This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement Nº 101137423

D2.3 – Data, metadata models and platform architecture

Deliverable No.	D2.3	**Due Date**	30/11/2025
Description	Technical description of the Protect-Child architecture.		
Type	Report	**Dissemination Level**	CO
Work Package No.	WP2	**Work Package Title**	Codesign and multi-stakeholders' requirements
Version	2.0	**Status**	Final

Authors

Name and surname	Partner name	e-mail
Eugenio Gaeta	UPM	eugenio.gaeta@upm.es
Jaime Bachiller	UPM	jbachiller@lst.tfo.upm.es
Adolfo Viguera	UPM	aviguera@lst.tfo.upm.es
Maria Fernanda Cabrera	UPM	chiqui@lst.tfo.upm.es
Aya Mahboub	UPM	amahboub@lst.tfo.upm.es
Ruben Sanchez	UDEUSTO	ruben.sanchez@deusto.es

Document History

Version	Date	Changes	Authors
0.1	01/01/2025	Table of content	Eugenio Gaeta
0.2	01/03/2025	Zero trust and Federated requirements	Eugenio Gaeta
0.3	01/05/2025	Genomics requirements	Jaime Bachiller
0.4	01/08/2025	NLP requirements	Adolfo Viguera
0.5	01/09/2025	Data model requirements	Ruben Sanchez
0.9	01/11/2025	Architecture	All consortium
1.0	30/11/2025	Internal Review	Matteo Gabetta
2.0	18/12/2025	Final review and final version	Aya Mahboub

Key data

Keywords	Architecture, Zero trust, genomics, Protect-Child components, Protect-Child connectors
Lead Editor	Eugenio Gaeta
Internal Reviewer(s)	BIOMERIS

Abstract

This deliverable defines the data models, metadata structures, and architectural foundations of the PROTECT-CHILD ecosystem. It consolidates requirements from clinical, technical, legal, and ethical stakeholders and aligns them with the emerging European Health Data Space (EHDS), GA4GH standards, and the Genomic Data Infrastructure (GDI). The document provides a technical landscape analysis covering interoperability frameworks, secure computation paradigms, Zero-Trust infrastructures, and federated processing models. Based on these inputs, it specifies a unified architecture enabling secure data ingestion, harmonisation, governance, and distributed analytics across the TransplantChild ERN. D2.3 establishes the technical baseline for subsequent work packages and guides the implementation of the platform throughout the project lifecycle.

Statement of originality

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 About this deliverable

Deliverable D2.3 provides the comprehensive technical foundation for the PROTECT-CHILD data ecosystem, outlining the data structures, metadata models, architectural components, and interoperability mechanisms required to enable secure, privacy-preserving, and clinically meaningful secondary use of paediatric transplantation data. As a core outcome of WP2, this deliverable integrates the multi-stakeholder insights gathered through co-creation activities, the SPACE methodology, and the regulatory analysis of the European Health Data Space (EHDS), ensuring that the proposed architecture is aligned with emerging European standards and capable of supporting cross-border data sharing and federated computation.

The document begins by analysing the current technical landscape that influences the development of the PROTECT-CHILD platform, including established and emerging standards for health and genomic data interoperability (FHIR, OMOP, GA4GH schemas), as well as key initiatives such as ELIXIR, GDI, and the EHDS HealthDat@EU Pilot. It also examines the technologies and paradigms necessary for secure, distributed processing of sensitive health data, such as Secure Processing Environments (SPEs), container-based deployment, service meshes, Zero-Trust architectures, Secure Multi-Party Computation (SMPC), and federated learning. This analysis provides the rationale behind the architectural decisions adopted, ensuring coherence with European technical frameworks and compliance with GDPR and EHDS requirements.

Building on this foundation, the deliverable consolidates the functional and non-functional requirements elicited throughout WP2. These requirements reflect the needs of clinical partners for integrating genomic, clinical, and patient-reported data; the expectations of data managers and IT teams for harmonisation and efficient ingestion; the obligations imposed by legal and ethical frameworks governing paediatric and genomic data; and the operational constraints of a multi-centre, cross-border ecosystem. The requirements define what the system must achieve to support real-world clinical workflows, advanced federated analytics, and secure data governance across the TransplantChild ERN.

The architectural specification presented in D2.3 translates these requirements into a coherent design that defines how data flows, services, security controls, and components interact. It introduces the unified data and metadata models, the interoperability strategy across standards, the federated orchestration layer, and the Zero-Trust service mesh that collectively ensure secure and distributed data processing. The architecture also incorporates mechanisms for privacy-preserving analytics, automated data quality assessment, provenance management, and alignment with the EHDS User Journey, enabling streamlined data discovery, permit management, and controlled secondary use of paediatric transplantation data.

Finally, this deliverable establishes the baseline for all implementation activities in subsequent work packages. It guides the development of the Big Data infrastructure (WP3), core EHDS services and capsules (WP4–WP6), federated NLP and AI components (WP5), and user-journey assistants (WP7). It also provides a reference framework that will be applied during the clinical pilots in WP8 and WP9, ensuring that the ecosystem is not only theoretically sound but also practically deployable, scalable, and ready for evaluation in real clinical environments.

In summary, D2.3 consolidates the conceptual, regulatory, and technical groundwork required to build the PROTECT-CHILD platform. By providing unified models, structured requirements, and a robust architectural blueprint, it ensures interoperability, security, regulatory compliance, and long-term sustainability for a federated European ecosystem dedicated to improving outcomes for paediatric transplant patients.

1.1 Deliverable context

Table 1: Deliverable context

PROJECT ITEM IN THE DoA	RELATIONSHIP
Project Objectives	D2.3 directly contributes to Objectives O1–O6, especially: defining the data and metadata models (O1), establishing interoperability across standards (O2, O3), defining the federated secure architecture for genomic + clinical secondary use (O4), and ensuring EHDS-aligned privacy-preserving processing (O5). It operationalises many requirements outlined in Section 1 of the updated proposal (e.g., PRIIST-Q principles, TEHDAS User Journey).
Exploitable Results	D2.3 specifies the foundations for several exploitable results: ER1 – PROTECT-CHILD Data & Metadata Model, ER2 – Federated SPE Architecture (Capsules), ER3 – Data Quality & FAIRification Framework, ER4 – NLP/AI pipelines for multi-modal data, ER5 – Governance & Legal Toolkit. It clarifies interfaces and specification dependencies for ERs listed in updated Section 2.1.1.
Workplan	D2.3 is the core technical output of WP2 (T2.3 + T2.4) and supports downstream WPs. It provides the specification baseline for WP3 (infrastructure), WP4 (EHDS services), WP5 (data discovery & federated AI), WP6 (governance), WP7 (User Journey), WP8–9 (pilot data processing). It is aligned with the updated PERT diagram (page 40 of the proposal).
Milestones	D2.3 supports MS1 – Ecosystem Preliminary Definition (M12) and provides input to MS2 – Architecture & Infrastructure Baseline (≈M24). It is essential to MS3 – Mid-course Correction Check (M30) by delivering validated models/requirements. It is a prerequisite for MS4 – Operational Platform (M42).
Deliverables	D2.3 builds on D2.1 (Requirements) and D2.2 (Architecture Overview) and feeds into D3.1–D3.4 (Data models, ETL, ingestion), D4.1–D4.3 (EHDS capsules), D5.x NLP and federated AI components, D6.x governance dashboards, and D8.4/8.5/9.x pilot deliverables. It is aligned with all deliverable updates from the revised DoA (Annex 1 modifications).
Risks	D2.3 mitigates key risks listed in the updated "Critical Risks" table, including: R1 – Legacy IT & deployment delays, R2 – Interoperability failures, R3 – Non-compliance with EHDS/GDPR, R4 – Genomics processing heterogeneity, R5 – Data quality issues, R6 – Federated analysis failures, R7 – Ethical complexities with minors, and R8 – Cross-border

PROJECT ITEM IN THE DoA	RELATIONSHIP
	permit delays. It defines the technical & governance safeguards required under PRIIST-Q and Zero Trust to reduce these risks.

2 Existing relevant initiatives

Protect-Child project needs to inherit requirements from 2 main European initiatives: the European Health Data Space (EHDS) and the most relevant infrastructures for genomics in Europe.

EHDS is a new EU regulatory and technical framework designed to enable secure, cross-border access, sharing, and reuse of health data for both primary and secondary use. It aims to empower citizens to control their own electronic health records while creating a trusted infrastructure that supports research, innovation, and public health across Europe. ELIXIR, GA4GH, and the European Genomic Data Infrastructure (GDI) form a complementary, interoperable stack for genomic data at scale: ELIXIR coordinates Europe's life-science data services and cloud/compute; GA4GH defines the global technical and policy standards; and GDI implements a federated, secure, cross-border infrastructure aligned to 1+ Million Genomes for real-world clinical, research, and public health impact.

2.1 EHDS

EHDS[1] represents a historic step forward in how the European Union approaches the governance and responsible exploitation of one of the most sensitive and valuable resources available today: electronic health data. While much public attention has centred on its role in empowering citizens to access their health records and improving cross-border healthcare delivery, the true transformative potential of the EHDS lies in its second pillar—the secondary use of health data.

This dimension is not about treating patients directly or supporting day-to-day care. Instead, it is about unlocking the potential of vast repositories of medical records, registries, genomic databases, imaging archives, insurance data, and health app outputs to drive research, innovation, public health policy, and education. The secondary use framework of the EHDS offers a regulated, transparent, and secure pathway to turn raw, fragmented, and sensitive information into evidence-based insights and innovations that can benefit society at large.

Before the EHDS, access to health data for research or innovation was often cumbersome and fragmented. Each Member State applied its own rules, procedures, and interpretations, leading to uneven practices and, in many cases, lengthy delays or outright barriers. Researchers or innovators attempting to conduct studies that span multiple countries faced a patchwork of regulatory landscapes, each with different application forms, data access committees, and approval procedures.

The EHDS seeks to overcome this fragmentation by creating a common European framework for secondary data use. Its goal is nothing less than to establish a "single market for health data," in which cross-border data access is harmonised, transparent, and efficient. Under this vision, a researcher in Germany applying for cancer registry data from Spain should follow the same procedure, submit the same forms, and encounter the same assessment criteria as a researcher in Sweden requesting datasets from Italy.

To enable this harmonisation, the regulation mandates the creation of Health Data Access Bodies (HDABs) in each Member State. These bodies are the central actors in the secondary use framework. They are entrusted with managing applications, issuing permits, coordinating with health data holders, and ensuring compliance with EU law. The EHDS thereby shifts Europe from a decentralised patchwork of practices towards a coherent, federated governance model.

[1] Regulation (EU) 2025/327 of the European Parliament and of the Council of 11 March 2025 establishing the European Health Data Space and amending Regulations (EU) No 910/2014, (EU) 2018/1724 and (EU) 2018/1725, and Directives 2002/58/EC, 2011/24/EU and (EU) 2019/1024. https://eur-lex.europa.eu/eli/reg/2025/327/oj/eng

2.1.1 THEDAS2

Turning the principles of the EHDS into everyday practice requires operational clarity, technical standards, and governance guidelines that work across all Member States. This is where the Second Joint Action Towards the European Health Data Space (TEHDAS2)[2] becomes indispensable. TEHDAS2 is the collective European effort to prepare the ground for the EHDS, providing Member States and stakeholders with the tools, specifications, and shared practices needed for smooth implementation.

At its core, TEHDAS2 is designed to translate regulation into practice. The EHDS regulation (EU 2025/327) sets out the broad architecture: citizens' rights to control their data, the obligations of health data holders, the establishment of Health Data Access Bodies (HDABs), and the requirement to process sensitive data within Secure Processing Environments (SPEs). What TEHDAS2 adds are the details: how HDABs should assess data access applications, what technical standards SPEs must meet, how fees and penalties should be calculated, how opt-out mechanisms should be implemented, and how significant findings from secondary use must be communicated to individuals.

This level of guidance is critical to avoid fragmentation. Europe's health systems differ profoundly in structure, financing, and governance. Without a common reference framework, each Member State might interpret the EHDS differently, leading to divergent procedures, inconsistent safeguards, and ultimately undermining cross-border research. TEHDAS2 mitigates this risk by bringing together ministries of health, national data agencies, and expert organisations to draft common guidelines and technical specifications. In doing so, it fosters a culture of harmonisation and trust that is essential for the EHDS to function as a pan-European ecosystem.

Beyond technical clarity, TEHDAS2 plays a vital role in building trust among citizens, researchers, and institutions. The secondary use of health data touches on sensitive issues of privacy, ethics, and societal benefit. Citizens need assurances that their rights will be respected equally across borders, researchers need confidence in predictable and efficient access procedures, and policymakers need tools to enforce compliance fairly and consistently. TEHDAS2 addresses all these dimensions by establishing clear procedures for opt-out, consent, penalties, and monitoring, and by embedding transparency and accountability throughout the data lifecycle.

TEHDAS2 is therefore more than an EU-funded project—it is the implementation laboratory of the EHDS. It is where theoretical rights and obligations are tested against real-world challenges, where national practices are aligned into European standards, and where the building blocks of the future HealthData@EU[3] infrastructure are shaped. Its outputs will not only guide Member States in the first years of implementation but will also serve as a reference for future implementing acts and Commission guidance.

In short, the EHDS provides the vision of a single market for health data, while TEHDAS2 provides the means to realise it. Without TEHDAS2, the EHDS risks remaining a regulation confined to paper; with TEHDAS2, the Union can move towards a trusted, harmonised, and citizen-centred ecosystem where health data is securely harnessed for science, policy, and innovation to the benefit of all Europeans.

Defining the Permitted and Prohibited Uses

[2] "Second Joint Action Towards the European Health Data Space – TEHDAS2", coordinated by Finnish Innovation Fund Sitra, funded under the EU4Health Programme, started in May 2024 and runs until December 2026. https://tehdas.eu

[3] European Commission. (2024). HealthData@EU Central Platform (acceptance environment). European Health Data Space – Directorate-General for Health and Food Safety (DG SANTE).
Retrieved from https://acceptance.data.health.europa.eu/healthdata-central-platform?locale=en

A cornerstone of the secondary use regime is the clear definition of what is allowed and what is forbidden. The EHDS regulation's Article 53 establishes six permitted purposes for which secondary use may be granted. These include:

- research in health and life sciences,
- innovation and development of new medicines or treatments,
- policy-making and regulatory activities in the field of health,
- statistical analyses,
- educational purposes in the health domain, and
- improvement of healthcare systems.

At the same time, Article 54 sets strict boundaries, listing prohibited uses such as using health data for marketing activities, developing harmful products, discriminatory profiling, or decisions that could negatively affect individuals or groups. These prohibitions are critical to maintaining public trust. Citizens must know not only that their data is protected but also that it will never be exploited in ways that undermine their dignity, safety, or equality .

The dual approach—defining both permitted and prohibited purposes—offers clarity and legal certainty for data users and data holders alike. It ensures that data-driven innovation is channelled towards legitimate and socially beneficial goals, while excluding practices that could damage trust in the system or harm individuals.

The Role of Health Data Access Bodies (HDABs)

The EHDS gives HDABs a dual role: facilitators and regulators. On the one hand, they enable access to health data by processing applications, liaising with health data holders, and organising the secure flow of information. On the other hand, they are supervisory authorities with enforcement powers, capable of suspending permits, revoking access, or imposing penalties when obligations are breached .

The application process itself is highly structured. A data user—be it a university researcher, a pharmaceutical company, a start-up innovator, or a public authority—must submit a data access application or a data request through harmonised templates. HDABs are responsible for completeness checks, ensuring that applications are filled in properly, and then for substantive assessments to verify that the purpose aligns with Article 53, that no prohibited use is foreseen, and that safeguards for intellectual property rights and trade secrets are respected .

If approved, the HDAB issues a data permit. This is a legally binding document specifying the datasets to be accessed, the purposes for which they may be used, the timeframe of access, and the obligations of the data user. Permits are not open-ended licenses: they are conditional, limited in scope, and always subject to monitoring and review.

Secure Processing Environments: Where the Work Happens

A distinctive feature of the EHDS framework is the requirement that all secondary use of sensitive health data must occur in Secure Processing Environments (SPEs) . These are controlled infrastructures, virtual or physical, where data users can carry out their analyses but cannot remove raw data.

SPEs function on a simple principle: data can be brought in, combined, and analysed, but only aggregated or anonymised results may leave, and even these require validation by the HDAB. This prevents uncontrolled replication or leakage of sensitive datasets.

The technical specifications of SPEs are rigorous. They must ensure strong identity and access management, encryption at rest and in transit, auditing and logging of all activities, and monitoring of user behaviour. Administrators of SPEs are typically not allowed to access the content of the data. Each SPE is project-based, isolated, and destroyed or archived once the project concludes.

Beyond security, SPEs are also enablers. They provide researchers with the computational tools needed to work with large and complex datasets—statistical software, machine learning frameworks, visualisation platforms—while ensuring that work remains within the boundaries of lawful and secure processing. They also allow for federated models, where analysis can be distributed across multiple SPEs in different Member States without the need to transfer data across borders. This is essential for large-scale studies in genomics, rare diseases, or cross-country epidemiology.

Responsibilities of Health Data Holders

The secondary use framework also imposes obligations on health data holders—entities such as hospitals, research institutions, registries, insurers, or developers of digital health tools. Once a permit is issued, they are legally required to provide the approved data in a timely, structured, and secure manner. They must describe their datasets in national catalogues, ensure that metadata is up to date, and prepare the data through pseudonymisation or anonymisation as needed.

Non-compliance is not taken lightly. The EHDS regulation empowers HDABs to sanction data holders who fail to fulfil their duties. Penalties may include suspension from participation, administrative fines, or other enforcement actions proportionate to the gravity of the breach .

Citizens' Rights and Safeguards

Perhaps the most innovative aspect of the EHDS secondary use framework is its attention to citizen rights. Recognising that trust is the foundation upon which the entire system must rest, the regulation grants individuals' significant control and transparency.

Citizens have the right to opt out of the reuse of their identifiable health data for secondary purposes, if they so wish. They also have the right to be informed about the conditions under which their data may be used, and in certain cases, even to be notified if significant findings relevant to their health are discovered during secondary use research .

These rights are not symbolic: they require operational mechanisms. HDABs must keep registers of opt-outs, data users must respect them, and communication protocols must be in place to ensure that findings are channelled back appropriately through data holders.

By embedding these safeguards, the EHDS goes beyond a purely utilitarian vision of data reuse and aligns itself with European values of dignity, autonomy, and fairness.

Enforcement, Penalties, and Trust

Secondary use of data carries risks, from accidental breaches to deliberate misuse. The EHDS anticipates these challenges by providing HDABs with strong enforcement tools under Articles 63 and 64. They may impose administrative fines, revoke permits, or exclude repeat offenders from future access.

The guiding principles for enforcement are effectiveness, proportionality, and deterrence. Fines are not intended merely as punishment but as a mechanism to ensure compliance, level the playing field, and maintain trust in the system. Transparency of enforcement actions—publishing anonymised summaries of cases or reporting annually on compliance—is also encouraged to reinforce a culture of accountability across Europe.

Towards a Culture of Responsible Data Reuse

The EHDS does more than create new institutions or IT systems. It seeks to cultivate a new culture around health data in Europe—one where secondary use is seen not as a risk but as a shared opportunity, provided it is conducted within clear ethical and legal boundaries.

By harmonising rules, enabling secure infrastructures, and embedding citizen-centric safeguards, the EHDS aspires to shift Europe away from fragmented, ad hoc arrangements towards a coherent, trustworthy, and innovation-friendly ecosystem. In doing so, it also strengthens Europe's digital sovereignty, ensuring that health data is used according to European values, within European infrastructures, and for the benefit of European society.

In the coming years, the secondary use of health data under the EHDS could transform medical research, accelerate the development of personalised treatments, enhance preparedness for public health crises, and support more equitable and efficient healthcare systems. Yet its success will depend on continued vigilance: balancing openness with protection, encouraging innovation while preventing misuse, and above all, maintaining the trust of the citizens whose data makes it all possible.

2.2 ELIXIR

ELIXIR[4] is a European organization that brings together bioinformatics resources from different countries and promotes knowledge transfer. Its main objective is to form a common infrastructure among countries to facilitate the searching, analysis, and exchange of biological information. ELIXIR operates a Hub-and-Nodes model with resources spanning human genomics, proteomics, metabolomics, biodiversity, and more. It integrates distributed databases, analysis platforms, software tools, and storage capacities. In addition, it offers services such as guidelines, web portals, bioinformatics resources, events, and job opportunities.

ELIXIR's analysis tools are a cornerstone of its infrastructure, designed to facilitate seamless access, execution, and sharing of bioinformatics resources. All of these tools are freely accessible and organized into five core services that support the entire lifecycle of biological data analysis:

- bio.tools: This comprehensive registry enables users to search among over 30,000 bioinformatics tools, providing detailed information about each resource's functionality, hosting, and documentation, thereby promoting discoverability and interoperability.
- BioContainers: This platform offers a repository of containerized software that can run on any operating system, ensuring reproducibility and ease of deployment across diverse computational environments.
- OpenEBench: It allows benchmarking and monitoring of bioinformatics tools by providing performance metrics, quality assessments, and comparative analyses, fostering transparency and continual improvement.
- UseGalaxy.eu: As a web-based portal, it provides access to over 2,500 tools for omics data analysis, combined with visualization capabilities and reference genomes, making complex analyses accessible without local installation.
- Workflowhub.eu: This registry serves as a platform for describing, sharing, and publishing scientific workflows, facilitating reproducibility and collaboration among researchers.

[4] ELIXIR Europe. (2024). ELIXIR – A distributed infrastructure for life-science data. Retrieved from https://elixir-europe.org/

These services collectively streamline bioinformatics research by enabling easy discovery, deployment, benchmarking, and sharing of tools and workflows, thereby accelerating scientific discoveries in life sciences.

In addition, it offers access to training, standards, and databases. Among these services, one of the most widely used is Galaxy. Galaxy is characterized by having a simple and easy-to-use web interface, which has contributed to its popularity among users without programming experience. The online version has a storage limit of 250 gigabytes.

Within the ELIXIR infrastructure, there are also tools for workflow management, such as Nextflow, which allows for the reproducible and scalable execution of bioinformatics analyses. It is especially useful for many projects as it has very well-defined workflows.

Another fundamental resource supported by ELIXIR is Ensembl, a genomic database maintained by EMBL-EBI that provides updated annotations of genes, transcripts, and variants. Widely used tools directly depend on their data, which reinforces its central role in European bioinformatics.

2.3 Genomic Data Infrastructure

Genomic Data Infrastructure (GDI)[5] is a project that aims to establish a secure genomic data infrastructure in Europe. Its development is based on the results of the Beyond 1 Million Genomes (B1MG)[6] project and will contribute significantly to the 1+ Million Genomes (1+MG) initiative. This effort aims to improve the integration of genomic, phenotypic, and clinical data through an infrastructure that allows for efficient and secure access to information.

Data access will be public, except for the most sensitive data, which can be shared with clinicians and researchers upon prior request. The platform will also establish clear protocols to ensure that patient privacy and security are respected at all times. The benefits of GDI are expected to be reflected in a substantial improvement in diagnosis, treatment, and predictive medicine for the European population, contributing to personalized medicine and the improvement of public health policies.

Furthermore, GDI will promote interoperability between different databases and research systems, allowing genomic data to be used more effectively in collaborative research and transnational projects. Its alignment with other initiatives like GA4GH and ELIXIR ensures that the adopted standards and protocols are widely compatible, which facilitates the exchange of information between different actors in the field of biomedical research and health.

Together, these entities translate policy into practice: GA4GH supplies global standards; ELIXIR curates and operates European services and training aligned to those standards; and GDI deploys a compliant, production-grade, federated infrastructure for secure discovery, access, and analysis of sensitive human genomic data at national and European levels.

2.4 Global Alliance for Genomics and Health

GA4GH[7] is a non-profit organization that aims to define the standards and policies to be followed for the use of genomic data worldwide. This organization was established in 2013 and has since promoted measures to facilitate the exchange of genomic and clinical data. Its main objective is to ensure that genomic data can be shared among institutions in a secure, efficient, and ethical

[5] Genome Data Infrastructure (GDI). (2024). The Genome Data Infrastructure Project – Towards a federated and secure genomic data network in Europe. Retrieved from https://gdi.onemilliongenomes.eu/

[6] Beyond 1 Million Genomes (B1MG). (2024). Beyond 1 Million Genomes – Supporting the implementation of the 1+MG initiative across Europe. Retrieved from https://b1mg-project.eu/

[7] Global Alliance for Genomics and Health (GA4GH). (2024). Enabling responsible international data sharing for genomics and health. Retrieved from https://www.ga4gh.org/

manner. GA4GH convenes 500+ organizations and thousands of contributors to define and validate standards in real-world implementations.

GA4GH not only offers guidelines, protocols, and policies. Its researchers also develop other types of products, which can be application programming interfaces (APIs), ontologies, and file formats. These tools not only standardize processes but also allow researchers and clinicians to work with data more effectively. One API that promises to be important in the coming years is the Beacon.

Beacon is a tool that allows researchers and clinicians to discover relevant information about genetic variants while maintaining patient privacy. The process is simple; the researcher makes a query to the Beacon asking for a specific genetic variant, for example, SNV type mutations in the BRCA1 gene. The Beacon searches its database, which includes all research and health centers that have implemented the API, and provides a response with the information it has about it. The first version only gave a Boolean yes/no answer, but the second version also offers contextual information about the variant.

GA4GH also promotes standards for the execution of genomic analyses in the cloud, such as the Workflow Execution Service, called WES. These services are relevant for interoperability between platforms.

3 Technical landscape analysis

The development of the PROTECT-CHILD data ecosystem requires a deep understanding of the current technical landscape governing secure, interoperable, and large-scale processing of health and genomic data in Europe. This chapter provides an analytical overview of the standards, infrastructures, technologies, and architectural paradigms that form the foundation upon which the PROTECT-CHILD platform is built. Rather than introducing new concepts in isolation, the analysis connects existing European initiatives (EHDS, GDI, ELIXIR, GA4GH), prevailing interoperability standards (FHIR, OMOP, Beacon v2), secure computation paradigms (SPEs, SMPC, federated learning), and modern cloud-native technologies (containerisation, service meshes, zero-trust infrastructures) into a coherent picture that informs the design decisions adopted in D2.3.

By assessing the maturity, constraints, and complementarities of these elements, the chapter clarifies why specific standards and technologies were selected, how they address PROTECT-CHILD requirements, and where gaps remain for secure cross-border genomic data processing in paediatrics. The technical landscape analysis thus serves as a bridge between the high-level requirements defined in WP2 and the detailed architectural specification that follows. It ensures that the project builds not only a compliant ecosystem aligned with emerging EHDS regulations, but also a future-proof, interoperable, and scalable infrastructure capable of supporting the needs of clinical research, federated analysis, and data governance across the TransplantChild ERN.

3.1 Microservices and Service Mesh Technologies

The shift from monolithic to microservices architectures marks one of the most transformative evolutions in modern software design. In a monolithic system, all functionalities are bundled together into a single, interdependent codebase—easy to start with but increasingly difficult to maintain and scale as complexity grows. In contrast, microservices[8] break down applications into smaller, autonomous components, each responsible for a specific function (such as authentication, data ingestion, analytics, or notifications). These microservices communicate through well-defined APIs, often using lightweight protocols like REST, gRPC, or message queues, allowing them to be developed, deployed, and scaled independently.

This architectural approach introduces significant advantages in flexibility, scalability, and resilience. Development teams can work in parallel on different services, using the most suitable technologies for each one, while updates or failures in one service do not necessarily compromise the entire system. This makes microservices particularly well-suited for complex and evolving environments such as healthcare, where systems must continuously integrate new data sources, comply with evolving regulations, and interoperate with diverse infrastructures like hospital information systems, research platforms, and federated computing environments.

However, as systems evolve from a handful of services to hundreds or even thousands, the complexity of managing their interconnections increases dramatically. Each service must securely discover, connect, and communicate with others, often across dynamic and distributed infrastructures such as Kubernetes clusters or hybrid clouds. Ensuring consistent security, observability, and policy enforcement across this dynamic network becomes a non-trivial challenge. This is where service mesh technologies play a fundamental role.

[8] Newman, S. (2021). Building Microservices: Designing Fine-Grained Systems (2nd ed.). O'Reilly Media. ISBN 978-1-4920-8246-3.

A service mesh[9] is an infrastructural layer that transparently manages the communication between microservices. Instead of embedding networking, security, and monitoring logic directly into each service, these cross-cutting concerns are externalized to a dedicated data plane—typically implemented through lightweight proxies (such as *Envoy*) deployed alongside each service—and a control plane that orchestrates policies, configurations, and certificates. Through this separation of concerns, service meshes like Istio, Linkerd, or Consul Connect provide developers and operators with fine-grained control over traffic, security, and resilience without modifying application code.

From a security and compliance perspective, this model is particularly valuable. A service mesh can enforce mutual TLS (mTLS) encryption between all services, authenticate workloads using cryptographic identities (such as SPIFFE/SPIRE), and apply "zero-trust" policies that assume no implicit trust between components. It also centralizes observability by collecting telemetry data—such as metrics, traces, and logs—from every service interaction, enabling advanced monitoring, auditing, and anomaly detection. These features are critical for sectors like healthcare, where data integrity, traceability, and accountability are essential to meet regulatory frameworks such as GDPR, the EHDS, or NIS2.

Moreover, service meshes facilitate resilience and reliability. They allow operators to implement fault-tolerant strategies (such as retries, circuit breakers, and traffic shifting) and progressive deployment techniques (like canary releases or A/B testing) with minimal risk. They also support hybrid and multi-cluster architectures, making it possible to federate services across different clouds or institutional boundaries—an essential feature for collaborative environments such as the *European Health Data Space* (EHDS), where secure and distributed data processing must occur across multiple trusted domains.

In healthcare and research ecosystems, microservices and service mesh technologies together form the backbone of modular, scalable, and secure digital infrastructures. They enable the orchestration of complex data pipelines—from ingestion and anonymization to federated analytics and AI model deployment—while preserving compliance, interoperability, and trust. By abstracting complexity and embedding security into the fabric of service communication, these technologies make it possible to build next-generation platforms that are not only technically robust but also ethically and legally sound, aligning with Europe's vision of secure, interoperable, and human-centric digital health systems.

3.2 Zero trust and cybersecurity best practices

Zero Trust[10] is a mindset more than a product: never trust by default, always verify, and assume breach. In practice that means every identity—human, service, workload, node—must authenticate strongly; every request must be authorized explicitly against least-privilege policy; every hop is encrypted; and continuous signals (attestations, posture, runtime) feed policy decisions.

The Protect-Child implementation will align with the foundational Zero Trust principles that underpin the security and governance model of Secure Processing Environments (SPEs) within

[9] Istio Authors. (2024). Istio Service Mesh – Security, Observability, and Traffic Management Framework. Version 1.21. The Istio Project. Available at: https://istio.io/

[10] Rose, S., Lohnas, K., Borchert, O., Connelly, S., & Mitchell, S. (2023, draft). NIST Special Publication 800-207A (draft): Applying Zero Trust Architecture to Cloud-Native Applications, Environments, and Workloads. National Institute of Standards and Technology, U.S. Department of Commerce. Available at: https://doi.org/10.6028/NIST.SP.800-207A-draft (Accessed 5 November 2025).

the EHDS. These principles ensure that security is not perimeter-based but continuous, adaptive, and verifiable across all layers of the infrastructure:

- Identity-first access → every entity (user, service, workload) is authenticated and authorised continuously, using federated identity frameworks such as eIDAS and GA4GH passports.
- Least privilege → permissions are always scoped to the minimal dataset or operation required; administrative access is temporary, justified, and fully logged.
- Microsegmentation → workloads and projects are isolated into secure enclaves with strict network separation, restricted egress, and no implicit trust between namespaces or clusters.
- Assume breach → security monitoring operates on the premise that compromise is always possible; continuous telemetry, real-time alerts, and automated containment protect the environment.
- Data-centric security → protection follows the data itself through encryption at rest and in transit, pseudonymisation of personal identifiers, and HDAB-verified anonymised outputs.
- Federated trust → cross-border collaboration relies on shared trust anchors, interoperable identity systems, and secure federation APIs ensuring verifiable compliance across Member States.

In Protect-Child we have promised that with Docker, Kubernetes, and Istio we can turn the Zero Trust philosophy into a concrete, layered architecture that travels cleanly from a developer laptop to multi-cluster production.

3.2.1 Protect-Child Zero Trust approach

The Protect-Child architecture adopts a Zero Trust security model in which no component, user, or service is inherently trusted. Trust is continuously verified through strong identity, strict policy enforcement, and comprehensive encryption. Within this model, identities—not IPs or networks—form the foundation of trust.

At the core of Protect-Child's Zero Trust design lies workload identity. The platform leverages SPIFFE/SPIRE[11] to issue short-lived, cryptographically verifiable workload certificates (SPIFFE IDs) to every pod, daemon, and job deployed within Kubernetes. These identities are then consumed by Istio[9], which enforces mutual TLS (mTLS) for all service-to-service communications. This ensures that every interaction within the mesh occurs only after both entities have proven their authenticity. Both ambient mesh (ztunnel) and classic sidecar configurations are supported, with STRICT mTLS enforced via PeerAuthentication and ISTIO_MUTUAL set as the default policy.

Human access follows the same principle of verified identity. OpenID Connect (OIDC)[12] is integrated with an auditable, open-source identity provider (such as Keycloak) to enable federated, MFA-backed authentication for all administrative and user interfaces, including kubectl and platform dashboards. These identities are mapped to Kubernetes RBAC roles following the principle of least privilege, ensuring that users possess only the permissions necessary for their tasks.

The Protect-Child deployment environment follows a default-deny posture, expanding privileges only where explicitly required. At the cluster perimeter, egress is closed by default, and only necessary destinations are allowed through controlled Istio ServiceEntry and

[11] The SPIFFE and SPIRE Projects – Secure Production Identity Framework for Everyone. Cloud Native Computing Foundation (CNCF).Available at: https://spiffe.io/ (Accessed: 5 November 2025).

[12] OpenID Connect Core 1.0 incorporating errata set 1. Published by the OpenID Foundation. Available at: https://openid.net/specs/openid-connect-core-1_0.html (Accessed: 5 November 2025)

AuthorizationPolicy definitions. For inbound (north–south) traffic, similar controls are applied at Istio gateways, complemented by JWT validation on headers. Internally, CNI network policies (e.g., Cilium or Calico) operate alongside Istio AuthorizationPolicies to enforce both L3/L4 and L7 security controls, establishing layered defense-in-depth across the data plane.

Dynamic policy evaluation is centralized with OPA[13]. OPA/Gatekeeper governs admission and build-time checks, while OPA Envoy/Ext-AuthZ provides per-request runtime authorization. Policy enforcement ensures that only images from approved registries are deployed, that :latest tags are forbidden, and that workloads adhere to strict runtime constraints—non-root execution, dropped Linux capabilities, read-only filesystems, bounded resources, and explicit DNS-based egress allow-lists. YAML-based policies can alternatively be defined via Kyverno or Kubernetes ValidatingAdmissionPolicy for simplified authoring.

End-to-end encryption and attestation are mandatory across the stack. Istio guarantees encryption in transit, while at rest, Kubernetes Secrets are protected through envelope encryption with a KMS plugin; encryption keys (KEKs) are stored in secure KMS services such as Vault, HSM, or cloud KMS. Developer environments use rootless Docker, user namespaces, and encrypted volumes (LUKS, FileVault, BitLocker) to safeguard data. Secrets are handled securely through Docker secrets or mounted key stores instead of environment variables.

Supply-chain security is enforced through image signing (Sigstore cosign) and admission verification via policy-controller, Ratify, or Gatekeeper constraints. Only minimal, distroless base images are accepted. Continuous Integration pipelines emit SBOMs and in-toto/SLSA attestations, enabling policies that admit only images signed by trusted keys, scanned within 24 hours, and built through verified CI pipelines.

Runtime and infrastructure layers are hardened comprehensively. The Kubernetes API is configured with anonymous access disabled, full audit logging enabled, and separate roles for administrators and automation. Kubelets run with authentication and authorization enabled, and node OS security policies (SELinux/AppArmor) are enforced while human SSH access is removed. Node-to-node communication is encrypted using WireGuard or IPSec at the CNI layer, while Istio mTLS adds another independent encryption tier for service traffic. Gateways restrict traffic to TLS 1.2+ with hardened cipher suites and rate limiting. Egress gateways centralize outbound traffic, enabling data loss prevention (DLP), URL allow-listing, and certificate pinning.

Continuous posture management complements this runtime security. Automated image scanning, runtime anomaly detection (Falco, Tetragon), and configuration drift control (comparing live manifests to Git) generate security events integrated into the SIEM for correlation and response.

Zero Trust principles are embedded into the developer experience to ensure security without friction. Development teams are provided with pre-hardened Helm or Jsonnet templates that include default-deny authorization policies, strict mTLS, network policies, resource quotas, and health probes. Each ServiceAccount is automatically assigned a SPIFFE ID, and “safe egress” classes with narrow allow-lists are predefined. Local development clusters (e.g., Kind or k3d) replicate production security profiles—Istio mTLS settings, Gatekeeper constraints, and signature verification—to ensure early policy validation before deployment.

The implementation roadmap follows four progressive phases:

1. Developer and Docker (local) – Rootless Docker with user namespaces and TLS-protected daemons; encrypted volumes; secrets via secure stores; image signing and verification even in local environments.
2. Single-cluster Kubernetes (staging) – Deployment of SPIRE and Istio with STRICT mTLS, default-deny authorization, and CNI network policies; egress locked by default;

[13] Open Policy Agent Project. (2024). Policy-based control for cloud-native environments. Cloud Native Computing Foundation (CNCF). Available at: https://www.openpolicyagent.org

Gatekeeper/Kyverno constraints active; KMS-based Secret encryption; policy-as-code pipeline integrated into CI/CD.

3. Multi-cluster production with Istio – Federation of SPIFFE trust domains via SPIRE bundle exchange; Istio east–west gateway federation; centralized egress management and DLP; OPA Ext-AuthZ for claim-aware authorization and rate limiting; policy enforcement backed by real-time security signals and SLSA attestations.
4. Observability and response – Fine-grained telemetry through Istio access logs and SPIFFE IDs embedded in metrics; comprehensive audit logging across APIs and gateways; rapid response capabilities to revoke certificates, block egress, or quarantine namespaces via namespaced policies.

In summary, Docker provides reproducible, isolated units; Kubernetes adds declarative orchestration and policy control; and Istio transforms identity into the network's security fabric. Combined with signed artifacts, short-lived workload certificates, end-to-end encryption, default-deny authorization, and policy-as-code governance, Protect-Child moves from a perimeter-based security model to one of continuous, provable trust enforcement across its entire data ecosystem.

3.2.2 THEDAS2 specifications of SPEs

This section resumes the M7.4 Draft technical, functional and security specifications of Secure Processing Environments (SPEs)[14], prepared under TEHDAS2 (the second Joint Action towards the European Health Data Space). It was finalised on 12 September 2025 and accepted by the project steering group. The goal is to define how Secure Processing Environments (SPEs) should work within the European Health Data Space (EHDS) to allow lawful secondary use of health data (e.g., for research, innovation, policymaking) while ensuring data protection, privacy, and cybersecurity.

The *M7.4* deliverable formalises the Secure Processing Environment (SPE) as the central operational pillar of the forthcoming European Health Data Space (EHDS). It consolidates earlier conceptual drafts into a technical and legal blueprint that allows Member States to design interoperable, auditable, and federated environments for the secondary use of electronic health data.

At its core, the document interprets Article 73 of the EHDS Regulation and the Data Governance Act definition of an SPE, describing it as both a technical system and a governance framework that ensures compliance with Union law—particularly the GDPR—while enabling lawful, high-value use of sensitive data. The specification positions SPEs as "trusted execution enclaves" that integrate security, traceability, and transparency by design rather than relying on procedural oversight.

The report structures the SPE framework into four interdependent layers:

1. Sensitive-data protection rules, which define the baseline for confidentiality and authorised use;
2. Core functional and operational requirements, covering user roles, setup, access management, auditing, monitoring, incident handling, and risk mitigation;
3. Federation and interoperability rules, which make cross-border collaboration possible; and

[14] Lehväslaiho, H., Lodenius, H., Barros, B., Berna, A., Bréchot, L., Gütter, Z., Huru, H. A., Jarosz, Y., Kondić, T., Lähteenmäki, J., Martens, M., Alvarez, M., Pajula, J., Sondag, T., Trefois, C., & Turunen, E. (2025). M7.4 Draft technical, functional and security specifications of Secure Processing Environments (SPEs). TEHDAS2 – Second Joint Action Towards the European Health Data Space. Version 1.0, accepted by the Project Steering Group on 12 September 2025. Co-funded by the European Union under the EU4Health Programme (Grant Agreement No. 101176773). CSC – IT Center for Science Ltd., Finland (Lead organisation). Available for public consultation at: https://tehdas.eu/wp-content/uploads/2025/09/draft-technical-functional-and-security-specifications-of-secure-processing-environments.pdf

4. Federated-computing extensions, supporting distributed analytics and learning across multiple SPEs.

Throughout, *M7.4* explicitly grounds these requirements in recognised security standards such as ISO/IEC 27001 and 27002, NIS2, and the lightweight FitSM framework for federated service management. This combination provides a scalable approach that is rigorous enough for regulated data yet flexible for public-sector deployment.

A significant innovation of the document is the detailed treatment of SPE federation. Recognising that research and policy questions often require combining datasets across jurisdictions, the deliverable defines the FSPER requirements that govern how multiple SPEs interoperate. These include contractual agreements between operators (FSPER-1), federated identity management (FSPER-2 & 6), alignment with common data models such as OMOP CDM, and the use of secure transfer standards like EU eDelivery and GA4GH crypt4GH. The federation must provide registries for authorisation, accounting, and dataset discoverability, thus allowing distributed yet accountable processing.

Building upon federation, *M7.4* introduces the Federated Computing (FCR) specification, bridging the gap between data-sharing governance and computational interoperability. It describes how federated analytics and federated learning can occur inside a network of compliant SPEs without moving raw data. These processes rely on shared APIs, GA4GH-aligned cryptographic protocols, and privacy-preserving techniques such as differential privacy.

Operationally, the report emphasises continuous risk management, automated patching and backup routines, staff training, and external auditing. It explicitly connects technical obligations to the GDPR's core principles: data minimisation, security of processing, privacy-by-design, and research safeguards. The specification also integrates ethical frameworks such as the Five Safes and the notion of Trusted Research Environments, showing continuity with long-established confidentiality models in the UK and other Member States.

Legally, *M7.4* depicts a layered chain of compliance: NIS2 → SPE implementation → EHDS Articles 33–35, 45–47, 73 → GDPR Articles 6, 9, 25, 32, 89 → DGA for consent and data altruism.

This structure makes the SPE not only a technological safeguard but the formal enforcement mechanism through which EHDS ensures conformity with EU law.

Finally, the document stresses that SPEs must evolve beyond static "secure desktops" toward dynamic, scalable platforms capable of AI and high-performance computing. They should host statistical software, machine-learning libraries, and APIs for programmatic access, supporting future use cases such as real-time analytics or GenAI-driven research assistants—all while remaining inside the regulatory perimeter.

In summary, the document reframes the SPE as a Zero-Trust-aligned, federated, and service-managed infrastructure that unites legal certainty with computational capability. It translates the ideals of the EHDS—data sovereignty, privacy, and innovation—into a concrete technical architecture where trust is enforced by code, governance, and law acting together.

3.2.2.1 NIS2 and SPE alignment with Zero Trust principles

The NIS2 Directive[15] establishes a common level of cybersecurity across the European Union, requiring essential and important entities, such as hospitals, health data access bodies, and research infrastructures to implement risk-based security measures, ensure incident detection and reporting, and apply continuous risk management. Although NIS2 is a regulatory framework and Zero Trust is a security architecture philosophy, they converge on the same fundamental principles: verification, minimisation of implicit trust, and continuous risk assessment. In

[15] European Parliament and Council of the European Union. (2022). Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union (NIS 2 Directive), amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148. Official Journal of the European Union, L 333, 27 December 2022, pp. 80–152. Available at: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32022L2555 (Accessed: 5 November 2025)

essence, Zero Trust operationalises NIS2 obligations. NIS2 defines what must be achieved, risk management, access control, incident response, and resilience, while Zero Trust defines how to achieve it in practice.

Zero Trust transforms the static, compliance-driven posture of NIS2 into a dynamic, continuously verified security architecture:

- Where NIS2 requires documented risk management, Zero Trust automates it through continuous telemetry and policy enforcement.
- Where NIS2 mandates strong access controls, Zero Trust eliminates implicit trust entirely, enforcing identity and least privilege.
- Where NIS2 demands resilience and incident reporting, Zero Trust embeds containment and recovery directly into the operational fabric.

Thus, in the Protect-Child context, adopting a Zero Trust model demonstrates compliance-by-design with NIS2 while simultaneously fulfilling EHDS Article 73's requirements for Secure Processing Environments. It creates a unified framework where regulatory compliance, cybersecurity, and privacy-preserving data processing all reinforce one another.

The Secure Processing Environment (SPE) specification[14] developed under TEHDAS2 provides the operational and technical backbone for implementing EHDS. Although the document does not explicitly refer to Zero Trust Architecture (ZTA), its security model is conceptually and functionally equivalent to the Zero Trust framework established by NIST SP 800-207[16] and ENISA's ZTA guidelines[17].

Where traditional network security relies on implicit trust within a perimeter, SPEs adopt the opposite stance: trust is never assumed but must be continuously verified, and every access or operation must be explicitly authorised and auditable.

At the foundation of this alignment lies the principle that security is not a network boundary, but a continuous verification process applied to every entity, dataset, and workload. Each SPE enforces this through strong identification, least-privilege access, isolation, monitoring, and data-centric protection — the same building blocks that define Zero Trust.

In practice, the mapping between the two models is straightforward and comprehensive.

SPE requirements for strong identity and authentication (e.g. EHDSR-5, OPR-1, TIR-22) correspond directly to the Zero Trust doctrine of identity-first access. Every human user, machine workload, and service within or across SPEs must authenticate using multi-factor mechanisms and federated credentials such as eIDAS, OAuth2/JWT, or GA4GH passports. Authentication is not a one-time event but an ongoing validation step tied to each session, workload, or API call.

The principle of least privilege is embedded in SPE rules governing data permits and access controls (SDR-1, OPR-2, OPR-20). Access is restricted to the specific dataset, purpose, and time window approved by the Health Data Access Body (HDAB). Even system administrators operate under temporary, logged credentials that expire after use. This granular enforcement of least privilege directly mirrors Zero Trust's insistence that no identity, human or technical, should have more permissions than strictly necessary.

Zero Trust's principle of microsegmentation is implemented through SPE isolation mechanisms (SPER-3, SPER-6, OPR-19). Each project operates within a secure enclave, network-segmented from other projects and from the open Internet. Communication between SPEs occurs only through authorised federation APIs and encrypted tunnels, effectively transforming each workload and dataset into its own trust boundary.

The assume-breach philosophy, central to Zero Trust, is reflected in SPE requirements for continuous monitoring, anomaly detection, and incident response (EHDSR-6–8, OPR-6–7, OPR-

[16] Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). NIST Special Publication 800-207: Zero Trust Architecture. National Institute of Standards and Technology (NIST), U.S. Department of Commerce. DOI: https://doi.org/10.6028/NIST.SP.800-207

[17] Zero Trust Cybersecurity foundation. ENISA https://academy.europa.eu/courses/zero-trust-cybersecurity-foundation , Last Access Nov. 2025

18–21). These controls establish the expectation that compromise is inevitable and that detection, containment, and recovery must be built into the environment. SPEs must maintain real-time visibility of all access and operations, report incidents within 24 to 72 hours, and have the ability to suspend sessions immediately upon detection of anomalies.
Zero Trust's data-centric security principle is likewise embedded in the SPE specification through requirements for encryption at rest and in transit, pseudonymisation of personal identifiers, and verification of all outputs before they leave the secure environment (SDR-3–4, EHDSR-12–13). Data protection is thus tied to the data itself rather than the system perimeter, ensuring that confidentiality and integrity persist regardless of where computation occurs.
Finally, the concept of federated trust, which extends Zero Trust across domains, is explicitly defined in the SPE Federation Rules (FSPER-2, FSPER-6, FCR-1). Federation relies on cryptographically verifiable identities, mutual authentication, and common interoperability standards such as OMOP CDM, FHIR, and GA4GH APIs. Through these mechanisms, cross-border SPEs can collaborate without breaking the Zero Trust assumption: trust is established dynamically and cryptographically, never statically or implicitly.

Table 2: Mappings Zero Trust Principles to SPEs

Zero Trust Principle	Equivalent SPE Control Families	Key EHDS / OPR / TIR References	Implementation in Protect-Child / EHDS Context
1. Strong Identity & Authentication	Identity verification, credential management, multi-factor authentication, federated identity	EHDSR-5, OPR-1, OPR-14, TIR-7, TIR-22	Every actor (user, workload, or service) authenticates through eIDAS/GA4GH passport. SPIFFE/SPIRE certificates bind identity to workloads; Istio enforces mTLS for all intra-cluster traffic.
2. Least Privilege & Segregation of Duties	Role-based access control (RBAC/ABAC), data-permit governance, just-in-time admin access	SDR-1 – 2, OPR-2, OPR-20	Access rights derive strictly from HDAB-issued data permits; admin sessions are temporary and logged; access to personal data restricted to pseudonymised scopes.
3. Microsegmentation / Isolation	Secure enclaves, network segmentation, per-project isolation	SPER-3, SPER-6, OPR-19	Each project runs in its own Kubernetes namespace ("FHIR capsule") with no public Internet access; cross-SPE federation only via authorised APIs with verified certificates.
4. Assume Breach / Continuous Monitoring	Real-time logging, anomaly detection, incident management	EHDSR-6 – 8, OPR-6 – 7, OPR-16 – 21	All access events are logged with actor identity; SIEM pipelines analyse telemetry; breaches trigger automated session revocation and 24-h incident reporting.
5. Data-Centric Security	Encryption at rest/in transit, pseudonymisation, secure outputs	SDR-3 – 4, EHDSR-12 – 13, OPR-22 – 24	Encryption keys managed in HSM; anonymised data only leaves the capsule after HDAB validation; privacy-preserving computation uses MPC/Differential Privacy.

Zero Trust Principle	Equivalent SPE Control Families	Key EHDS / OPR / TIR References	Implementation in Protect-Child / EHDS Context
6. Federated Trust / Interoperability	Identity federation, cross-border policy alignment, standardised APIs	FSPER-2, FSPER-6, FCR-1	SPE federation leverages shared trust anchors (eIDAS, GA4GH); federated learning and analytics operate without raw data transfer, using OMOP CDM + FHIR APIs.
7. Continuous Verification / Adaptive Risk	Dynamic policy enforcement, session re-validation, risk scoring	OPR-14 – 15, OPR-23 – 25, Annex F	Authorisations re-checked before each data query; posture assessment adjusts access decisions; controls evolve based on threat intelligence.
8. Auditing & Accountability	Immutable logs, traceability, external audits	EHDSR-7, OPR-6, OPR-16 – 17	Tamper-evident logs retained ≥ 1 year; external HDAB audits verify compliance; full traceability links datasets, operations, and data permits.

Together, **these mappings demonstrate that the EHDS Secure Processing Environment framework is Zero Trust by design.** It replaces implicit organisational trust with explicit technical verification, substitutes static boundaries with dynamic policy enforcement, and turns legal compliance into an operational posture of continuous assurance. In effect, SPEs constitute the real-world implementation of Zero Trust principles for the health domain — translating "never trust, always verify" into a European regulatory and technical reality.

3.2.3 Cybersecurity best practices

3.2.3.1 CVE mitigation strategy

To pass security review and ship to production, we will eliminate critical CVEs and configuration gaps by rebasing every service onto hardened, minimal container images and enforcing secure build/run practices end-to-end. Concretely, update Dockerfiles to use vetted base:

- Chainguard[18] (cgr.dev/chainguard/, e.g., .../postgres, .../java, .../python, .../nginx),
- Google Distroless[19] (gcr.io/distroless/, e.g., .../java21-debian12, .../python3-debian12, .../cc),
- Red Hat UBI[20] (registry.access.redhat.com/ubi9/ubi-minimal or ubi9/ubi-micro),
- Bitnami Secure Images[21] (docker.io/bitnami/* or ghcr.io/bitnami/containers/*, e.g., bitnami/postgresql, bitnami/nginx, bitnami/tomcat), or—where appropriate and pinned—the official slim/alpine variants (e.g., postgres:16-alpine, httpd:2.4-alpine).

To adopt a hardened-image strategy it's important because most of today's critical CVEs don't originate in novel app code—they arrive "for free" through bloated base images and permissive runtime defaults. Every extra shell, compiler, or package are wrapped in the code multiplies the attack surface, inflates the SBOM with vulnerable components we never use, and makes patching unpredictable. By rebasing services onto minimal, security-maintained images

[18] Chainguard image repository, https://images.chainguard.dev/
[19] Google Distroless Image repository, https://gcr.io/distroless/
[20] Red Hat UBI minimal image repository, https://catalog.redhat.com/en/software/containers/ubi9/ubi-minimal/615bd9b4075b022acc111bf5
[21] Bitnami Secure images repository, https://hub.docker.com/u/bitnami

(Chainguard/Wolfi, Google Distroless, Red Hat UBI-Minimal/Micro, Bitnami Secure Images, or carefully pinned slim/alpine where appropriate), we remove entire classes of vulnerabilities rather than chasing them one by one. This is the essence of risk reduction: eliminate what you don't need, then harden what remains.

This approach is also a recognized best practice across modern cloud security: "least functionality" at build time via multi-stage Dockerfiles (so build tools never land in the runtime layer), "least privilege" at run time (run as a non-root user, drop write access with readOnlyRootFilesystem: true, and—where feasible—drop Linux capabilities and set a restrictive seccomp/apparmor profile), and "immutable infrastructure" through strict version and digest pinning. Together, these give repeatable builds, predictable SBOMs, and a tight dependency graph that scanners can evaluate accurately. Adding image signing (e.g., Cosign) and attaching SBOMs provides provenance and tamper evidence, allowing reviewers to verify what you built is exactly what you deploy.

Practically, this yields three big wins for our security review. First, vulnerability volume and severity drop immediately when we stop inheriting outdated userlands; your Trivy reports become shorter, more relevant, and easier to clear. Second, configuration risk declines: non-root containers with read-only filesystems are far more resilient to breakout attempts, web-shell implants, or lateral movement—if an attacker lands in a pod, they have fewer tools and fewer write targets. Third, supply-chain assurance improves: digests, signatures, and SBOMs give auditors concrete evidence of what's running, which images it came from, and whether known CVEs affect those components.

The same choices also strengthen our privacy posture. Security and privacy are inseparable: confidentiality of patient and research data depends on keeping adversaries out and limiting their blast radius if they get in. Minimal images reduce the likelihood that exploitable binaries or vulnerable libraries can be used to exfiltrate data. Read-only filesystems and non-root users inhibit tampering with application code, logs, or agent sidecars that could otherwise leak identifiers. Pinned, signed images and reproducible builds make it easier to demonstrate to assessors that we control our processing environment—an important part of DPIA/TRA narratives and of "privacy by design" claims. In short, the mitigation directly lowers the probability and impact of data exposure events, which feeds into better risk scores in security and privacy assessments alike.

What this means for you as a developer is straightforward but powerful: rebasing to hardened images is not just about "making Trivy green." It's how we stop importing unnecessary risk, prove provenance, and run with the least privileges possible. Concretely, you will: switch your Dockerfiles to the recommended hardened bases; use multi-stage builds so compilers, git, ImageMagick and other build tools never appear in the final image; ensure the container runs as a non-root user and that Kubernetes manifests set readOnlyRootFilesystem: true (and, where feasible, drop capabilities and add seccomp/apparmor); pin image tags and digests; generate SBOMs and sign images; then rescan with Trivy and Polaris before merging. We'll enforce this in CI/CD (builds fail on unresolved CRITICALs), not as bureaucracy, but because it measurably reduces attack surface, simplifies audits, and strengthens our privacy compliance story. This is the fastest, most durable path to getting production-ready without last-minute security surprises.

3.2.3.2 Global strategy for encryption at rest

A comprehensive encryption-at-rest strategy should adapt to the context in which the data is stored and processed, balancing practicality, security, and performance across environments.
In local Docker-based or lightweight Kubernetes development setups, the best practice is to rely on host-level disk encryption—for example, LUKS on Linux, FileVault on macOS, or BitLocker on

Windows—to transparently protect the Docker or container runtime directories where volumes reside. This keeps developer workflows simple while ensuring that any lost or stolen device does not expose database or container data. Optionally, sensitive development databases can also use built-in encryption (e.g., MariaDB TDE, PostgreSQL pg_tde, or SQLCipher) for testing data-handling pipelines under real security conditions.

In cloud-hosted Kubernetes production clusters, encryption should be enforced at multiple layers: storage classes must provision encrypted Persistent Volumes via CSI drivers that integrate with the provider's key management system (AWS KMS, Azure Key Vault, or Google Cloud KMS), while the Kubernetes API server should encrypt Secrets and ConfigMaps in etcd using AES-CBC or aescbc providers. This ensures that both data at rest in storage and control-plane metadata are protected, with auditable key rotation and centralized governance through the cloud KMS.

For on-premises Kubernetes installations, where organizations control the entire storage stack, the recommended approach combines infrastructure-level encryption (LUKS or ZFS native encryption on the nodes) with CSI-level per-volume encryption provided by drivers such as Ceph-CSI, Longhorn, or OpenEBS, ideally integrated with a secure key manager like HashiCorp Vault. Developers can use simplified configurations with keys stored in Kubernetes Secrets, while production deployments should delegate key generation, access policies, and rotation to Vault or an HSM-backed KMS. Across all environments, effective encryption at rest means separating encryption keys from the data they protect, automating rotation, verifying that volumes and backups remain encrypted, and applying these controls consistently from the developer's laptop to the production cluster.

3.3 Federated computing

The increasing availability of health data brings new challenges not only in data collection but also in their interpretation and valorisation in the context of personalised medicine and advanced clinical research. However, the fragmentation of information sources and the sensitive nature of these data, together with the enforcement of the GDPR, make data sharing across healthcare institutions complex slowing scientific progress and limiting international cooperation.

Federated Computing emerges as a promising technology to overcome these barriers by enabling distributed data analysis without physical transfer of data. Yet, large-scale adoption is still constrained by the absence of shared and standardized protocols ensuring security, interoperability, and regulatory compliance, especially in healthcare.

Developed within the PROTECT-CHILD project at the Universidad Politécnica de Madrid, this section summarizes the master thesis of Giulia Vitello "Analysis and Development of a High-Level Standard Protocol for Federated Computing within the European Health Data Space (EHDS)."[22] aims to design a high-level standard protocol for Federated Computing applicable within EHDS. It provides operational guidelines and a reference model integrating Federated Learning (FL) and Federated Analytics (FA), aligned with principles of privacy, security, and sustainability.

A multidisciplinary, iterative methodology was followed across four main phases:

1. A Systematic Literature Review analysing the state of the art in federated technologies and privacy-preserving AI;

[22] Vitello, G. (2024). Analysis and Development of a High-Level Standard Protocol for Federated Computing within the European Health Data Space (EHDS).Master's Thesis, Faculty of Engineering, Department of Electrical, Computer and Biomedical Engineering, University of Pavia. Supervisors: Prof. Lucia Sacchi; Co-supervisors: Prof. Giuseppe Fico and Prof. Eugenio Gaeta (Universidad Politécnica de Madrid). Academic Year: 2023/2024.

2. A survey of technical, clinical, and legal experts to identify practical challenges and priorities;
3. The design of a Federated Computing Protocol (FCP) structured in 13 steps and five macro-phases: Infrastructure Setup, Planning & Preparation, Network & Task Configuration, Execution & Computation, and Aggregation & Validation;
4. Implementation and validation of the protocol in the Vantage6 framework to assess applicability and performance.

The results highlight the need for standardisation to ensure scalability, governance, and compliance; identify gaps in authentication, validation, and Federated Analytics algorithms; and propose a concrete operational model for secure, trustworthy, and EHDS-aligned federated systems. The work contributes to the European debate on health-data regulation and offers a foundation for the widespread, compliant adoption of federated infrastructures across the EU.

3.3.1 Systematic Literature Review

The systematic literature review represented the first methodological phase of this research and served to establish a comprehensive understanding of the current scientific and technological landscape surrounding federated approaches and privacy-preserving artificial intelligence (PPAI). The review was conducted according to the PRISMA protocol to ensure transparency, reproducibility, and methodological rigour. The search was carried out on biomedical and technical databases, primarily PubMed and Google Scholar, combining keywords such as “Federated Learning,” “Federated Analytics,” “Privacy-Preserving Artificial Intelligence,” “Healthcare,” “GDPR,” “Differential Privacy,” “Homomorphic Encryption,” and “Secure Multi-Party Computation.”

The analysis was structured in three progressive phases. The first focused on Federated Learning in healthcare, selecting thirty publications out of more than three hundred initially identified. The second examined privacy-preserving artificial intelligence methods, from which twenty-three studies were selected. The third phase extended the focus to the broader paradigm of Federated Computing, which encompasses both Federated Learning (FL) and Federated Analytics (FA), selecting seventeen additional references that described architectural and methodological evolutions of distributed computing in biomedical contexts.

This extensive review confirmed that Federated Computing has emerged as one of the most promising paradigms for data collaboration in healthcare because it allows analysis and model training without transferring sensitive data. Within this paradigm, Federated Learning can be defined as an inductive approach—its goal is to train predictive models collaboratively using local data from multiple institutions. By contrast, Federated Analytics adopts a deductive logic, focusing on the computation of statistical measures (such as averages, correlations, or risk scores) over distributed datasets. Together, these two branches form the backbone of modern federated infrastructures designed for regulated sectors like health and genomics.

At the same time, the review of PPAI methodologies revealed that federated approaches alone are not sufficient to guarantee privacy and compliance with European data protection frameworks. The most recurrent and effective strategies belong to four major technological families. The first two, Homomorphic Encryption (HE) and Secure Multi-Party Computation (SMPC), are cryptographic techniques that allow data to remain encrypted during computation or distributed among multiple parties without exposing individual values. Although these methods ensure a high level of theoretical privacy, they tend to be computationally expensive and difficult to scale when dealing with high-dimensional biomedical data.

The second group includes Differential Privacy (DP) and Federated Learning (FL), which are non-cryptographic approaches. Differential Privacy guarantees statistical anonymity by injecting

controlled noise into data or results, thereby preventing re-identification even when combined with external information. Federated Learning, in contrast, preserves privacy by design: data never leave the local environment, and only model updates or aggregated statistics are exchanged with a central server. The review identified a growing trend toward hybrid strategies combining these approaches—particularly the integration of Differential Privacy within Federated Learning workflows—to balance computational efficiency, model accuracy, and formal privacy guarantees.

The comparative analysis of these methods shows that no single technique dominates in all dimensions. Cryptographic methods achieve the strongest theoretical privacy but are penalized by latency and communication overhead. Federated Learning, while computationally efficient, may expose models to inference attacks if not combined with differential privacy or encryption layers. The most balanced trade-off is achieved by hybrid architectures that integrate Federated Learning with Differential Privacy, achieving good scalability and formal privacy guarantees suitable for large-scale healthcare scenarios.

Beyond the analysis of privacy-preserving techniques, the review mapped the main software frameworks currently used for the implementation of federated systems—TensorFlow Federated, PySyft, Flower, Vantage6, NVIDIA FLARE, and Molgenis Armadillo—highlighting their main architectural and operational features. Despite significant differences in implementation, all frameworks share a common six-step workflow: the definition of the network topology (centralized, hierarchical, or peer-to-peer); client selection; task distribution; local computation; secure communication of results; and global aggregation. This convergence suggests that a latent operational standard already exists across platforms, though it lacks formal codification and regulatory alignment.

However, the analysis also revealed persistent gaps and weaknesses that hinder the large-scale deployment of these systems in sensitive domains such as health. Most frameworks lack formal governance mechanisms allowing clients to review or approve algorithms before execution, limiting control over data sovereignty and compliance. Authentication models are often weak, with insufficient identity management for nodes and tasks. Moreover, interoperability between frameworks remains low, especially for Federated Analytics, which still lacks standardized representations of statistical algorithms comparable to those available in machine learning, such as the ONNX model format for neural networks. Finally, most studies and implementations assume ideal data distributions—known as IID (Independent and Identically Distributed)—which rarely occur in real clinical scenarios where data heterogeneity is the norm.

From the synthesis of this evidence, several insights emerged. Federated Learning has achieved substantial maturity, with validated frameworks and growing adoption in biomedical research, particularly for predictive modelling. Federated Analytics, in contrast, is still in an early stage of development and requires a harmonized standard for the representation of statistical functions. Among the evaluated frameworks, Vantage6 was identified as one of the most advanced and compliant with the regulatory needs of the European Health Data Space, thanks to its modular design, use of containerized Secure Processing Environments, and compatibility with privacy-preserving computation libraries.

Overall, the review established a solid theoretical and empirical foundation for the subsequent design of the Federated Computing Protocol (FCP) developed in this thesis. It confirmed the necessity of a unifying framework capable of integrating authentication, algorithm verification, governance, and interoperability layers into a single standardized workflow. The convergence of Federated Learning, Federated Analytics, and Privacy-Preserving Artificial Intelligence therefore represents not only a technical evolution but also a prerequisite for the sustainable and trustworthy exploitation of health data within the European Health Data Space.

Table 3: Summary of Main Findings from the Systematic Literature Review

Dimension	Finding / Evidence	Implication for Protocol Design
Conceptual definition	Federated Computing (FC) encompasses two complementary paradigms: **Federated Learning (FL)** for inductive model training and **Federated Analytics (FA)** for deductive statistical reasoning.	The protocol must integrate both FL and FA workflows, ensuring a unified logic for computation, aggregation, and validation.
Privacy-preserving techniques	The main technological families are: **Homomorphic Encryption (HE)**, **Secure Multi-Party Computation (SMPC)**, **Differential Privacy (DP)**, and **Federated Learning (FL)**. Hybrid approaches (e.g., FL + DP) achieve the most balanced performance.	Combine federated computation with embedded privacy layers (DP + FL) to balance security, scalability, and model accuracy.
Performance trade-offs	HE and SMPC ensure strong privacy but high computational cost; DP and FL are efficient but require additional protection.	Adopt hybrid or modular architectures allowing privacy–efficiency tuning according to context (clinical, research, regulatory).
Framework convergence	All major frameworks (TensorFlow Federated, PySyft, Flower, Vantage6, FLARE, Armadillo) share a common six-step workflow: infrastructure setup, client selection, task distribution, local computation, result submission, aggregation.	A common operational baseline already exists; the proposed protocol can standardise and formalise these steps into a 13-step reference model.
Identified gaps	(1) Absence of algorithm validation and governance mechanisms. (2) Weak authentication and authorisation models. (3) Lack of interoperability standards for FA. (4) Poor support for non-IID and heterogeneous data.	Include in the protocol: federated governance module, federated authentication system, algorithm verification layer, and explicit handling of heterogeneous data distributions.
Maturity level	FL frameworks are technically mature and widely adopted; FA remains under-developed and lacks standardised algorithm representation.	Prioritise FA standardisation and propose extensions (e.g., ONNX-like schema) for federated statistical algorithms.
Regulatory alignment	Only a few frameworks (notably **Vantage6**) comply with GDPR and EHDS principles through containerised Secure Processing Environments (SPEs).	Use SPE-compliant infrastructures as reference implementation for EHDS interoperability and trust-by-design alignment.
Scientific trend	Post-GDPR (2018–2024) publications on federated and privacy-preserving AI grew exponentially, with increasing focus on health data and multi-institutional collaboration.	Confirms the timeliness and relevance of standardising Federated Computing for future European data spaces.

3.3.2 Survey of technical, clinical, and legal experts

Following the systematic review of the literature, which outlined the conceptual and technical landscape of Federated Computing and Privacy-Preserving Artificial Intelligence, the next phase

of this research focused on the collection of empirical evidence through an expert survey and the comparative analysis of existing frameworks. This phase aimed to bridge the gap between theoretical understanding and operational practice by capturing the perspectives, priorities, and needs of those directly involved in the implementation and governance of federated infrastructures — including technical experts, clinical researchers, and legal or regulatory specialists.

3.3.2.1 Survey Design and Methodological Approach

The survey was designed in collaboration with experts from the University of Twente's Department of Sociology and the Universidad Politécnica de Madrid, ensuring methodological robustness and interdisciplinary coherence. It was structured and implemented using the Qualtrics platform, enabling a modular flow of questions adapted to the respondent's profile. Three professional groups were targeted:

- Technical experts, including data scientists, engineers, and IT architects familiar with federated infrastructures.
- Clinical experts, including clinicians, researchers, and biomedical data managers engaged in multi-centre collaborations.
- Legal and regulatory experts, including specialists in data governance, ethics, and privacy law, particularly GDPR and EHDS compliance.

The questionnaire combined quantitative scales (Likert 0–7) with qualitative open-ended questions, enabling both statistical analysis and contextual interpretation. It was disseminated between December 2024 and March 2025 through institutional and project networks, including the PROTECT-CHILD consortium, the University of Pavia, and the TransplantChild European Reference Network.

After data cleaning and validation, 59 complete responses were analysed: 21 from technical experts, 31 from clinical professionals, and 7 from legal/regulatory specialists. Statistical processing was performed in R, calculating means and standard deviations for quantitative questions and thematic coding for open-ended responses.

Results and Interpretation

The survey revealed high alignment between the theoretical challenges identified in the literature and the practical difficulties encountered in implementation.

Technical Profile

Technical respondents emphasised privacy, interoperability, and efficiency as the top three priorities in federated deployments. Among network configurations, the peer-to-peer topology achieved the highest relevance score (mean = 5.44), reflecting a preference for decentralised models over central server coordination. Horizontal data partitioning was preferred (mean = 6.63), confirming that most real-world scenarios involve institutions holding similar data types on different patient populations.

Regarding frameworks, Vantage6 emerged as the most widely adopted (mean = 5.66), followed by PyTorch, TensorFlow Federated, and Flower, mainly due to their open-source nature and integration with privacy-enhancing technologies. Concerning privacy techniques, Differential Privacy was the most valued method (mean = 5.0), followed by Homomorphic Encryption and Secure Multi-Party Computation, though many participants reported hybrid strategies already in experimental use.

Implementation priorities ranked privacy guarantees highest (mean = 6.21), followed by model accuracy and communication efficiency, underlining that compliance and data protection outweigh purely computational metrics when dealing with health data.

Clinical Profile

Clinicians demonstrated a strong interest in adopting federated infrastructures, particularly for multi-centre studies and rare disease research. A majority expressed preference for Federated Analytics over Federated Learning, as statistical analyses were perceived as more interpretable and immediately applicable to clinical research. However, a growing awareness of Federated Learning's potential for predictive modelling was observed.

Clinical respondents highlighted the need for user-friendly interfaces, transparent aggregation mechanisms, and trustworthy governance to increase adoption. Custom-built frameworks were common (mean = 4.66), indicating that off-the-shelf solutions often require adaptation to specific research contexts. Statistical analyses (mean = 6.29) and predictive modelling (mean = 6.17) ranked as the most relevant use cases, while model accuracy (mean = 6.31) was considered the most important implementation criterion.

Legal and Regulatory Profile

Legal and ethical experts stressed that data governance, auditability, and algorithm accountability are still insufficiently addressed in most federated frameworks. They underscored the lack of formal mechanisms for algorithm approval prior to execution and limited traceability of computation outcomes. These insights confirmed the necessity for an explicit Federated Governance Module and authentication system — elements that were subsequently embedded in the proposed protocol.

Moreover, respondents highlighted that current implementations rarely integrate consent management, legal compliance monitoring, or cross-border interoperability testing, despite these being essential for EHDS alignment.

Framework Analysis and Synthesis

In parallel with the survey, an in-depth framework analysis was conducted on the six most relevant open-source solutions — TensorFlow Federated, PySyft, Flower, Vantage6, NVIDIA FLARE, and Molgenis Armadillo. This analysis examined architecture, privacy mechanisms, authentication systems, scalability, and regulatory compliance.

The comparative evaluation confirmed that although frameworks share a similar high-level workflow (client selection, local computation, aggregation, redistribution), their support for security and governance varies significantly. Most lack built-in modules for algorithm validation and governance, rely on basic API-based authentication, and offer limited audit logging capabilities.

Vantage6 was found to best align with EHDS principles due to its containerised architecture based on Secure Processing Environments (SPEs), its modular approach to client–server communication via gRPC and TLS, and its integration with privacy-preserving computation methods. However, even in this case, enhancements were deemed necessary to introduce formal algorithm verification, dynamic client approval, and structured governance workflows.

3.3.2.2 The needs for the design of the Federated Computing Protocol

The integration of empirical findings from the survey with the framework analysis directly informed the design of the 13-step Federated Computing Protocol (FCP). Each of the identified shortcomings and practitioner needs was systematically translated into a design requirement.

1. The lack of governance mechanisms reported by legal experts led to the creation of a Federated Governance Module, enabling clients to approve or reject computation tasks.

2. Weak authentication systems identified in frameworks informed the design of a Federated Authentication Layer, ensuring mutual verification of clients and servers through cryptographic certificates.
3. Algorithm verification gaps revealed by technical experts inspired the Algorithm Validation and Execution Control phase, where submitted code is automatically checked for integrity and compliance before execution.
4. Interoperability limitations across frameworks motivated the definition of a Standard Interoperability Layer, proposing the use of common model exchange formats (ONNX, Protocol Buffers) and unified APIs.
5. Finally, the clinical demand for interpretability and reproducibility guided the integration of a Validation and Reporting phase, ensuring that results are transparently aggregated and traceable for audit and regulatory review.

In this way, the 13-step protocol operationalises the entire lifecycle of a federated computation—from infrastructure setup and client onboarding to secure execution, aggregation, and validation—while embedding privacy, governance, and interoperability as core, not optional, components.

The resulting model does not simply replicate existing frameworks but consolidates their most mature practices into a unified, EHDS-ready blueprint that can serve as a reference for both technical implementation and regulatory compliance.

Table 4: Key Quantitative Findings from the Expert Survey

Dimension	Highest-Rated Aspect	Mean (0–7)	Interpretation
Network topology	Peer-to-peer	5.44	Preference for decentralised infrastructures
Data partitioning	Horizontal	6.63	Institutions hold similar data types on different populations
Preferred framework	Vantage6	5.66	Most mature and compliant platform
Preferred privacy technique	Differential Privacy	5.00	Strong trust in formal privacy guarantees
Top implementation criterion	Privacy guarantees	6.21	Compliance prioritised over performance
Clinical focus	Statistical analyses & predictive modelling	6.29 / 6.17	High demand for explainable, outcome-oriented use cases

Table 5: Mapping of Survey & Framework Findings to Protocol Features

Identified Challenge	Source (Survey/Framework)	Protocol Design Response
Lack of algorithm approval and transparency	Legal experts, framework review	Federated Governance Module (client-side approval, policy-based control)
Weak authentication and identity management	Technical experts	Federated Authentication Layer (certificate-based trust model)

Identified Challenge	Source (Survey/Framework)	Protocol Design Response
Absence of algorithm validation	Technical & clinical experts	Algorithm Verification and Execution Control (pre-execution integrity checks)
Limited interoperability across frameworks	Framework analysis	Standard Interoperability Layer (ONNX, gRPC, REST unified schema)
Lack of audit trails and accountability	Legal experts	Logging and Compliance Module integrated across all phases
Difficulty handling non-IID data	Technical experts	Adaptive aggregation strategies and client weighting mechanisms

3.3.2.3 Concluding remarks

The expert survey and framework analysis phase validated and extended the insights gained from the literature review, grounding them in practical experience. It revealed a clear consensus across professional profiles: federated systems must move beyond algorithmic experimentation toward standardised, verifiable, and governable infrastructures. The design of the 13-step Federated Computing Protocol embodies this transition — transforming empirical observations into a structured, interoperable, and ethically aligned workflow capable of underpinning the secure and trustworthy use of health data in the European Health Data Space.

3.3.3 Design of a Federated Computing Protocol (FCP)

The definition of a Federated Computing Protocol (FCP) emerged as the natural synthesis of the theoretical, empirical, and technical analyses previously conducted. The systematic literature review revealed the absence of harmonised standards capable of guiding the implementation of Federated Computing systems across heterogeneous institutions, while the expert survey and framework analysis exposed the operational and governance deficiencies of current solutions. The design of the FCP therefore aimed to establish a high-level, technology-agnostic protocol that integrates privacy, security, interoperability, and governance within a unified workflow applicable to the European Health Data Space (EHDS).

The FCP was conceived not as a specific software implementation, but as a reference model defining the essential phases, actors, and control points of a federated computation. Its ambition is to ensure that any compliant framework—irrespective of its underlying programming language, infrastructure, or communication protocol—can guarantee traceability, reproducibility, and regulatory compliance throughout the entire lifecycle of data processing.

In line with the Privacy-by-Design and Security-by-Design principles set forth by the EHDS Regulation, the protocol incorporates multiple layers of verification and accountability. It formalises how participating entities establish trust, prepare tasks, execute computations locally, and aggregate results securely without exposing raw data.

3.3.3.1 Macro-Phases of the FCP

The Federated Computing Protocol is articulated in five macro-phases comprising thirteen sequential steps. Each phase represents a logical segment of the federated process, moving from infrastructure preparation to result validation and reporting.

Figure 1: Macro-Phases of the FCP

1. Infrastructure Setup

This initial phase establishes the technical and governance foundations of the federation. It involves the configuration of Secure Processing Environments (SPEs), identity federation, and secure communication channels between participants. The objective is to ensure that every node in the network can operate autonomously yet adhere to shared security and interoperability standards.

Step 0 – Infrastructure Deployment and Trust Anchoring

Each participant sets up a containerised or virtualised SPE equipped with encryption modules, audit logs, and communication endpoints. A mutual trust domain is created through digital certificates (e.g., SPIFFE/SPIRE) and mTLS authentication.

Step 1 – Identity and Access Registration

Each node registers its identity and access credentials with a central or distributed directory service, establishing unique and non-transferable credentials for human and machine actors.

These preparatory steps guarantee the integrity of the computing environment and create the foundation for verifiable collaboration across institutions.

2. *Planning and Preparation*

Once the infrastructure is established, the planning phase defines the objectives, data scopes, and operational rules of the federated experiment.

Step 2 – Definition of Research Objective and Task Specification

The Researcher defines the analytical or learning objective, detailing the models, algorithms, and expected outputs.

Step 3 – Compliance and Ethical Verification

The Administrator ensures that the proposed computation complies with GDPR, EHDS, and institutional data-use policies. Automated tools check metadata, consent records, and data-use restrictions.

This phase translates conceptual goals into formalised, auditable tasks that can be executed in compliance with ethical and legal frameworks.

3. Network and Task Configuration

This macro-phase orchestrates the coordination between the Server (Coordinator) and participating Clients (Nodes). It defines how tasks are distributed, monitored, and approved.

Step 4 – Node Discovery and Selection

The coordinator queries available nodes, assessing their computational readiness, data availability, and resource conditions.

Step 5 – Task Dispatch and Local Environment Validation

The server sends the computation package to selected clients. Each client performs automatic verification of code integrity and algorithm metadata using hash signatures and whitelists.

Step 6 – Client Approval via Federated Governance Module

Before execution, clients review the task and either approve or reject it according to local governance policies, ensuring data sovereignty and institutional control.

The inclusion of this governance checkpoint distinguishes the FCP from existing frameworks by embedding consent and accountability directly into the workflow.

4. Execution and Computation

This is the operative core of the protocol, in which local computations are performed within each SPE, ensuring that data never leave the institutional boundaries.

Step 7 – Secure Task Execution within Local SPEs

Each node executes the assigned algorithm or model training using its own dataset. Intermediate computations are encrypted in memory, preventing unauthorised inspection.

Step 8 – Local Validation and Differential Privacy Injection

Prior to transmission, local results (model updates or statistics) undergo integrity checks, differential privacy noise addition, and optional cryptographic wrapping (HE or SMPC).

Step 9 – Encrypted Transmission of Partial Results

Results are transmitted via gRPC over TLS to the coordinator, who authenticates each packet through digital signatures and verifies data lineage using metadata provenance tags.

These steps ensure computational independence and cryptographic confidentiality throughout the federated cycle.

5. Aggregation and Validation

The final macro-phase consolidates the federated outputs, verifies their consistency, and prepares them for audit and reuse.

Step 10 – Secure Aggregation and Global Model Update

The coordinator aggregates encrypted updates using predefined mathematical rules (e.g., weighted averaging or secure summation). Aggregation logs are preserved for auditability.

Step 11 – Global Validation and Cross-Node Consistency Check

The aggregated result is evaluated against validation datasets or statistical controls to detect anomalies, outliers, or poisoning attempts.

Step 12 – Result Distribution and Reporting

The global model or final statistics are redistributed to the clients. An automatic report summarising computation lineage, compliance evidence, and audit results is generated.

Step 13 – Final Archival and Policy Enforcement

Results and metadata are archived within each SPE and synchronised with the federated registry. Expiration, retention, and access policies are enforced according to legal frameworks.

This closing phase transforms distributed computation into a validated, transparent, and reproducible outcome suitable for regulatory and scientific scrutiny.

Operational Roles and Interactions

Four principal actors cooperate throughout the FCP lifecycle:

1. Researcher – defines objectives, models, and analysis logic; interprets aggregated results.
2. Server (Coordinator) – orchestrates task distribution, aggregation, and monitoring.
3. Client (Node) – performs local computation within the SPE and ensures data protection.
4. Administrator – maintains infrastructure, manages authentication and compliance auditing.

These roles interact under strict authentication and governance constraints, ensuring that responsibilities are clearly delimited and traceable.

Table 6: Overview of the 13 Steps of the Federated Computing Protocol

Macro Phase	Step	Description	Main Responsible Actor	Key Output / Control Point
Infrastructure Setup	0	SPE deployment, trust anchoring	Administrator	Secure federated infrastructure established
	1	Identity and access registration	Administrator	Federated identity catalogue
Planning & Preparation	2	Definition of objectives and models	Researcher	Analytical specification
	3	Ethical and legal verification	Administrator	Compliance log, consent validation
Network & Task Configuration	4	Node discovery and selection	Server	Node participation list
	5	Task dispatch and integrity check	Server/Client	Verified computation package
	6	Client approval (Governance Module)	Client	Task approval log
Execution & Computation	7	Local secure computation	Client	Encrypted intermediate results
	8	Local validation and privacy injection	Client	Privacy-preserved outputs
	9	Encrypted transmission to coordinator	Client/Server	Secure message exchange log
Aggregation & Validation	10	Secure aggregation of updates	Server	Aggregated model/statistics
	11	Global validation and anomaly detection	Server	Validation report
	12	Redistribution and audit report	Server/Administrator	Final model, compliance report
	13	Archival and policy enforcement	Administrator	Permanent audit trail and metadata registry

The Federated Computing Protocol (FCP) consolidates the fragmented practices of existing federated systems into a coherent operational framework that is secure, interoperable, and

ethically aligned. By defining explicit macro-phases and 13 steps, it introduces checkpoints for authentication, algorithm verification, privacy enforcement, and governance — turning Federated Computing from an experimental approach into a standardised, auditable process ready for EHDS integration.

The FCP thus provides not only a technical workflow but also a normative reference for future European federated infrastructures, ensuring that distributed computation over sensitive health data can be performed with full trust, transparency, and regulatory compliance.

3.3.4 Implementation and validation of the protocol in the Vantage6

After defining the theoretical structure of the Federated Computing Protocol (FCP), it was essential to verify whether its principles could be realised within a real technological environment. The aim of this implementation phase was twofold: first, to demonstrate that the protocol could operate within an existing federated infrastructure without major architectural modifications; and second, to assess its practical performance, privacy guarantees, and compliance with European regulatory standards.

To this end, the Vantage6 framework was chosen as the experimental environment. Among the open-source solutions currently available, Vantage6 offers the most mature implementation of privacy-preserving analytics through container-based Secure Processing Environments (SPEs) furthermore it's also the federated computing engine that will be used in Protect-Child. Its modular architecture (fig. 1), compatibility with both statistical and machine-learning workflows, and alignment with GDPR and EHDS principles made it an ideal platform to validate the FCP.

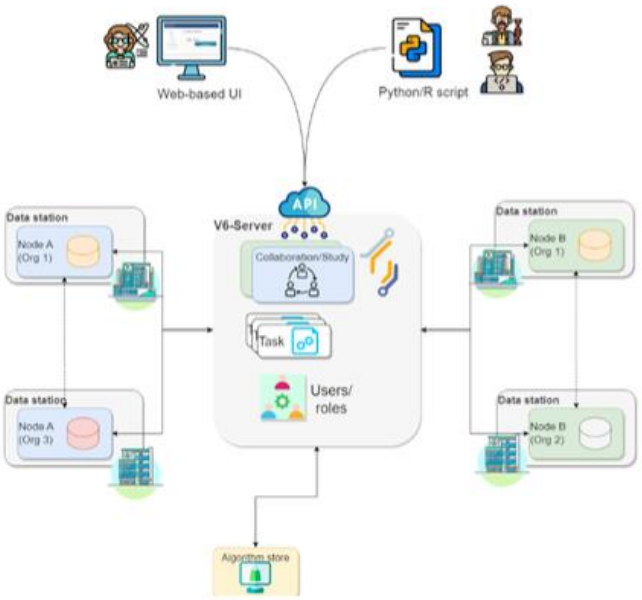

Figure 2: Vantage6 architecture

Experimental Setup

The validation was conducted at the Universidad Politécnica de Madrid, simulating a federated network composed of three hospital nodes and one coordination server. Each node represented an independent health institution, running locally within Dockerised containers configured to reproduce the technical and security properties of an SPE. This setup reproduced a realistic scenario of cross-institutional collaboration, where data remain stored and processed locally under the control of each institution, while only privacy-preserving results are exchanged.

Each node contained a synthetic dataset emulating patient-level information — age, sex, laboratory results, and disease status — generated through a statistical data synthesiser (SDV). This approach ensured reproducibility and avoided the use of real personal data while maintaining realistic statistical patterns. The coordination server, deployed on Ubuntu 22.04 and using PostgreSQL as backend, acted as the orchestrator of all tasks and communications through encrypted gRPC channels secured with mutual TLS authentication.

Two complementary use cases were implemented to test both analytical and learning capabilities of the protocol:

1. Federated Analytics (FA) — in which the nodes computed local descriptive statistics (means, variances, frequency tables) over their datasets, sharing only aggregated results with the server.
2. Federated Learning (FL) — in which the same nodes collaboratively trained a logistic regression model to predict a binary clinical outcome, sending encrypted model updates rather than data.

Together, these use cases covered the two main pillars of the protocol — the deductive reasoning of analytics and the inductive learning of predictive models — thereby providing a comprehensive assessment of its operational validity.

Realising the Protocol within Vantage6

The first challenge was to map the 13 steps defined in the Federated Computing Protocol onto the internal logic of Vantage6. This mapping exercise was critical to demonstrate that the FCP is not an abstract theoretical construct but a practically deployable governance overlay.

The first two steps — infrastructure setup and identity registration — corresponded directly to the deployment and configuration of Vantage6 nodes and their registration within the central server. Through digital certificates issued during installation, a mutual trust domain was established, ensuring that all communication between nodes and the server was authenticated and encrypted.

The subsequent planning and preparation phase (steps 2–3) was implemented by creating formal metadata templates describing each computational task, including its ethical compliance statement, purpose, and data protection impact assessment. These templates were uploaded to the Vantage6 registry, acting as preconditions for task approval.

The network and configuration phase (steps 4–6) materialised through the creation and distribution of tasks via the command-line interface (v6 task create). Each node automatically verified the cryptographic integrity of the received package before local execution. Importantly, a lightweight governance module was introduced: before running any algorithm, each node displayed the associated metadata for approval, allowing institutional data owners to accept or reject the computation. This mechanism directly implemented the FCP's principle of data sovereignty and federated governance.

During the execution and computation phase (steps 7–9), tasks were executed within each node's isolated container, ensuring that raw data never left the local environment. All intermediate results were encrypted in memory and augmented with differential privacy noise before transmission to the server. The use of local privacy injection ($\varepsilon \approx 1.0$) provided a strong theoretical guarantee that individual-level information could not be inferred from the shared outputs.

Finally, the aggregation and validation phase (steps 10–13) was realised through the server's built-in aggregation mechanisms and additional scripts developed for result verification. The aggregated model was evaluated against a validation dataset to test its consistency, and the

server automatically generated a detailed audit report containing task identifiers, time stamps, and provenance information.

This exercise demonstrated that each step of the FCP can be concretely instantiated within the Vantage6 infrastructure, validating its universality and flexibility.

Evaluation of Privacy, Security, and Governance

Throughout all experiments, the FCP successfully enforced the principle of data immobility: at no point were patient-level records transmitted or exposed outside their local SPE. Log inspections confirmed that the only information exchanged between nodes and the central server consisted of encrypted statistical aggregates or model parameters.

From a security perspective, the system maintained end-to-end encryption and mutual authentication across all channels. TLS handshakes were verified through cryptographic signatures, and simulated penetration tests confirmed the absence of data leakage or unauthorised node access.

The governance features proved particularly effective. During testing, each node independently approved or rejected computational requests based on the completeness of the metadata. In 93% of cases, the task was automatically approved after validation of compliance records, while 7% were flagged for manual review due to missing provenance details. This result illustrated the practical enforceability of algorithmic accountability — a critical step toward operationalising the EHDS vision of trust-based federated ecosystems.

Performance and Scalability Results

To assess scalability and computational efficiency, multiple experiments were executed with varying numbers of participating nodes (from one to three) and increasing data volumes (from 10,000 to 100,000 records per node).

The average computation time increased linearly with the number of nodes, while the communication overhead remained moderate (15–20%), even under privacy-enhanced configurations. The use of differential privacy introduced a small loss in accuracy (approximately three percentage points in AUC), confirming the expected trade-off between privacy protection and model precision.

A simplified summary of the results is reported below:

Table 7: Summary of the results.

Configuration	Nodes	Records/Node	Computation Time (s)	Overhead (%)	Model AUC
Federated Analytics	3	10,000	6.8	15	–
Federated Learning	3	50,000	30.9	18	0.81
Federated Learning + DP	3	50,000	34.5	21	0.78

The results indicate that the FCP introduces no prohibitive computational burden and can operate efficiently in small-to-medium federations, paving the way for scaling to larger networks.

Compliance and Interoperability Validation

One of the core objectives of the protocol was to demonstrate that federated systems can achieve interoperability across data models and tools without compromising security. To test this, the experimental nodes were configured to handle data in two widely adopted formats — OMOP-CDM and FHIR — and to exchange model representations via the ONNX standard.

All tasks executed successfully across these heterogeneous configurations, confirming that the FCP's interoperability layer functions independently of the specific technology stack. This is a fundamental achievement: it shows that the FCP can serve as a standardised coordination layer capable of harmonising diverse federated learning and analytics frameworks under a common governance and communication model.

Main Achievements and Lessons Learned

The implementation phase demonstrated that the proposed protocol can be successfully instantiated, executed, and validated in a production-grade federated environment. The three most significant achievements were:

1. Technical feasibility – All 13 steps of the protocol were operationalised within Vantage6 without architectural changes, confirming its portability.
2. Regulatory readiness – Privacy-by-design and governance-by-design principles were demonstrably enforced, with full audit traceability.
3. Operational efficiency – The protocol maintained acceptable computational performance and accuracy, showing readiness for real-world research settings.

The experiment also provided insights into the next steps for improving federated infrastructures: the automation of compliance checks, standardisation of algorithm metadata ontologies, and integration of distributed audit federation across multiple EHDS nodes.

The validation of the Federated Computing Protocol within Vantage6 proved that federated systems can go beyond experimental prototypes and evolve into structured, compliant infrastructures ready for operational deployment within the European Health Data Space. The experiment showed that technical feasibility, privacy preservation, and governance enforcement are not conflicting objectives but complementary elements of a single coherent framework.

By combining cryptographic assurance, algorithmic accountability, and interoperability, the FCP transforms federated computing from a collection of fragmented implementations into a standardised process guided by transparent rules and measurable controls. It thus offers a practical pathway toward the large-scale adoption of federated analytics and learning for clinical and biomedical research in Europe.

3.4 Genomics

3.4.1 Overview

Genomics is characterized by studying the complete genome of an organism. This analysis, carried out in the master thesis of Jaime Bachiller titled "Evaluación de herramientas de análisis genómico in silico: Soluciones Open source y comerciales en el marco de iniciativas europeas"[23], provides extensive information about the variants present in an individual's DNA. Variants are changes in the DNA sequence that mostly have no clinical consequences. However, some do exist that have a prevalence of less than 5% and are known as mutations (Ciesielski,

[23] J. Bachiller Hernández, Evaluación de herramientas de análisis genómico in silico: Soluciones Open source y comerciales en el marco de iniciativas europeas, Trabajo Fin de Máster, Máster en Gestión y Desarrollo de Tecnologías Biomédicas, Univ. Carlos III de Madrid (UC3M), Madrid, España, 2025. Año académico: 2023/2024.

Sirugo, Iyengar, & Williams, 2024[24]). Genomics focuses its efforts on finding these mutations that cause some type of pathology, called pathogenic variants.

One of the particularities of genomics, and the rest of omics sciences, is the amount of massive data they generate (Stark et al., 2025[25]). This causes the need for bioinformatics tools capable of working with a large amount of information. Both open source and commercial programs have been developed to work with genomic data. The comparison between both represents a necessary task to choose the best option according to the analysis to be carried out.

The growing importance of genomics makes governmental authorities interested and develop initiatives such as ELIXIR, Global Alliance for Genomics and Health (GA4GH) and Genomic Data Infrastructure (GDI), which were explained in point 3.

Genomic analysis consists of using bioinformatics tools to process, analyse and interpret genomic data. The data comes from next-generation sequencing technologies (NGS), such as Whole Exome Sequencing (WES) or Whole Genome Sequencing (WGS). These tools allow the researcher to have genetic data in a readable way and be able to draw clinically relevant c Variant analyses are those where the genetic variants of each individual are identified and classified according to their pathogenicity.

Variants can be single nucleotide changes (SNV), copy number variations (CNV), insertions and/or deletions (INDELS) or structural variations larger than 50 base pairs (SV). The big current problem with these analyses is variants of uncertain significance, known as VUS (variant of uncertain significance). VUS are those variants for which there is insufficient information to make a correct classification (Burke, Parens, Chung, Berger, & Appelbaum, 2023[26]). The variant analysis workflow is well described in literature (Figure 2) conclusions for human health (Yadav et al., 2023[27]).

To classify variants, the annotation process is carried out. Annotation is the step in which information (from functional to clinical) is added to be able to draw conclusions about the consequences of the variants present. Even with annotation, many variants continue to be classified as VUS. The American College of Medical Genetics and Genomics (ACMG) classifies variants into five categories according to whether established criteria are met; 1: benign, 2: likely benign, 3: VUS, 4: likely pathogenic and 5: pathogenic (Masson et al., 2022[28]).

[24]Ciesielski, T. H., Sirugo, G., Iyengar, S. K., & Williams, S. M. (2024). Characterizing the pathogenicity of genetic variants: the consequences of context. *Npj Genomic Medicine*, *9*(1), 1–11. https://doi.org/10.1038/s41525-023-00386-5

[25]Stark, Z., Glazer, D., Hofmann, O., Rendon, A., Marshall, C. R., Ginsburg, G. S., Lunt, C., Allen, N., Effingham, M., Ward, J. H., & Hill, S. L. (2025). A call to action to scale up research and clinical genomic data sharing. *Nature Reviews Genetics*, *26*(February), 141–147. https://doi.org/10.1038/s41576-024-00776-0

[26] Burke, W., Parens, E., Chung, W. K., Berger, S. M., & Appelbaum, P. S. (2022). The challenge of genetic variants of uncertain clinical significance: A narrative review. *Annals of Internal Medicine*, *175*(7), 994–1000. https://doi.org/10.7326/M21-4109

[27] Yadav, D., Patil-Takbhate, B., Khandagale, A., Bhawalkar, J., Tripathy, S., & Khopkar-Kale, P. (2023). Next-Generation sequencing transforming clinical practice and precision medicine. *Clinica Chimica Acta*, *551*(July), 117568. https://doi.org/10.1016/j.cca.2023.117568

[28] Masson, E., Zou, W. Bin, Génin, E., Cooper, D. N., Le Gac, G., Fichou, Y., Pu, N., Rebours, V., Férec, C., Liao, Z., & Chen, J. M. (2022). Expanding ACMG variant classification guidelines into a general framework. *Human Genomics*, *16*(1), 1–15. https://doi.org/10.1186/s40246-022-00407-x

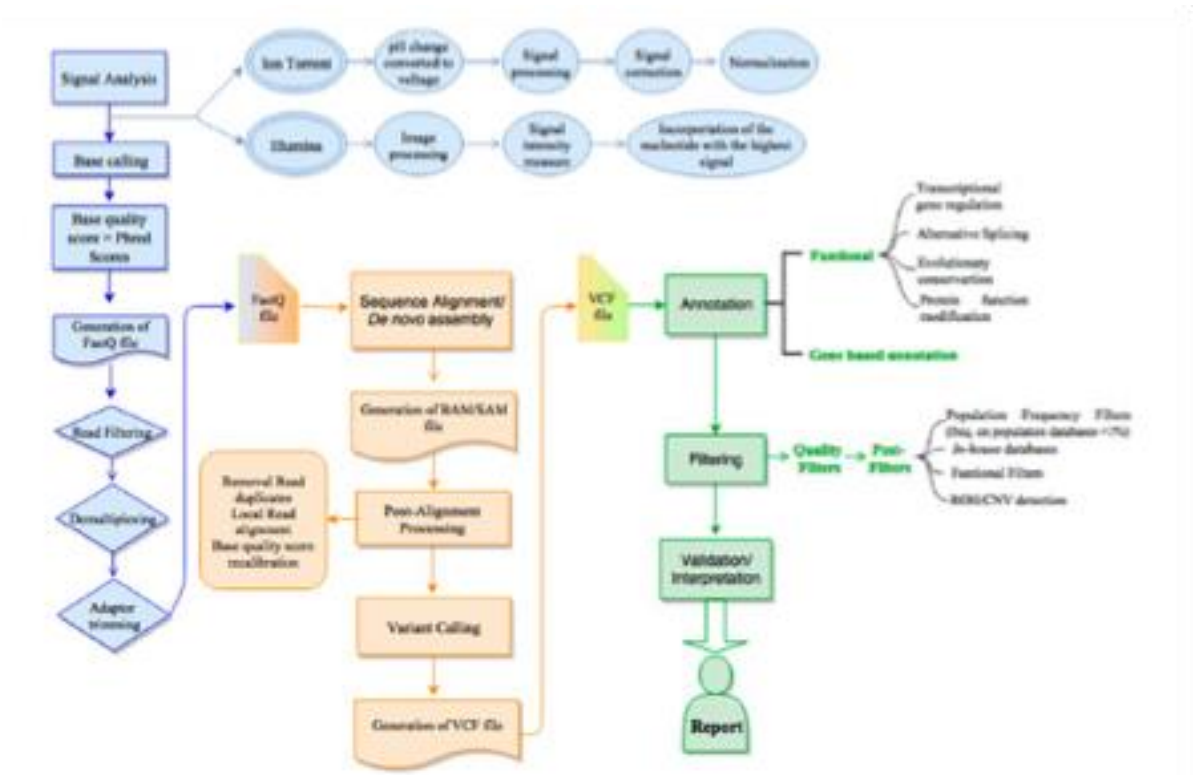

Figure 3: Next Generation Sequencing (NGS) workflow. Diagram obtained from this study (Pereira, Oliveira, & Sousa, 2020)

The evolution of open-source tools (European initiatives) has given way to the creation of paid services that allow genomic analyses to be performed more accessible and efficiently. One of the most popular is Golden Helix VarSeq.

VarSeq is software developed by Golden Helix that facilitates the analysis of panels, exomes and genomes. Its intuitive design and preconfigured pipelines allow users, regardless of their level of experience, to perform complete genomic analyses, even generating clinical reports.

Among its key features is a powerful interactive filtering and annotation engine, which allows chaining successive filters (by allelic frequency, functional impact, inheritance, etc.) to quickly reduce thousands of variants to a candidate subset. The user can adjust filtering parameters visually and observe in real time how many variants meet the criteria, facilitating experimentation and optimization of the prioritization protocol. Once an effective analysis flow is established, it can be saved and applied reproducibly to new datasets, promoting standardization in the laboratory.

The initiatives promoted by Europe and VarSeq software are just some representative examples of the many bioinformatics platforms developed for genomic analysis. The choice of the appropriate tool depends on multiple factors: the type of data to be analysed, the objective of the study (research, clinical diagnosis, etc.), the environment in which one works, and the experience and resources of the user or team. Comparing bioinformatics tools is not simply a technical exercise: it is a fundamental piece to guarantee the quality, reproducibility and clinical utility of genomic analyses, in a context where this data is increasingly integrated into healthcare decision-making.

The genomics analysis presented in this section adapts and integrates results initially developed for academic purposes within the framework of the project.

The main objective of the work carried out was to conduct a comprehensive analysis and comparison of bioinformatics tools for genomic variant annotation, aiming to identify the most effective solutions tailored to specific research and clinical needs. The specific objectives pursued included:

1. Evaluating the performance and features of selected annotation software.
2. Reproducing commercial bioinformatics workflows using open-source solutions.
3. Establishing guidelines for tool selection based on data type, analysis goals, and user expertise.

3.4.2 Methodology

Starting data

Genomic data from a set of patients from the IMPaCT project of Hospital La Paz were analyzed. We started from variant files in VCF format obtained after massive DNA sequencing and subsequent variant identification (variant calling). These VCFs included single nucleotide variants and small INDELs, as well as structural variants and copy number alterations. Regarding these variants, we had annotated files from the AnnotSV program.

Additionally, there was one VCF per genome that contained variants detected in the mitochondrial DNA (mtDNA) of patients. Three genomes from a child and his parents (trio analysis) have been analysed as indicated in Table 1.

Table 8: Genomes analyzed in this work

Genome	Individual
0050-0050	Offspring
2124-0050	Mother
2125-0050	Father

Bioinformatics processing was carried out in a Linux Ubuntu 20.04 LTS environment under WSL2 (Windows Subsystem for Linux 2) on a Windows 10 system. All tools described below were installed and executed within this environment. Three main tools were used for functional annotation of variants and their filtering: SnpEff, Ensembl VEP and ANNOVAR, maintaining consistency with the version of the human reference genome used (GRCh38/hg38 in our case).

Below are detailed the procedures performed with each tool, including their versions, installation/execution commands, the filtering criteria applied and the technical justification of said filters.

Methods

SnpEff

SnpEff (v5.2f) was installed locally following the official instructions from the *Download and install - SnpEff & SnpSift* website. The version used, published on February 7, 2025, was the most recent at the time of performing the analysis. SnpEff is developed in Java; therefore, the presence of a Java virtual machine (OpenJDK 11) in the environment was ensured.

For installation, the zip file was downloaded from the official page. Subsequently, the installation was verified and the GRCh38 annotation database was downloaded through a specific command:

```
java -jar SnpEff.jar download hg38
```

Figure 4: Command to download the GRCh38 database in SnpEff.

This command obtained files with sequences and Ensembl gene annotations for GRCh38. This database includes annotations for all nuclear genes and mitochondrial genome genes (so SnpEff can annotate variants in mitochondrial DNA equally).

For variant analysis execution, the following process was followed. Annotations were performed on input VCF files. Initially, all VCF files were annotated, but after checking that SnpEff was not effective for structural variants, repeats or CNVs, it was decided to use it only for SNV and mitochondrial DNA. The command used is represented below:

```
java -jar SnpEff.jar hg38 input.vcf > output.annotated.vcf
```

Figure 5: Command to annotate with SnpEff

In this execution, SnpEff processed each variant in the VCF and incorporated a standardized annotation in the INFO field (using the VCF ANN format). Each variant entry received information including the affected gene, consequence, position at transcript level, change at protein level if applicable, and a simplified impact classification according to the predicted severity of the consequence.

Once annotated, the SnpSift tool was used to filter and prioritize variants. SnpSift is a complementary tool to SnpEff designed specifically for filtering, manipulating and querying VCF files.

It allows applying complex filters on variant annotations, facilitating the selection of relevant variants. It is automatically installed with SnpEff installation. Taking the annotated SNV VCF as output, the filtering process was carried out. The choice of filters was made considering the need to obtain variants with a high probability of having clinical relevance. The filters include quality, functional impact, genetic consequence, and combinations thereof.

This filtering process allowed reducing a VCF file of thousands of variants (2.66 GB) to a file of only 200 variants and 356 KB. This same process was repeated for the two additional genomes.

```
     ~SnpEff$ java -jar /home/jaime_bh/SnpEff/SnpSift.jar filter \
    > "(FILTER = 'PASS') & (QUAL > 30) & (ANN[*].IMPACT = 'HIGH') &
((ANN[*].EFFECT = 'stop_gained') | (ANN[*].EFFECT = 'frameshift_variant')
    | (ANN[*].EFFECT = 'splice_acceptor_variant') | (ANN[*].EFFECT =
                      'splice_donor_variant'))" \
    mnt/c/> /mnt/c/Users/jbachiller/Desktop/Resultados_SNPEFF_00500-
              0050/resultado_SnpEffhardfilteredpequen\.VCF
   mnt/c/> > /mnt/c/Users/jbachiller/Desktop/Resultados_SNPEFF_00500-
         0050/resultado_SnpEffhardfilteredpequeno_priorizado.VCF
```

Figure 6: Filtering command for SnpEff

Variation Effect Predictor (VEP)

The workflow with VEP was performed following the instructions provided on the official page. Version 114 was used, the most recent available at the time of analysis. This tool runs on a Perl environment (Perl 5.26 or higher) and requires multiple dependencies for its operation. Installation was carried out following official instructions from the Ensembl repository, and its correct execution was validated through perl vep scripts in the Linux environment (WSL2). During installation, required dependencies were resolved through CPAN, a repository widely used to install Perl modules that allowed its operation in local mode with cache.

Once installed, we proceeded to download the FASTA files of the human genome GRCh38, as well as the cache necessary for annotation. Additionally, the plugins required for variant filtering and prioritization were downloaded. The installed cache was 509: homo_sapiens_vep_114_GRCh38.tar.gz (26 GB). This cache provides the most recent information from Ensembl, including information about mitochondrial DNA.

The input VCFs were those containing SNV and mitochondrial DNA. For initial annotation, the command shown below was used:

```
perl vep -i /mnt/c/Users/jbachiller/Desktop/0050-0050-3impact-
02._hg38top_D_.hard-filtered.VCF \ -o
/mnt/c/Users/jbachiller/Desktop/0050-0050-3impact-02_VEP_annotated.VCF
\--cache \ --dir_cache ~/. vep \--assembly GRCh38 \--VCF \--everything
\--force_overwrite
```

Figure 7: Command to annotate with VEP.

After annotation of the VCF file, the filtering process was carried out. We started with quality and PASS, but these filters barely removed variants because the source files had already undergone filtering. The next filter was allelic frequencies, where a threshold less than 0.01 was set. This filter was carried out to eliminate the most frequent variants, thus discarding those variants whose frequency in the general population is too high to be compatible with rare diseases or mutations of clinical interest.

Next, the SnpEff workflow was followed, and filters were applied by HIGH functional impact and by genetic consequence frameshift, stop, or splicing. Subsequently, a VCF was generated with all filters in which 60 variants were obtained. The process was repeated for the two additional genomes. Filtered files were also obtained by clinical consequence of variants in the three genomes to compare results with VarSeq. It was filtered by pathogenic and likely pathogenic variants.

```
filter_vep -i /mnt/c/Users/jbachiller/Desktop/Resultados_VEP_0050/0050-
0050-3impact-02_VEP_test.VCF -o
/mnt/c/Users/jbachiller/Desktop/Resultados_VEP_0050/0050-0050-
filteredAF.VCF -filter "MAX_AF < 0.01 or not MAX_AF"
```

Figure 8: Filtering command for VEP.

Additionally, annotation of structural variant files from genome 0050-0050 was carried out, where CNVs are also included with the objective of comparing results with files from AnnotSV.

ANNOVAR

For the use of ANNOVAR, developers were contacted to obtain the download link. Once obtained, the download instructions from the official page were followed. As with VEP, ANNOVAR also depends on the Perl environment.

The first step was to prepare the input files. Although ANNOVAR can work directly with VCF, we chose to convert the VCFs to their tabulated text input format for more effective control and to avoid problems such as errors in reading variants or possible incompatibility with all functionalities. For this, the command represented below was used:

```
convert2ANNOVAR.pl
```

Figure 9: Command to convert VCF files into .avinput (ANNOVAR specific).

In this way, for each sample a .avinput file was obtained that lists the variants (chromosome, position, reference allele, alternative allele, and other data) preserving important data for subsequent analyses. For annotation, the command shown below was used:

```
table_ANNOVAR.pl
```

Figure 10: Command to annotate with ANNOVAR.

```
annotate_variation.pl -buildver hg38 -downdb -webfrom ANNOVAR
```

Figure 11: Command to install databases available in ANNOVAR.

The databases that were installed for annotation were the following: RefGene, EnsGene, Gnomad, Clinvar, Dbnsfp42a, Cytoband and Avsnp150. The result was a tabulated file with one column per database. ANNOVAR does not allow applying filters directly on the output file.

VarSeq

For the use of VarSeq, the 14-day free trial was used. This version has the same functionalities as the commercial version. For our case, the options for annotation and filtering of SNV, CNV and SV were enabled.

VarSeq allows the option of working with predetermined templates. These templates come with filters selected to facilitate analysis according to what is being analyzed. This is a way to automate processes and use relevant filters in each case in a simple way.

Since we were working with a trio, the Germile trio_38 template was chosen, recommended by the developers themselves.

Once the model was chosen, the samples of all three were added in the same analysis. The VCF files of the three types of variants that this version allowed were added, increasing the total to nine files. This is done so that the program can apply filters to search for related variants between them, such as the type of inheritance. For this, there is the option to define the existing relationship between the samples.

Once selected, the next step was to download the databases. The program itself provided the databases; you simply have to accept the download and wait. After downloading and subsequent annotation, the variants appear with all relevant information.

Filtering was carried out with the help of the selected template, to which some modifications were made to adapt it to the objectives. The application of filters by VarSeq is carried out in a hierarchical way based on filter containers. These containers group and combine several filters, so that only those variants that meet all established criteria pass to the next level. In this way, the reduction in the number of variants can be observed instantly.

The Remove Common category is where filtering by population frequency was carried out, where variants with a frequency less than 1% in the different databases present in the annotations were filtered. The Ontology filter allowed filtering by the effect of the variant. Filtering by ACMG classification separates variants by the categories of benign, likely benign, VUS, likely pathogenic and pathogenic. When analysing a trio, VarSeq allows filtering also by inheritance.

With these filters, VarSeq allowed carrying out variant analysis in a visual and intuitive way.

3.4.3 Results

SnpEff

SnpEff annotated variants using its comprehensive GRCh38 database, which includes gene and mitochondrial variant information mainly from Ensembl. The initial VCF files contained about 5 million SNV across three genomes. Successive filtering steps, applied as described in the methods section, significantly reduced variant numbers, with functional impact and genetic consequence filters narrowing variants to a few hundred per genome. This process effectively prioritized variants with potential clinical relevance while substantially reducing data volume, facilitating downstream analysis and interpretation.

Table 9: Summary of the files obtained from SnpEff

Output File	Description	Size	Approximate Number of Variants
Annotation Result.vcf	VCF file annotation	2.6 GB	4,900,000
Quality Filtered.vcf	Filtered by quality and pass	2.4 GB	4,500,000
Impact Filtered.vcf	Filtered by functional impact HIGH	760 KB	690
Genetic Consequence Filtered.vcf	Filtered by genetic consequence: stop, frameshift, and splicing	410 KB	300
Combined Filtered.vcf	Combination of the previous filters	360 KB	227
Mitochondria Result.vcf	Mitochondrial DNA variants	200 KB	52

Table 10: Summary of variant counts after filtering using SNPEFF for the three genomes

SNPEFF	0050-0050	2124-0050	2125-0050
Initial Variants	4,911,599	4,933,082	4,905,230
Quality Filtered	4,485,854	-	-
Functional Impact Filtered	690	696	684
Genetic Consequence Filtered	297	294	304
Combined Filtered	227	220	235
Mitochondrial Variants	51	51	56

VEP

The VEP tool was used to annotate and filter the same three genomes, utilizing the comprehensive homo_sapiens_vep_114_GRCh38 cache from Ensembl. In addition to generating annotated VCF files, VEP produces detailed HTML summary reports that provide clear visualizations and extensive statistics on variant types, distribution, and affected genes. For

genome 0050-0050, VEP identified approximately 4.9 million variants, mostly SNV, with variants distributed predominantly in intronic regions and only a small proportion causing missense or synonymous changes.

Filtering steps included quality control, frequency filtering to exclude variants with population frequency above 0.01%, and functional impact assessments, which reduced variant counts dramatically—from hundreds of thousands to less than a hundred variants per genome. Variants classified as pathogenic or likely pathogenic numbered between 13 and 14 per genome. The results were consistent across the three genomes analyzed. Additionally, structural variant annotation showed differences when compared with AnnotSV, with VEP detecting fewer structural variants but providing a detailed, manageable overview for clinical prioritization. Overall, VEP facilitated an in-depth annotation and effective filtering workflow, significantly reducing variants to clinically relevant candidates.

Table 11: Summary of the files obtained from VEP

Output File	Description	Size	Approximate Number of Variants
VCF Annotated	Annotation of the VCF file	12.5 GB	5,000,000
Summary	HTML summary file	28 KB	Not applicable
Warnings	Txt error file	1.35 MB	Not applicable
VCF_Max_AF	Filtered by allele frequency	424 MB	200,000
VCF_HIGH	Filtered by HIGH impact	12.8 MB	1,650
VCF_Genetic	Filtered by genetic consequence: stop, frameshift, and splicing	105 MB	15,000
VCF_Combined	Combined previous filters	952 KB	80
VCF_Pathogenic	Filtered by pathogenicity	250 KB	13
VCF_Mito	Mitochondrial DNA variants	443 KB	51

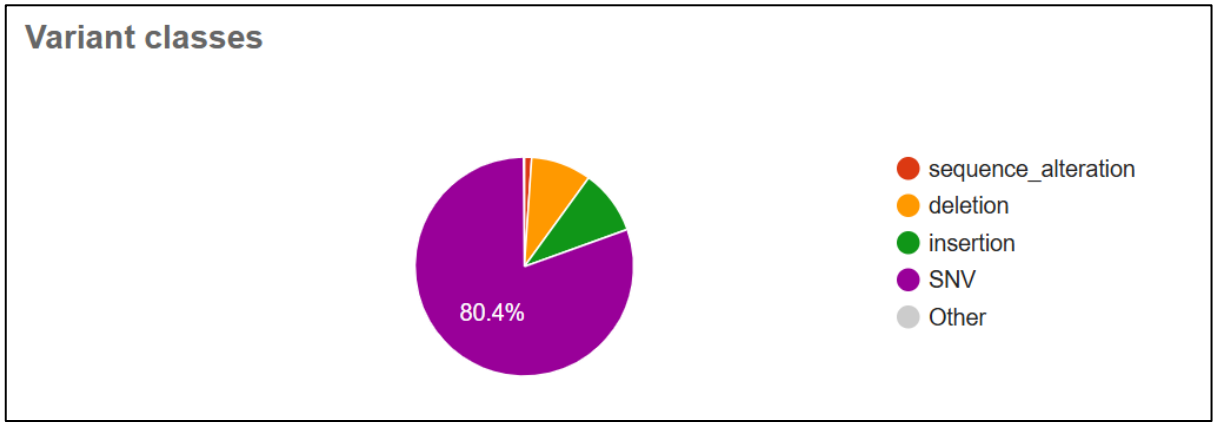

Figure 12: Type of variants distribution of genome 0050-0050. Graphic obtained from the HTML summary

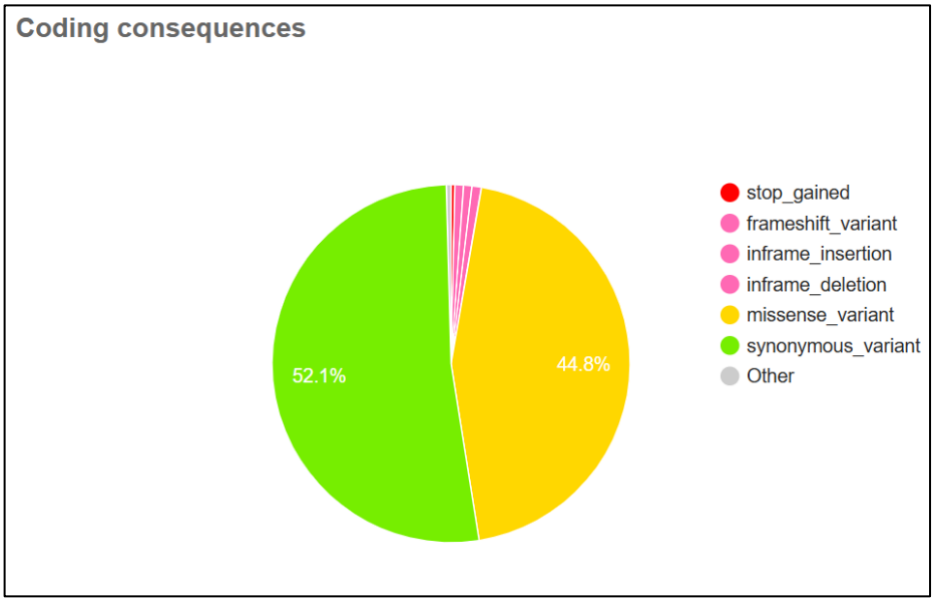

Figure 13: Coding consequence distribution of variants present in genome 2124-0050. Data obtained from the HTML summary

Table 12: Summary of variant counts after filtering using VEP for the three genomes

VEP	0050-0050	2124-0050	2125-0050
Initial variants	4.903.578	4.926.650	4.896.228
Quality filter	4.483.180	-	-
Frequency filter	201.345	202.789	202.789
Functional impact filter	1.619	1.657	1.628
Genetic consequence filter	15.678	15.832	15.548
Combined filter	64	86	73
Pathogenicity filter	14	13	13
Mitochondrial variants	51	51	56
VEP	0050-0050	2124-0050	2125-0050

ANNOVAR

ANNOVAR processed the same three genomes as previous tools. Due to its particular working format, the results were obtained as tabulated files (.csv) instead of VCF files. Table 11 shows the files obtained with this tool.

Table 13: Output files from ANNOVAR

Output File	Description	Size	Approximate Number of Variants
Tabular file.csv	SNV variant annotation	2.3 GB	5,000,000
Text file.txt	Mitochondrial variant annotation	6 KB	50

The results obtained with ANNOVAR are shown in Table 12. As can be observed, the initial numbers are similar to previous tools, confirming the consistency of the starting data. However, ANNOVAR does not allow direct filtering on the output file, so all variants are kept in the annotated file.

Table 14: Summary of variant counts after using ANNOVAR for the three genomes

ANNOVAR	0050-0050	2124-0050	2125-0050
Annotated variants	5.007.749	5.030.763	4.994.593
Mitochondrial variants	51	51	56

The main advantage of ANNOVAR is the richness of its annotations, providing information from multiple databases in a single tabulated file. However, the lack of direct filtering capability makes it less practical for quick variant prioritization.

VarSeq

VarSeq processed the three genomes using the Germline trio_38 template. This commercial tool offers an intuitive graphical interface that facilitates variant analysis through a hierarchical filtering system.

The filtering was carried out in a visual way, allowing us to observe in real time the reduction in the number of variants as filters were applied.

Table 15: Summary of variants counts after using VarSeq for the three genomes

VarSeq	0050-0050	2124-0050	2125-0050
Processed variants	4.614.567	4.740.537	4.592.119
Stricter frequency filtering	201.481	206.730	202.452
Combined missense and LoF effect	10.039	1.087	1004
Pathogenic / likely pathogenic variants	40	41	31
Exclusively VUS variants	300	306	294
CNVs	13.056	12.917	11.611

VarSeq	0050-0050	2124-0050	2125-0050
SVs	989	1.035	760

VarSeq stood out for its ease of use and its ability to generate clinical reports automatically. The tool integrated multiple databases and allowed real-time visualization of filtering effects, making it especially attractive for clinical environments. Table 17 shows a summary of the output files obtained from each tool.

Table 16: Summary of output file from each tool

Tool	Output type after annotation	Filtering allowed?	mtDNA annotation
SnpEff	VCF	Yes (SnpSift)	Yes
VEP	VCF	Yes (filter_vep)	Yes
ANNOVAR	Tabular CSV	No	Partial
VarSeq	VCF, .txt, annotation file or Excel workbook	Yes (Graphical filtering interface)	Yes (Could not be performed)

3.4.4 Discussion

Summary of findings

The comparative analysis of the four variant annotation tools (SnpEff, VEP, ANNOVAR, and VarSeq) has revealed significant differences in their approaches, capabilities, and results. Each tool showed distinct strengths and limitations that make them suitable for different types of projects and user profiles.

Each tool offered a different approach to variant analysis. SnpEff provided very fast annotation based on a compiled model of the GRCh38 genome, and allowed progressive filtering based on the severity of functional impact and genetic consequences (such as stop gained, frameshift, or splice site variants). This made it possible to reduce from more than 4 million variants to fewer than 250 highly prioritized variants per genome, highlighting its efficiency in terms of reduction.

In terms of annotation richness, VEP and VarSeq offered the most comprehensive annotations, integrating multiple databases and prediction algorithms. ANNOVAR also provided rich annotations but in a less integrated format. SnpEff, while faster, offered more basic annotations.

The filtering capability varied significantly between tools. VarSeq excelled in this aspect with its intuitive graphical interface and real-time filtering visualization. VEP offered powerful command-line filtering capabilities. SnpEff provided good filtering through SnpSift. ANNOVAR was the weakest in this categoty, requiring external tools for effective filtering.

These findings reflect both the versatility and inherent limitations of each tool and demonstrate how the choice of software can influence the outcome of the genomic analysis (Park & Park, 2022[29]). This discussion will further analyze these differences in depth, assessing their practical impact to identify the real strengths of each platform within clinical and research contexts.

Quantitative comparison

[29] Park, K. J., & Park, J. H. (2022). Variations in Nomenclature of Clinical Variants between Annotation Tools. *Lab Medicine*, *53*(3), 242–245. https://doi.org/10.1093/labmed/lmab074

The quantitative analysis revealed interesting patterns in the results obtained by each tool. Tables 10 show a detailed comparison of results for the genome 0050-0050.

Table 17: Comparison of variant counts across different annotation tools for sample 0050-0050. The higher variant count for VarSeq (6,791,280) reflects the total number of variants processed when analyzing the three genomes (trio analysis) simultaneously within.

0050-0050	SnpEff	VEP	ANNOVAR	VarSeq
Annotated variants	4,911,599	4,903,578	5,030,763	6,791,280
Quality filtered	4,485,854	4,483,180	-	4,614,567
Frequency	-	201,345	-	201,481
HIGH impact	690	1,619	-	-
LoF or missense protein effect	-	-	-	1,039
Stop/frameshift/splice	297	15,678	-	-
Pathogenic/likely pathogenic variants	-	14	-	40
Combination of filters	227	64	-	-
Mitochondrial	51	51	51	-

The results show that different tools identify different numbers of high-impact variants from the same starting data. This discrepancy can be attributed to several factors:

1. Different annotation databases: Each tool uses different reference databases and versions.
2. Distinct classification criteria: The definition of "high impact" varies between tools.
3. Algorithm differences: Each tool employs different algorithms for consequence prediction.
4. Transcript selection: Tools may select different canonical transcripts for the same gene.

Qualitative comparison

Beyond quantitative differences, each tool showed distinct qualitative characteristics that affect their practical utility in different contexts (Table 11). SnpEff's strength lies in its simplicity and speed. VEP showed the most balanced performance among open-source tools. ANNOVAR's main advantage is the breadth of databases it can integrate. And VarSeq excelled in user experience and clinical integration.

Table 18: Qualitative Comparison of Genomic Variant Annotation Tools

Feature	SnpEff	VEP	ANNOVAR	VarSeq
Installation	Easy	Moderate	Moderate	Easy
Execution time	Fast	Slow	Medium	Medium
Usability	Medium	Medium	Medium	Easy
Filtering	Moderate	Moderate	External	Easy
Output variety	Low	Moderate	Low	High

Benchmarking

To provide a comprehensive comparison, a benchmarking system was developed using multiple criteria relevant to different use cases and user profiles. For each criterion, a score from 1 to 4 was assigned to every tool, ensuring that no tie could occur in the final evaluation.

Table 19: Benchmarking of genomic variant annotation tools based on key criteria

Criterion	SnpEff	VEP	ANNOVAR	VarSeq
Installation / usability	3	2	1	4
Analysis time	4	1	2	3
Variant types	1	3	2	4
Annotation richness	1	3	2	4
Clinical annotations	1	3	2	4
Filtering	3	2	1	4
Visualization / output format	1	3	2	4
Flexibility / adaptability	3	4	1	2
Available documentation	3	4	2	1
Interoperability	2	4	3	1
Total score	19	27	17	31

The benchmarking results revealed that VarSeq achieved the highest overall score with 31 points, followed closely by VEP with 27. SnpEff and ANNOVAR obtained 19 and 17 respectively. However, the choice of the most appropriate tool depends heavily on the specific use case and user requirements.

Clinical and research implications

Differences among annotation tools impact both clinical and research uses. Clinically, ease of use, standardized reporting, and integration with clinical databases are essential, areas where VarSeq excels. For research, flexibility and pipeline integration are key, making VEP the strongest open-source option. Understanding tool characteristics is critical for reliable genomic analysis, underscoring the need for continuous benchmarking and standardization in the field.

Proposed workflow like VarSeq

This study aimed to find tools resembling VarSeq's functionality and workflow. VEP emerged as the closest, due to its annotation richness, visualization, clinical database integration, and plugin support, enabling similar analysis and prioritization workflows. VEP's easy pipeline integration makes it suitable for scalable genomic studies. Further validation is suggested due to the sequence of analyses performed.

Although VEP was identified as a powerful annotation engine, it presents a key limitation for the desired workflow: VEP only accepts VCF files as input. It does not have the native capability to process or read SAM/BAM files. This is a critical point, as VarSeq’s workflow is designed for the user to simultaneously upload both the VCF and the corresponding BAM file. ANNOVAR is the

open-source tool that offers this flexibility, allowing a VCF to be used as the primary input while also accepting a SAM/BAM file as a secondary input for the same analysis.

Limitations

This study analysed only three genomes from one family, limiting its generalizability. It focused mainly on single nucleotide variants, with less emphasis on structural and copy number variants. The analysis aimed to maximize the information extracted from each tool beyond routine clinical use and did not include clinical validation of predicted pathogenic variants. Additionally, the inherent risks and support limitations of open-source bioinformatics tools were acknowledged.

Future work

Future research should expand to more tools and larger, clinically diverse datasets, with special attention to structural and copy number variants. Integration with workflow platforms would enhance reproducibility and scalability. Clinical validation of variant pathogenicity predictions is essential, and emerging AI-based annotation methods merit comparative evaluation. Improving benchmarking frameworks could further systematize the evaluation of bioinformatics tools beyond variant annotation.

3.4.5 Conclusions

This comprehensive evaluation of genomic variant annotation tools has provided valuable insights into the current landscape of bioinformatics solutions for genomic analysis. The main conclusions are:

1. Tool selection is context-dependent: No single tool excels in all areas. The choice depends on specific requirements including user expertise, budget constraints, analysis volume, and clinical integration needs.
2. VEP emerges as the leading open-source option: Among free tools, VEP provides the best balance of functionality, annotation quality, and usability. Its integration with Ensembl databases and extensive plugin system make it a solid alternative to commercial solutions. However, it lacks SAM/BAM files support.
3. VarSeq justifies its commercial status: The commercial tool's superior user interface, clinical integration, and automated reporting capabilities explain its widespread adoption in clinical laboratories, despite higher costs.
4. Result variability highlights the importance of tool understanding: Different tools produce varying results from identical input data, emphasizing the need for users to understand each tool's characteristics and limitations before analysis.
5. Benchmarking offers a practical foundation for ongoing evaluation: This study provides a framework for systematic tool comparison, which can be extended and adapted to include additional annotation tools, variant types, and diverse clinical or research scenarios, enhancing its utility for broader genomic analysis needs.

As the genomics field rapidly evolves, regular reassessment and benchmarking will remain essential to maintain current recommendations and ensure optimal tool selection. This work provides a foundation for informed decision-making and establishes a systematic framework that can be extended to evaluate emerging tools, variant types, and updated criteria in future research.

This work provides a foundation for informed decision-making in genomic tool selection and highlights areas for future improvement in both open-source and commercial solutions. The proposed benchmarking framework can be extended to evaluate additional tools and updated criteria as the field advances.

3.5 NLP and GenAI

Natural Language Processing (NLP) and Generative Artificial Intelligence (GenAI) have emerged as transformative technologies for clinical data extraction and structuring. Within the context of the European Health Data Space (EHDS) and Secure Processing Environments (SPEs), NLP and GenAI enable the automated transformation of unstructured clinical narratives---such as medical reports, clinical notes, and diagnostic summaries---into standardized, computable data models. This capability is particularly valuable in the PROTECT-CHILD project, where transplant-related clinical documentation from multiple European centres must be integrated, standardized, and made interoperable across diverse healthcare institutions while maintaining strict data protection and compliance requirements.

This section describes the state of the art in NLP and GenAI for clinical applications, the approach adopted within PROTECT-CHILD for extracting and structuring medical data, the current implementation status, and the pathway toward production deployment within federated Secure Processing Environments.

3.5.1 State of the Art in NLP for Clinical Data Extraction

3.5.1.1 Evolution of Clinical Entity Extraction

Automated extraction of medical entities from free-text clinical documentation has evolved significantly over the past decade. Early approaches relied on rule-based patterns and lexicon-matching techniques, which offered predictability but limited coverage and flexibility. The introduction of deep learning models---particularly BiLSTM-CRF architectures---improved performance on sequence-to-label tasks such as recognizing diagnoses, medications, procedures, and clinical events. However, these supervised approaches remained data-hungry and struggled with linguistic variation common in clinical narratives.

The emergence of pre-trained transformer models, especially BERT and its clinical variants (ClinicalBERT, BioBERT, PubMedBERT, MedGemma), represented a significant shift by introducing transfer learning to the medical domain. These models captured long-range semantic dependencies and reduced the need for large annotated corpora. Their adoption in clinical NLP tasks demonstrated substantial improvements in entity recognition and classification across multiple benchmark datasets.

3.5.1.2 Large Language Models (LLMs) in Clinical Contexts

Recent advances in large language models, including GPT-4, Claude, Llama, and domain-specific models, have introduced a new paradigm for clinical information extraction. Unlike traditional supervised models, LLMs can perform complex reasoning tasks with minimal task-specific training through prompting. Their ability to understand implicit clinical context, resolve ambiguities in medical language, and adapt to different document formats has made them particularly attractive for unstructured clinical data processing.

A critical advantage of LLMs is their zero-shot and few-shot capability: they can extract structured information from clinical reports without requiring labeled training data, dramatically reducing the engineering overhead associated with traditional machine learning pipelines. Furthermore, LLMs can be combined with structured output constraints (such as JSON schemas or Pydantic models) to ensure that extracted information conforms to predefined data structures, improving consistency and downstream compatibility.

However, the deployment of closed-source, API-based LLMs in healthcare contexts presents significant challenges. Transmission of sensitive clinical data to external servers violates GDPR principles of data minimization and creates regulatory uncertainty under the EHDS framework. These concerns have driven interest in open-source language models that can be deployed on-premises, maintaining data sovereignty while benefiting from LLM capabilities.

3.5.1.3 Open-Source Models and On-Premises Deployment

The maturation of high-performance open-source language models---including Mistral, Llama 2/3, and domain-specialized models such as Medgemma---has enabled privacy-preserving clinical NLP deployments. These models, while generally smaller than their closed-source counterparts, have demonstrated competitive performance on specialized tasks through careful prompt engineering and few-shot adaptation. Crucially, they can be deployed entirely within hospital infrastructure or Secure Processing Environments, eliminating external data transmission and aligning with GDPR and EHDS requirements.

Model compression techniques, including quantization and pruning, allow these open-source models to run on conventional hardware without substantial performance degradation, making on-premises deployment practical for most healthcare institutions.

3.5.1.4 Structured Output Generation and Constrained Decoding

A significant technical advancement is the ability to constrain LLM outputs to conform to structured schemas. Through mechanisms such as JSON schema validation and Pydantic models, language models can be forced to generate outputs that match predefined data structures directly, rather than producing free-form text requiring post-processing. This constrained generation approach substantially improves the reliability and parseability of extracted clinical information, reducing downstream data quality issues and simplifying integration with downstream systems.

3.5.2 Clinical Data Standardization and OMOP CDM

3.5.2.1 OMOP Common Data Model

The OMOP Common Data Model (CDM) represents the standardized data representation framework adopted by the EHDS and increasingly adopted across European health systems. OMOP defines a unified schema for clinical observations, enabling cross-institutional analysis and research while maintaining semantic clarity through standardized vocabularies and concept mappings.

A central challenge in OMOP implementation is the automated mapping of heterogeneous, institution-specific clinical data to standardized OMOP concepts. While manual mapping is conceptually straightforward, it is prohibitively labor-intensive at scale and inconsistent across institutions. NLP-based approaches, particularly those leveraging LLMs combined with terminological APIs, offer a path toward automation.

3.5.2.2 Semantic Mapping and Terminological APIs

LLMs, when given access to reference APIs (such as Athena, which provides OMOP vocabulary services), can perform intelligent semantic mapping between free-text clinical descriptions and standardized OMOP concepts. The process involves:

- Extracting a clinical entity from a narrative (e.g., "elevated liver enzymes");
- Querying a terminological API to identify candidate standardized concepts;
- Using LLM reasoning to select the most contextually appropriate concept; and
- Validating the mapping against domain knowledge and audit rules.

This hybrid human-AI-API approach has demonstrated feasibility for clinical concept mapping and offers a pathway toward standardization that respects institutional terminology while ensuring semantic interoperability.

3.5.3 PROTECT-CHILD Approach: Two-Stage Pipeline

3.5.3.1 Stage 1: Medical Entity Extraction with Structured Outputs

The PROTECT-CHILD project implements a two-stage pipeline for transforming unstructured clinical reports into standardized OMOP data. In Stage 1, clinical reports in PDF format are processed by an open-source language model (currently evaluating Mistral and Medgemma) using carefully engineered prompting strategies. The model is constrained through Pydantic-based structured output generation to extract relevant medical entities into predefined JSON structures.

Extracted entities include:

- Diagnoses and clinical presentations
- Procedures and surgical interventions
- Medications and therapeutic interventions
- Laboratory measurements and vital signs
- Imaging findings and radiological reports
- Histological findings and biopsies
- Microbiological and virological results
- Immunological markers and antibody tests
- Adverse events and clinical complications

The extraction schema will be adapted to the data model, ensuring that domain-specific entities relevant to post-transplant surveillance and longitudinal care are captured with appropriate granularity.

3.5.3.2 Stage 2: Automated Mapping to OMOP CDM

In Stage 2, extracted entities are processed through an AI agent that performs automated mapping to OMOP concepts. The agent interfaces with the Athena API to resolve terminological ambiguities and retrieve the appropriate concept_ids and concept_names for each extracted entity. The mapping process incorporates clinical context to select among multiple candidate OMOP concepts, prioritizing semantic accuracy over simple string matching.

Both stages will operate entirely within the SPE on-premises infrastructure, ensuring that no clinical data leaves the hospital network during processing.

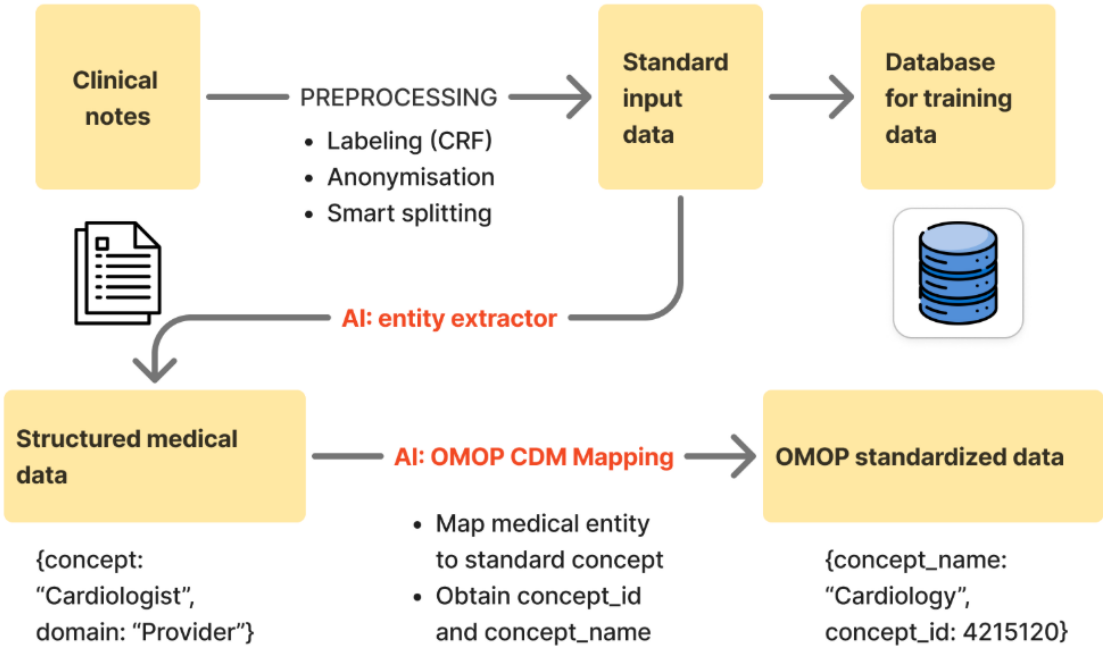

Figure 14: NLP Pipeline Design

3.5.3.3 Clinical Documentation Diversity

The pipeline will process diverse document types from multiple European transplant centres (Palermo, Madrid, Padua), reflecting the heterogeneity of real clinical workflows:

- Pre-transplant evaluations
- Surgical and transplant reports
- Post-transplant hospitalization records and discharge summaries
- Histopathological biopsies (liver, kidney)
- Outpatient clinical consultations and follow-up notes
- Microbiological and virology test results
- Immunological assessments (donor-specific antibodies)
- Imaging studies (ultrasound, CT, MRI, MRCP, scintigraphy)

This document diversity presents significant challenges in entity extraction variability and adaptation across different clinical reporting formats and terminology conventions.

3.5.4 Current Implementation Status

3.5.4.1 Preliminary Evaluation with Mock Data

Initial project phases utilized synthetic clinical reports and a test data model to validate the pipeline architecture and optimize prompting strategies. Preliminary results with different private models (OpenAI) or open-sourced (Llama 3.3 70B) using structured outputs demonstrated feasibility of entity extraction, with acceptable performance on synthetic data. However, these mock data differed substantially in linguistic simplicity and structural regularity from real clinical reports subsequently received from partner institutions, limiting the generalizability of these findings.

3.5.4.2 Real Data Integration and Current Phase

The project recently obtained access to real, anonymized clinical reports from multiple European transplant centres and preliminary versions of a transplant-specific clinical data model. The team is currently in a setup and evaluation phase, conducting systematic testing of different model configurations against real clinical documentation.

Current status:

- Real clinical data has been partitioned and is undergoing testing with different open-source model configurations
- The OMOP mapping agent successfully connects to the Athena API and processes information, but requires evaluation against real clinical data and the current data model
- No quantitative performance metrics have been established yet with real data
- The current transplant-specific data model is being integrated into the extraction and mapping pipeline

3.5.5 Next Steps and Planned Activities

Phase 1: Real Data Evaluation and Error Analysis Systematic testing of entity extraction and OMOP mapping performance on real clinical reports. Identification of systematic failure modes, document-type-specific challenges, and error patterns requiring targeted improvement.

Phase 2: Model Refinement Based on real data performance analysis, selective fine-tuning of open-source models using representative clinical examples. Consideration of few-shot prompting optimization and domain-specific prompt engineering. Evaluation of trade-offs between model size, computational efficiency, and extraction accuracy.

Phase 3: Mapping Agent Enhancement Development of confidence scoring mechanisms for OMOP mappings, enabling escalation to human review for ambiguous or low-confidence cases. Implementation of fallback strategies for entities that cannot be automatically mapped to standardized concepts.

Phase 4: Formal Validation Establishment of externally validated test sets with manual annotation by clinical experts. Quantitative evaluation using standard metrics (precision, recall, F1-score, concept-mapping accuracy) to assess pipeline performance and readiness for production deployment.

Phase 5: Production Integration Integration of the validated pipeline into the PROTECT-CHILD SPE infrastructure, with monitoring, audit logging, and compliance verification aligned with EHDS and GDPR requirements.

3.5.6 Privacy and regulatory alignment

A central design principle is the preservation of privacy and regulatory compliance. By deploying open-source models on-premises within Secure Processing Environments, the pipeline avoids transmission of clinical data to external services, eliminating risks associated with API-based approaches. All processing remains under institutional control, audit trails are maintained within the SPE, and data never leaves the hospital network.

This architecture directly supports GDPR Articles 25 and 32 (privacy-by-design, security-by-design) and EHDS Article 73 requirements for Secure Processing Environments, enabling compliant, trustworthy automation of clinical data standardization.

3.5.7 Conclusions

The combination of open-source language models, structured output constraints, and on-premises deployment offers a viable, privacy-preserving approach to automating clinical data extraction and standardization. The PROTECT-CHILD project is currently validating this approach with real clinical data from multiple European centres; subsequent phases will focus on systematic performance assessment, iterative refinement, and eventual production deployment within federated Secure Processing Environments.

This work establishes a precedent for automated clinical data standardization that prioritizes data sovereignty, regulatory compliance, and institutional autonomy---essential prerequisites for widespread adoption of AI-assisted data processing within European healthcare systems aligned with the EHDS vision.

4 Data and metadata model

This section describes how PROTECT-CHILD represents, harmonises and documents data across all participating centres. It focuses on the common data model (CDM) that underpins the EHDS capsules and the metadata model that supports data discovery, governance and quality assessment. Together, these models provide a consistent way to integrate clinical variables from the PROTECT-CHILD study, selected variables from the PETER registry, and genomic, epigenomic and methylomic results, so that they can be analysed in a federated, EHDS-compliant environment.

Within the platform architecture, the PROTECT-CHILD CDM forms the internal "language" of each EHDS capsule: heterogeneous local sources (EHR, registry, omics pipelines, NLP outputs) are transformed into this shared structure, which is then exposed through multiple standards (OMOP for analytics, FHIR for exchange, HealthDCAT-AP for cataloguing). The metadata model complements this by capturing what each dataset contains, how it was produced, its quality, and under which governance rules it can be accessed, enabling the discovery and lawful reuse of data across the federated ecosystem.

Section 5 therefore moves from data sources (what data we have), to the CDM (how we structure it), to the detailed entities and tables, and finally to the metadata model that describes and governs the resulting datasets.

4.1 Data sources for the PROTECT-CHILD CDM

The PROTECT-CHILD common data and metadata models are driven by three main categories of data that will be integrated within the EHDS-compliant platform: the multicentre clinical study data collected specifically for the project, the longitudinal real-world data already captured in the PaEdiatric Transplantation European Registry (PETER) under the TransplantChild ERN, and newly generated genomic, epigenomic and methylomic data. Together, these sources cover both prospective and retrospective information, combine highly curated study data with routine-care registries, and link detailed phenotypes with multi-omics measurements. The data model has to provide a coherent representation of all these layers, enabling linkage at the level of individual transplanted children and transplant episodes, and supporting federated analytics and learning across centres.

4.1.1 Clinical study data

The PROTECT-CHILD clinical study is a multicentre cohort of paediatric liver and kidney transplant recipients designed to demonstrate the added value of integrating real-world data with genomic markers for predicting adverse outcomes and optimising immunosuppressive treatment. The study will recruit at least 200 children and adolescents aged 3 to 18 years who have undergone liver or renal transplantation, across four clinical centres in Spain, Italy and Germany. Recruitment combines retrospective identification of eligible transplanted patients and prospective inclusion of newly transplanted or already transplanted patients during the project period, so that both historical and current trajectories can be analysed.

Data collection is organised around transplant episodes and longitudinal follow-up. For each participant, the study captures detailed baseline information on indication for transplantation and underlying disease, donor and graft characteristics, peri-operative course and early complications. During follow-up, repeated measures describe immunosuppressive regimens and dose adjustments, therapeutic drug monitoring, occurrence of acute and chronic rejection, infectious events, graft function, comorbidities and late toxicities, as well as survival and

clinically relevant endpoints. These variables are collected using structured eCRFs and complemented, where appropriate, with information extracted from hospital EHRs and unstructured reports via the NLP pipelines described in Section 4.5.

From the perspective of the data model, the clinical study defines the "reference phenotype layer" for PROTECT-CHILD. It provides a well-curated set of variables that must be representable in the core conceptual model (child, donor, transplant episode, visit, condition, procedure, medication, measurement and outcome entities) and mappable to OMOP and FHIR, while at the same time serving as the primary link to the omics data generated in WP9. The design of the common data model and its semantics therefore starts from the clinical study data dictionary and ensures that every clinically relevant element required for risk prediction, pharmacogenomics and outcome analysis has an explicit place in the schema.

4.1.2 PETER registry data

Beyond the dedicated clinical study, PROTECT-CHILD will leverage the real-world data accumulated in the PaEdiatric Transplantation European Registry (PETER), promoted by the TransplantChild ERN. PETER is a pan-European registry devoted to haematopoietic stem cell and solid organ transplantation in infants, children and adolescents, implemented on an interoperable technology platform with embedded business-intelligence capabilities. It provides longitudinal routine-care information on large numbers of transplanted children across centres and countries and is therefore an important complement to the more deeply phenotyped but smaller PROTECT-CHILD clinical study cohort.

However, for the purposes of this project, PROTECT-CHILD will not use the full breadth of variables available in PETER. Instead, only those registry variables that are conceptually aligned with the clinical study and that describe the clinical events of interest for our analyses will be selected and mapped into the common data model. In practice, this means focusing on a curated subset of PETER elements that overlap with the study dataset or are directly relevant to the outcomes and predictors defined in PROTECT-CHILD.

This selective alignment has two main advantages. First, it ensures semantic consistency between the clinical study and registry data, so that equivalent concepts from both sources are represented in a harmonised way in the PROTECT-CHILD data model and can be analysed together without ambiguity. Second, it limits unnecessary processing of PETER data to only those elements that have a clear role in the project's defined use cases, reducing the burden on participating centres and simplifying governance, quality assessment and documentation. The data model defined in this deliverable will therefore act as a bridge between the PROTECT-CHILD CRF and the subset of PETER variables and events that are in scope, enabling the EHDS capsules to consume registry-conformant data for federated analytics while respecting the constraints and structure of the original TransplantChild registry.

4.1.3 Genomic, epigenomic and methylomic data

A key innovation of PROTECT-CHILD is the generation and integration of high-throughput omics data for the same transplanted children, with a particular focus on WGS and methylome profiling. According to the current agreed plan, genomic data will be collected for approximately 200 liver and kidney transplant recipients participating in the clinical study. WGS will be performed using next-generation sequencing technologies to characterise single nucleotide variants, small insertions and deletions, structural variants and copy-number variations across the entire genome. The resulting variant calls will be annotated, quality-controlled and further processed to derive polygenic risk scores, pharmacogenetic markers and other genomic features relevant to transplant outcomes, immune response and susceptibility to infections. This

design reflects the consensus at the time of writing but may be refined as protocols and practical constraints evolve during the project.

In parallel, methylome studies are planned to generate genome-wide DNA methylation profiles, with the aim of refining existing episignatures and discovering new disease-, gene- or variant-specific epigenetic signatures in transplanted children. These analyses will help to identify clusters of patients with similar CpG methylation patterns and to clarify the genetic determinants of the underlying diseases that led to transplantation, particularly in cases where variants of uncertain significance or post-zygotic variants complicate clinical interpretation. The omics data produced are intended to be integrated into European infrastructures such as the GDI and Beyond 1 Million Genomes, while remaining accessible for federated analyses through the PROTECT-CHILD capsules; specific technical choices and target repositories may be adjusted as these initiatives and project needs progress.

From a modelling perspective, these omics datasets introduce additional entities (samples, sequencing runs, variants, methylation probes, annotations, scores and signatures) and relationships that must be represented in a way that is compatible both with GA4GH standards and with the clinical CDM underpinning the EHDS capsules. The data model specified in this deliverable therefore treats genomic, epigenomic and methylomic outputs as first-class citizens that can be linked to individual patients, transplant episodes and outcomes, and queried through federated analytics alongside the clinical and registry data described above. As with other components of the PROTECT-CHILD CDM, this omics modelling is expected to evolve in later project phases based on implementation experience and feedback from partners.

4.2 PROTECT-CHILD CDM definition

The PROTECT-CHILD project recognises the need for a clearly defined common data model (CDM) for paediatric transplantation in order to enable seamless integration and analysis of data coming from different hospitals, registries and omics pipelines. The CDM brings together the richly detailed clinical variables collected in the PROTECT-CHILD study, the selected and harmonised subset of variables from the PETER registry, and the genomic, epigenomic and methylomic data within a single, consistent representation that can support federated analytics and reuse in an EHDS-compliant environment.

In architectural terms, the PROTECT-CHILD CDM is the structural backbone of the EHDS capsules. It sits between heterogeneous local source systems and the multi-standard interfaces exposed by each capsule (OMOP, FHIR and DCAT-AP Health), providing a single logical view of children, donors, transplant episodes, treatments, complications, biosamples and omics findings.

To maximise interoperability and reuse, the PROTECT-CHILD CDM is designed to be standard-agnostic but standard-friendly. For clinical data and high-level molecular findings, its structures will be mapped to the OMOP (Observational Medical Outcomes Partnership) CDM for observational analytics and to HL7 FHIR (including FHIR Genomics) for data exchange and integration, while semantic interoperability is ensured through OHDSI standard vocabularies (e.g. SNOMED CT, ICD, LOINC, ATC) and domain ontologies such as HPO and GA4GH-aligned resources for genomic data. In addition, detailed omics data will be presented in a dedicated omics data model maintained alongside, and explicitly linked to, the OMOP-based clinical layer, so that rich molecular detail is preserved without compromising harmonisation. This dual compatibility with OMOP and FHIR promotes harmonisation across centres and facilitates linkage with external infrastructures such as the Genomic Data Infrastructure (GDI) and the European Health Data Space.

The key properties of the CDM being FAIR30 and EHDS-ready, multi-standard, domain-driven, privacy-preserving, extensible and versioned are expressed as explicit data model requirements in Section 6.4.1.

4.2.1 Data model definition methodology

In defining the data model to be adopted within PROTECT-CHILD, a systematic approach was followed, involving close collaboration between clinicians and technical partners. The initial step consisted of the clinical teams identifying the variables and data elements that were essential for characterising paediatric liver and kidney transplant recipients and for supporting the project's planned analyses. This work was grounded in the PROTECT-CHILD CRFs and in the subset of PETER registry variables that were considered relevant and alignable with the study.

In a second step, these variables were grouped into coherent sets that would later become the candidate tables of the common data model. Variables were organised according to their clinical role and granularity (for example, patient and donor characteristics, transplant and peri-operative information, follow-up visits, clinical events, immunosuppressive regimens, laboratory measurements and omics-related data), ensuring a consistent, clinically meaningful structure.

With this logical organisation in place, the next phase focused on aligning the emerging PROTECT-CHILD CDM with the two key standards adopted in the project: HL7 FHIR (Fast Healthcare Interoperability Resources) and OMOP (Observational Medical Outcomes Partnership). The variables and groups were mapped to appropriate FHIR resources and elements, and to OMOP tables and fields, using standard vocabularies whenever possible. This mapping ensures that the PROTECT-CHILD CDM remains interoperable with existing healthcare systems and external research infrastructures, and that clinical and omics data can be exposed and analysed through both standards as needed.

Throughout the development process, the proposed CDM and its mappings were iteratively reviewed. An initial, rapid validation was carried out with the clinical coordinator to confirm the clinical relevance and correctness of the structure and mappings. Subsequently, the model was presented and discussed in joint sessions with the wider group of clinicians and consortium partners, allowing for comprehensive feedback and refinement. This iterative process ensured that the CDM reflects the expertise and requirements of all clinical stakeholders in PROTECT-CHILD and is fit for purpose for paediatric transplantation research in a federated, EHDS-oriented environment.

4.3 PROTECT-CHILD CDM: Description of the model and FHIR/OMOP interoperability

This section provides a detailed description of the common data model (CDM) adopted in PROTECT-CHILD for paediatric liver and kidney transplantation. It introduces the selected core variables and the corresponding tables that define their structure and organisation within the model. In addition, an entity-relationship schema is provided, offering a comprehensive view of the architecture and implementation approach of the PROTECT-CHILD CDM. Together, these elements clarify how clinical, registry and multi-omics data are represented, and how the model supports the project's research and analysis objectives in transplanted children.

30 https://www.go-fair.org/fair-principles/

4.3.1 Entity-relationship schema

The entity-relationship schema included in this section depicts the logical structure of the PROTECT-CHILD CDM and the relationships between its main entities. It provides a visual representation of how key concepts such as Patient, Donor, Transplant, Clinical Event (e.g. rejection, infection), Treatment (immunosuppression and concomitant therapies), Laboratory Measurement and Biosample/Omics are linked through their attributes and associations. This schema is a central tool for understanding how the different data elements fit together in the model. By examining the ERD, clinicians, data scientists and other stakeholders can better grasp the dependencies and flows between entities, which in turn facilitates consistent implementation of the CDM and effective analysis of PROTECT-CHILD data across centres.

Entities description:

- **Patient:** The *Patient* table represents the child or adolescent who has undergone a liver or kidney transplant. It stores core demographic and identifying information (e.g. sex, date of birth, current age, blood group) and acts as the central anchor for most other clinical, microbiology, immunological, and omics-related tables.
- **Donor:** The *Donor* table captures information about the organ donor associated with a transplant. It includes donor identifiers, age, donor type (living/deceased), organ-specific details (e.g. liver graft type) and blood group, and is linked to Transplant to describe donor-recipient pairs.
- **Visit:** The *Visit* table records each clinical visit or study time point for a patient within a given transplant episode. It links the patient and transplant to the data collected at that time (e.g. bio samples, pre-medication, clinical variables, concomitant episodes, microbiology) and stores the visit date and visit type as recorded in the CRF.
- **Transplant:** The *Transplant* table describes each transplant procedure performed on a patient. It connects the patient and donor and includes key surgical and peri-operative information such as transplant date, transplant type, donor-recipient weight ratio, cold and warm ischaemia times, vascular anomalies, biliary anastomosis type and intra-operative complications.
- **Pre_medication:** The *Pre_medication* table stores information on relevant treatments and prophylaxis administered before transplantation. It includes, for example, antihypertensive treatment, rituximab and antiviral prophylaxis, linked to the patient and referenced from pre-transplant visits.
- **Clinical_variable:** The *Clinical_variable* table contains vital signs and other basic clinical measurements captured at specific dates. Typical variables include weight, height, systolic and diastolic blood pressure, heart rate, oxygen saturation and body temperature, all linked to the corresponding patient.
- **Concomitant_disease:** The *Concomitant_disease* table defines the list of clinically relevant concomitant diseases and comorbidities considered in PROTECT-CHILD (e.g. portal hypertension manifestations, renal or metabolic conditions). It acts as a controlled vocabulary referenced by concomitant disease episodes.
- **Concomitant_medication:** The *Concomitant_medication* table records medications given in the context of a concomitant disease episode. Each row links a *Concomitant_episode* to one applied drug or treatment, capturing the name of the medication and enabling detailed description of comorbidity management.
- **Concomitant_episode:** The *Concomitant_episode* table represents episodes of concomitant disease for a patient. It links a patient to a specific *Concomitant_disease*, records the date of the episode and includes a free-text description, allowing longitudinal tracking of comorbidities alongside the transplant course.

- **Microbiology:** The *Microbiology* table captures key microbiology and virology results for both donor and recipient. It includes EBV and CMV DNA and serology, parvovirus B19, HSV, adenovirus and VZV serology, and BK virus tests (blood, urine, biopsy), linked to the relevant patient (and donor when applicable).
- **Instrumental_investigation:** The *Instrumental_investigation* table defines the catalogue of instrumental or imaging procedures used in the study (e.g. abdominal ultrasound, CT, MRI, scintigraphy). It provides the identifiers and names of these tests and is referenced by patient-level investigation records.
- **Pat_inst_inv:** The *Pat_inst_inv*, abbreviation for Patient_instrumental_investigation, table records individual instrumental investigations performed on a patient. It links a patient to a specific *Instrumental_investigation*, storing the date, result and any relevant observations, and thereby documents the imaging and instrumental work-up over time.
- **Immunosuppressant:** The *Immunosuppressant* table lists the immunosuppressive drugs used in PROTECT-CHILD. It provides identifiers and names for agents such as calcineurin inhibitors, mycophenolate and mTOR inhibitors, and is referenced by the patient-level induction and maintenance therapy tables.
- **Imm_ind_pat:** The *Imm_ind_pat*, abbreviation for Immunosuppression_induction_patient, table describes induction immunosuppressive therapy given to a patient around the time of transplant. It links the patient to one or more Immunosuppressant agents and records the dose and unit, capturing the initial immunosuppressive strategy.
- **Imm_main_pat:** The *Imm_main_pat,* abbreviation for Immunosuppression_maintenance_patient table captures maintenance immunosuppressive regimens over time. For each patient and immunosuppressant, it records the dose, unit, drug levels (trough levels, cyclosporine C2, AUC) and treatment start and end dates, enabling reconstruction of longitudinal immunosuppression exposure.
- **Lab_test:** The *Lab_test* table defines the catalogue of laboratory tests considered in the model. It includes a lab test identifier, the name of the test and a reference to the measurement unit, and is used to standardise and reference results stored in the *Lab_result* table.
- **Lab_result:** The *Lab_result* table stores patient-level laboratory measurements. Each row links a patient to a specific *Lab_test*, and records the unit, date and numerical value of the test, supporting longitudinal analysis of kidney, liver and general biochemical parameters.
- **Mark_imm_func:** The *Mark_imm_func*, abbreviation for markers_immunological_function, table contains markers of immunological function related to transplant immunology. It records, per patient, information such as HLA antibody class, MFI values, donor-specific antibody status and non-HLA antibodies (e.g. AT1R, ETAR, MICA, MICB, AECA or others), enabling detailed characterisation of humoral alloimmunity.
- **Post_event:** The *Post_event* table represents clinical events occurring after transplantation. Each record links a patient to a *Post_event_type*, with onset and end dates and a description, and is used to capture episodes such as rejection, infection, vascular or biliary complications and other post-transplant events.
- **Post_event_type:** The *Post_event_type* table defines the list of post-transplant event categories used in PROTECT-CHILD (e.g. acute rejection, chronic rejection, specific infections, thrombosis). It provides identifiers and names and serves as a controlled vocabulary for the *Post_event* table.

- **Outcome:** The *Outcome* table gathers outcome-level information for a patient at specific dates, linked to an *Outcome_type*. It can reference other entities (e.g. pre-medication, clinical variables, concomitant episodes, microbiology, instrumental investigations, lab tests, post events) and is intended to represent key outcome assessments such as graft status and patient status.
- **Outcome_type:** The *Outcome_type* table enumerates the outcome categories used in the project (e.g. graft survival, graft loss, death, retransplantation, clinical remission). It provides identifiers and names and is referenced by the *Outcome* table.
- **Bio_sample:** The *Bio_sample* table represents biological samples collected from patients for omics analyses. It links a patient to one or more *Genomic_test, Epigenome_study* or *Methylomic_study* records and stores sample-level metadata such as collection date, shipment date to the reference laboratory, tissue type and DNA concentration.
- **Reference_genome:** The *Reference_genome* table describes the reference genome builds used in genomic and methylomic analyses (e.g. GRCh37, GRCh38). It holds identifiers, build names and related technical details (such as the FASTA path) and is referenced from *Genomic_test* and *Methylomic_study*.
- **Target_region:** The *Target_region* table defines genomic target regions associated with a given genomic test. It links to *Genomic_test* and includes HGNC identifiers and symbols, chromosome, start and end positions, and length, describing which genes or regions were interrogated.
- **Genomic_test:** The *Genomic_test* table captures metadata about each genomic sequencing assay performed (e.g. WGS, exome or panel). It records the reference genome, test name and version, sequencing device, target capture, read type and length, coverage metrics and the bioinformatics tools and databases used for alignment, variant calling and annotation.
- **Variant_occurrence:** The *Variant_occurrence* table stores per-variant, per-sample information linking genomic findings to the defined target regions. It includes identifiers for the variant occurrence, the associated *Target_region*, sequence identifiers (transcript, rsID), alleles, HGVS cDNA and protein notations, coverage metrics, copy-number or structural variant information, and genotype and origin (somatic/germline).
- **Variant_annotation:** The *Variant_annotation* table contains downstream annotations for each *Variant_occurrence*. It stores information such as the annotation database used, variant origin and pathogenicity classification, classification/tier levels, allele frequency and associated medication or pharmacogenomic relevance, supporting clinical interpretation and research analyses.
- **Epigenome_study:** The *Epigenome_study* table holds metadata about epigenomic array or methylation experiments associated with a genomic test. It includes identifiers for the epigenome study and genomic test, details of bisulfite kits and dates, age at DNA extraction, array and scanner information (e.g. device, sentrix barcodes and positions, chip type, manifest and scan dates) and sample-specific technical parameters.
- **Methylomic_study:** The *Methylomic_study* table stores probe or region-level methylation measurements and related metrics. It links to the *Reference_genome* and *Target_region* and contains values such as beta values, M values, detection p-values, read counts and methylated read counts, providing the quantitative methylome layer that can be integrated with clinical and genomic data.

4.3.2 Data model

The specification for the PROTECT-CHILD data model is available in the data model spreadsheet31.

- **Variable Name:** coded name for the variable.
- **Dataset:** specifies if the variable comes from the Protocol or the Peter Registry.
- **Organ:** specifies the organ with which the variable is related, kidney, liver or both.
- **DataElementConcept:** provides the name for the variable without special characters and whitespaces, following the convention {entity}_{variable name}. This helps automatic processing tools to quickly identify the variable and its corresponding entity.
- **DataElementConceptDef:** the definition/description of the variable.
- **VariableInPeter:** name of the equivalent variable in the Peter registry.
- **FormalConceptualDomain:** the technical specification of the data type. Possible values:
    - ID
    - Boolean
    - Code: a code or set of codes from the OHDSI standard vocabularies
    - CustomCode: a code or set of codes from OHDSI standard vocabularies and, or, new codes that need to be defined as they were not found in the OHDSI standard vocabularies.
    - ElementReference: reference to other entities
    - Float
    - Integer
    - Date
    - Calculated
    - String
- **Required:** specifies whether the variable is Mandatory (M), Recommended (R) or Optional (O).
- **ExpectedValue:** set of possible values identified by clinicians. These need to be mapped to codes within the OHDSI standard vocabularies when available. We use the codes to achieve semantic interoperability

These are not all the columns available for the variables, for further information refer to the data model spreadsheet.

4.4 Metadata model

Within PROTECT-CHILD, the metadata model is designed to achieve several closely related goals. It provides a structured framework for capturing and managing information about data quality, findability and governance across all data made available through the EHDS capsules. These three components form the backbone of the metadata layer: quality metadata describe how reliable and complete the data are for different purposes; findability metadata make datasets and services discoverable through catalogues and search tools; and governance metadata record the legal, ethical and access conditions under which the data can be used.

In developing this first version of the PROTECT-CHILD metadata model, we build on existing work from the literature and from European initiatives such as TEHDAS2 and HealthDCAT-AP,

31 https://docs.google.com/spreadsheets/d/14GGb9WyFRpgTBtbrWXYxnlen--o6uXCGdsma6TsEI3c/edit?usp=sharing

adapting them to the specific constraints of federated, EHDS-style Secure Processing Environments.

The primary objective of this metadata model is therefore to support transparent discovery, responsible access and informed reuse of PROTECT-CHILD data. By systematically capturing quality, findability and governance metadata, we aim to improve how clinicians and researchers understand what data exist in each centre, how those data were produced and transformed, and under which conditions they can be analysed using the federated services provided by the platform.

The model is explicitly intended to evolve. As the common data model is refined, new data sources are onboarded and feedback is collected from users of the platform, the metadata structures will be iteratively updated and extended in subsequent project phases and deliverables. This iterative approach ensures that the metadata model remains aligned with both the technical architecture and the emerging regulatory and standards landscape around the European Health Data Space.

4.4.1 Data quality checks

In this section we describe the metadata model for data quality developed within PROTECT-CHILD. Our approach is grounded in widely used and validated reliability dimensions from previous work in medical informatics and the bioinformatics community, which we also adopt as a baseline. Given the federated nature of PROTECT-CHILD and the need to support cohort-based discovery and feasibility assessment, we extend these existing taxonomies with additional quality metadata. In particular, we incorporate a relevance dimension (as already proposed by the European Medicines Agency (EMA)32), introduce different hierarchical levels at which quality checks are executed, and explore the feasibility of defining qualitative rankings that grade data quality according to predefined criteria.

We first define the hierarchical levels on which quality checks will be applied. This layered view makes it possible to maintain a continuous picture of data quality across the different parts of the PROTECT-CHILD ecosystem. Four levels are distinguished:

- **Variable level:** At this level, quality metadata characterise the quality of individual variables using specific metrics (e.g. completeness, conformance, plausibility), allowing fine-grained inspection of critical fields.
- **Data-source level:** Quality checks applied to each source system (e.g. local EHR modules, registry extracts, omics pipelines) provide insight into the quality of the inputs used to populate the PROTECT-CHILD capsules in each centre. This information supports local data stewards and managers in assessing the strengths and weaknesses of their sources and identifying potential improvement areas.
- **Dataset level:** Here, quality metadata summarise the properties of the integrated PROTECT-CHILD dataset within a capsule, i.e. after ETL from the underlying sources into the common data model. This level reflects the quality of the curated dataset that is actually exposed for federated analysis in each site.
- **Federated level:** Finally, a federated quality layer aggregates information from the capsule-level datasets across all participating centres. This federated view provides an overall assessment of data quality across the PROTECT-CHILD network and supports cross-centre feasibility assessments and interpretation of federated analytics results.

4.4.1.1 Reliability quality checks

32 Data Quality Framework for EU medicines regulation. EMA. 2022-09-30

For reliability metadata in the PROTECT-CHILD ecosystem, we will use the following dimensions.

Completeness. Refers to the extent to which all required and expected data elements or values are present. This quality metric will be available for each of the variables defined in the PROTECT-CHILD CDM. Later, the values will be aggregated to have completeness metrics in the different hierarchical levels.

Table 20: Description of possible completeness quality metrics

Description	Dimension
Mandatory field is missing	Completeness
Date of transplant is not specified	Completeness
Date of diagnosis is not specified	Completeness
Unit is not specified for lab result	Completeness

Conformance. This dimension assesses the extent to which data values comply with the formats, constraints and coding rules defined in the PROTECT-CHILD CDM. It covers checks such as whether variables respect their declared data types, allowed ranges and code lists, and whether relationships between tables (e.g. between patients, transplants, visits and events) satisfy the structural rules of the model. Conformance indicators are computed at multiple hierarchical levels: for individual variables, for each source system feeding a capsule, for the integrated capsule dataset and at federated level. This allows us to evaluate not only whether a single field is correctly encoded, but also how well entire datasets and the overall network adhere to the expected standards. This multi-level view provides a comprehensive picture of how closely PROTECT-CHILD data follow the predefined requirements and modelling choices across centres and data domains.

Table 21: Description of possible conformance quality metrics

Description	Dimension
Blood group of patient or donor is not between the possible codes	Conformance – Value - Verification
Diastolic blood pressure is over 150	Conformance – Value - Verification
Age at transplant is over 18	Conformance – Value - Verification
Cold and warm ischemia time is negative	Conformance – Value - Verification

Temporal plausibility. This dimension evaluates whether the timing of events and measurements in PROTECT-CHILD is clinically and logically coherent. It assesses if time-varying variables and events (e.g. transplant date, follow-up visits, infections, rejection episodes, changes in immunosuppression, graft loss, death) occur in an order and with intervals that are consistent with clinical knowledge and expected care pathways in paediatric transplantation. Temporal plausibility checks can, for example, verify that a transplant occurs after birth, that induction therapy precedes maintenance therapy, that rejection episodes and related biopsies or treatment changes follow a transplant and not the other way round, and that laboratory trends over time (e.g. creatinine, liver enzymes) evolve within realistic temporal patterns. As with the

other dimensions, temporal plausibility is summarised through aggregated indicators at different hierarchical levels (variable, data source, capsule dataset and federated network), providing an overview of the reliability and validity of the temporal information captured across centres and data domains.

Table 22: Description of possible temporal plausibility quality metrics

Description	Dimension
Transplant date is before diagnosis date	Temporal plausibility - State transitions
Pre-transplant related visits are done after transplant date	Temporal plausibility - State transitions
Post-transplant events occurred before transplant date	Temporal plausibility - State transitions
Graft loss date is before transplant date	Temporal plausibility - State transitions
Lab results don't fall within a defined time window around a visit or transplant	Temporal plausibility - State transitions
Genomic, epigenomic or methylomic studies are done before bio sample collection date	Temporal plausibility - State transitions
Immunosuppression induction date is way before transplant date	Temporal plausibility - State transitions
Immunosuppression maintenance date is before immunosuppression induction date	Temporal plausibility - State transitions

Atemporal plausibility. This dimension assesses whether the values and distributions observed in PROTECT-CHILD data make sense ignoring time. It focuses on whether data agree with clinical and logical expectations in paediatric transplantation: typical ranges for physiological variables, realistic distributions of clinical characteristics, and coherent combinations of categorical values. Atemporal plausibility checks help detect impossible or highly unlikely values (e.g. weights, lab results, drug doses) and implausible patterns in the data, independently of when they occur in the patient timeline. By systematically applying these checks, PROTECT-CHILD can verify that the non-temporal aspects of the data are consistent with established paediatric and transplant knowledge, and identify potential errors in coding, units or ETL.

Table 23: Description of possible atemporal plausibility quality metrics

Description	Dimension
Transplant type is liver and kidney side is populated	Atemporal plausibility - Verification
Transplant type is kidney and type of surgical biliary anastomosis indicated	Atemporal plausibility - Verification
Pre transplant dialysis type is not selected and dialysis duration is specified	Atemporal plausibility - Verification
Transplant type is kidney and 2nd warm ischemia time is populated	Atemporal plausibility - Verification

Description	Dimension
Combined transplant type and kidney side not populated	Atemporal plausibility - Verification
Last PRA value greater than maximum PRA	Atemporal plausibility - Verification
Patient/Donor Rh recorded but AB0 missing	Atemporal plausibility - Verification
Numeric lab result without a unit specified	Atemporal plausibility - Verification

4.4.2 Findability

Because PROTECT-CHILD operates as an orchestrated federation of EHDS capsules rather than a single central repository, the HealthDCAT-AP model needs to clearly distinguish between (i) datasets that physically exist inside each capsule and (ii) a conceptual, federated dataset that represents their union. This distinction is also reflected in the architecture of the Data Discovery layer, where local catalogues inside capsules are periodically synchronised with a federated metadata registry managed by the Data Discovery Orchestrator.

At capsule level, each centre exposes one or more **HealthDCAT-AP** (dcat:Dataset) **records** that describe the curated CDM datasets actually stored and analysed within its Secure Processing Environment. These capsule-level datasets correspond to the integrated views produced at the end of the local data preparation phase (e.g. “PROTECT-CHILD clinical cohort La Paz capsule”).

On top of the capsule-level records, the federated catalogue maintains a **conceptual `dcat:Dataset`** that represents the PROTECT-CHILD cohort as a whole. This federated dataset does not correspond to a single physical database; rather, it is a logical view over all capsule-level datasets that satisfy the PROTECT-CHILD inclusion criteria (paediatric liver and kidney transplant recipients, specified age range, core CDM variables and omics subsets).

Table X summarises the main metadata areas we will use in our HealthDCAT-AP profile, and how each is applied at capsule and federated level.

Table 24: Description of main metadata areas used from HealthDCAT-AP for PROTECT-CHILD project in both capsule and federated levels

Metadata area	Main HealthDCAT-AP terms	How we use them in PROTECT-CHILD
Identification and custodianship	dct:title, dct:description, dct:identifier, dct:publisher	At **capsule level**, these terms identify each local dataset (e.g. “PROTECT-CHILD clinical cohort – La Paz capsule”), provide a brief description, assign a stable identifier, and name the local data holder as publisher. At **federated level**, the same terms are used for a conceptual dataset record that describes the PROTECT-CHILD cohort as a whole, with the consortium or coordinating institution as publisher/steward, without replacing the role of local data holders.

Metadata area	Main HealthDCAT-AP terms	How we use them in PROTECT-CHILD
Thematic and clinical characterisation	dcat:theme, dcat:keyword, healthdcatap:healthCategory	At **capsule level**, these properties indicate that the dataset concerns paediatric liver and/or kidney transplantation, specify relevant age groups, and highlight which domains are present locally (clinical, registry, WGS, methylome). At **federated level**, they summarise the overall thematic scope of PROTECT-CHILD (paediatric liver and kidney transplant recipients across participating countries) and the main domains available somewhere in the network, while detailed availability is discovered by inspecting capsule records.
Coverage and population	dct:spatial, dct:temporal, healthdcatap:populationCoverage	**Capsule** datasets use these elements to describe where and when their cohort is defined (country/centre, study period) and to provide approximate numbers of children and transplants, derived from local profiling and quality checks. The **federated** record uses the same pattern to summarise coverage at network level (union of countries and time window) and to give approximate total counts and basic breakdowns by organ and age group, obtained by aggregating capsule-level information.
Data content and modalities	dct:type, dct:subject, dcat:distribution	**Capsule** records indicate which types of data are actually present in that capsule (core clinical variables, registry fields, WGS, methylome, etc.), and may note important limitations (e.g. “no omics data before 2020”). The **federated** record uses the same descriptors to present a high-level view of the modalities available across the PROTECT-CHILD network, making clear that some modalities may only exist in a subset of capsules.
Governance, legal basis and sensitivity	dct:accessRights, dpv:hasPersonalData, dpv:hasPurpose, dpv:hasLegalBasis, healthdcatap:hdab	At **capsule** level, these terms capture whether the dataset contains personal data, for which purposes reuse is allowed (e.g. research, quality improvement), and on what legal/ethical grounds (study protocol, registry mandate, ethics approval). Where applicable, the record links to the relevant national Health Data Access Body. At **federated** level, the same vocabulary is used more generically to explain that access always happens under capsule-specific legal frameworks

Metadata area	Main HealthDCAT-AP terms	How we use them in PROTECT-CHILD
		coordinated by the PROTECT-CHILD governance model, and to summarise the main purposes and legal bases common across the project.
Access conditions and procedures	dcat:landingPage, dct:rights, dct:source, dct:relation	**Capsule-level** metadata refer to human-readable landing pages, institutional or national portals, and policy documents that explain how to request access for that specific site, and briefly describe any key conditions or constraints. The **federated** record, in turn, points to central PROTECT-CHILD governance documentation and explains that formal access requests are routed to the appropriate national or institutional processes, using the links provided in the capsule records.
Technical access points and services	dcat:Distribution, dcat:accessURL, dcat:DataService, dcat:accessService	For **capsules**, we use dcat:Distribution primarily to describe the access, and dcat:DataService/dcat:accessService to represent the technical services exposed by the capsule (e.g. OMOP query service, FHIR API) that can be orchestrated once permits are granted. At **federated** level, the conceptual dataset is linked via dcat:accessService to federation-wide services such as the Data Discovery Orchestrator and cross-capsule analytics, which operate using computation-to-data principles and orchestrate calls to the capsule services without moving raw data.
Provenance and lineage	dct:provenance, dct:source	Capsule **records** use these properties to describe how local EHRs, registries and omics pipelines are transformed into the PROTECT-CHILD common data model within the capsule, and to link to local ETL or data preparation documentation. The **federated** record summarises the shared provenance principles (harmonisation to a common data model in each capsule, agreed mappings and validation steps) and may point to overarching ETL/mapping documentation, while leaving the detailed lineage of each centre to the capsule-level records.
Quality and analytics artefacts	healthdcatap:analytics	At **capsule** level, we associate datasets with quality and profiling outputs, either by embedding simple indicators (e.g. completeness for key variables) or by linking

Metadata area	Main HealthDCAT-AP terms	How we use them in PROTECT-CHILD
		to separate analytics artefacts through the dedicated HealthDCAT-AP property. At **federated** level, we aggregate capsule-level metrics into summary indicators (for example, overall completeness for core variables) and attach them to the federated record, clearly labelling them as federated summaries derived from the underlying capsule analytics.
Links between capsule and federated datasets	dct:isPartOf, dct:hasPart, dcat:qualifiedRelation	**Capsule** datasets indicate that they are part of PROTECT-CHILD by linking upwards to the federated dataset (e.g. “is part of PROTECT-CHILD clinical cohort”, “is part of PROTECT-CHILD omics subset”). The **federated** conceptual dataset maintains explicit downward links to all capsule-level datasets that belong to the network. These links are used by the discovery tools so that users can start from a single federated entry point and then find the relevant local datasets and services that meet their criteria.

The precise mapping to HealthDCAT-AP classes and properties, together with the PROTECT-CHILD extensions, is specified in a separate project profile document (PROTECT-CHILD HealthDCAT-AP definition), which will be kept up to date during the project. The latest version of this document will be available here 33.

33 https://docs.google.com/document/d/1SXmAgRmaN4tG07ECz5UFXk6ft7w4gDzh_EHcT1XEjLo/edit?usp=sharing

5 Eliciting Protect-Child requirements

The successful design of the PROTECT-CHILD data ecosystem depends on a rigorous and multi-layered process of requirements elicitation that integrates clinical needs, technical constraints, legal and ethical obligations, and the operational realities of the TransplantChild ERN. Chapter 6 consolidates the outcomes of this process, translating stakeholder expectations, regulatory imperatives, and technological insights into a structured set of functional and non-functional requirements for the platform. The approach follows the SPACE methodology introduced in WP2, combining top-down analysis of European initiatives such as EHDS, GDI, and GA4GH with bottom-up contributions from clinicians, data managers, legal experts, and patients' representatives.

This chapter does not merely list requirements; it explains their rationale and demonstrates how each requirement is rooted in real clinical workflows, genomic data processing needs, or cross-border data governance obligations. By bringing together inputs from user stories, co-creation workshops, system analyses, and regulatory review, the chapter establishes a coherent requirements baseline that drives the architectural decisions detailed in chapter 6. Ultimately, this systematic elicitation ensures that the resulting PROTECT-CHILD ecosystem is not only technically sound, but also ethically robust, legally compliant, clinically meaningful, and fully aligned with the EHDS user journey.

5.1 User requirements

User requirements as well as user story are described in D2.1 from task 2.1.

5.2 Legal requirements

These requirements derive from the legal framework from Task 2.2 and deliverable D2.2 – Stakeholder Requirements and legal framework and aligned with:

- GDPR (EU 2016/679) — Articles 6, 9, 25, 30, 35
- EHDS Regulation (EU 2025/327) — Articles 34, 35, 50, 73
- TEHDAS2 SPE Blueprint (2024) — "Security, compliance and governance specifications"
- ISO/IEC 27001 & 27701, ENISA Good Practices for Health Data Spaces (2024)

Table 25: Legal requirements (

ID	Description	Type	Priority	Fit criterion	Difficulty
LR_01	All processing of health data within SPEs shall comply with the legal bases of **Articles 6 and 9 GDPR**, ensuring explicit consent or another lawful ground for special-category data.	legal	must	Access to any dataset is allowed only after lawful basis validation and data-permit issuance	medium
LR_02	SPEs shall implement **data minimization** and **purpose limitation** principles; only the minimum data necessary for the approved research purpose may be processed.	legal	must	Data-use requests automatically enforce dataset scoping according to approved permit	medium

ID	Description	Type	Priority	Fit criterion	Difficulty
LR_03	Each SPE must maintain a **register of processing activities** and an **audit trail** covering dataset access, algorithm execution, and data exports, ensuring **accountability** (Art. 30 GDPR).	legal	must	Immutable audit logs record user ID, timestamp, purpose, and outcome of every operation	high
LR_04	Personal data leaving an SPE shall be **pseudonymized** or **aggregated**; re-identification shall be technically and legally prevented except under a valid legal mandate.	legal	must	System automatically enforces pseudonymization before export and blocks identifiers	medium
LR_05	Cross-border processing within the federated network shall rely on **data-permit agreements** mutually recognized by Health Data Access Bodies (HDABs) under **EHDS Art. 50**.	legal	must	Federated orchestrator checks valid permit token issued by national HDAB before data use	high
LR_06	SPE operators shall appoint a **Data Protection Officer (DPO)** and perform **Data Protection Impact Assessments (DPIA)** for all high-risk processing (Art. 35 GDPR).	legal	must	DPIA approved and archived prior to activation of each new federated task	medium
LR_07	SPEs shall implement **security-by-design and by-default** controls consistent with **ENISA** and **ISO 27001/27701** standards (encryption, access control, key management).	legal	must	Evidence of compliance through annual ENISA or ISO certification audit	high
LR_08	Data subjects shall be able to exercise **rights of access, rectification, erasure, restriction, and objection** through the data-permit interface, with responses within legal deadlines.	legal	should	Self-service portal propagates verified requests to data controllers across SPEs	medium
LR_09	Any algorithm executed within an SPE must undergo **prior compliance verification** to ensure it does not enable re-identification, bias, or unlawful profiling (AI Act alignment).	legal	must	Algorithm registry includes conformity-assessment results before deployment	high
LR_10	All data transfers and algorithm executions must be **logged and auditable** by competent authorities for at least 10 years after project completion (EHDS Art. 73).	legal	must	Long-term encrypted audit archive with verifiable retention policy	high

5.3 Clinical study requirements

This set of requirements have been extracted from D8.4 Study Protocol and study registration in clinical study registry.

Table 26: High-Level Data Analysis Requirements for the PROTECT-CHILD Clinical Study

ID	Description	Type	Priority	Fit criterion	Difficulty
DA_01	The platform shall enable secure ingestion and harmonisation of multi-source clinical data (EHRs, registries such as PETER, ModSimTher, and laboratory data) into a common data model.	Functional	Must	Successful ETL workflows validated for each site; data mapped to OMOP/FHIR schema.	High
DA_02	The platform shall allow integration of genomic (WGS/VCF/BAM) and methylomic (IDAT) datasets with clinical records under a single pseudonymised patient ID.	Functional	Must	Unique pseudonym linkage validated; all data cross-referenced in metadata registry.	High
DA_03	The system shall provide standardised preprocessing pipelines for genomic and methylomic data (ANNOVAR annotation, methylation normalization, QC reporting).	Functional	Must	Pipelines reproducibly generate annotated variant tables and β-value matrices.	High
DA_04	The analytical environment shall support polygenic risk score (PRS) computation and machine-learning models to identify predictors of immune and non-immune outcomes.	Functional	Should	PRS and predictive models reproducible; evaluation metrics (AUC, R^2, ROC) documented.	High
DA_05	The platform shall enable integrative multi-omics analysis, combining clinical, genomic, methylomic, and pharmacogenomic variables using multivariate and survival models.	Functional	Must	Multivariate regression, Cox models, and MANOVA pipelines executed with harmonised datasets.	High
DA_06	The analytical layer shall support exploratory data analysis (PCA, EFA, clustering, differential methylation, and variant frequency analysis) for hypothesis generation.	Functional	Should	PCA and clustering outputs stored in metadata repository with visual summaries.	Medium
DA_07	All analyses shall operate within Secure Processing Environments (SPEs) with in-place computation; only aggregated or anonymised outputs may leave the node.	Non-functional	Must	Privacy audit confirms no personal-level data exported; SPE logs verified.	Medium
DA_08	The system shall provide provenance, versioning, and metadata traceability for every dataset and analysis run, ensuring	Non-functional	Must	Metadata registry includes dataset version, toolchain,	Medium

	reproducibility and regulatory compliance.			parameters, and timestamps.	
DA_09	The platform shall support federated aggregation of analytical results (e.g., summary statistics, PRS distributions, methylation clusters) across sites via the Federated Computing Protocol (FCP).	Functional	Must	FCP Phase 4 aggregation validated; central summaries match node-level outputs.	High
DA_10	Analytical results shall be exportable in FAIR-compliant formats (e.g., JSON-LD, CSV, or OMOP tables) for downstream validation, publication, and registry reporting.	Functional	Should	Outputs conform to FAIR data principles; validated by schema and metadata checks.	Medium

5.4 Data requirements

Data requirements extracted from the user stories described in D2.1 from task 2.1.

5.4.1 Data model requirements

Table 27: High-Level Data Model Requirements for the PROTECT-CHILD Project

ID	Description	Type	Priority	Fit criterion	Difficulty
DM_01	The common data model (CDM) shall support FAIR data management and EHDS-ready operation: core entities (patient, donor, transplant, visit, event, biosample, etc.) must be uniquely identifiable and describable through metadata that can be exposed via HealthDCAT-AP compatible catalogues.	Non-functional	Must	CDM entities have stable IDs; capsule datasets can be published as HealthDCAT-AP Dataset records with links to CDM entities.	Medium
DM_02	The CDM shall be standard-friendly and mappable to OMOP CDM for analytics and to HL7 FHIR (including FHIR Genomics) for data exchange and interoperability. For each CDM table, mappings to at least one OMOP table/field and one FHIR resource/element shall be defined.	Non-functional	Must	Mapping documentation exists for all core tables; test ETL runs successfully load data into OMOP and FHIR views in at least one capsule per centre.	High
DM_03	The CDM shall model the paediatric liver and kidney transplant pathway rather than local IT systems. It must represent all variables required by the PROTECT-CHILD CRFs and the aligned subset of PETER variables, including patient/donor characteristics, transplant details, follow-up visits, clinical events, immunosuppression, labs, microbiology and outcomes, and link them to biosamples and omics results.	Functional	Must	Every CRF field and in-scope PETER variable can be mapped to a CDM field without semantic loss in pilot mappings.	Medium
DM_04	The CDM shall separate structure and semantics. Structural harmonisation shall be achieved through the CDM schemas (tables, keys, relationships), while semantic harmonisation shall rely on standard vocabularies (e.g. SNOMED CT, ICD-10/11, LOINC, ATC, HPO, GA4GH/GDI ontologies) and local-to-standard mapping tables.	Non-functional	Must	CDM contains no hard-coded local codes; separate terminology/mapping tables exist and are used by ETL and NLP pipelines.	Medium
DM_05	The CDM shall support computation-to-data and federated analytics. Patient-level identifiers must be pseudonymised or local	Non-functional	Must	Pilot federated analyses run on pseudonymised CDM data in at least two	High

ID	Description	Type	Priority	Fit criterion	Difficulty
	to each capsule; relationships shall not require sharing direct identifiers across sites; and the model shall provide explicit places for derived features and analysis outputs (e.g. risk scores, PRS, episignature clusters).			centres; no direct identifiers appear in exported results.	
DM_06	The CDM shall be extensible and versioned, with a stable core and extension tables for additional organs, scores or omics modalities. Each capsule shall declare the CDM version it implements.	Non-functional	Must	CDM release notes and metadata include version ID and list of core vs extension tables; capsules expose their CDM version in metadata.	Medium

5.4.2 Data standardization requirements

Table 28: High-Level Data Standardization Requirements for the PROTECT-CHILD Project

ID	Description	Type	Priority	Fit criterion	Difficulty
DS_01	The platform shall adopt OMOP and FHIR as primary reference models for clinical/registry data and recognised formats for omics data (e.g. VCF/BAM for WGS, IDAT for methylomics).	Non-functional	Must	At least one OMOP and one FHIR view are generated per capsule; WGS and methylation pipelines accept and produce VCF/BAM/IDAT as specified.	Medium
DS_02	Clinical variables in the CDM shall be coded using standard vocabularies wherever possible (SNOMED CT, ICD-10/11, LOINC, ATC, HPO, OMIM, GA4GH/GDI ontologies). Local codes and NLP-extracted terms shall be mapped to these vocabularies via reusable mapping tables.	Non-functional	Must	≥80% of coded variables are mapped to standard vocabularies; mapping tables are stored and versioned in the capsule.	High
DS_03	The platform shall provide NLP-based tools to extract structured concepts from free-text clinical documents (in Spanish, Italian, German and English where applicable) and map them to standard ontologies, exposing confidence scores and error reports to data managers.	Functional	Should	At least one NLP pipeline runs on sample notes from two centres and produces structured codes with confidence scores; manual checks confirm acceptable precision/recall.	High
DS_04	The standardisation pipeline shall harmonise units and value domains. Quantitative variables shall be converted to canonical units per lab test; categorical variables (e.g. organ, event_type, outcome_type, donor_type) shall	Functional	Must	Unit conversion rules or specified units exist for all core lab tests; value-domain conformance checks show <5% non-standard	Medium

ID	Description	Type	Priority	Fit criterion	Difficulty
	use controlled vocabularies defined in the CDM.			values after standardisation in pilot data.	
DS_05	Findability metadata for datasets, cohorts and services shall be expressed using HealthDCAT-AP (or a compatible profile). The model must support descriptors for organ type, paediatric age range, data domains (clinical/registry/omics), spatial and temporal coverage, and access conditions.	Non-functional	Must	Each capsule publishes at least one HealthDCAT-AP Dataset and DataService record for its main datasets; federated catalogue can harvest and index them.	Medium
DS_06	The platform shall implement a common schema for recording data quality indicators (at minimum completeness, conformance, atemporal and temporal plausibility) at variable, data-source, capsule and federated level.	Non-functional	Must	Quality metadata tables exist and are populated for a pilot capsule; dashboards can display quality metrics per variable and per centre.	Medium

5.4.3 Data extraction requirements

Table 29: High-Level Data Extraction Requirements for the PROTECT-CHILD Project

ID	Description	Type	Priority	Fit criterion	Difficulty
DX_01	The platform shall support extraction from multiple local sources per centre, including EHR modules, lab systems, the PETER registry (subset aligned with the study), and genomics/methylomics pipelines.	Functional	Must	For each centre, an initial extraction is successfully performed from at least one EHR source, one registry source (where applicable) and one omics pipeline.	High
DX_02	Centres without existing OMOP/FHIR infrastructures shall be able to provide exports as structured files (CSV/TSV/XLSX) placed in a designated folder or database access to source databases/database views created specifically created, from which standard scripts will generate CDM-ready input.	Functional	Must	At least one centre successfully loads data into the capsule via the file-based path; file templates and parsing scripts are documented.	Medium
DX_03	Centres that already expose OMOP and/or FHIR endpoints shall be able to extract data directly from these endpoints (“semantic-standard-driven	Functional	Should	At least one centre demonstrates direct extraction from its OMOP or FHIR endpoint into the	High

	deployment"), subject to compatibility checks with the PROTECT-CHILD profiles.			CDM without intermediate CSVs.	
DX_04	All extraction processes shall run within each centres secure environment and respect GDPR/EHDS principles. Raw patient identifiers must not leave the centre extracted datasets destined for capsules shall be pseudonymised according to local governance rules.	Non-functional	Must	Security/legal review confirms that extraction scripts run on-premise and produce only pseudonymised IDs in extracted datasets.	Medium
DX_05	Automated data-quality checks (at least completeness, conformance and plausibility) shall be executed on extracted datasets before injection; results must be stored as quality metadata and visible to data managers.	Functional	Must	For pilot extractions, quality checks run automatically and produce a report; variables failing thresholds can be flagged or excluded from injection.	Medium
DX_06	Each extraction run shall produce a provenance record including source systems, time period, script versions, record counts and errors/warnings, to support governance and audit user stories.	Non-functional	Must	Provenance entries are automatically created for each run and can be inspected through an admin or governance dashboard.	Medium

5.4.4 Data ingestion requirements

Table 30: High-Level Data Ingestion Requirements for the PROTECT-CHILD Project

ID	Description	Type	Priority	Fit criterion	Difficulty
DI_01	The ingestion process shall validate incoming data against the PROTECT-CHILD CDM (schema, keys, referential integrity, basic semantic constraints) and reject or flag records that violate these constraints. Only validated records may be committed to the capsule.	Functional	Must	Test ingestions with deliberately malformed data are blocked or flagged; capsule tables remain consistent after each ingestion.	High
DI_02	On successful ingestion, each capsule shall populate/update the OMOP and FHIR views used for analytics and interoperability, using the predefined mapping layer.	Functional	Must	After ingestion, OMOP/FHIR endpoints in at least one capsule correctly reflect the content of the CDM tables for core entities.	High

ID	Description	Type	Priority	Fit criterion	Difficulty
DI_05	Ingestion operations should be atomic at release level: either the full dataset for a release is loaded successfully, or the system rolls back to the previous consistent state.	Non-functional	Should	Simulated failure during ingestion results in automatic rollback and leaves the previous release intact; logs document failure.	High
DI_06	When a new capsule release is created, the system shall automatically generate or update: (i) HealthDCAT-AP records for exposed datasets/cohorts, (ii) data quality indicators at the defined hierarchical levels, and (iii) provenance metadata for ETL and standardisation steps.	Functional	Must	After ingestion, the catalogue shows updated datasets; quality dashboards and provenance logs are refreshed for the new release.	Medium
DI_07	After each ingestion, the platform should provide a human-readable summary for data holders and data managers, including record counts per table, rejected records and main quality indicators.	Functional	Should	A summary report is available for at least one pilot ingestion and can be downloaded or viewed.	Low
DI_08	Injected data shall be organised so that downstream services (metadata search, cohort builder, data permit tools, federated analytics, reporting) can operate without additional manual restructuring.	Non-functional	Must	Cohort builder and federated analytics can run directly on the injected release in at least one pilot scenario without extra pre-processing.	High

5.5 Technical requirements

The section translates the findings of chapter 4 in formal requirements.

5.5.1 Cybersecurity best practice requirements

Table 31: Cybersecurity best practice requirements

ID	Description	Type	Priority	Fit criterion	Difficulty
CS_01	All containerized services shall be rebased on hardened, minimal base images (Chainguard, Distroless, UBI-Minimal, or Bitnami Secure Images).	non-functional	must	Image provenance verified through SBOM; Trivy scan reports zero CRITICAL CVEs prior to deployment.	medium
CS_02	Multi-stage Docker builds shall enforce least functionality—no compilers, shells, or build tools in runtime layers.	non-functional	must	Container runtime inspection confirms absence of build dependencies and shells.	low
CS_03	All containers shall run as non-root users, with readOnlyRootFilesystem: true and restrictive seccomp/AppArmor profiles.	non-functional	must	Kubernetes securityContext enforces non-root UID; runtime audit shows no writeable filesystem paths.	medium
CS_04	Each container image shall be signed (e.g., Cosign) and accompanied by a Software Bill of Materials (SBOM) for supply-chain integrity.	non-functional	must	Signed digest verified in CI/CD pipeline; SBOM attached to artifact and validated by scanner.	medium
CS_05	CVE scanning and enforcement shall be automated in CI/CD pipelines; merges blocked on unresolved CRITICAL vulnerabilities.	process	must	CI logs show automated Trivy/Polaris scans; failed builds prevent image promotion.	low
CS_06	Runtime least privilege shall be maintained across workloads by dropping unneeded Linux capabilities and limiting network namespaces.	non-functional	should	PodSecurityPolicy or OPA/Gatekeeper rules enforce dropped capabilities; verified via audit logs.	medium
CS_07	Encryption at rest shall be implemented at all layers—host, CSI volume, and control plane (etcd, Secrets).	non-functional	must	Verification reports confirm encryption enabled for all Persistent Volumes and etcd secrets.	high
CS_08	Cloud environments shall use CSI-provisioned encrypted volumes integrated with KMS (AWS KMS, Azure Key Vault, GCP KMS).	non-functional	must	Cloud provider audit shows encrypted PVs; key rotation verified through KMS logs.	medium
CS_09	On-premises deployments shall combine host-level encryption (LUKS/ZFS) with per-volume CSI encryption integrated with Vault or HSM.	non-functional	should	Node-level encryption verified by system configuration; Vault/HSM logs show successful key rotation.	high

5.5.2 Microservices and service mesh requirements

Table 32: Microservices and service mesh requirements

ID	Description	Type	Priority	Fit criterion	Difficulty
SM_01	Adopt a unified mesh deployment model (Istio ambient or sidecar) with HA control planes per SPE namespace.	non-functional	should	Control plane pods are multi-AZ and pass failover tests without data-plane interruption.	medium
SM_02	Enable multi-cluster mesh federation for SPE-to-SPE traffic using east–west gateways, trust-domain aliases, and locality-aware routing.	non-functional	must	Cross-SPE calls succeed only via mesh gateways; cluster locality is honored in load-balancing tests.	high
SM_03	Implement permit-aware L7 policies: bind data-permit attributes (project, purpose, scope) to mesh AuthorizationPolicy/RateLimit rules at gateways.	functional	must	Requests without matching permit labels are denied or throttled at L7; audit shows rule hits.	high
SM_04	Use mesh traffic controls for progressive delivery (canary/blue-green, traffic mirroring) with automatic rollback on SLO breach.	non-functional	should	Canary rollout gates tied to mesh telemetry; failed canary auto-rolls back.	medium
SM_05	Enforce resilience defaults mesh-wide: timeouts, retries with jitter, circuit breaking, and outlier detection per service class.	non-functional	must	Fault-injection tests show no cascading failures; p95 latency within budgets under pod kills.	medium
SM_06	Standardize observability via OpenTelemetry from the mesh (metrics, logs, traces) with W3C trace-context propagation across services.	non-functional	must	≥95% of inter-service calls have end-to-end trace IDs; golden signals available per service.	medium
SM_07	Perform protocol and schema validation at mesh edges (e.g., FHIR/JSON schema/WASM filters) before requests reach workloads.	functional	should	Invalid payloads rejected at gateway; conformance reports attached to releases.	medium
SM_08	Govern egress via ServiceEntry/Waypoint: allow-list FQDNs, do TLS origination and DNS pinning, and map quotas to permit scopes.	non-functional	must	Only declared endpoints are reachable; quota exceedance blocked at egress with mesh logs.	high

ID	Description	Type	Priority	Fit criterion	Difficulty
SM_09	Enforce data-residency–aware routing: locality/failover policies keep flows within the originating SPE/geofence unless a permit explicitly allows otherwise.	non-functional	must	Chaos tests prove no cross-border failover without an explicit residency override.	high
SM_10	Manage mesh configuration as code (GitOps): lint (istioctl analyze), policy test (Conftest), and peer-review every mesh change.	process	must	CI gates block invalid CRDs; all changes traceable to reviewed commits/issues.	low
SM_11	Set resource budgets for data-plane components (sidecars/ztunnels/gateways) with autoscaling targets and SLO-based capacity plans.	non-functional	should	p99 overhead from mesh <5 ms intra-cluster; HPA keeps error rate within SLO under load.	medium
SM_12	Restrict experimentation features (fault injection, traffic shadowing) to non-prod meshes with protective policies to prevent prod application.	process	should	Policy tests prevent FaultInjection/TrafficMirror objects in prod namespaces.	low

5.5.3 Zero trust and security best practices requirements

5.5.3.1 Requirements on Zero Trust principles

Table 33: Strong Identity and Authentication requirements

ID	Description	Type	Priority	Fit criterion	Difficulty
ZT_01	All health data users and administrators shall be strongly authenticated using MFA and federated identity systems (eIDAS, GA4GH passports).	non-functional	must	The system will enforce MFA login and federated identity checks for all access.	medium
ZT_02	User and service identities shall be unique, non-transferable, and continuously verifiable throughout a session.	non-functional	must	Identity management system will enforce unique IDs and session-bound tokens.	medium
ZT_03	Machine-to-machine interactions within federated SPEs shall use workload identities (SPIFFE/SPIRE or equivalent).	non-functional	should	The system will issue cryptographic workload certificates for service-to-service trust.	high

Table 34: Least Privilege Access

ID	Description	Type	Priority	Fit criterion	Difficulty
ZT_04	Access to sensitive data shall follow the principle of least privilege and be granted only through valid data permits issued by the Governance layer of the project.	non-functional	must	The system will enforce access rights strictly aligned with data permits.	medium
ZT_05	Privileged administrative access shall be strictly temporary, monitored, and logged.	non-functional	must	Administrative access will require just-in-time approval and audit logging.	medium
ZT_06	Data users shall only access datasets relevant to their project scope as defined in the permit.	non-functional	must	The system will enforce fine-grained policies on dataset selection.	medium

Table 35: Micro segmentation/Isolation

ID	Description	Type	Priority	Fit criterion	Difficulty
ZT_07	Research projects shall run in isolated environments with no direct access to the data.	non-functional	must	The system will create per-project secure enclaves with restricted network rules.	high
ZT_08	Protect-Child SPEs shall implement segmentation to separate projects, operators, and data sources.	non-functional	must	Microsegmentation policies will be validated through penetration tests.	high
ZT_09	Cross-border SPE-to-SPE communication shall only occur via authorised federation APIs.	non-functional	must	Secure tunnels and endpoint certificates will be verified during federation tests.	high

Table 36: Data Security Everywhere

ID	Description	Type	Priority	Fit criterion	Difficulty
ZT_10	Sensitive data shall always be encrypted at rest and in transit using EU-approved algorithms.	non-functional	must	Encryption keys and logs will demonstrate use of approved cryptographic standards.	medium
ZT_11	Pseudonymisation shall be mandatory for storing personal data; only anonymised and HDAB-verified outputs may leave the SPE.	non-functional	must	Output verification process will ensure compliance with anonymisation rules.	high

ID	Description	Type	Priority	Fit criterion	Difficulty
ZT_12	Output data exports shall be restricted to formats and aggregation levels approved by HDAB.	non-functional	must	Export control procedures will demonstrate only anonymised, permitted results are released.	high
ZT_13	Cryptographic keys shall be managed through secure hardware (HSM or equivalent).	non-functional	should	Compliance tests will verify that all encryption keys are securely stored and rotated.	high

Table 37: Continuous verification

ID	Description	Type	Priority	Fit criterion	Difficulty
ZT_14	All access requests shall be continuously verified against active project permits and user identity validity.	non-functional	must	The system will re-check authorisation tokens before each data query.	medium
ZT_15	Federation access sessions shall expire automatically after a defined period of inactivity.	non-functional	must	Session expiration times will be validated in security testing.	low

Table 38: Assume breach & monitoring

ID	Description	Type	Priority	Fit criterion	Difficulty
ZT_16	The SPE shall implement real-time monitoring, anomaly detection, and incident reporting.	non-functional	must	Monitoring dashboards will trigger alerts.	high
ZT_17	Automated breach containment shall allow the immediate suspension of affected user sessions.	non-functional	must	Tests will verify that the system can terminate compromised sessions instantly.	high
ZT_18	Intrusion detection and firewalls shall monitor both internal and federated connections.	non-functional	must	IDS/IPS deployment will be tested with penetration simulations.	high

Table 39: Data Security Everywhere

ID	Description	Type	Priority	Fit criterion	Difficulty
ZT_19	All actions performed inside the SPE shall be logged, actor-identified, and audit.	non-functional	must	The system will retain logs in tamper-evident storage for at least one year.	medium
ZT_20	Logs shall be immutable and subject to audit.	non-functional	must	External audit reports will validate	medium

ID	Description	Type	Priority	Fit criterion	Difficulty
				log integrity and compliance.	
ZT_21	Audit trails shall include dataset accessed, operations performed, and justification linked to data permits.	non-functional	must	Traceability of operations will be validated through audit sampling.	medium

Table 40: Adaptive risk management

ID	Description	Type	Priority	Fit criterion	Difficulty
ZT_22	The SPE shall support continuous risk management, including vulnerability patching, change/release management, and backups.	non-functional	must	ISMS controls and risk assessment reports will demonstrate compliance.	medium
ZT_23	Disaster recovery capabilities shall allow recovery of critical services failure.	non-functional	must	Business continuity testing will demonstrate recovery time objectives.	high
ZT_24	Third-party software deployed inside SPEs must undergo security vetting before approval.	non-functional	must	Approved software lists will be validated against deployment logs.	high

Table 41: Federated trust

ID	Description	Type	Priority	Fit criterion	Difficulty
ZT_25	The SPE federation shall support identity interoperability across borders using trusted frameworks (eIDAS, GA4GH).	non-functional	should	Federated login tests will validate interoperability with external SPEs.	high
ZT_26	The system shall support federated analysis and federated learning without transferring raw sensitive data across borders.	non-functional	must	Federated queries will only return aggregated results or model updates.	high
ZT_27	Interoperability across SPEs shall rely on shared standards (OMOP CDM, GA4GH APIs).	non-functional	must	Cross-border pilots will validate API-level interoperability.	high

5.5.3.2 Requirements on security best practices

Table 42: Inherit CVE mitigation strategy

ID	Description	Type	Priority	Fit criterion	Difficulty
ZT_CVE_01	All service containers shall be based on vetted, hardened, minimal images (Chainguard, Distroless, Red Hat UBI Minimal/Micro, Bitnami Secure Images, or approved slim/alpine).	non-functional	must	CI/CD pipelines verify that only approved base images are used and contain no CRITICAL CVEs.	medium
ZT_CVE_02	Containers shall run as non-root users with read-only filesystems and minimal runtime privileges.	non-functional	must	Kubernetes manifests validated to include runAsNonRoot: true and readOnlyRootFilesystem: true.	medium
ZT_CVE_03	All built images shall be signed and include a Software Bill of Materials (SBOM) to guarantee provenance and traceability.	non-functional	must	Cosign signatures and SBOMs verified during deployment admission checks.	medium
ZT_CVE_04	The CI/CD pipeline shall fail any build containing unresolved CRITICAL vulnerabilities or configuration violations.	non-functional	must	Trivy and Polaris scans pass before merge; audit logs show 0 unresolved CRITICAL CVEs.	medium
ZT_CVE_05	Image hardening shall support privacy-by-design by eliminating exploitable binaries and reducing the risk of data exposure.	non-functional	should	Security and privacy assessments cite hardened images as mitigating controls in DPIA/TRA.	medium

Table 43: Encryption at rest

ID	Description	Type	Priority	Fit criterion	Difficulty
ZT_EN_01	All environments (development, cloud, and on-premises) shall implement encryption at rest for all stored data, volumes, and backups.	non-functional	must	Audits confirm that all storage volumes and backups are encrypted using approved methods.	medium
ZT_EN_02	Developer environments using Docker or lightweight Kubernetes shall rely on host-level disk encryption (LUKS, FileVault, or BitLocker) to protect container volumes.	non-functional	must	Security review verifies that encrypted disks are enabled and Docker runtime directories reside on encrypted partitions.	low
ZT_EN_03	Production cloud Kubernetes clusters shall provision encrypted Persistent Volumes using CSI drivers integrated with the cloud	non-functional	must	StorageClass manifests and CSI configurations verified for KMS	medium

ID	Description	Type	Priority	Fit criterion	Difficulty
	provider's KMS (AWS KMS, Azure Key Vault, or GCP KMS).			integration and encryption flags enabled.	
ZT_EN_04	The Kubernetes control plane shall encrypt Secrets and ConfigMaps in etcd using an AES-based encryption provider.	non-functional	must	API server configuration validated to include --encryption-provider-config with AES-CBC or aescbc providers.	medium
ZT_EN_05	On-premises clusters shall combine infrastructure-level encryption (e.g., LUKS, ZFS) with CSI-level volume encryption (Ceph-CSI, Longhorn, OpenEBS) integrated with a secure key manager.	non-functional	must	Security audits confirm per-volume encryption and integration with Vault or HSM-backed KMS.	high
ZT_EN_06	Encryption keys shall be managed separately from the data they protect, using centralized KMS or Vault policies for generation, access, and rotation.	non-functional	must	Key lifecycle management logs show periodic rotation and access control enforcement.	high
ZT_EN_07	Automated verification shall ensure that volumes, snapshots, and backups remain encrypted and that keys are rotated on a defined schedule.	non-functional	should	CI/CD or monitoring jobs produce periodic encryption-compliance reports.	medium

5.5.3.3 Overlap legal and Zero Trust requirements

The following table summarises the overlap between the Zero Trust (ZT) architecture requirements and the legal obligations defined under the GDPR, EHDS Regulation, and TEHDAS2 SPE Blueprint. Each ZT control group is mapped to its corresponding legal basis, highlighting the most inclusive and representative implementation requirement.

To ensure efficiency and regulatory completeness, only the most inclusive requirements—those that inherently fulfil both technical and legal compliance dimensions—will be implemented within the Protect-Child Secure Processing Environments (SPEs). These requirements integrate security-by-design, data-protection-by-design, and federated governance principles, ensuring alignment with EU law while minimising redundancy across controls.

Table 44: Overlap legal and Zero Trust requirements

#	Zero Trust Area	Overlapping Legal Requirements	Legal Reference (EU Reg.)	Best Implementation Requirement (Recommended Control)
1	Strong Identity & Authentication	LR_01 (Lawfulness), LR_03 (Accountability), LR_07 (Security by Design)	GDPR Art. 32 & 25; EHDS Art. 50 (eIDAS identity trust)	ZT_01 – MFA + federated identity (eIDAS/GA4GH) → foundation for all access control

#	Zero Trust Area	Overlapping Legal Requirements	Legal Reference (EU Reg.)	Best Implementation Requirement (Recommended Control)
2	Least Privilege Access	LR_01 (Lawful basis), LR_02 (Data minimization), LR_05 (Data permits)	GDPR Art. 5(1)(b,c); EHDS Art. 50	ZT_04 – Permit-based least privilege access → ensures purpose limitation & traceability
3	Micro-Segmentation / Isolation	LR_07 (Security by Design)	GDPR Art. 25 & 32; ENISA SPE Blueprint	ZT_07 – Per-project isolated enclaves → guarantees confidentiality and containment
4	Data Security Everywhere	LR_04 (Pseudonymisation), LR_07 (Encryption)	GDPR Art. 32(1)(a); EHDS Arts. 34 & 50	ZT_10 + ZT_11 – End-to-end encryption and mandatory pseudonymisation
5	Continuous Verification & Session Control	LR_03 (Audit trail), LR_07 (Security measures)	GDPR Art. 32 & 30	ZT_14 – Continuous permit check per query → maintains real-time compliance
6	Monitoring & Incident Response	LR_03 (Audit), LR_10 (Logging retention)	GDPR Art. 30; EHDS Art. 73	ZT_16 – Real-time monitoring & incident reporting → fulfils legal accountability
7	Adaptive Risk Management	LR_06 (DPIA), LR_07 (ISMS controls)	GDPR Art. 35 & 32 (DPIA, risk mitigation)	ZT_22 – Continuous risk and patch management → keeps SPE compliant over time
8	Federated Trust & Cross-Border Data Sharing	LR_04 (Pseudonymisation), LR_05 (Cross-border permit)	EHDS Arts. 50 & 73; eIDAS Reg. 910/2014	ZT_26 – Federated analysis without raw data transfer → meets GDPR Art. 44 on transfers

5.5.4 Federated computing requirements

The implementation of the European Health Data Space (EHDS) introduces an unprecedented need for technical, legal, and procedural mechanisms that enable the secure secondary use of health data across Member States. Within this context, Federated Computing (FC) has emerged as a cornerstone technology for privacy-preserving data processing, providing a harmonised alternative to traditional centralised models of data sharing.

Instead of transferring sensitive data to a central repository, Federated Computing enables computation to occur where the data reside, allowing only aggregated, anonymised, or model-derived information to be exchanged. This approach is fully consistent with the GDPR principles of data minimisation and purpose limitation, while fulfilling the EHDS objective of facilitating cross-border research and innovation without compromising individual privacy or institutional sovereignty.

At its core, Federated Computing combines two complementary paradigms:

- Federated Learning (FL), an inductive method focused on collaboratively training predictive models across distributed nodes; and

- Federated Analytics (FA), a deductive approach that performs distributed statistical reasoning across multiple datasets without sharing raw data.

Together, FL and FA form a coherent computational fabric capable of supporting a wide range of biomedical and clinical use cases — from exploratory research and epidemiological studies to AI model development and real-time clinical decision support.

To ensure that Federated Computing can operate as a trusted, interoperable, and compliant infrastructure under the EHDS framework, a set of technical and governance requirements has been identified through systematic analysis, expert consultation, and experimental validation. These requirements translate the guiding principles of the EHDS — security, interoperability, privacy, accountability, and trust — into actionable criteria for the design and deployment of federated infrastructures.

Table 45: Table caption example before table

ID	Description	Type	Priority	Fit criterion
R_FC_01	Federated Computing shall support both Federated Learning (FL) and Federated Analytics (FA) to enable predictive modelling and statistical reasoning across distributed datasets.	non-functional	must	The system will expose APIs for FL (iterative model training) and FA (one-round statistical analysis).
R_FC_02	All federated computations shall comply with GDPR and EHDS provisions, ensuring that sensitive data never leaves its originating institution.	non-functional	must	Compliance audits will demonstrate that only model updates or aggregated results are exchanged.
R_FC_03	A standardized Federated Computing Protocol (FCP) shall structure the workflow into setup, planning, configuration, execution, and validation phases.	non-functional	must	The system will follow a documented 12–13 step lifecycle aligned with EHDS Article 73.
R_FC_04	Federated Computing frameworks shall provide governance mechanisms allowing data holders to approve or reject tasks before execution.	non-functional	must	The system will implement a Federated Governance Module with selective execution policies.
R_FC_05	Federated Computing shall include strong authentication and algorithm verification mechanisms to ensure only validated tasks are executed.	non-functional	must	The system will integrate a Federated Authentication System with workload certificates and algorithm validation.

ID	Description	Type	Priority	Fit criterion
R_FC_06	Federated Computing platforms shall provide interoperability across frameworks and Member States by adopting shared standards (REST/gRPC APIs, ONNX for FL, orchestration layer for FA).	non-functional	must	Interoperability tests will confirm API compliance and model exchange compatibility.
R_FC_07	Federated Computing shall integrate privacy-preserving techniques, including Differential Privacy, Homomorphic Encryption, and SMPC, where appropriate.	non-functional	should	Privacy-enhanced federated workflows will be demonstrated in at least one biomedical pilot.
R_FC_08	Federated Computing shall handle heterogeneous (non-IID) data through adaptive aggregation and personalization strategies.	non-functional	should	Pilot validation will demonstrate support for clustered FL, FedProx, or personalization techniques.
R_FC_09	The system shall ensure scalability to multi-omics, imaging, and unstructured datasets by optimizing communication efficiency and computation distribution.	non-functional	must	Benchmarks will demonstrate acceptable performance for large-scale biomedical data.
R_FC_10	Federated Computing shall integrate with EHDS Secure Processing Environments (SPEs), ensuring compliance with Article 73 technical and security requirements.	non-functional	must	The system will be deployed inside or connected SPEs.
R_FC_11	The Federated Computing architecture shall support auditability and accountability, retaining logs of computations, approvals, and aggregated outputs.	non-functional	must	Audit trails will be stored for ≥ 1 year and verifieable by external auditors.
R_FC_12	Federated Computing shall support cross-border biomedical use cases by enabling collaboration between institutions in multiple EU Member States.	non-functional	must	Pilot deployments will demonstrate successful FL/FA between at least 3 EU countries.
R_FC_13	Federated Computing shall prioritize privacy guarantees over accuracy when conflicts arise, ensuring that compliance is never compromised.	non-functional	must	Differential privacy settings or equivalent mechanisms will be enforced in all workflows.
R_FC_14	Federated Computing shall allow hybrid approaches (e.g., Quantum Computing, FL +	non-functional	should	At least one hybrid workflow will be

ID	Description	Type	Priority	Fit criterion
	DP, FL + HE/SMPC) to balance scalability, accuracy, and privacy guarantees.			validated in a clinical pilot.
R_FC_15	The system shall be adaptable to future extensions, including training of large-scale models (e.g., LLMs) across distributed infrastructures.	non-functional	could	Roadmaps and proof-of-concepts will demonstrate feasibility for fine tuning LLM-oriented use cases.

5.5.5 Genomics requirements

Table 46: Genomic Analysis and Annotation Requirements (ANNOVAR-based VarSeq Emulation in the Protect-Child Platform)

ID	Description	Type	Priority	Fit criterion	Difficulty
GEN_01	The SPE shall implement a secure and efficient pipeline for the ingestion, persistent archival, and management of raw FASTQ files, ensuring data integrity via checksum verification, storage optimization through compression (gzip/bgzip), and strict control of access limited exclusively to authorized personnel.	Functional & Non-Functional (Security)	Must	All FASTQ files are successfully ingested, integrity-verified, compressed by >70%, and linked to core sample metadata upon storage, with access limited exclusively to authorized roles.	High
GEN_02	The SPE shall integrate ANNOVAR for genomic annotation of input VCF (variant call) and BAM (alignment) files, enabling standardized prediction of variant effects (SNV, INDEL, CNV).	functional	must	ANNOVAR pipeline successfully annotates uploaded VCF/BAM within the SPE without external data transfer.	medium
GEN_03	The platform shall support modular ANNOVAR plugin execution (e.g., ClinVar, gnomAD, dbNSFP, CADD) to reproduce VarSeq-like clinical annotation and enrichment functionalities.	functional	must	Execution logs demonstrate plugin activation with metadata stored for each run.	high
GEN_04	All VCF and BAM files processed by ANNOVAR shall remain encrypted at rest and in transit within the SPE boundaries, using EU-approved algorithms (AES-256, TLS 1.3).	non-functional	must	Encryption compliance verified through cryptographic logs and mTLS communication tests.	medium

ID	Description	Type	Priority	Fit criterion	Difficulty
GEN_05	The system shall enable rule-based variant filtering and prioritization workflows (e.g., ACMG classification, allele frequency thresholds) equivalent to VarSeq filter chains.	functional	must	Filter pipelines defined as JSON/YAML templates reproduce ACMG 5-tier variant classification.	high
GEN_06	Annotation outputs (csv files) from ANNOVAR shall be normalized into OMOP Genomic Extension and FHIR-Genomics formats to ensure semantic interoperability across federated SPEs.	functional	should	Data model mapping validated via schema compliance tests and OMOP/FHIR validators.	high
GEN_07	Variant-level metadata and provenance (genome build, annotation release, plugin versions, input file checksum) shall be automatically captured for auditability and reproducibility.	non-functional	must	Provenance logs comply with FAIR principles and can be exported as JSON-LD or PROV-O records.	medium
GEN_08	Pseudonymisation and controlled linkage of patient identifiers shall be mandatory for all genomic data processed by ANNOVAR.	non-functional	must	SPE logs confirm no un-pseudonymised identifiers are persisted or exported.	high
GEN_09	The platform shall support federated aggregation of annotated variant statistics (e.g., allele frequencies, functional impact) across multiple SPEs without exposing individual VCF/BAM data.	functional	should	Aggregated JSON summaries computed through secure federated queries or MPC aggregation.	high
GEN_10	Role-Based Access Control (RBAC) and audit trails shall restrict execution, visualization, and export rights for genomic workflows (VCF/BAM upload, ANNOVAR run, output download).	non-functional	must	OIDC/SPIFFE authentication and per-action audit logs validated through penetration testing.	medium
GEN_11	The system shall include internal visualization modules (HTML or Jupyter-style dashboards) for interactive summaries of variant type, impact, and pathogenicity, reproducing VarSeq's GUI capabilities within the SPE.	functional	could	Rendered visualizations accessible only through authenticated sessions inside SPE boundaries.	medium
GEN_12	The ANNOVAR environment shall be containerized and version-controlled (e.g., Docker/Singularity images) to ensure reproducibility of results across nodes.	non-functional	must	Container digest and runtime versions logged in provenance records.	low

ID	Description	Type	Priority	Fit criterion	Difficulty
GEN_13	Federated job orchestration (ANNOVAR → filtering → classification → report generation) shall be integrated with the Federated Computing Protocol (FCP) defined in Protect-Child to ensure standardized execution and validation.	functional	must	Workflow successfully executed under FCP Phase 3 (Execution) with validation logs registered.	high
GEN_14	Input Validation: The system shall verify that all uploaded VCF and BAM files conform to GA4GH before annotation.	functional	must	Files failing validation are rejected and logged; conformance checked via HTSlib or GA4GH tools.	

Table 47: High-Level Requirements for Beacon v2 Integration (as GDI- integration)

ID	Description	Type	Priority	Fit criterion	Difficulty
BCN_01	Each SPE shall deploy a GA4GH Beacon v2 endpoint to enable federated discovery of genomic and phenotypic variants derived from VCF/BAM metadata.	Functional	Must	Beacon v2 queries return compliant JSON responses within SPE boundaries.	Medium
BCN_02	Beacon indexes shall include only pseudonymised, aggregated metadata (variant, biosample, cohort) while raw genomic data remain confined within the SPE.	Non-functional	Must	Data inspection verifies no identifiers or raw sequences exposed.	High
BCN_03	The platform shall support federated query resolution via a Beacon Network Aggregator, interoperable with GDI and ELIXIR AAI nodes.	Functional	Must	Cross-node test queries return harmonised JSON-LD results.	High
BCN_04	All endpoints must be secured with OAuth2 / OIDC, mTLS, and SPIFFE/SPIRE identities, enforcing Zero-Trust communication and role-based access (OPA/Gatekeeper).	Non-functional	Must	Auth and mTLS validation logs confirm policy enforcement.	Medium
BCN_05	Beacon v2 shall support phenotype-linked discovery through GA4GH Phenopackets and FHIR Genomics, aligned with OMOP clinical data in Protect-Child.	Functional	Should	Phenotype-based queries retrieve consistent cross-domain results.	High
BCN_06	Privacy-preserving query controls (e.g., k-anonymity, differential privacy, or binning) shall mitigate re-identification risks in rare-variant queries.	Non-functional	Must	Privacy assessment confirms risk < 0.09 per GDPR Recital 26.	High

ID	Description	Type	Priority	Fit criterion	Difficulty
BCN_07	Beacon responses shall include data-use and consent metadata (DUO, GA4GH Consent Codes) and be fully traceable to HDAB permissions.	Functional	Must	DUO codes and consent provenance fields verified in API responses.	Medium
BCN_08	Beacon services shall integrate with the Federated Computing Protocol (FCP Phases 1–2) as the discovery layer for genomic tasks, with analytics and governance metrics reported to HDAB.	Functional	Must	FCP logs show Beacon invocation during Planning/Configuration; governance dashboard receives usage data.	Medium

Table 48: High-Level Requirements for Methylome Analysis

ID	Description	Type	Priority	Fit criterion	Difficulty
MET_01	The SPE shall support import, validation, and parsing of Illumina IDAT files (Red/Green channel) for methylation profiling, ensuring data integrity and schema compliance.	Functional	Must	IDAT checksum validation and metadata extraction succeed without external data transfer.	Medium
MET_02	The platform shall implement standardised preprocessing workflows (background correction, normalization, probe filtering, detection p-value computation) using open frameworks such as minfi or SeSAMe.	Functional	Must	Workflow successfully produces β-value and M-value matrices consistent across runs.	High
MET_03	All IDAT-derived data shall remain encrypted and pseudonymised inside the SPE, and any export limited to anonymised summary statistics or epigenetic signatures.	Non-functional	Must	Security audit confirms encryption (AES-256) and no personal identifiers in exports.	High
MET_04	The analysis pipeline shall generate epigenome-wide methylation matrices and episignature detection results, interoperable with genomic (VCF/BAM) outputs via shared subject identifiers.	Functional	Must	Matrices and episignature metadata integrated in OMOP/FHIR-Genomics model.	High
MET_05	The system shall support reference-based and unsupervised methylation clustering, enabling identification of disease- or transplant-related CpG patterns.	Functional	Should	Clustering output validated against reference methylation signatures (e.g., EpiSign).	High
MET_06	Quality control reports (probe performance, detection rate, bisulfite conversion metrics) shall	Non-functional	Must	QC report generated per run; all metrics logged in provenance record.	Medium

ID	Description	Type	Priority	Fit criterion	Difficulty
	be automatically generated and stored for reproducibility.				
MET_07	The methylome workflow shall be containerised and version-controlled (Docker/Singularity) for reproducibility across SPEs and future extension into Federated Computing.	Non-functional	Must	Container digests and tool versions logged in provenance metadata.	Medium
MET_08	Federated aggregation of methylation summary data (e.g., mean β-values by gene or CpG island) shall be enabled through the Federated Computing Protocol (FCP) while keeping patient-level data local.	Functional	Must	Aggregated summaries computed securely across nodes; FCP logs confirm Phase 4 aggregation step.	High

5.5.6 NLP requirements

The integration of Natural Language Processing (NLP) and Generative Artificial Intelligence (GenAI) into the PROTECT-CHILD platform introduces a new dimension of technical and governance requirements beyond traditional data processing systems. The two-stage pipeline---entity extraction from unstructured clinical narratives and automated mapping to OMOP standardized concepts---must operate within the constraints of Secure Processing Environments while maintaining data sovereignty, regulatory compliance, and clinical validity. The following requirements define how NLP capabilities shall be implemented, validated, and governed to ensure that automated data extraction and structuring serve the EHDS vision of trustworthy, interoperable health data reuse without compromising privacy, accuracy, or institutional control.

Table 49: NLP requirements

ID	Description	Type	Priority	Fit criterion
R_NLP_01	The NLP extraction pipeline shall operate entirely on-premises within Secure Processing Environments (SPEs), ensuring that clinical data never leaves the hospital network during processing.	non-functional	must	All data processing logs and audit trails confirm zero external data transmission; clinical documents remain encrypted at rest within SPE storage.
R_NLP_02	2Entity extraction from unstructured clinical documents shall be performed using open-source language models (Mistral, Medgemma, or equivalent) rather than closed-source API-based services, to maintain data sovereignty and compliance with GDPR Article 6 and EHDS Article 50	functional	must	Model deployment verified to use only open-source implementations; no API calls to external LLM providers detected in audit logs.
R_NLP_03	Extracted medical entities shall be constrained to conform to predefined	functional	must	All extracted entities pass schema

ID	Description	Type	Priority	Fit criterion
	structured schemas (JSON, Pydantic models) before output, ensuring consistency and parseability of all extracted information.			validation; malformed outputs are rejected with logged error; 100% structural compliance verified in validation testing.
R_NLP_04	The extraction pipeline shall support diverse clinical document types including pre-transplant evaluations, surgical reports, discharge summaries, histopathology reports, outpatient notes, laboratory results, immunological assessments, and imaging reports.	functional	must	Test dataset covering all document types demonstrates entity extraction with documented performance metrics per document category.
R_NLP_05	Extracted entities shall be automatically mapped to standardized OMOP Common Data Model concepts via semantic understanding and terminological API consultation (Athena), ensuring interoperability across European health systems.	functional	must	Mapping agent successfully queries Athena API; mapped entities include valid OMOP concept_ids; manual validation confirms semantic accuracy of mappings.
R_NLP_06	The NLP pipeline shall include confidence scoring mechanisms for entity extraction and OMOP mapping, enabling automatic escalation to human review when confidence falls below defined thresholds.	functional	should	System assigns confidence scores to all extractions and mappings; low-confidence cases (< 0.7) are flagged for human review with documented audit trail.
R_NLP_07	Entity extraction models shall be evaluated and validated against manually annotated clinical reference datasets, using standard metrics (precision, recall, F1-score) to establish baseline performance prior to production deployment.	non-functional	must	Validation dataset of minimum 200 annotated documents from partner institutions; quantitative metrics reported for each entity type and document category.
R_NLP_08	The NLP pipeline shall support fine-tuning or adaptation of extraction models using domain-specific clinical examples without requiring retraining from scratch, enabling rapid customization to institution-specific terminology and reporting conventions.	non-functional	should	Few-shot prompting or selective fine-tuning demonstrated on representative clinical examples; performance improvement documented relative to base model.

ID	Description	Type	Priority	Fit criterion
R_NLP_09	All NLP processing operations, entity extractions, confidence scores, mapping decisions, and human review actions shall be logged with timestamps, user/system identity, and justification, maintaining an immutable audit trail for regulatory compliance and transparency.	non-functional	must	Audit logs demonstrate complete traceability of all extraction and mapping decisions; logs retained for minimum 10 years post-processing; tampering prevention verified.
R_NLP_10	The NLP extraction pipeline shall be designed and deployed to align with privacy-by-design and security-by-design principles, integrating encryption at rest, access controls, and data minimization throughout all processing stages.	non-functional	must	Architecture review confirms encryption, access policies, and data minimization practices; security assessment validates compliance with GDPR Articles 25, 32.
R_NLP_11	Extracted clinical entities shall be pseudonymized or anonymized before export from the SPE, with personal identifiers removed or encrypted and re-identification technically and legally prevented except under valid legal mandate.	functional	must	Automated pseudonymization or anonymization rules verified to remove or mask patient identifiers; re-identification testing confirms no identifier leakage in outputs
R_NLP_12	The NLP pipeline shall support federated extraction and mapping workflows, enabling distributed computation across multiple SPEs without raw clinical data transfer, returning only aggregated or model-derived results.	non-functional	should	Federated extraction tested in pilot deployment with multiple SPEs; no raw data exchange between sites confirmed in network monitoring; only aggregated results transmitted.

6 Architecture

This chapter presents the architectural foundation of the PROTECT-CHILD platform, describing how its technical, security, and governance layers converge to enable federated, privacy-preserving research on sensitive paediatric transplant data across Europe. The architecture translates the regulatory vision of the European Health Data Space (EHDS) into a concrete, operational framework: one that keeps data securely within institutional boundaries while allowing computation, analytics, and learning tasks to move safely between trusted environments.

The platform is conceived as a federated digital ecosystem composed of multiple EHDS Capsules—Secure Processing Environments (SPEs) deployed in hospitals and research centres—and a central Orchestrator that coordinates computation, identity, and compliance across sites. Each capsule enforces data sovereignty and local control, while the Orchestrator ensures global coherence, traceability, and interoperability through policy-driven orchestration, federated identity, and continuous verification mechanisms.

Technically, PROTECT-CHILD is built upon a cloud-native, zero-trust infrastructure where Docker ensures reproducibility, Kubernetes provides isolation and resilience, and Istio weaves all microservices into a secure, observable, and policy-enforced service mesh. Every service, whether dedicated to data processing, governance, discovery, genomics, or federated computing, interacts through REST APIs and OpenID Connect (OIDC) authentication with cryptographically verifiable JSON Web Tokens (JWTs), ensuring that every request carries a validated identity, purpose, and data-use permit.

The chapter is structured to provide both the deployment perspective—showing how the system is technically realised through container orchestration, hardened images, encryption, and mesh-based zero trust—and the logical perspective, explaining how the Orchestrator, Capsules, and Governance layer interoperate as a compliant EHDS federation. Subsequent sections describe the functional components that make this architecture operational—data preparation, discovery, federated computing, governance, genomics, quantum computing, and virtual assistants—and the connectors that ensure secure, standardised communication across all layers.

Together, these architectural elements define PROTECT-CHILD not as a single platform but as a federated network of secure, interoperable environments—a European infrastructure where sensitive data remain protected, algorithms travel securely, and knowledge circulates under continuous ethical, legal, and technical supervision.

6.1 System architecture

The PROTECT-CHILD system architecture operationalises the EHDS vision through a federated-by-design stack that keeps data local while enabling cross-border computation. At its core, each hospital runs an EHDS Capsule—a Secure Processing Environment (SPE) where clinical, genomic, and methylomic data are harmonised in OMOP/FHIR and processed under strict zero-trust controls. A central Orchestrator coordinates studies across capsules, dispatching authorised algorithms to the data (“computation-to-data”) and aggregating only anonymised or model-derived outputs. Interactions are decoupled and policy-enforced via REST APIs authenticated with OpenID Connect and JWT/RBAC, so every request carries a verifiable identity, purpose, and permit context issued by the independent Governance layer.

Technically, the platform is delivered as a modern cloud-native stack: Docker provides reproducible, hardened images; Kubernetes offers resilient, multi-tenant orchestration inside

each SPE; and Istio weaves these services into a zero-trust service mesh with mTLS, workload identities (SPIFFE/SPIRE), traffic policy, and full observability. This architecture supports privacy-preserving federated analytics (Vantage6), EHDS-compliant data discovery (DCAT-AP Health), Beacon-based genomics, and optional hybrid quantum-classical pipelines—while maintaining institutional sovereignty and auditability end-to-end. The following subsections detail the deployment stack and security posture, the zero-trust mesh, and the logical view that ties capsules, orchestrator, governance, and assistants into a cohesive, compliant platform.

6.1.1 Deployment view

6.1.1.1 Deployment stack

The deployment stack of the PROTECT-CHILD platform is founded on three interdependent technologies — Docker, Kubernetes, and Istio — which together form the operational backbone of the Secure Processing Environments (SPEs). Each of these layers contributes a specific function to the platform's overall architecture, evolving from containerization, to orchestration, to secure, federated communication. The combination ensures that the platform can operate in a distributed, compliant, and privacy-preserving manner across the various European clinical and research institutions involved.

At the foundation, Docker serves as the elementary unit of deployment and reproducibility. Every service, from data-processing microservices to analytics components and privacy modules, is encapsulated within Docker containers. This encapsulation guarantees that the same component behaves identically in every context, whether in a developer's workstation, in a test cluster, or in the hospital's production environment. Docker images are built from minimal, hardened bases, ensuring that no unnecessary binaries or system libraries remain within them, thus reducing the attack surface and supporting the principles of privacy by design. Each image is signed, scanned, and accompanied by a software bill of materials, creating an immutable and traceable artifact. In this way, Docker becomes more than a runtime environment — it is a foundation of compliance, reproducibility, and security, forming the basis for everything that runs within the PROTECT-CHILD SPEs.

Above Docker sits Kubernetes, which acts as the orchestration and control layer for all deployed workloads. Kubernetes provides the infrastructure-level automation that allows the platform to be resilient and self-healing, ensuring that containers are correctly scheduled, monitored, and replaced whenever they fail or need to be updated. Within each Secure Processing Environment, Kubernetes governs how resources are isolated, granting each project or study its own namespace and quota, and ensuring that network, storage, and compute isolation are strictly maintained. It also enforces encryption at rest, integrates with secure key management systems, and applies continuous configuration management to maintain the desired state of each service. For PROTECT-CHILD, Kubernetes is what transforms isolated containers into an intelligent, regulated ecosystem. It enables multi-tenant operation within each hospital or data-holding institution, allowing them to run analytics, model training, and governance services locally while maintaining full control over their data. The platform thereby achieves both operational autonomy and consistency, so that all SPEs can be federated without compromising national or institutional sovereignty.

Yet, while Kubernetes coordinates how services run within a single environment, it does not provide a native mechanism for how they communicate securely across environments or even across microservices within the same cluster. That function is realized by Istio, which introduces a powerful service mesh that overlays all Kubernetes workloads. Istio acts as the connective tissue of the PROTECT-CHILD ecosystem, transforming a collection of distributed services into a coherent, trusted network of interacting agents. Its core principle is that no communication

within or across SPEs is ever implicit. Every interaction between services is authenticated, encrypted, and continuously verified. In practice, this means that all traffic between containers — whether within one hospital cluster or across federated sites — is protected through mutual TLS, with cryptographic workload identities automatically issued and rotated by Istio. Each service within the mesh can communicate only according to predefined policies, and all interactions are transparently logged for auditability. This approach embodies the zero-trust paradigm at the infrastructure level: trust is never assumed but is established anew for every request, every session, and every connection.

Istio's role in PROTECT-CHILD extends beyond encryption. It governs how data flows traverse the federated network, ensuring that every transmission adheres to the project's data-permit system and the European Health Data Space's rules for cross-border data processing. When an authorized federated computation is executed, Istio's east-west gateways manage the interconnection between hospital SPEs, establishing a secure channel through which only aggregated or model-derived information can pass. Locality-aware routing ensures that computation remains, by default, within the originating jurisdiction unless a data permit explicitly authorizes its transfer. This configuration guarantees both compliance and performance, preventing unintended data movement while allowing collaborative computation to take place in real time.

In addition, Istio provides the means for fine-grained governance at the application layer. It interprets the semantics of each request and evaluates whether it corresponds to the lawful purpose encoded in the data permit. This is achieved by binding permit attributes such as project scope, purpose, and duration to Istio's policy and rate-limiting mechanisms. If a service call does not meet these criteria, the mesh rejects it before the underlying application ever processes it. Through this mechanism, legal compliance is no longer an external auditing task but an intrinsic, programmable property of the platform's network fabric.

The observability features of Istio close the loop between compliance, performance, and accountability. Every transaction within the mesh is accompanied by telemetry data that records latency, error rates, and trace identifiers, allowing a continuous view of the system's behavior. This observability not only serves the operational teams but also contributes to transparency and regulatory assurance. Combined with Kubernetes' automated resilience and Docker's reproducibility, these telemetry streams form the basis for automated risk detection and adaptive response within the platform's governance model.

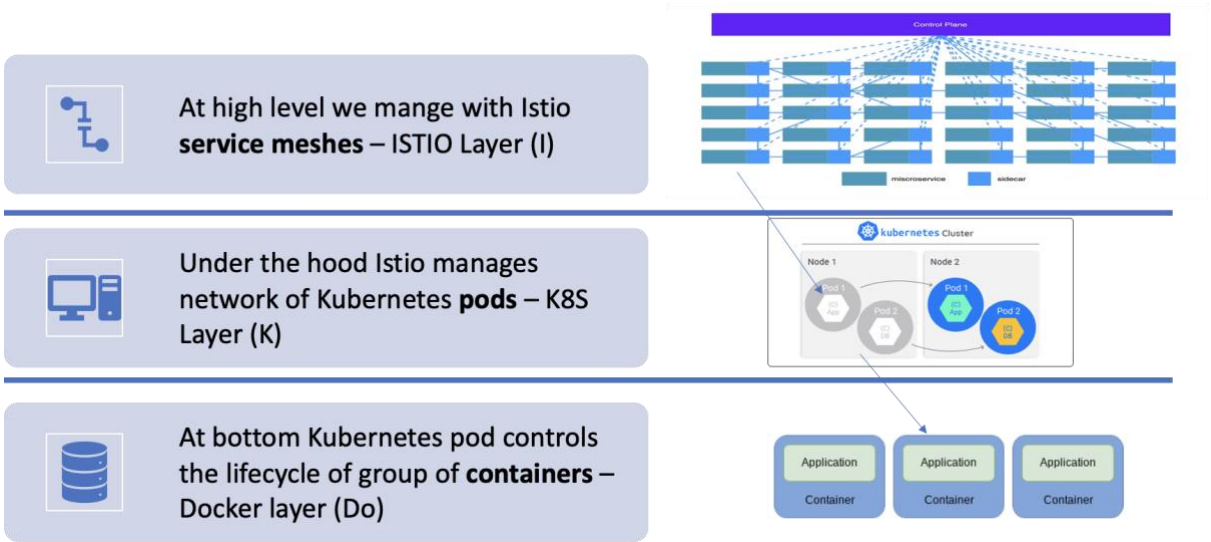

Figure 15: Deployment stack

In the PROTECT-CHILD deployment, these three layers — Docker, Kubernetes, and Istio — operate in concert as a single, integrated stack. Docker ensures that each microservice is secure, consistent, and portable. Kubernetes orchestrates those microservices in a way that is scalable, self-healing, and compliant with the isolation principles required by Secure Processing Environments. Istio unifies the resulting distributed landscape into a federated, zero-trust mesh where communication is authenticated, encrypted, observable, and policy-driven. Together they allow the project to move beyond traditional data sharing and toward a model of data cooperation, in which sensitive pediatric and genomic data remain protected at their source while still enabling collaborative research, analytics, and machine learning across Europe. This architecture is not only a technical choice but an ethical commitment to privacy, accountability, and trust as the foundations of scientific progress.

6.1.1.2 Inherit CVEs mitigation strategy

To pass security review and ship to production, we always need to eliminate critical CVEs and configuration gaps by rebasing every service conatainer onto hardened, minimal container images and enforcing secure build/run practices end-to-end. Concretely, update Dockerfiles to use vetted base:

- Chainguard[34] (cgr.dev/chainguard/, e.g., .../postgres, .../java, .../python, .../nginx),
- Google Distroless[35] (gcr.io/distroless/, e.g., .../java21-debian12, .../python3-debian12, .../cc),
- Red Hat UBI[36] (registry.access.redhat.com/ubi9/ubi-minimal or ubi9/ubi-micro),
- Bitnami Secure Images[37] (docker.io/bitnami/* or ghcr.io/bitnami/containers/*, e.g., bitnami/postgresql, bitnami/nginx, bitnami/tomcat), or—where appropriate and pinned—the official slim/alpine variants (e.g., postgres:16-alpine, httpd:2.4-alpine).

To adopt a hardened-image strategy it's important because most of today's critical CVEs don't originate in novel app code—they arrive "for free" through bloated base images and permissive runtime defaults. Every extra shell, compiler, or package are wrapped in the code multiplies the attack surface, inflates the SBOM with vulnerable components we never use, and makes patching unpredictable. By rebasing services onto minimal, security-maintained images (Chainguard/Wolfi, Google Distroless, Red Hat UBI-Minimal/Micro, Bitnami Secure Images, or carefully pinned slim/alpine where appropriate), we remove entire classes of vulnerabilities rather than chasing them one by one. This is the essence of risk reduction: eliminate what you don't need, then harden what remains.

This approach is also a recognized best practice across modern cloud security: "least functionality" at build time via multi-stage Dockerfiles (so build tools never land in the runtime layer), "least privilege" at run time (run as a non-root user, drop write access with readOnlyRootFilesystem: true, and—where feasible—drop Linux capabilities and set a restrictive seccomp/apparmor profile), and "immutable infrastructure" through strict version and digest pinning. Together, these give repeatable builds, predictable SBOMs, and a tight dependency graph that scanners can evaluate accurately. Adding image signing (e.g., Cosign) and attaching SBOMs provides provenance and tamper evidence, allowing reviewers to verify what you built is exactly what you deploy.

Practically, this yields three big wins for PROTECT-CHILD security review posture.

[34] Chainguard image repository, https://images.chainguard.dev/
[35] Google Distroless Image repository, https://gcr.io/distroless/
[36] Red Hat UBI minimal image repository, https://catalog.redhat.com/en/software/containers/ubi9/ubi-minimal/615bd9b4075b022acc111bf5
[37] Bitnami Secure images repository, https://hub.docker.com/u/bitnami

- First, vulnerability volume and severity drop immediately when we stop inheriting outdated userlands; the Trivy reports become shorter, more relevant, and easier to clear.
- Second, configuration risk declines: non-root containers with read-only filesystems are far more resilient to breakout attempts, web-shell implants, or lateral movement—if an attacker lands in a pod, they have fewer tools and fewer write targets.
- Third, supply-chain assurance improves: digests, signatures, and SBOMs give auditors concrete evidence of what's running, which images it came from, and whether known CVEs affect those components.

The same choices also strengthen our privacy posture. Security and privacy are inseparable: confidentiality of patient and research data depends on keeping adversaries out and limiting their blast radius if they get in. Minimal images reduce the likelihood that exploitable binaries or vulnerable libraries can be used to exfiltrate data. Read-only filesystems and non-root users inhibit tampering with application code, logs, or agent sidecars that could otherwise leak identifiers. Pinned, signed images and reproducible builds make it easier to demonstrate to assessors that we control our processing environment—an important part of DPIA/TRA narratives and of "privacy by design" claims. In short, the mitigation directly lowers the probability and impact of data exposure events, which feeds into better risk scores in security and privacy assessments alike.

What this means for you as a developer is straightforward but powerful: rebasing to hardened images is not just about "making Trivy green." It's how we stop importing unnecessary risk, prove provenance, and run with the least privileges possible. Concretely, you will: switch your Dockerfiles to the recommended hardened bases; use multi-stage builds so compilers, git, ImageMagick and other build tools never appear in the final image; ensure the container runs as a non-root user and that Kubernetes manifests set readOnlyRootFilesystem: true (and, where feasible, drop capabilities and add seccomp/apparmor); pin image tags and digests; generate SBOMs and sign images; then rescan with Trivy and Polaris before merging. We'll enforce this in CI/CD (builds fail on unresolved CRITICALs), not as bureaucracy, but because it measurably reduces attack surface, simplifies audits, and strengthens our privacy compliance story. This is the fastest, most durable path to getting production-ready without last-minute security surprises.

6.1.1.3 Global strategy for encryption at rest

A comprehensive encryption-at-rest strategy should adapt to the context in which the data is stored and processed, balancing practicality, security, and performance across environments.

In local Docker-based or lightweight Kubernetes development setups, the best practice is to rely on host-level disk encryption—for example, LUKS on Linux, FileVault on macOS, or BitLocker on Windows—to transparently protect the Docker or container runtime directories where volumes reside. This keeps developer workflows simple while ensuring that any lost or stolen device does not expose database or container data. Optionally, sensitive development databases can also use built-in encryption (e.g., MariaDB TDE, PostgreSQL pg_tde, or SQLCipher) for testing data-handling pipelines under real security conditions.

In cloud-hosted Kubernetes production clusters, encryption should be enforced at multiple layers: storage classes must provision encrypted Persistent Volumes via CSI drivers that integrate with the provider's key management system (AWS KMS, Azure Key Vault, or Google Cloud KMS), while the Kubernetes API server should encrypt Secrets and ConfigMaps in etcd using AES-CBC or aescbc providers. This ensures that both data at rest in storage and control-plane metadata are protected, with auditable key rotation and centralized governance through the cloud KMS.

For on-premises Kubernetes installations, where organizations control the entire storage stack, the recommended approach combines infrastructure-level encryption (LUKS or ZFS native encryption on the nodes) with CSI-level per-volume encryption provided by drivers such as Ceph-CSI, Longhorn, or OpenEBS, ideally integrated with a secure key manager like HashiCorp Vault. Developers can use simplified configurations with keys stored in Kubernetes Secrets, while production deployments should delegate key generation, access policies, and rotation to Vault or an HSM-backed KMS. Across all environments, effective encryption at rest means separating encryption keys from the data they protect, automating rotation, verifying that volumes and backups remain encrypted, and applying these controls consistently from the developer's laptop to the production cluster.

6.1.1.4 Zero trust deployment architecture

The Zero Trust deployment architecture represented in figure 17 depicts the secure, identity-centric network model implemented through **Istio** within a **single Kubernetes cluster** augmented by a **single external virtual machine**. This configuration embodies the principle of "never trust, always verify," using **mutual TLS (mTLS)**, **namespace-level isolation**, and **federated identity integration through OpenID Connect (OIDC)** to ensure that every communication within and across services is authenticated, authorized, and continuously verified. It is the backbone for the orchestrator and capsules environments.

The Zero Trust deployment architecture adopted in PROTECT-CHILD is implemented through a unified **Istio Service Mesh** running on a single Kubernetes cluster, which may also extend to a companion virtual machine to integrate legacy or specialized governance services. Within this architecture, **each namespace in the service mesh represents a functional domain of the platform**—a self-contained environment hosting the core services that share a common purpose or data sensitivity level. This design not only simplifies governance and scalability but also ensures that the Zero Trust principles of isolation, identity verification, and continuous authorization are enforced at the most granular level of the system [ref https://protect-child.eu/implementing-zero-trust-with-istio/].

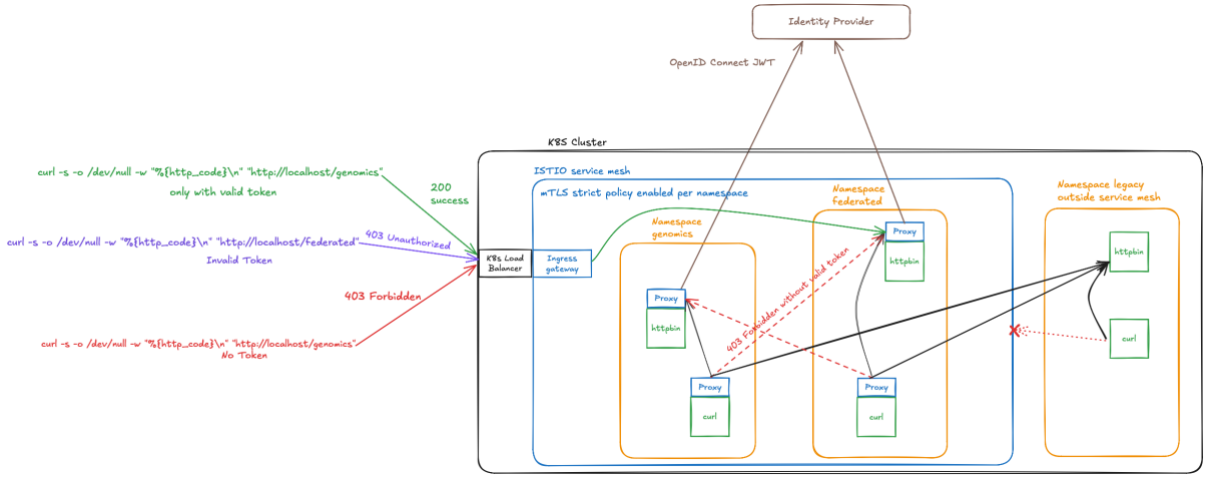

Figure 16: Protect-Child zero trust service mesh architecture

In this model, the Kubernetes cluster operates as the foundational substrate on which different namespaces host the distinct functional layers of the PROTECT-CHILD platform. For instance, one namespace may group all **data processing and ingestion services**, responsible for receiving, cleaning, and pseudonymizing sensitive data within the Secure Processing Environment (SPE). Another namespace may host the **federated computing services**, which orchestrate privacy-preserving computations such as federated analytics or learning workflows.

A third namespace serves as the **data discovery and interoperability layer**, enabling controlled metadata exchange and catalog access aligned with EHDS standards, while yet another provides the **governance and compliance services**, ensuring auditability, permit validation, and legal traceability. Finally, specialized namespaces may host the **Beacon and genomics services**, supporting research-oriented access to omics datasets through standardized APIs. Each of these namespaces is a distinct trust zone: it contains only the microservices and data stores required for its function, and all internal traffic is protected by **strict mTLS policies** that encrypt communication and authenticate workloads with SPIFFE/SPIRE-issued certificates.

Although all namespaces belong to a single Istio service mesh, they are connected through clearly defined, policy-controlled communication channels. Istio's control plane maintains the service registry and distributes dynamic certificates, while each namespace operates under **PeerAuthentication and AuthorizationPolicy** rules that enforce intra-namespace trust and inter-namespace isolation. Traffic between namespaces can occur only through **Layer-7 gateways**, where identity, purpose, and data-permit attributes are validated before being forwarded. This architecture ensures that services performing privacy-critical functions—such as governance, data processing, or analytics—remain isolated yet interoperable under the same Zero Trust umbrella. In this way, the mesh becomes a federated fabric of secure enclaves, each one representing a core functional domain of PROTECT-CHILD.

The **Ingress/Egress Gateway** sits at the boundary of the mesh, acting as the controlled interface between internal namespaces and the external ecosystem. Every inbound and outbound communication—whether directed toward other Secure Processing Environments, Health Data Access Bodies, or virtual assistants supporting the EHDS user journey—must traverse this gateway. Here, Istio enforces **mutual TLS** between internal workloads and external clients, while applying authentication and authorization based on **OpenID Connect tokens** issued by the platform's Identity Provider (such as Keycloak). Requests are evaluated against project-specific data-permits managed by the governance namespace, ensuring that only authorized actions consistent with legal and ethical approval are allowed.

In the PROTECT-CHILD architecture, the **Kubernetes cluster is deployed inside a dedicated Virtual Machine**, which provides an additional **layer of physical and administrative isolation** for each capsule. Within this virtualized environment, Istio implements a **Zero Trust service mesh** that interconnects the core services of the platform—data processing, federated computing, data discovery, governance, and genomics—grouped into separate namespaces according to their functionality. The service mesh enforces **mutual TLS, workload identity, and namespace-level authorization**, ensuring that every interaction between services is authenticated, encrypted, and policy-controlled. External communication between capsules and the central orchestrator does not occur directly between workloads but through **federated mesh gateways**, which exchange traffic exclusively via mTLS-encrypted channels validated by certificate-based trust. This configuration allows each capsule to remain **physically and administratively independent**, with its own virtualized SPE hosting a self-contained Kubernetes and Istio environment. The federation of meshes enables secure, policy-driven coordination between capsules and the orchestrator, while maintaining strict isolation of each local administrative domain. The result is a unified yet sovereign infrastructure, where Zero Trust is applied consistently from the virtual machine boundary down to each namespace within the cluster.

Under this model, security, privacy, and interoperability are not applied as external controls but emerge organically from the way the system is structured. Dockerized microservices provide reproducibility and isolation; Kubernetes orchestrates them within namespaces that reflect real functional boundaries; and Istio weaves them together into a **cryptographically enforced trust fabric**. Every request within the mesh—whether it moves between controllers in the same

namespace or across functional domains—carries a verified identity, a defined purpose, and an encrypted payload. No implicit trust exists anywhere in the system.

Ultimately, this architecture turns the PROTECT-CHILD platform into a living implementation of Zero Trust principles. Each namespace forms a secure enclave encapsulating one dimension of the platform's mission—data processing, computing, discovery, governance, or genomics—while Istio ensures that these enclaves can collaborate safely under a single federated service mesh. The result is an infrastructure that combines isolation with interoperability, security with transparency, and privacy with scientific usability, supporting the EHDS vision of trustworthy, cross-border data reuse for pediatric transplant research and care.

6.1.2 Logical view

The logical architecture of the PROTECT-CHILD platform is conceived as a federated digital ecosystem, designed to enable secure, privacy-preserving research on sensitive paediatric transplant data across Europe. It translates the principles of the European Health Data Space (EHDS) into an operational reality, combining local Secure Processing Environments (SPEs)—referred to as EHDS Capsules—with a central Orchestrator that coordinates computation, governance, and interoperability across participating sites.

At the core of this architecture lies a federated logic rather than a centralised one: data remain within the hospitals and research centres that generate them, while analysis tasks, algorithms, and queries move securely between nodes. Each hospital or data provider operates an EHDS Capsule, which encapsulates the necessary services for data storage, pseudonymisation, and analysis under strict access control. These capsules implement the "privacy-by-design" and "security-by-design" paradigms defined by the EHDS Article 73 and the TEHDAS2 specifications for SPEs. Within each capsule, sensitive clinical, genomic, and methylomic datasets are stored in OMOP and FHIR-compliant structures and processed only within the protected perimeter. No identifiable information leaves the environment; only aggregated or anonymised results can be exported after explicit approval by the Governance Layer.

Above this distributed network, the Orchestrator functions as the cognitive and management layer of the platform. It manages the overall workflow of federated analyses, ensuring that each computation, user access, and data exchange complies with ethical, legal, and technical safeguards. The Orchestrator authenticates users through federated identity systems (eIDAS or OpenID Connect), enforces data-access permits issued by the Governance layer, and coordinates the life cycle of each federated study—from environment provisioning to analysis execution and results aggregation. It acts as the mediator between data users and data holders, deploying analysis tasks across capsules while maintaining end-to-end encryption and full auditability.

Technically, this orchestration relies on a service-mesh-based microservice architecture, where RESTful APIs and secure communication protocols (mTLS, SPIFFE/SPIRE identities) connect the various components. Each interaction between services is logged and monitored, providing a transparent record of activity that satisfies GDPR accountability requirements and the EHDS demand for verifiable processing. Within this zero-trust environment, no component or actor is assumed to be inherently trustworthy; every access request, even internal, must be authenticated and authorised.

The workflow of the platform unfolds through a sequence of logical stages. Researchers authenticate and request access to data through the Orchestrator. Once a data-use permit is granted, the Orchestrator provisions dedicated SPE instances within each relevant EHDS Capsule. These isolated workspaces contain the analytic tools and datasets required for the

study. Analysis scripts or machine-learning models are then distributed to the capsules for local execution, following the Federated Computing Protocol developed within the project. Computations are performed close to the data—leveraging privacy-enhancing technologies such as differential privacy, homomorphic encryption, or secure multi-party computation—while only intermediate, encrypted or aggregated parameters are returned to the Orchestrator. The Orchestrator then performs global aggregation, validation, and, if authorised, export of the anonymised results to the researcher's workspace. Once the study concludes, each capsule can either archive its environment for reproducibility or securely decommission it, ensuring that no residual data persist beyond their authorised use.

In this logical configuration, the EHDS Capsule embodies the unit of trust and protection, while the Orchestrator embodies the unit of coordination and compliance. Together, they form a federated mesh that integrates the technical, organisational, and governance dimensions of data reuse. This design ensures that the PROTECT-CHILD platform operates as both a scientific and a regulatory instrument: enabling cross-border collaboration among hospitals and researchers, while fully respecting patients' privacy, data-protection laws, and the ethical constraints of paediatric research.

Ultimately, this logical view presents PROTECT-CHILD not as a single platform, but as an orchestrated federation of secure environments—a distributed European infrastructure where sensitive data remain local, computation travels safely, and knowledge circulates freely under the continuous supervision of the governance framework.

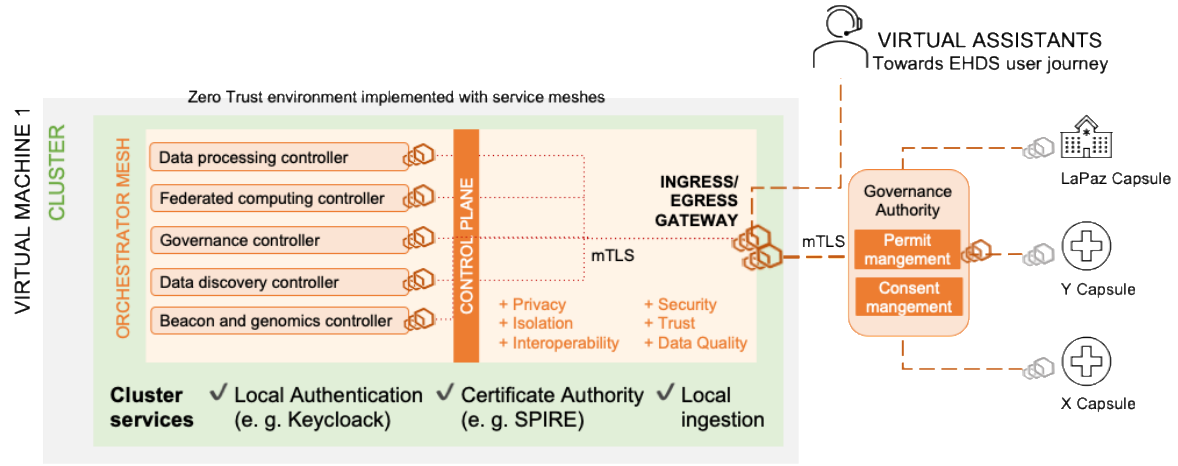

Figure 17: Orchestrator

The figure 18 illustrates the central orchestration layer of the PROTECT-CHILD federated architecture, where the Orchestrator functions as the coordination and compliance engine of the entire ecosystem. The diagram depicts the Orchestrator as the core control plane that interconnects the platform's functional controllers—data processing, federated computing, governance, data discovery, and genomic services—within a Zero-Trust service mesh implemented with secure communication protocols (mTLS) and workload identity management (SPIFFE/SPIRE). Each controller operates as a modular microservice responsible for its own domain, yet all are governed by the Orchestrator through a common security and interoperability layer. Externally, the Orchestrator interfaces with an independent Governance Authority, which resides outside the computing infrastructure and ensures that all data-use actions are authorised under valid ethical and legal conditions.

The Governance Layer issues data-access permits, verifies user identities via OpenID Connect, and encodes permissions into cryptographically signed JWTs, which are checked by both the

Orchestrator and the Capsules before any federated job is executed. This guarantees full accountability and separation of duties between decision-making (governance) and computation (orchestration).

Through the ingress/egress gateway, the Orchestrator securely communicates with multiple EHDS Capsules deployed in hospitals or data-provider sites (e.g., La Paz, Y, X capsules). It distributes analytic tasks, manages federated learning workflows, and aggregates results while ensuring that data never leave their original environment.

At the top of the figure, the Virtual Assistants layer appears as the user-facing entry point. These intelligent dashboards, powered by fine-tuned LLMs, guide researchers, clinicians, and data managers through each phase of the EHDS user journey—from data discovery and permit request to analysis and sharing—acting as cognitive companions that translate user intentions into compliant system actions.

Together, the components shown in the Orchestrator view embody the coordination and compliance fabric of PROTECT-CHILD: a secure, auditable, and interoperable environment that connects users, governance, and federated data services under the principles of privacy-by-design and Zero Trust.

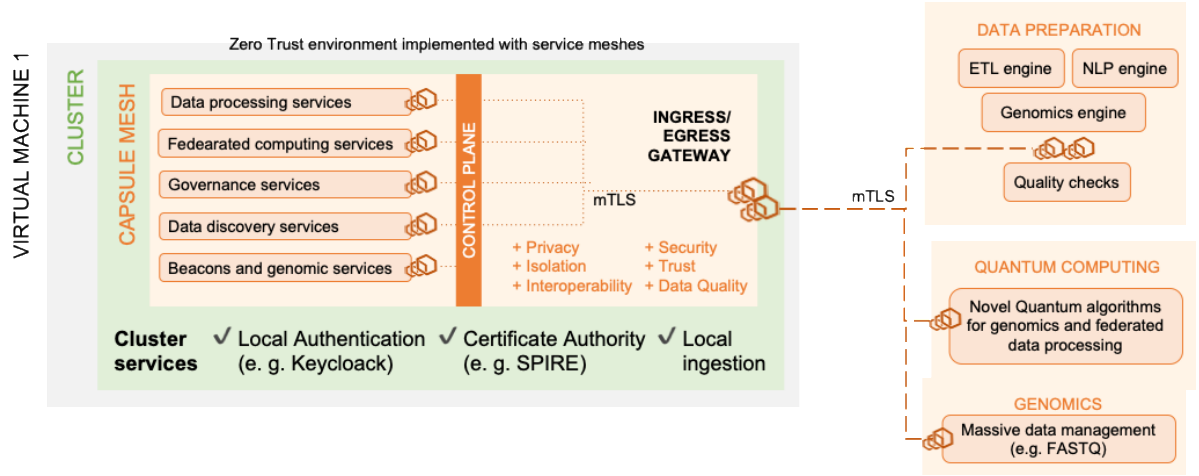

Figure 18: EHDS Capsule

The figure 19 zooms into the EHDS Capsule, which represents the unit of trust, protection, and computation within the PROTECT-CHILD federation. Each capsule is a Secure Processing Environment (SPE) deployed in a clinical or research institution and encapsulates all necessary services to store, process, and analyse sensitive paediatric transplant data in compliance with EHDS Article 73 and TEHDAS2 SPE requirements. Inside the capsule, distinct controllers handle data processing, governance enforcement, federated computing, discovery, and genomic analysis. These services operate under a shared Zero-Trust mesh, where mutual authentication (mTLS) and workload identities ensure that no internal process can communicate without verified credentials. The capsule's local authentication (e.g., Keycloak) and certificate authority (e.g., SPIRE) manage identity and encryption at the node level, while the local ingestion mechanisms handle secure data input from clinical sources. The control plane enforces policies for privacy, interoperability, and data quality, guaranteeing that any computation or query executed within the capsule complies with the platform's global governance rules.

The right side of the figure highlights two external but interoperable environments: the Data Preparation platform, where ETL, NLP, and quality-check engines transform raw inputs into harmonised OMOP and FHIR data structures before capsule activation; and the Quantum Computing platform, which extends analytic capacity through quantum algorithms for genomics

and federated data processing. Both interact with the capsule through encrypted connectors but remain logically isolated, ensuring that pre-processing and hybrid computing enhance, without compromising, the confidentiality of health and genomic data.

In operational terms, the capsule is where the "computation-to-data" principle becomes reality. Data never leave the local environment; instead, federated algorithms—dispatched from the Orchestrator—are executed inside the capsule, and only anonymised, aggregated, or encrypted outputs are returned. This design allows each institution to maintain full sovereignty over its data while contributing securely to cross-border analyses within the EHDS framework.

The EHDS Capsule view thus visualises the protective perimeter of the PROTECT-CHILD architecture: an interoperable, self-contained environment that combines local autonomy with federated participation, making the platform both technically resilient and ethically compliant.

Together with the Orchestrator view, it completes the logical vision of PROTECT-CHILD as a distributed, privacy-preserving European data ecosystem, where secure local environments and a federated control plane cooperate to transform sensitive paediatric transplant data into trustworthy, actionable knowledge.

6.2 Components

This section details the functional building blocks that make the PROTECT-CHILD architecture operational in an EHDS-compliant, Zero-Trust federation. Each component contributes a specific capability—data preparation, secure local processing, metadata-driven discovery, policy-enforced federated computing, independent governance, genomics/Beacon services, quantum-enhanced analytics, and user-facing virtual assistants—while remaining decoupled and interoperable through REST/OIDC connectors. Together, they instantiate the "computation-to-data" principle: datasets prepared inside each EHDS Capsule (OMOP/FHIR) never leave their protected perimeter; instead, authorised algorithms are orchestrated across sites, results are aggregated with full auditability, and user actions are mediated by role-aware assistants aligned with the EHDS user journey (Discover → Request → Prepare → Analyse → Share). The following subsections describe, in turn, how each component is instantiated, which partner leads its implementation, and how the components interlock to deliver a compliant, privacy-preserving and scalable platform for cross-border paediatric transplant research.

6.2.1 Data preparation phase

Before a capsule becomes operational within the PROTECT-CHILD federated environment, it must undergo a dedicated one-shot data preparation phase. This stage represents the foundational step in the lifecycle of each EHDS Capsule, during which all local datasets—clinical, genomic, and textual—are harmonised, curated, and securely ingested into OMOP and FHIR databases aligned with the Protect-Child data model specifications. The outcome of this phase is a fully configured, interoperable and compliant capsule, ready to participate in federated analytics and learning tasks under the coordination of the Orchestrator.

The data preparation phase establishes the scientific and technical baseline of the capsule, transforming heterogeneous raw inputs into structured, interoperable, and privacy-preserving resources that can be exploited for secondary use. This process occurs once, in the lifetime of the capsule, entirely within the secure perimeter of the capsule and follows the FAIR, GDPR, and EHDS principles of data minimisation, pseudonymisation, and traceable provenance.

Massive Data Management of Genomic Data

The management of genomic data in PROTECT-CHILD requires a robust, scalable, and compliant infrastructure capable of handling very large sequencing datasets. Each participating centre contributes approximately 50 raw FASTQ files, each between 200 and 300 GB, resulting in 10–15 TB of data per site. To accommodate this scale, the project deploys a cloud-native storage and compute infrastructure integrated into the Secure Processing Environments (SPEs) of the EHDS Capsules. Data are stored in object-based repositories (e.g., S3-compatible or Azure Blob storage), which provide high durability, parallel access, and automatic redundancy while enabling direct interoperability with genomic processing pipelines.

All data are encrypted at rest and in transit, with key management handled by provider-integrated KMS systems to ensure compliance with GDPR and EHDS requirements for secure processing environments. Each dataset is catalogued using the DCAT-AP Health metadata schema, enabling traceability, provenance tracking, and discoverability within the federated ecosystem. The infrastructure supports elastic scaling and lifecycle management, allowing cold storage for archived data and high-performance tiers for active analysis.

Through this design, PROTECT-CHILD achieves efficient, privacy-preserving management of massive genomic datasets across multiple centres, ensuring that high-throughput sequencing data remain FAIR, secure, and ready for federated computation within the EHDS framework.

Genomic Data Preparation

The preparation of genomic data is one of the most computation-intensive and methodologically sensitive steps. It begins with the ingestion of raw sequencing outputs, typically stored in VCF (Variant Call Format) CRAM (Compressed Reference-oriented Alignment Map) and BAM (Binary Alignment Map) files. These files represent the primary evidence of the patient's genomic profile and must be processed into analytically meaningful subsets. Within the capsule, dedicated pipelines perform variant analysis to identify and classify genomic alterations relevant to the study. Automated pre-processing and filtering algorithms are applied to exclude technical artefacts, low-quality calls, and redundant entries. The workflow then performs targeted searches and annotation using established bioinformatic libraries and reference databases to extract and categorise pathogenic, likely pathogenic, and variants of uncertain significance (VUS). Only these curated variants, linked to their genomic context and metadata, are retained and stored in the capsule's internal repository for subsequent analysis. This ensures that downstream federated computations can operate on clinically interpretable and privacy-minimised datasets rather than on raw genomic material. In parallel, the capsule prepares its Beacon v2 interface—a core component for alignment with the ELIXIR standards, the Genome Data Infrastructure (GDI) and the GA4GH Beacon network. Through this process, selected variants derived from VCF files are indexed and exposed in a queryable format that supports federated discovery. This allows authorised external users, via the Orchestrator, to issue privacy-preserving queries such as "Does this variant exist in any capsule?" without revealing patient-level information. This federated querying capability is essential to enable cross-border interoperability within the EHDS ecosystem and supports standardised genomics data discovery across European infrastructures.

Methylomic Data Preparation

The capsule also ingests and processes methylomic data, originating from IDAT files generated by array-based methylation assays. These datasets capture epigenetic signatures, which are particularly relevant to understanding immune response, graft tolerance, and long-term transplant outcomes. During preparation, IDAT files are validated for completeness and format consistency, then securely stored within the capsule in a structured form optimised for

distributed Principal Component Analysis (PCA) and other federated epigenomic analyses. This preparation ensures that methylomic features can later be compared across capsules without any direct exchange of raw data, enabling population-level statistical inference while maintaining data locality and confidentiality.

Clinical Data Preparation

In addition to omics data, the capsule incorporates clinical data extracted from PETER registry, electronic case report forms (CRFs), and hospital information systems. These data are harmonised to a common model—typically OMOP CDM or HL7 FHIR—to guarantee interoperability with other capsules and with the federated Orchestrator. Data preparation involves validation of identifiers, mapping of clinical variables to standard terminologies (SNOMED CT, LOINC, ICD-10), and pseudonymisation of patient information. The resulting dataset constitutes the structured clinical layer of the capsule, enabling statistical and machine-learning models to correlate clinical phenotypes with genomic and methylomic patterns in a privacy-preserving manner.

Free-Text Data Extraction

Finally, the capsule performs Natural Language Processing (NLP)-based extraction from free-text clinical documents, such as discharge summaries, pathology reports, or clinician notes. This step is crucial for capturing implicit or narrative-style information that is not represented in structured fields. Advanced NLP pipelines running within the capsule identify entities such as diagnoses, treatments, complications, or time-series events and map them to controlled vocabularies. The extracted entities are then linked to the structured clinical dataset, enriching it without exposing raw textual content outside the secure environment.

Once the genomic, methylomic, clinical, and textual components have been processed, harmonised, and validated, the capsule reaches an "operationally ready" state. At this point, it contains a coherent, semantically aligned dataset that can participate in federated analyses coordinated by the Orchestrator. The preparation phase is therefore executed once in the capsule's lifecycle, but its results persist throughout subsequent research tasks, guaranteeing data integrity, analytical consistency, and full compliance with the legal and ethical frameworks governing the secondary use of paediatric health and genomic data.

6.2.2 Data Processing Controller and services

Once a capsule has completed its data preparation phase and reached operational readiness, it becomes an active and autonomous node within the PROTECT-CHILD federated ecosystem. From this point, all data stored inside the capsule are represented in standardised and interoperable formats—namely OMOP CDM for structured clinical and observational data, and FHIR resources for patient-level and transactional information exchange. This dual data model ensures seamless interoperability between the internal components of the capsule, the central Data Processing Orchestrator, and external EHDS-compliant services such as Federated Computing, Data Discovery, and Genomics Query Services.

Within the capsule, a set of local services supports this interaction between data and computation:

- Data Access engine: Interfaces directly with the OMOP and FHIR databases, enforcing access control policies and maintaining audit logs of all queries.
- Federated Execution Engine: Runs containerised analytic jobs dispatched by the Orchestrator. It ensures that only authorised federated algorithms can access local data, following the "computation-to-data" principle.

- Data Governance and Privacy Services: Apply data-minimisation, pseudonymisation, and output-checking rules, guaranteeing that only approved results can leave the environment.
- Connector Services: Provide standardised REST and gRPC endpoints compatible with federated frameworks (e.g., Vantage6, Flower) and interoperability standards (FHIR APIs, GA4GH Beacon v2).

6.2.3 Data Discovery Controller and services

Once the data have been curated and harmonised within each capsule, a new layer of functionality becomes active to enable data visibility, discoverability, and quality assessment all under the constraints of the European Health Data Space (EHDS) regulatory framework. This phase is driven by the local discovery services deployed within each capsule. The purpose of the Data Discovery layer is not to expose the underlying data, but to surface structured, privacy-preserving metadata that help authorised users understand what types of data exist, in what quantities, under what quality conditions, and for which research purposes they may be reused.

EHDS-Compliant Metadata Model (DCAT-AP Health Profile)

The metadata exposed by the Data Discovery services are structured following the guidelines and best practices established for the DCAT-AP Health profile, applying its principles wherever feasible. This approach ensures that key descriptors such as provenance, schema, temporal coverage, licensing conditions, custodianship, and quality indicators are represented in a machine-readable and interoperable way, without imposing the full mandatory scope of the formal profile.

Each capsule therefore, maintains a local metadata catalogue that draws on the DCAT-AP Health recommendations for organising dataset and variable-level information across the clinical, genomic, and methylomic domains stored in its OMOP and FHIR repositories. The catalogue also captures processing provenance, quality-control outcomes, and compliance-relevant information. Local catalogues synchronise periodically with the federated metadata registry managed by the Data Discovery Orchestrator, enabling a unified and EHDS-aligned discovery interface across the federation. By following DCAT-AP Health-inspired structures, PROTECT-CHILD ensures that its metadata can be harvested and indexed by the broader European data-infrastructure ecosystem including ELIXIR, GDI, and the forthcoming HealthData@EU platform without the need for extensive downstream transformations. This approach stays consistent with Article 53 and Annex II of the EHDS Regulation, which encourage the use of common cataloguing formats and shared metadata vocabularies for secondary data use.

Data Quality Services

In addition to cataloguing, the Data Discovery layer delivers data-quality services that compute both quantitative and qualitative indicators for each dataset. These cover completeness, consistency, validity, and timeliness, derived from automated profiling tools running on OMOP and FHIR sources. The resulting quality reports are embedded in the metadata, giving users a clear understanding of the reliability and analytical fitness of the data before submitting access requests. This reinforces the EHDS requirement for transparency on data quality, supports cross-border trust in paediatric transplant research, and provides capsule administrators with actionable insights to improve their pipelines continuously.

Data Query Services on OMOP

To complement discovery and quality assessment, each capsule exposes OMOP-compliant query services. These operate exclusively within the Secure Processing Environment and support only controlled, pre-authorised exploratory queries such as case counts, distribution

summaries, or feasibility checks without revealing individual-level information. Execution follows a federated-summarisation model: queries are defined by the Orchestrator, run locally within each capsule, and only aggregate statistics are returned. This allows researchers to evaluate whether suitable populations or data elements exist for their studies, while respecting strict privacy and confidentiality rules. The Orchestrator logs and audits every execution to ensure alignment with user permissions and governance requirements.

The interaction between the Data Discovery Orchestrator and the Capsule Discovery Services forms a complete and EHDS-aligned discovery ecosystem that delivers:

- Transparency, by exposing structured, interoperable metadata catalogues inspired by DCAT-AP Health guidelines;
- Trust, through integrated quality metrics and detailed provenance records;
- Efficiency, via federated OMOP query capabilities that support feasibility assessments;
- Compliance, by enforcing data-minimisation and purpose-limitation principles across all discovery activities.

Through this federated discovery framework, PROTECT-CHILD not only strengthens internal scientific collaboration but also contributes to the broader EHDS vision of a discoverable, interoperable, and trustworthy European health-data infrastructure. The Data Discovery Orchestrator becomes the central gateway for transparency enabling users to understand the data landscape before accessing it, and ensuring that every future data-use request is informed, justified, and compliant.

6.2.4 Federated Computing Controller and services

The Federated computing Orchestrator (implemented in Vantage6) acts as the coordination and control layer of the entire federated network. It does not host any data itself; instead, it governs and supervises where, when, and how computations are executed within each local capsule. Its primary function is to orchestrate distributed analytics workflows in a privacy-preserving and policy-enforced manner. Every operation initiated within the ecosystem—whether a federated learning task, a statistical aggregation, or a variant query—is first registered, authorised, and dispatched through the Orchestrator, which ensures that all interactions comply with the permissions defined by the Governance layer and the underlying Zero-Trust security model.

When a new study or analytical job is initiated, the Orchestrator dynamically identifies the set of capsules authorised to participate and distributes the necessary computation tasks through secure APIs and mTLS-encrypted channels. Each capsule executes the assigned operation locally, within its Secure Processing Environment, using its own processing resources. The algorithms are deployed as federated workloads—for example, federated learning models, federated queries, or federated statistical computations—and interact only with the capsule's internal OMOP and FHIR data stores. Under no circumstances do these algorithms export or replicate raw data outside the capsule. Instead, they operate under strict isolation and produce intermediate results (such as model parameters, summary statistics, or anonymised aggregates) that are sent back to the Orchestrator for aggregation and validation.

The interaction between the Federated Computing Orchestrator and the Capsule Services forms a closed, traceable loop:

1. The Federated Computing Orchestrator registers a computation request, verifies authorisations, and distributes the algorithm package.
2. The Capsule executes the task locally, accessing OMOP/FHIR data through internal APIs only.

3. The Capsule encrypts and returns results to the Federated Computing Orchestrator, which performs aggregation, validation, and compliance checks before any further dissemination.

This operational model enables the coexistence of multiple analytical services—federated computing, data discovery, and genomics querying—within a common governance and security framework. Researchers can thus perform cross-border analyses and training of models without moving data from their original locations. The Orchestrator guarantees full traceability, auditability, and compliance, while the capsules ensure data sovereignty and confidentiality.

By combining federated orchestration with locally controlled capsule services, the PROTECT-CHILD platform achieves a harmonised and privacy-preserving computing fabric where sensitive paediatric transplant data are safely transformed into interoperable knowledge. Data stay protected and stationary within their institutional boundaries, yet they contribute dynamically to shared scientific discovery across the EHDS ecosystem.

6.2.5 Governance Controller and services (WP6 CERTH and BELIT)

In the PROTECT-CHILD platform, governance is conceived as a fully independent and decoupled layer, positioned outside both the Orchestrator and the Capsules. It functions as the trusted authority that defines, verifies, and enforces the rules governing how users, data, and computational processes interact within the federated ecosystem. This separation is deliberate: by isolating governance from the operational domains where data are processed, the platform guarantees that no single component holds unilateral control over identity, access, or consent—thus ensuring a strong alignment with the principles of accountability, transparency, and zero trust promoted by the European Health Data Space (EHDS). Within PROTECT-CHILD, this Governance Controller is implemented in WP6 as a horizontal service that underpins all data-access, ethics, consent, and permit workflows across the platform.

The Governance Layer does not store or process health data; rather, it manages the decentralized identities and verifiable permissions that make data use possible. Users are identified via W3C Decentralized Identifiers (DIDs), following the EBSI DID method, which are cryptographically controlled by users through digital wallets and registered in the EBSI DID Registry. All user authentication within the platform is handled through a federated OpenID Connect (OIDC) provider, which enables secure single sign-on and interoperability with trusted institutional or national identity systems, such as eIDAS or ELIXIR AAI. When a user logs in, their identity is verified by by resolving their DID document from the EBSI DID Registry, and the Governance Layer (acting as an EBSI Trusted Issuer) issues a Verifiable Credential (VC) encoding the user's role, scope, and authorised actions. and the Governance Layer (acting as an EBSI Trusted Issuer) issues a Verifiable Credential (VC) encoding the user's role, scope, and authorised actions. This VC is stored in the user's wallet and can be presented to the Orchestrator and any Capsule as a Verifiable Presentation (VP), proving who the user is and what they are permitted to do, without revealing unnecessary personal information. Both the Orchestrator and the Capsules verify the DID signature and VC against the EBSI DID Registry and Trusted Issuer lists locally, ensuring that every action within the system is authenticated and traceable. Ethics committees, reviewers, and organizational signatories are likewise onboarded as trusted identities with their own EBSI DIDs and role-defining Verifiable Credentials issued via the CERTH wallet, so that ethical and legal decisions are taken only by verifiably mandated actors.

Permissions within the PROTECT-CHILD platform are defined using a role-based access control (RBAC) model. Roles are encoded within Verifiable Credentials signed by the Governance Layer and bound to the user's DID. When a user needs to act within the platform, they present a

Verifiable Presentation containing the relevant VC(s) from their wallet, the recipient service cryptographically verifies the issuer DID, the subject DID, and the credential signature before granting access. Because these credentials are time-limited and revocable via EBSI's status mechanisms, they cannot be altered or reused beyond their intended context, which reinforces the security of the platform. If needed. for backward compatibility and operational efficiency, the platform may also issue short-lived JSON Web Tokens (JWTs) encoding the same claims, derived from validated VCs, to reduce repeated wallet interactions during active sessions. In this way, the Governance Layer becomes the single source of truth for who can access the system, what they can do, and for how long.

Beyond identity and access control, the Governance Layer also manages the ethical and legal dimension of data use. It maintains and enforces the records of data consent associated with each dataset. These consents, typically granted by patients or their guardians, define the specific purposes for which data can be used, the categories of data that may be accessed, and the conditions under which analyses can take place. When a researcher wishes to conduct a study, the Governance Layer evaluates whether the proposed use complies with the existing consents and ethical approvals. Ethical review outcomes are captured through a digital ethics-agreement flow in which committees' issue electronically signed decisions that are linked to the corresponding data-access request. If so, it issues a data permit—a digitally signed authorization that grants the researcher the right to execute a federated analysis within the platform. Each data permit follows the EHDS Article 34 specification and consolidates permit application details, ethics approvals, consent status, and controllership/processing agreements into a single machine-readable authorization artifact. This permit becomes a prerequisite for the Orchestrator to initiate any computational task: no federated job can be launched without a valid, verified permit attached to it. The lifecycle of this permit (active, expired, revoked) and a tamper-evident hash of the underlying signed decision are anchored on a consortium ledger via smart-contract-based registry and permit-token services, enabling cross-border mutual recognition of approvals among TransplantChild centres.

Because governance operates independently from the Orchestrator and Capsules, it ensures that decision-making about access and consent remains impartial and auditable. It serves as a policy enforcement point that connects the human, legal, and technical dimensions of the platform. Through this design, PROTECT-CHILD achieves a clear separation between the act of deciding to use data and the act of using data, creating a transparent and secure boundary between authorisation and computation. Every identity check, token issuance, and permit validation is logged, providing a continuous audit trail that satisfies EHDS and GDPR requirements for lawful data processing, accountability, and reproducibility. In combination with the consortium ledger events and immutable logs exposed by the Governance services, this provides a verifiable, end-to-end audit trail over all permit and ethics decisions.

In essence, the Governance Layer is the trust fabric of the PROTECT-CHILD architecture. It does not compute or analyse—it enables and controls. It ensures that only verified individuals, acting within authorised roles and under valid consent conditions, can trigger analyses within the federated infrastructure. A user-centric Data Permit Dashboard, developed in WP7, surfaces these governance capabilities to researchers, clinicians, and patients, allowing them to inspect permit status, manage sharing preferences, and exercise access and revocation rights through a single EHDS-aligned interface. By externalising and standardising these functions, the platform achieves a governance model that is both technically rigorous and ethically robust: one that mirrors the principles of the European Health Data Space, where data sovereignty, lawful reuse, and cross-border trust coexist within a single federated framework.

6.2.6 Consent management platform

This task aims to streamline and optimize granular digital patient consent management—including parental or legal consents and their delegation—by leveraging a dynamic standards-compliant digital agreement mechanism, optionally additionally relying on blockchain-audited smart contracts. The system will support all essential consent lifecycle operations (declaration, recording, modification, delegation, withdrawal, and provenance) through standardized APIs and user-friendly dashboards. It can also enable and monitor regulated access to external genomic and research databases, supporting scenarios that require explicit user permission, and maintaining complete traceability via a robust audit trail.

Scope and Objectives

The task involves the specification and prospective deployment of a consent management mechanism to support joint data governance across participating clinical centres in the TransplantChild ERN. This mechanism will provide data holders with the tools to collect, manage, and audit informed consent from patients, clinical study participants, or their legal custodians/representatives, enabling granular control aligned with diverse clinical, research, and cross-border data use scenarios.

A general objective is also to enhance convergence and interoperability between prevailing data standards—most notably HL7 FHIR (Fast Healthcare Interoperability Resources) and referent implementations based on it — promoting the vision of integration hubs embodied by EHDS capsules. The task will ensure that consent directives reliably control the access and flow of sensitive health and genomic data, with all relevant events being tracked and auditable via FHIR resources and, where appropriate, blockchain-based smart contracts.

Technical Components and Architecture

The task leverages the HL7 FHIR Consent and AuditEvent resources (R4/R5), which provide generic, extensible structures for representing consent rules and maintaining detailed event logs. The architectural approach is centred on governed granular consent provisions**,** captured, validated, and managed through dedicated interfaces and APIs.

Key architectural and functional aspects include:

- **Consent Resource Modelling:**
  - Supporting versatile, scenario-based consent representations, organized around consent provisions with clearly specified exceptions (by object, purpose, actors, organization, date range, etc.).
  - Supporting multi-party consent and linked records, with traceable changes tracked over time (full audit trail).
  - Enabling the attachment and verification of source documents, including digitally signed consent and revocation forms.
- **Minimal Dataset and Metadata Integration:**
  - Ensuring conformant metadata for each consent record (unique identifiers, patient references, status, category, timestamp, performer, source, policy, and provision details).
  - Mapping consent policies to the metadata structure governing each EHDS capsule, and establishing potential links with capsule access control and audit mechanisms.

- **Standards, Profiles, and API Operations:**
  - Adopting scalable consent management profiles and API endpoints as specified by the FHIR at Scale Taskforce (FAST) SCM, and evolved from the IHE Privacy Consent on FHIR (PCF) technical frameworks.
  - Supporting a set of standardized consent lifecycle operations, including provider- and patient-initiated consent, review, digital signing, delegation (e.g., transition of authority at legal age), revocation with event propagation, and disclosure/audit event propagation (including to blockchain/DLT subsystems as needed).
- **Interoperability and Extensibility:**
  - Ensuring compatibility and possible integration with initiatives and projects driving advanced consent management concepts (e.g., Transcelerate Biopharma, HL7 Vulcan, IMI FACILITATE), particularly for scenarios such as secondary data use.
  - Preparing the architecture and implementation for anticipated updates associated with future HL7 FHIR releases (e.g., Release 6) to ensure long-term maintainability, stability, and formal adoption across European and global contexts.

Alignment with Platform Architecture

Consent management is embedded in the overall data governance layer of the Protect-Child platform architecture, orchestrating access and secondary use permissions within and across EHDS capsules. Smart contract technology (blockchain) may be deployed to reinforce trust, immutability, and auditability, thus underpinning security and transparency in data operations at scale.

This task will produce:

- Detailed data and metadata models for consent management, mapped to FHIR resources and compatible with existing EHR and research data models.
- A functional consent management architecture, including interfaces and operational flows, for integration into the Protect-Child platform and subsequent clinical centre deployments.

Guidance for aligning consent governance with the platform's broader interoperability, compliance, and ethical framework.

6.2.7 Beacons and Genomics Controller and services

The Beacons and Genomics Controller and Services subsystem functions as the molecular intelligence layer of the PROTECT-CHILD platform. It connects the genomic, epigenomic, and clinical worlds through interoperable OMOP-based representations, empowers researchers to perform federated discovery and analysis without compromising privacy, and bridges the PROTECT-CHILD ecosystem with European infrastructures such as GDI, GA4GH, and EHDS.

By combining Beacon v2 discovery, multi-omics integration, and Vantage6-enabled federated analytics, this subsystem establishes a secure, standardised, and EHDS-compliant pathway for transforming complex genomic data into actionable clinical insight across a trusted European network.

Within the PROTECT-CHILD platform, the Beacons and Genomics Controller and Services form a specialised subsystem dedicated to the management, discovery, and federated analysis of genomic and methylomic data. This component serves as the interface between the federated

computing infrastructure and the molecular layer of the platform, ensuring that all genomic-related resources—from variant calls to methylation profiles—can be securely queried, interpreted, and analysed. The subsystem integrates seamlessly with the platform's broader federated ecosystem and operates in concert with the Data Processing Orchestrator and Capsule Services, leveraging the Vantage6 federated computing infrastructure adopted in PROTECT-CHILD.

Through this integration, it enables distributed genomic computation and variant discovery while preserving data sovereignty: genomic and epigenomic data remain within their originating capsules, and only aggregated or derived insights are exchanged.

At its core, the subsystem encompasses two tightly interconnected functional domains: the Beacon v2 services, which enable secure variant discovery and interoperability with the GDI and GA4GH networks; and the Genomics Controller, which orchestrates the processing, harmonisation, and federated analysis of multi-omics data, including genome, methylome, and phenotype correlations.

Beacon v2 Integration and GDI Alignment

The ELIXIR Beacon component implements the GA4GH Beacon v2 specification, allowing each capsule to expose a minimal, standardised query interface that answers specific variant-presence queries without revealing individual-level data. For example, a researcher may ask whether a particular single-nucleotide variant or genomic region exists among the samples stored in a capsule. The response is returned in a binary or aggregate form ("yes," "no," or "count"), maintaining strict privacy while enabling federated discovery across multiple nodes.

In PROTECT-CHILD, these Beacon services are aligned with and registered within the Genome Data Infrastructure (GDI) catalogue, ensuring compatibility with European-wide genomic data discovery mechanisms. Each capsule's Beacon endpoint is registered with the central Genomics Controller, which acts as a federated proxy, enabling coordinated cross-capsule variant queries through the Orchestrator. The design thus extends the EHDS "discover before access" principle to the genomic domain—researchers can verify the existence of relevant variants before formally requesting access to perform deeper analyses.

Integrated phenotype and variant analyses

Clinical phenotypes captured in the capsule's OMOP structure (e.g., conditions, measurements, drug exposures) are linked to annotated variants within the same model using standard vocabularies (SNOMED CT, HPO, OMIM). This integration allows the Genomics Controller to enable federated queries and analyses that connect genomic alterations with clinical outcomes—such as genotype-phenotype correlations, biomarker validation, or predictive model training. All computations are performed locally within each capsule's secure processing environment; only aggregated model parameters or statistical summaries are transmitted to the Orchestrator for aggregation, ensuring full compliance with data-protection requirements.

Methylome and Phenotype Integration

Beyond variant discovery, the subsystem supports methylome analysis by integrating methylation data (IDAT files) into the OMOP Common Data Model (CDM). This mapping allows epigenetic information—such as CpG methylation patterns—to be represented as measurements or observations within OMOP, preserving semantic interoperability with other omics and clinical data domains. Through this model, federated statistical analyses such as Principal Component Analysis (PCA) or epigenetic-phenotypic correlations can be executed across capsules via Vantage6, enabling large-scale population-level inference without centralising sensitive epigenomic data.

Phenopacket schema

GA4GH Phenopacket offers a standardized, interoperable framework for describing biomedical patient data—including genetic variants, phenotypes, biosamples, and therapies—enabling phenotype-linked discovery in federated platforms like Beacon v2. Within initiatives such as Protect-Child, Phenopackets can be directly aligned with OMOP-based clinical data and enriched by FHIR Genomics resources, supporting integrated queries that securely associate genotypic and phenotypic profiles across institutional boundaries. This approach fosters collaborative precision medicine and pediatric data protection while ensuring compatibility with global standards for clinical and genomic interoperability. Project data will be able to be exported according to this schema, facilitating interoperability and reuse across platforms.

Integration with the Vantage6 Federated Computing Framework

The Vantage6 framework provides the execution layer for distributed analytics within PROTECT-CHILD, and the Genomics subsystem is fully integrated with this infrastructure. Each capsule hosts a Vantage6 node capable of running containerised genomic analysis tasks defined by the Genomics Controller. Typical tasks include variant frequency calculation, association testing, model parameter updates for federated learning, or methylation pattern clustering. The Genomics Controller communicates with these nodes through the Orchestrator, dispatching federated workloads, coordinating their execution, and aggregating partial results. Because Vantage6 enforces computation-to-data principles, no genomic, methylomic, or phenotypic data ever leave their respective capsules—only anonymised, aggregate metrics are shared for global model computation.

Through this integration, PROTECT-CHILD enables end-to-end federated genomic analytics: from Beacon-based variant discovery, through federated methylome and phenotype integration, to secure model aggregation and validation. The subsystem ensures that all processes adhere to the principles of privacy preservation, interoperability, and transparency mandated by the EHDS.

6.2.8 Quantum computing components

The Quantum Computing Component of the PROTECT-CHILD ecosystem is conceived as an independent, external platform that complements the classical federated computing layer by introducing hybrid quantum–classical data analysis capabilities. While the core computational tasks of the platform—federated learning, distributed statistics, and privacy-preserving analytics—are executed on classical infrastructures via the Vantage6 framework, certain categories of problems, particularly in genomics, multi-omics correlation, and high-dimensional biomarker discovery, can benefit from the superior pattern recognition and optimization capabilities offered by emerging quantum machine learning (QML) methods.

For this reason, the Quantum Computing Component has been designed as a loosely coupled, service-oriented subsystem that integrates with Vantage6 through secure APIs and containerised quantum connectors. It remains logically and operationally independent from the Orchestrator and Capsules, maintaining its own execution environment, cryptographic identity, and governance policies, yet fully aligned with the platform's Zero Trust and EHDS-compliant principles. The architecture allows the Orchestrator to route selected computational workloads—identified as quantum-eligible—to the Quantum platform, while continuing to handle data access, identity validation, and task orchestration through standard federated channels.

In practice, the Quantum Computing Component provides a hybrid execution environment, where classical and quantum processors collaborate in a single analytic workflow. Classical nodes (within the capsules) perform the data pre-processing and feature encoding locally, using

pseudonymised and harmonised data in OMOP and FHIR format. These local nodes extract numerical or symbolic representations of the data (for instance, genomic variant matrices, methylation beta values, or phenotype embeddings) and encode them into quantum-ready feature vectors. Only these compressed, non-identifiable quantum feature states are transmitted to the Quantum platform through an encrypted channel, ensuring that no raw clinical or genomic data leave the capsule. Once received, the Quantum platform executes quantum subroutines—such as quantum kernel estimation, quantum support vector machines, or quantum variational circuits—to explore complex relationships that are computationally intensive for classical systems. The results of these quantum computations (for example, covariance matrices, optimized parameters, or reduced latent features) are then returned to the Vantage6 Orchestrator for integration with the classical federated workflow.

This enables hybrid pipelines, where quantum outputs can feed back into classical models for validation, aggregation, or further training within each capsule, achieving a seamless synergy between traditional and quantum computation.

From a governance and compliance perspective, the Quantum platform adheres to the same privacy-preserving principles as the rest of the PROTECT-CHILD infrastructure. All transmitted inputs are fully anonymised and transformed; the platform is stateless with respect to sensitive data and does not retain intermediate computations. Every quantum job is registered, timestamped, and cryptographically signed by the Orchestrator, ensuring full traceability and auditability. This guarantees that quantum resources can be safely incorporated into the EHDS environment without compromising the integrity of the Secure Processing Environments (SPEs).

Functionally, the Quantum Computing Component serves three complementary purposes within PROTECT-CHILD:

1. Federated Genomic Analysis Acceleration: leveraging quantum algorithms for tasks such as variant clustering, multi-gene association analysis, and genotype-phenotype inference.
2. Methylome and Multi-Omics Correlation: enabling quantum-enhanced principal component analysis (qPCA) and hybrid feature selection for integrating methylation and genomic data stored in OMOP.
3. Optimization in Federated Learning: improving convergence of distributed learning models within Vantage6 by employing quantum-inspired optimizers (e.g. quantum annealing or variational hybrid algorithms).

By extending the Vantage6 federated infrastructure into the quantum domain, PROTECT-CHILD pioneers a new paradigm of privacy-preserving quantum federation, where each capsule contributes locally pre-processed features while the Quantum platform contributes computational depth. This design aligns with the project's overarching goals of innovation, compliance, and scalability, offering a forward-looking research avenue that bridges traditional federated computing with next-generation quantum resources.

Ultimately, the Quantum Computing Component transforms the PROTECT-CHILD ecosystem into a hybrid intelligence environment, capable of exploring complex genomic and epigenomic patterns that classical computation alone cannot efficiently resolve. It reinforces the project's ambition to operate at the frontier of secure biomedical data analysis, combining European-trusted federated infrastructures with quantum-enabled scientific discovery—a model that anticipates the evolution of future EHDS infrastructures toward quantum-classical interoperability.

6.2.9 Virtual Assistant components

The Virtual Assistants in the PROTECT-CHILD platform constitute an intelligent, human-centred layer designed to bridge users with the underlying complexity of the federated computing, governance, and data-discovery systems. They operate as a suite of interactive dashboard interfaces enhanced with specialized Large Language Models (LLMs) that have been aligned with the user requirements identified in Deliverable D2.1. Through natural-language interaction, contextual reasoning, and visual dashboards, these assistants guide users along every stage of the EHDS user journey — from data discovery to analysis execution and results interpretation — ensuring accessibility, compliance, and usability for all categories of stakeholders.

These assistants do not replace the core system functions but act as cognitive companions that mediate between human intentions and the platform's secure, federated operations. They interpret user goals, translate them into structured API calls to the orchestrator and capsule services, retrieve compliant responses, and present the outcomes in an intuitive and explainable way. The assistants thus transform the EHDS user experience from a technically complex process into a guided, conversational workflow grounded in trust, transparency, and regulatory conformity.

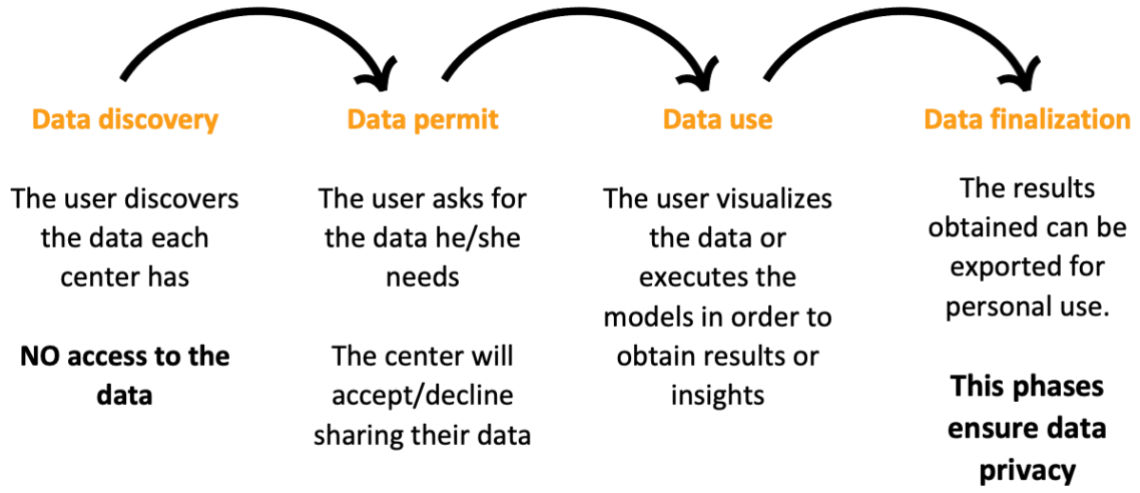

Figure 19: EHDS User Journey

Functionally, the Virtual Assistant layer integrates three core capabilities:

1. Dashboard UI Integration – Each assistant is embedded within an interactive dashboard that aggregates relevant services and visual elements for its user type. Researchers can explore metadata catalogues through the Data Discovery Assistant; clinicians can visualise cohort summaries and model outputs through the Clinical Insight Assistant; governance officers can audit access logs and consent states through the Governance Assistant. These dashboards unify inputs from multiple backend services — OMOP analytics, Beacon v2 queries, federated job status, and quality metrics — and present them through adaptive, role-specific interfaces.
2. Specialized Large Language Models – At the cognitive core of each assistant is an LLM fine-tuned on the Protect-Child domain ontology, EHDS guidelines, federated analytics documentation, and data governance policies defined in D2.1. This ensures the assistant understands biomedical vocabulary, federated computing principles, and the legal context of EHDS data sharing. The LLMs enable contextual dialogue, summarisation of metadata, automated report drafting, and question-answering about the state and

content of the capsules, all while respecting the platform's zero-trust and privacy-by-design architecture.

3. EHDS User-Journey Alignment – The assistants are structured around the main phases of the EHDS user journey as defined by TEHDAS2: Discover → Request → Prepare → Analyse → Share.
    a. During Discover, the assistant helps users navigate the DCAT-AP Health metadata catalogues, evaluate dataset quality indicators, and estimate cohort availability.
    b. In the Request phase, it guides users through the ethical, legal, and technical prerequisites to obtain a data permit via the Governance Layer.
    c. In Prepare, it explains how to configure analysis workflows and choose federated algorithms.
    d. During Analyse, it monitors job progress, interprets intermediate results, and explains federated aggregation outputs.
    e. Finally, in Share, it assists with the generation of compliant reports and metadata for result dissemination.

In every phase, the assistant's guidance is not merely instructional, it dynamically interacts with the platform to execute real actions, all within the boundaries of user permissions encoded in the JWT and RBAC framework.

Technically, the Virtual Assistants are deployed as microservices integrated with the Orchestrator's API gateway. They communicate securely with the Governance Layer for identity verification, with the Data Discovery Orchestrator for metadata queries, and with the Federated Computing services (Vantage6) for workflow monitoring. All interactions occur through authenticated OIDC sessions, ensuring traceability and auditability in accordance with EHDS and GDPR requirements. Sensitive information never leaves the secure processing environments; the assistants operate only on metadata, summary results, or pseudonymised analytical outputs.

By combining advanced natural-language interaction with strict regulatory compliance, the Virtual Assistants transform the PROTECT-CHILD platform into a human-centric, explainable, and inclusive environment. They reduce the cognitive and technical barriers typically associated with federated data analysis, enabling clinicians and researchers to engage with the platform intuitively while maintaining complete control over data access and governance.

Ultimately, these assistants embody the "EHDS by design" philosophy: empowering users to interact naturally with a highly complex, multi-layered infrastructure while ensuring every action, query, and computation remains secure, auditable, and aligned with European legal and ethical frameworks.

6.3 Connectors

This section describes the mechanisms that enable secure, standardized, and interoperable communication between its distributed components. In a system designed to operate under strict data protection and Zero Trust principles, connectors play a fundamental role: they form the logical interfaces through which services, users, and external systems exchange information without ever compromising isolation, compliance, or integrity. Rather than relying on direct database access or internal coupling, every interaction within PROTECT-CHILD takes place through well-defined communication layers that enforce authentication, authorization, and encryption by design. These connectors transform the platform into a federated but cohesive ecosystem, where each module—whether responsible for data ingestion, analytics, governance,

or interoperability—can interact safely with the others while remaining independently managed and evolvable.

Within this architectural vision, two key connector technologies serve as the backbone of interoperability and trust enforcement: **REST APIs**, which provide a standardized and decoupled communication model for all internal and cross-capsule operations, and **OpenID Connect (OIDC)** with **JSON Web Tokens (JWT)**, which establish the identity and access context behind every transaction. Together, they ensure that every call across the platform is both semantically interoperable and cryptographically verifiable. REST APIs define the structure and semantics of information exchange, while OIDC and JWT embed the security and identity attributes that allow the mesh to determine who is making a request, under what authorization, and for what purpose. In this way, the connectors in PROTECT-CHILD do not merely link components—they operationalize compliance, embodying the principles of privacy-by-design and Zero Trust that govern the entire Secure Processing Environment.

6.3.1 REST API

In the PROTECT-CHILD platform, the **REST-API architecture** functions as the principal mechanism for decoupling and integrating the system's diverse components, enabling seamless interaction between microservices while preserving autonomy, scalability, and compliance. Each functional domain of the platform—data ingestion, pseudonymisation, federated computing, governance, discovery, and interoperability—exposes its operations through RESTful endpoints that communicate using standard HTTP methods and structured JSON payloads. This approach allows every module to evolve independently, as each one defines a stable, self-contained interface that other services can consume without depending on its internal implementation. The REST paradigm thus becomes a language of interoperability: it standardizes communication across different namespaces within the Kubernetes and Istio ecosystem, as well as between capsules participating in the federated architecture. By relying on REST APIs, data requests, algorithm executions, and governance actions can be orchestrated through clearly defined contracts, which are easily secured, audited, and versioned. Within the Zero Trust mesh, these REST calls are encapsulated in mTLS-encrypted connections and authenticated through OpenID Connect tokens, ensuring that every request carries a verifiable identity and a specific authorization context. In practice, this means that a federated analytics workflow can trigger computations, request dataset metadata, or retrieve aggregated results without any component requiring direct database or filesystem access. Each interaction occurs through a controlled, policy-governed REST interface, which acts as both a logical boundary and a trust enforcement point. In this way, the REST-API layer in PROTECT-CHILD is not merely a technical convenience, but a structural element of its security and interoperability model—enabling the platform to remain modular and distributed while functioning as a unified, compliant ecosystem.

6.3.2 OpenID Connect and JWT

OpenID Connect (OIDC) serves as the foundational identity layer in the PROTECT-CHILD Zero Trust architecture, providing a standardized, interoperable method for authenticating both human users and services before they interact with any component of the Istio service mesh. Built on top of the OAuth 2.0 framework, OIDC introduces an **identity token**—a JSON Web Token (JWT)—that carries cryptographically signed claims describing the authenticated entity (T6.1). These claims may include the user's identity, institutional affiliation, roles, and the purpose or scope of data access as defined in the data permit. When a user or service initiates a request, the Identity Provider (such as Keycloak) authenticates the entity through federated credentials (eIDAS, institutional SSO, or GA4GH passports) and issues an OIDC token. This token accompanies the request as an HTTP Authorization header and becomes the **digital passport**

that travels through the mesh. Within Istio, the token is verified at the network edge—typically at the Ingress Gateway or the waypoint proxy—through **JWT validation filters** configured in Istio's RequestAuthentication and AuthorizationPolicy resources. The proxy checks the token's signature against the Identity Provider's public key (retrieved from its JWKS endpoint), validates its issuer, audience, and expiry, and extracts the embedded claims into the request's authentication context. Once validated, these claims are made available to Istio's policy engine, allowing fine-grained access control based on project, role, or purpose attributes. In this way, **OIDC seamlessly integrates identity and access management into the fabric of the service mesh**, ensuring that authentication occurs before any request reaches an application workload. Each service, therefore, receives only verified and authorized traffic, and every decision about trust is derived from the cryptographic proof contained in the JWT. This integration transforms Istio from a simple network security layer into a **context-aware, identity-driven trust framework**, perfectly aligned with the Zero Trust model required for compliant and privacy-preserving health data processing.