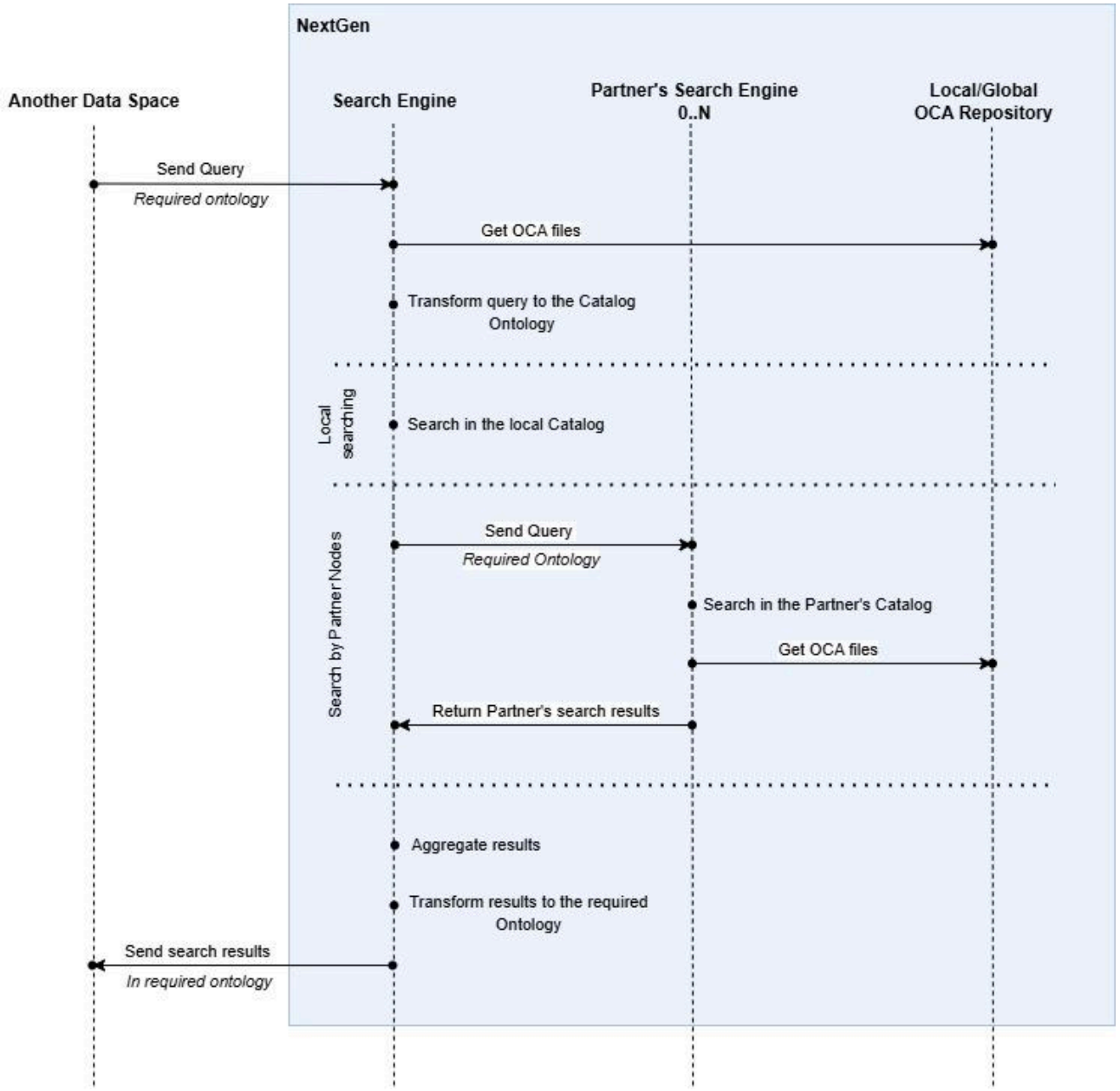

Figure 23: How external Data Spaces can search for data in NextGen

Third-party Data Spaces can use any ontology. It specifies their desired result ontology and format (including MMIO) in the query, ensuring interoperability. Inferring and mapping rules for these ontologies must be stored in the OCA Repository.

Example

If EHDS uses DCAT-AP and NextGen stores data in DCAT-3, EHDS can query the data in DCAT-AP format. The data will be processed in DCAT-3 and converted back to DCAT-AP before returning the results to EHDS. All ontology transformations between DCAT-AP and DCAT-3 will be handled by MMIO.

4.6.7 Federated Learning Process

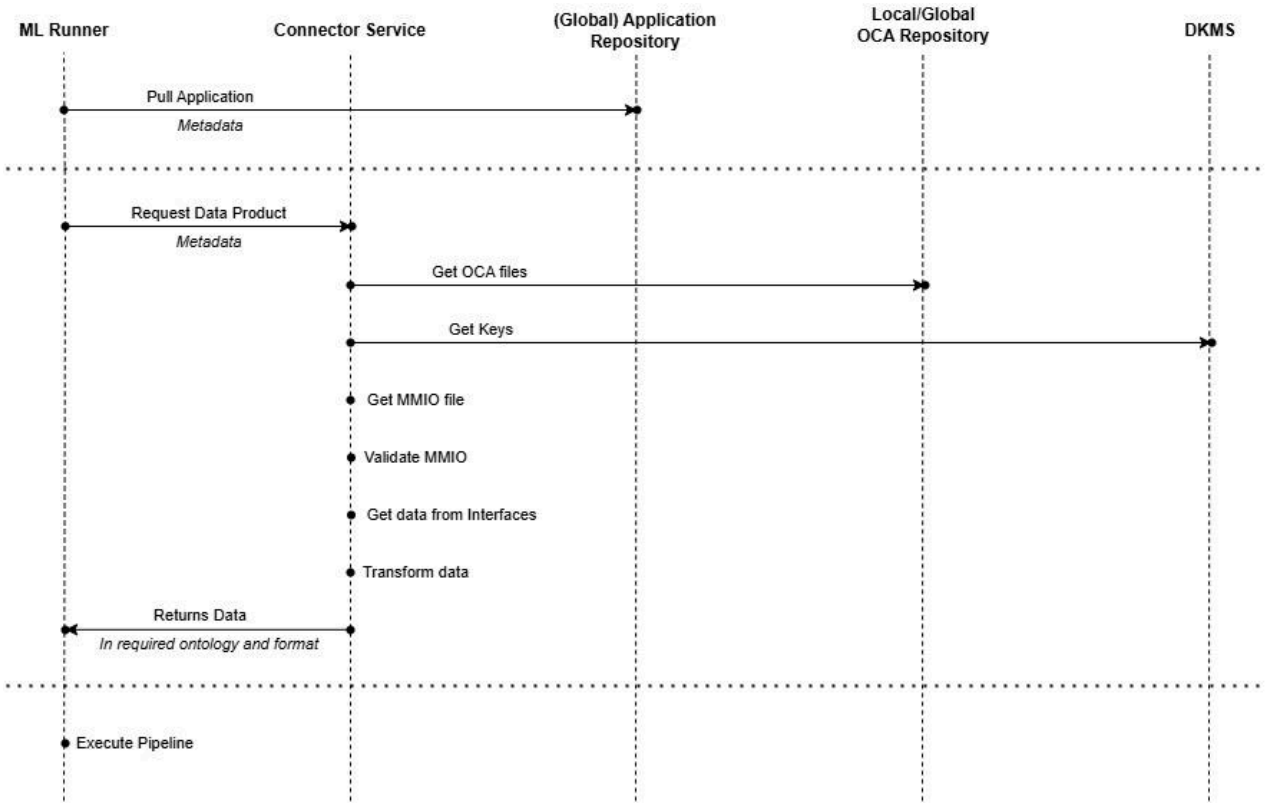

Figure 24: MMIO in Federated Learning process

1. During a Federated Learning Pipeline, the ML Runner requests:
    a. Data products from the Connector Service.
    b. Applications from the Application Repository.
2. The Connector Service retrieves the MMIO file and validates it using keys from the DKMS.
3. The Connector retrieves actual data from the Data Layer through Interfaces, and transforms the data into the required ontology and format using the OCA files from the OCA Repository.
4. The Connector sends the actual data to the ML Runner.
5. The ML Runner executes the Pipeline.

Applications are pulled directly from the Application Repository without MMIO validation and transformation.

4.7 Implementation (D2.2)

As part of Deliverable D2.2, HIRO developed a working demonstration to showcase the current progress and implementation of its core services within the demonstration of MVP Technology-1.

In this demonstration of MVP Technology-1, HIRO implemented three foundational services, leveraging the Multimodal Integration Object (MMIO) framework to validate the functional capabilities of the Pathfinder platform. The deployment was designed to simulate a **decentralized environment**, reflecting the real-world architecture envisioned by the project.

The services were deployed across multiple virtual nodes running within a Kubernetes infrastructure. Each node represents an independent data provider and hosts local instances of key services, including:

- **Search Service** – enables decentralized discovery and querying of catalog items across federated nodes.
- **Catalog Service** – manages the creation, update, and deletion of catalog items.
- **Knowledge Graph** – stores and structures catalog items based on predefined ontologies.

This distributed setup validates the modularity, scalability, and interoperability of the Pathfinder architecture, and demonstrates the foundational capabilities required for federated data discovery.

Table 10: Data Space implemented Services & Repo links

Service	Description	Status	Github Repository link
Search Service	-Receives the search query from the user and executes the query across distributed Catalogs. - Aggregates the search results from different distributed catalogs and sends them back to the user.	In progress	link[27]
Catalog Service	-Allows you to create, update, read, and delete Catalog Items. -Allows you to retrieve Catalog Items based on search	In progress	link[28]

[27] https://github.com/HIRO-MicroDataCenters-BV/ds-search-service
[28] https://github.com/HIRO-MicroDataCenters-BV/ds-catalog

	queries. -UnpacksMMIOs and gets the semantic metadata to be indexed in the knowledge graph.		
Knowledge Graph	-Graph database designed to store Catalog Items according to a specific ontology	In progress	link[29]

4.8 Interoperability

Interoperability is a key part of the NextGen platform, allowing it to connect different data ecosystems. By following widely used standards and supporting both semantic and protocol-level interoperability, NextGen makes it easy for users to access, share, and analyse data across countries, industries, and platforms. This interoperability improves the platform's usefulness and supports European initiatives like the EHDS and GDI

Governance and Auditability
The platform supports verifiable credentials and decentralized authentication to provide secure and controlled access to data.
NextGen's MMIOs allow data to be shared and processed across different systems while keeping governance and access control in place.

Metadata Interoperability
NextGen supports DCAT (and its subsequent developments), which are widely used metadata standards in European data spaces. NextGen follows in particular the current development of Health-DCat for interoperability with EU health related initiatives like EHDS and GDI.
Metadata in NextGen is designed to work with these standards, making it easy to find and share datasets across different platforms.

Data Formats and Standards
NextGen works with many data formats, including OMOP, FHIR, and CDISC, which are widely used.
The platform leverages MMIOs to map (i.e.transform) and harmonize data into user-defined formats, ensuring compatibility with different systems and workflows.

Semantic Interoperability
NextGen uses a decentralized semantic architecture that allows different ontologies and vocabularies to work together, so data from different sources can be understood and used consistently.
The platform leverages the MMIO and OCA Repository for mapping and aligning ontologies, making it easier to connect and interpret data across different fields.

[29] https://github.com/HIRO-MicroDataCenters-BV/Neo4j-With-Neosemantics

Protocols for Secure Data Exchange
NextGen supports secure communication using HTTPS and uses JSON-LD for structured data exchange, ensuring data is shared safely and in a standardized way.

4.9 Testing

As part of the NextGen project, several testing approaches will be used to ensure the reliability and resilience of the system:

- **Unit tests** – used to verify individual components (functions, methods, classes) in isolation. These tests run quickly and serve as the first line of defense against regression issues.
- **Integration tests** – designed to test interactions between system modules, including database operations, external APIs, and other services. These tests help ensure that components work correctly together.
- **End-to-End tests** – focus on testing the entire system from the user's perspective. These will be used in a limited scope for key user scenarios.

5 Summary and Next Steps

The NextGen platform showcases the creation of a Data-Oriented Architecture and a federated Data Space, offering a secure, scalable, and interoperable foundation that supports collaboration among various stakeholders.

This blueprint outlines the architecture to how federated learning, decentralized identity management, and semantic interoperability can be leveraged to facilitate efficient and privacy-preserving data exchange. The implementation of Multimodal Integration Objects and policy-driven access control further ensure that data remains protected while maximizing its utility for scientific discovery and research.

Next Steps
As the project progresses, the following key areas will be the focus of subsequent development and validation:

- **Implementation and Testing**
  The architectural components described in this deliverable will be implemented and tested in controlled environments to validate their performance, scalability, and security.
- **Pilot Deployment and User Validation**
  Further pilots maybe deployed to evaluate platform functionality in other settings. Feedback from stakeholders will be used to refine functionalities.
- **Sustainability and Long-Term Strategy**
  A roadmap for the long-term sustainability of NextGen will be developed, ensuring that the platform continues to provide value beyond the project's duration through partnerships, funding models, and community engagement.

By following these next steps, NextGen aims to contribute to the development of standards for secure and collaborative data-driven research.

6 Glossary of Technical Terms

Term	Description
Data Harmonisation	Data harmonisation refers to aligning and integrating data from diverse sources to ensure consistency, compatibility, and coherence. It involves resolving differences in data formats, structures, semantics, and standards to create a unified and standardised data representation. Data harmonisation aims to enable seamless data integration, exchange, and analysis across systems, platforms, and domains. Organisations can achieve better data consistency, accuracy, and reliability by harmonising data, facilitating effective decision-making, and promoting interoperability among different data sources and stakeholders.
Data Interoperability	Data interoperability refers to the ability of different systems, applications, or platforms to exchange and use data seamlessly without restrictions or compatibility issues. It involves harmonising data formats, structures, and protocols to ensure that data can be shared, accessed, and understood by various systems or entities. Data interoperability enables the efficient and effective exchange of information, promotes collaboration, and facilitates the integration of data from diverse sources, allowing for improved data-driven decision-making and insights.
Data sovereignty	Data sovereignty means that data generated within a country's borders is governed by that nation's laws and regulatory frameworks. Data sovereignty materialise in NextGen through the control over the nodes remaining with the research organisations.
Data Catalogue Vocabulary (DCAT, DCAT-AP, Health-DCAT)	An RDF vocabulary designed to facilitate interoperability between data catalogues on the web, aiding metadata discoverability and alignment. DCAT has multiple profiles tailored for specific domains. In NextGen we will consider for example DCAT-AP, a profile for EU applications and Health-DCAT.
Multimodal Integration Object (MMIO)	A component responsible to achieve the higher level of interoperability and data portability in a data ecosystem. With respect to the information discovery mechanism, the MMIO is an envelope that encapsulates any data (i.e. any modality) with its semantic and additional relevant information (i.e. purpose, consent, rules etc). Integrated into the NextGen architecture, the MMIO enables (i) conversion of the underlying constellation of multimodal formats (ii) application of site-specific governance and regulatory requirements, and (iii) embedded authentication, audit, and integrity functionality.
OCA	Overlays Capture Architecture (OCA) refers to a core technology framework that enables the harmonization and integration of data and semantics across various data models and representation formats. OCA allows for the capture and overlaying of additional semantic information onto existing data, enhancing its discoverability, interoperability, and understanding. It provides a flexible and extensible approach to enriching data with contextual meaning, enabling better data integration, searchability, and analysis in distributed data ecosystems. OCA plays a crucial role in facilitating semantic interoperability and promoting the effective utilization of data across different systems and domains.
OCA Bundle	Within an OCA architecture, the OCA bundle is the structured schema representing a data object. The integrity of an OCA Bundle can be verified independently. As such the OCA bundle is a core element of the OCA Ecosystem enabling data harmonisation.

OCAFILE	The OCA File is a concept that enables the expression of OCA Objects using a Domain Specific Language (DSL).
Schema	A dataset schema refers to the structured representation and organisation of the elements, attributes, relationships, and constraints that define the format and structure of a dataset. It serves as a blueprint or template that outlines the expected data types, properties, and their interconnections within a dataset. The dataset schema provides a framework for ensuring data consistency, interoperability, and understanding by defining the rules and guidelines for data entry, storage, and exchange. It helps to standardize the structure and semantics of the data, facilitating effective data management, integration, and analysis across different systems and applications
Data Space	A Common European Data Space is a secure, privacy-preserving digital ecosystem underpinned by interoperable infrastructures and governance frameworks, enabling organisations and individuals to pool, share, process and reuse data across specific sectors under fair, transparent and non-discriminatory access.
Data-oriented architecture (DOA)	A design approach that prioritizes data organisation, accessibility, and processing efficiency, structuring systems around data flow and transformations rather than the control flow of processes.
Decentralized architecture	A system structure/architecture where data, processes, and control are distributed across multiple nodes or devices, rather than being concentrated in a central server or location.
Data Layer Data-oriented architecture	A part of DOA responsible for managing, storing, and providing access to data across the system. It acts as the foundation where data is organized, governed, and made available for use by various services and applications..
Interface	A definition of the protocols that will be used between the systems or services to interact with each other programmatically.
Infrastructure	A foundational system that provides the necessary services, resources, and technologies to support the development, deployment, and operation of software applications.
Data Product	A Data Asset that is refined and structured in such a way that it is ready for use by end users or applications. It has been processed, organized, and presented with a specific purpose or use case in mind.
Data Asset	A Data Asset is any collection of data that has intrinsic or potential value to an organization due to its ability to generate insights, support decision-making, improve processes, or create economic benefit.
Application	An algorithm packaged that is used to build machine learning (ML) or federated computing Pipelines.
Ontology	A structured way of organizing and defining concepts and relationships within a specific area of knowledge. It acts like a map, showing how different ideas connect and providing a shared vocabulary to help people and systems understand and work with data consistently.

Policy	A set of rules that define who can access and use a Data Product or Application. It ensures compliance with security, governance, and regulatory requirements.
Contract	An agreement between a Data Consumer and a Data Provider (for Data Products) or an Application Provider (for Applications) that follows specific Policies. It defines the conditions under which a Data Product or Application can be accessed, shared, or used in processes like federated learning.
Data Processing	Data processing refers to the systematic collection, organization, transformation, and analysis of raw data into meaningful and usable information. It involves a series of operations—such as data entry, validation, sorting, aggregation, and analysis—often carried out using computational methods to support decision-making, knowledge generation, and automation across various domains.
Model Training	The process of adjusting a model's parameters on a dataset to minimize error and learn underlying patterns.
Pipeline	In the context of Machine Learning, a structured sequence of data processing and model training steps that automate the workflow to a predictive model. It streamlines and organizes the various stages involved in building a machine learning model, ensuring that each stage receives inputs in a standard format and produces outputs required for the next stage.
Pipeline Contract	A contract created by the Data Consumer that has the other Contracts for catalog items and user signature to be used and validated by the Data Product and Application Provider.
NextGen Node	A core system of the NextGen project, serving as a node in a decentralized network. It manages the Data Holder's Data Catalog, ensures secure access to Data Products and Applications for federated learning, and provides data format compatibility, data usage control, and data usage history tracking. Each partner operates their own instance of the NextGen Node.
Marketplace	A centralized logical system that is shared between the nodes which includes essential systems such as a Training Builder, Escrow Locker, Global Application Repository, Global OCA Repository, Node Registry, and Recommendation System. The functionality of the Marketplace is defined by the set of systems it includes.
Data Layer Architecture	A system that stores the actual data at a NextGen Node that is stored in our Data Holder's physical infrastructure.
Global OCA Repository	A system for storing and sharing data objects like OCA Bundles, Capture Bases, and Overlays.
Interfaces System	A system that acts as a bridge between the Connector Service and the Data Layer. It should be implemented and deployed by the Data Holder within their own infrastructure, providing interfaces for reading and writing data to the Data Layer. This ensures that the Data Holder retains full control over the access that the NextGen Node has to their data.
Training Builder	A system designed to construct both federated learning and machine learning Pipelines. The Training Builder delegates the execution of the Pipeline to the ML Runner, utilizing a Kubernetes Job object; this Job runs within the node where the execution occurs.

Escrow Locker	A system responsible for locking and unlocking the money after ensuring that governance requirements for both the Data Provider and Data Consumer are in place. It receives the payment from the Data Consumer and holds it until the process is completed. Once validation is confirmed, it transfers the payment to the Data Provider and notifies both parties that the process was successfully completed. In NextGen scope it will be responsible for auditing the transactional logs stored in the clearing house in different nodes without any money transactions or locking or unlocking of money.
Global Application Repository	A centralized storage for Applications, which are used by the Training Builder and shared across all nodes. It provides these Applications during Pipeline execution in the ML Runner.
Node Registry	A bootstrap node for new participants in NextGen. During onboarding, a new participant receives the initial list of distributed NextGen Nodes from the Node Registry.
Recommendation System	A system designed to provide personalized suggestions to users by leveraging machine learning models. This system plays a key role in enhancing user experience by offering recommendations for Applications and Data Products that align with user needs and interests.
User	An active participant in the system who interacts with the platform through a user interface (UI). This component provides access to system features based on roles.
User Interface (UI)	A system that handles specific functions or displays certain content. It connects the front-end design with back-end services, allowing users to interact with the system efficiently.
Monitoring & Logging System	A system that collects telemetry metrics and logs, stores them, and visualizes data on dashboards. It gathers information from all NextGen systems, services, and the underlying infrastructure. Additionally, it manages alert configurations.
Gateway	A service that serves as the central access point, handling requests from external services to the NextGen Node, verifying authorization and authentication through the Authentication & Authorization service, and proxying requests to downstream internal services.
Governance	A group of services designed to handle user authentication, authorization, and policy validation. It plays a critical role in ensuring secure and policy-compliant interactions across the system.
Policy Engine	A service that checks if users comply with the rules or policies defined by the node owner. It ensures that user actions align with the established policies.
Authentication & Authorization	A service that verifies user identity and determines their access rights to system resources. It integrates with DKMS to manage keys and ensure security, and integrates with Policy engine to validate Policies.
DKMS (Decentralized Key Management System)	A system for securely creating, storing, and managing cryptographic keys in a decentralized way. It ensures secure identity verification, encryption, and data integrity without relying on a central authority.

Recommender	A service that is responsible for getting recommendations from the Recommendation System to the user, enabling personalized recommendation delivery.
ML Runner	A service designed to execute both federated learning and machine learning Pipelines inside the NextGen Node.
Catalog	A group of services that stores and provides Catalog Items (metadata of Data Products and Applications) from the Knowledge Graph.
Catalog Item	A metadata of the Data Product or Application that will be stored in the Catalog, and it will be shared to the Partners. It will be following an Ontology.
Catalog Service	A system that manages Catalog Items. It lets you create, update, read, and delete items through the Gateway.
Knowledge Graph	A graph database designed to store Catalog Items according to a specific ontology.
Quality Control	A group of services responsible for verifying that a Data Product complies with baseline requirements for quality, completeness, and documentation.
Scoring Service	A service that evaluates Data Products by collecting metadata from the Catalog Service and data from the Connector Service. It sends the data to one or more Data Product Validator services, which return scores. Then, it combines the scores and sends the final result to the Catalog Service for storage.
Data Product Validator	A service that checks the quality of a Data Product and assigns it a score. It helps the Scoring Service measure how good the data is.
Connector Service	A service that provides a unified interface for NextGen Node internal services to interact with the Data Access Layer.
Local Application Repository	A service that stores and distributes Applications, which can be used in a Pipeline. The applications are stored locally on the node of the application owner.
Contract Engine	A group of services that enables the creation and validation of Contracts for the use of Catalog Items (metadata of Data Products and Applications).
Contract Generator	A service that creates Contracts for selected Catalog Items. Its main task is to generate a Contract that describes the terms and rules for transactions between participants. The Contract is machine-readable and can be validated using cryptographic methods.
Contract Validator	A service that is responsible for validating the integrity, authenticity, and compliance of Contracts. It ensures that Contracts are cryptographically signed, have not been tampered with, and adhere to the policies and rules.
Clearing House	A distributed service in NextGen Node designed to collect and store transaction logs related to creating Contract, executing Contract, and accessing Data Products and Applications. It ensures transparency and enables future audits.
Data Holder	Any natural or legal person, entity or body who has the right or obligation to make certain electronic health data available for secondary use.

Genomic Data Infrastructure (GDI)	The project is enabling access to genomic and related phenotypic and clinical data across Europe. It is doing this by establishing a federated, sustainable and secure infrastructure to access the data. It builds on the outputs of the Beyond 1 Million Genomes (B1MG) project and is realising the ambition of the 1+Million Genomes (1+MG) initiative.

7 References

[1]	C4Model, "The C4 model for visualising software architecture", 2018. [Online]. Available: https://c4model.com/.
[2]	W3C, "Data Catalog Vocabulary (DCAT) - Version 3", 2024. [Online]. Available: https://www.w3.org/TR/vocab-dcat-3/
[3]	SEMIC, "DCAT-AP 3.0", 2024. [Online]. Available: https://semiceu.github.io/DCAT-AP/releases/3.0.0/.
[4]	EHDS, "European Health Data Space", 2022. [Online]. Available: https://health.ec.europa.eu/ehealth-digital-health-and-care/european-health-data-space_en.
[5]	GDI, "Providing access to genomic data to improve research, policy making and healthcare across Europe", 2022. [Online]. Available: https://gdi.onemilliongenomes.eu/.
[6]	GDI, "Genomic Data Infrastructure", 2023. [Online]. Available: https://github.com/GenomicDataInfrastructure.
[7]	GDI, "GDI User Portal - Docs", 2024. [Online]. Available: https://genomicdatainfrastructure.github.io/gdi-userportal-docs/.
[8]	IDSA, "The International Data Spaces Association", 2017. [Online]. Available: https://internationaldataspaces.org/
[9]	M. Fowler, "Data Monolith to Mesh", 2019. [Online]. Available: https://martinfowler.com/articles/data-monolith-to-mesh.html.
[10]	[2] OpenContainers, "Distribution Specification v1.0.1", 2021. [Online]. Available: https://github.com/opencontainers/distribution-spec/blob/v1.0.1/spec.md.
[11]	Christian Cabrera, Andrei Paleyes, Pierre Thodoroff, Neil D. Lawrence, "Real-world Machine Learning Systems: A survey from a Data-Oriented Architecture Perspective", 2023. [Online]. Available: https://www.researchgate.net/publication/368392884_Real-world_Machine_Learning_Systems_A_survey_from_a_Data-Oriented_Architecture_Perspective.
[12]	Rajive Joshi, "Data-Oriented Architecture: A Loosely-Coupled Real-Time SOA", 2007. [Online]. Available: https://community.rti.com/sites/default/files/archive/Data-Oriented_Architecture.pdf.
[13]	Andrey Ganyushkin, "Data-Oriented Architecture", 2023. [Online]. Available: https://full-stack.blog/blog/data_oriented_architecture/.
[14]	Eyas's Blog, "Data-Oriented Architecture", 2020. [Online]. Available: https://blog.eyas.sh/2020/03/data-oriented-architecture/.
[15]	MLatcl, "Data-Oriented Architectures for AI-Based Systems", n.d. [Online]. Available: https://mlatcl.github.io/projects/data-oriented-architectures-for-ai-based-systems.html.
[16]	DataMesh Manager, "What is a Data Product?", n.d. [Online]. Available: https://www.datamesh-manager.com/learn/what-is-a-data-product.
[17]	Xavier Gumara Rigol, "Data as a Product vs. Data Products: What Are the Differences?", 2021. [Online]. Available:

	https://towardsdatascience.com/data-as-a-product-vs-data-products-what-are-the-differences-b43ddbb0f123.
[18]	K2View, "What is a Data Product?", 2024. [Online]. Available: https://www.k2view.com/what-is-a-data-product/.
[19]	K2View, "What is Data Mesh?", 2025. [Online]. Available: https://www.k2view.com/what-is-data-mesh/.
[20]	Qlik, "Data Products", n.d. [Online]. Available: https://www.qlik.com/us/data-management/data-products.
[21]	GetRightData, "Data Products 101: What is a Data Product?", 2024. [Online]. Available: https://www.getrightdata.com/resources/data-products-101-what-is-a-data-product.
[22]	Micah Horner, Product Marketing Manager, TimeXtender, "The Ultimate Guide to Data Products", 2023. [Online]. Available: https://www.timextender.com/blog/product-technology/the-ultimate-guide-to-data-products.
[23]	Luzmo, "Data Products", 2024. [Online]. Available: https://www.luzmo.com/blog/data-products.
[24]	OneData, "What is a Data Product?", n.d. [Online]. Available: https://onedata.ai/what-is-a-data-product/#:~:text=Data products turn raw data,business users and data experts.
[25]	GetRightData, "Getting Started with Data Products Series – Part Two", 2023. [Online]. Available: https://www.getrightdata.com/blog/getting-started-with-data-products-series-part-two.
[26]	CNCF Distribution, "Distribution API Specification", n.d. [Online]. Available: https://distribution.github.io/distribution/spec/api/.
[27]	THC Lab, "OCA Repository OpenAPI Specification", 2023. [Online]. Available: https://github.com/THCLab/oca-repository-rs/blob/main/openapi.yml.
[28]	Kubernetes, "Kubernetes Documentation", n.d. [Online]. Available: https://kubernetes.io/.
[29]	Docker, "Docker Documentation", n.d. [Online]. Available: https://docs.docker.com/.
[30]	JSON-LD Community, "JSON-LD", n.d. [Online]. Available: https://json-ld.org/.
[31]	Colossi Network, "Overlays Capture Architecture", n.d. [Online]. Available: https://oca.colossi.network/.
[32]	Colossi Network, "Decentralised Key Management System", n.d. [Online]. Available: https://dkms.colossi.network/.
[33]	KERI Project, "KERI: Key Event Receipt Infrastructure", n.d. [Online]. Available: https://keri.one/.
[34]	W3C, "ODRL Information Model", 2018. [Online]. Available: https://www.w3.org/TR/odrl-model/.
[35]	HCF, "Decentralized Knowledge Management System", 2023. [Online]. Available: https://hackmd.io/@hcf/HyUE6WVnn.
[36]	Neo4j, "Graph Database Platform", n.d. [Online]. Available: https://neo4j.com/.

[37]	Flower AI, “Flower AI Platform”, n.d. [Online]. Available: https://flower.ai/.