Project acronym:	Lasers4MaaS
Project title:	Laser-as-a-Service Digital Platform with Dynamic Beam Shaping for Acceleration of Smart, Decentralised and Sustainable Factory of the Future
Call	HORIZON-CL4-2024-TWIN-TRANSITION-01-03 Manufacturing-as-Service: technologies for customised, flexible, and decentralised production on demand
Grant agreement No:	101178719
Work-package 5:	Digital platform and database management
Deliverable D5.2:	Architecture of the Laser4Maas Platform Database
Owner:	FUTURICE
Due date:	31/05/2026

Type		
R	Document, report (excluding the periodic and final reports)	
DEM	Demonstrator, pilot, prototype, plan designs	
DEC	Websites, patents filling, press & media actions, videos, etc	
DATA	Data sets, microdata, etc	
DMP	Data management plan	
ETHICS	Deliverables related to ethics issues	
SECURITY	Deliverable related to security issues	
OTHER	Software, technical diagram, algorithms, models, etc	x

Dissemination level		
PU	Public, fully open, e.g. project website	x
SEN	Sensitive, limited under the conditions of the Grant Agreement	
Classified R-UE/EU-R	EU RESTRICTED under the Commission Decision No2015/444	

Disclaimers: this project has received funding from the European Union's HORIZON EUROPE research and innovation programme under grant agreement No. 101178719. The content of this publication is the sole responsibility of the Consortium partners listed herein and does not necessarily represent the view of the European Commission or its services.

HISTORY OF CHANGES

Version	Publication date	Change
1.0	18/05/2026	First draft
2.0	28/05/2026	Final version

LIST OF AUTHORS

Name	Organization	Email
Denis Washington	FUTURICE	denis.washington@futurice.com
Christoph Nützel	FUTURICE	christoph.nuetzel@futurice.com
Jakob David Neuhauser	TUW	jakob.neuhauser@tuwien.ac.at
Volkher Onuseit	USTUTT	volkher.onuseit@ifsw.uni-stuttgart.de
Marco Cinelli	ULEI	m.cinelli@luc.leidenuniv.nl
Linda Eller-Shein	CIVAN	linda.shein@civanlasers.com
Yao Ren	RACE	yao.ren@ukaea.uk
Piergiuseppe Gianotti	CORNAGLIA	piergiuseppe.gianotti@cornaglia.com
Tobias Hohmann	FFT	tobias.hohmann@fft.de
Pasquale Franciosa	WMG	p.franciosa@warwick.ac.uk
Ivan Moretti	ECOR	ivan.moretti@ecor-international.com
Rinaldo Rigon	ECOR	rinaldo.rigon@ecor-international.com

REVIEWED BY

Name	Organization	Email
Pasquale Franciosa	WMG	p.franciosa@warwick.ac.uk

TABLE OF CONTENTS

1. Executive summary 9
2. Introduction, Goals & Constraints 9
2.1. Data Architecture Goals 9
2.2. Data Architecture Constraints 9
3. Technical Context 10
3.1. Data Storage & Distribution Approaches 10
3.1.1. Centralized Database 10
3.1.2. Data Federation & Data Spaces 10
3.1.3. Centralized Ledger 11
3.1.4. Distributed Ledger / Blockchain 11
3.2. Data Streaming and Messaging Protocols 12
3.2.1. MQTT (Message Queuing Telemetry Transport) 12
3.2.2. AMQP (Advanced Message Queuing Protocol) 13
3.2.3. Kafka (Apache Kafka) 13
3.2.4. NATS & NATS JetStream 13
3.3. Industrial Data Interoperability Standards 14
3.3.1. ISA-95 (International Society of Automation - 95) 14
3.3.2. OPC UA (Open Platform Communications Unified Architecture) 15
3.3.3. Asset Administrative Shell (Industrie 4.0) 15
4. Platform Database Architecture 16
4.1. Choice of Database Technology 16
4.2. Data Location & Ownership 17
4.3. Data Model 17
4.3.1. Materials 18
4.3.2. Welding Task Specification 18
4.3.3. Process Parameters 19
4.3.4. Process Development 20
4.3.5. Welding Stations 21
4.3.6. Remote Monitoring 21
5. Platform Data Interoperability & Integrations 22
5.1. Interoperability Approach 22
5.2. JSON-Based Data Exchange Formats 22
5.2.1. Welding Tasks & Process Parameters 23
5.2.2. Platform Service APIs 23
5.3. Experiment Series Import Spreadsheet Format 24
5.4. Beam Shape Import & Export Format 24
5.5. Monitoring Data Ingestion 24
6. Impact 26

7. Exploitation opportunities 26
8. Outlook 27

LIST OF FIGURES

Figure 1: Levels of the ISA-95 Model....14
Figure 2: Basic Structure of the Asset Administration Shell....16
Figure 3: Submodels of the Lasers4MaaS Platform Database data model and their relationships....17
Figure 4: UML class diagram for the *Welding Task Specification* data submodel....18
Figure 5: UML class diagram for the *Process Parameters* data submodel....19
Figure 6: UML class diagram for the *Process Development* data submodel....20
Figure 7: UML class diagram for the *Welding Stations* submodel....21
Figure 8: UML class diagram depicting the Remote Monitoring sub-model....21
Figure 9: Excerpt from an example welding task represented as JSON....23
Figure 10: Excerpt from the Process Selector API documentation, generated from the OpenAPI specification....23
Figure 11: Example of an experiment series spreadsheet that can be imported into the platform....24
Figure 12: Example of the CIVAN beam shape definition format....24
Figure 13: Data acquisition flow from local sensors to the Lasers4MaaS platform....24
Figure 14: Overview of the platform's Remote Monitoring Pipeline....25

LIST OF TABLES

Table 1: Data formats and protocols used for interoperability 22

LIST OF ABBREVIATIONS

AAS	Asset Administration Shell
API	Application Programming Interface
DoE	Design of Experiment
DPP	Digital Product Passport
ESPR	European Sustainable Products Regulation
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
IP	Intellectual Property
SaaS	Software as a Service
SQL	Structured Query Language

1. Executive summary

This report describes the architecture of Laser4Maas platform database and related data interoperability concerns, such as the used or developed data models, formats, and exchange protocols.

Evidence is provided for the following work package tasks:

- **T5.1** (*Review of current cloud services, software interfaces and APIs*), with regards to data storage and data transmission from laser welding stations to the platform, mainly in sections 3 and 5.5. Additional details can be found in Deliverable 1.3 (*Current standards and approaches for interoperability of data streams*).

- **T5.3** (*Database architecture, IIoT connectivity, and manufacturing data space management*). Section 4 discusses the data architecture, including choice of database technology choices (section 4.1), data location and ownership (section 4.2), and the data stored to support the platform's use cases (section 4.3). Section 5 discusses data interoperability.

2. Introduction, Goals & Constraints

The storage and interopable exchange of data is at the heart of the CONNECT objective addressed by the Lasers4MaaS digital platform. The platform's approach to data collection, storage, and sharing, as well as the data models, formats and protocols behind these activities, are critical to reaching this objective, and are discussed in this report.

The data architecture is structured to support the three-layer framework presented at the I-ESA 26 conference, which bridges the "Physicality Gap" in decentralized manufacturing. The database serves as the digital repository for the "Digital Handshake," ensuring that validated process "recipes" can be securely stored and shifted between partners. This methodology was peer-reviewer and presented at the I-ESA 26 conference (April 2026):

"Case Study: Servitisation as the Driver for Supply Chain Resilience – Enabled by Multi-Level Interoperability in the Lasers4MaaS Platform" (Nützel et al., 2026, Forthcoming).

The platform's data strategy is operationalized across three distinct layers designed to integrate federated networks into a cohesive manufacturing environment. At the micro level of shop floor interoperability, the architecture relies on industry-standard protocols such as OPC UA and MQTT to ensure hardware-agnostic data collection from diverse laser vendors, thereby supporting the project's objective of software-defined physical execution. At the meso level, the platform provides sovereign data management to protect highly confidential welding recipes that constitute critical trade secrets. This is achieved through a PostgreSQL database schema designed for multi-tenancy, which providing isolated data slices for individual companies while remaining architecturally ready for future integration into federated manufacturing data spaces such as Catena-X. For macro-level supply chain resilience, the architecture integrates NATS JetStream for high-performance, cloud-native messaging to enable real-time data streaming for predictive maintenance and the population of data fields required for DPPs to automate compliance with EU regulatory frameworks.

2.1. Data Architecture Goals

- **Interoperability:** Connect the platform to machines and services through well-defined interfaces and data formats, building on established industry standards where possible and sensible, to ease data exchange and integration into the wider digital landscape.

- **Portability:** Architecture the platform database to be deployable in most cloud and on-premise environments, to avoid dependence on particular hosting providers (especially non-EU-based ones), maximize data protection, and open the door to future decentralized deployment scenarios like Data Spaces

2.2. Data Architecture Constraints

The following issues shape the design of the platform's data aspects:

- **Digital sovereignty:** In a world of growing geopolitical stability, strategic autonomy and resilience in the digital realm is becoming a critical concern for the European manufacturing industry. The design of the platform must take this reality into account, for instance by avoiding dependence on proprietary technologies of US hyperscalers.

- **Intellectual property protection:** Companies developing laser welding processes consider these as critical trade secrets. The platform must ensure that process development data is stored securely and not inadvertently exposed to competitors.

3. Technical Context

This section surveys the data storage technologies, protocols, models and formats considered in the design of the Lasers4MaaS database architecture. For convenience, it includes content excerpts from deliverables 1.2 (*Current standards and approaches for interoperability of digital twins for automation workflows*) and 1.3 (*Current standards and approaches for interoperability of data streams*).

3.1. Data Storage & Distribution Approaches

Multiple approaches to storing and sharing data were considered for the Lasers4MaaS platform. These are described and compared below.

3.1.1. Centralized Database

In this model, a single database hosted by the platform provider (usually in the cloud) serves multiple companies, or "tenants". The platform provider manages the infrastructure, while tenants access a private slice of the database through the platform's user interfaces and APIs. This is the approach taken by cloud-based Software-as-a-Service (SaaS) solutions.

Classic database management systems like PostgreSQL (relational) or MongoDB (document-based) are good fits for this model. They allow multi-tenacy to be achieved through query patterns – for instance, by filtering for a "tenant ID" data field to attached to each database record – or through their ability to host multiple isolated instances of a database on the same server or cluster, enabling each tenant to be assinged to a separate instance.

Key benefits of this approach are:

- **Efficiency:** A single, centralized deployment can be more efficiently operated due to economies of scale, reducing the platform's overall economic cost and environmental impact.

- **Low adoption cost:** Companies do need to set up and maintain their own database and platform infrastructure, which also total .

The main downside is that tenants must fully trust the platform provider not to access, leak, or manipulate their proprietary data, especially to competitors also using the platform. This can present a barrier to adoption.

3.1.2. Data Federation & Data Spaces

In the federation approach, each company using the platform hosts a separate instance of the database – and, usually, the platform itself – in its own cloud or on-premise infrastructure, giving it full control and sovereignity over its own data. Data exchange between different companies is done through import / export features or direct peer-to-peer communication using standardized protocols and data formats.

Data Spaces build on this approach by adding an identity and governance layer, creating an ecosystem where participants can exchange data and services with agreed-upon, clearly defined terms of usage. For instance, Catena-X[1] is a data space for the automative industry that aims to enable use cases such as the automated collection of Product Carbon Footprint (PCF) data across the whole supply chain, by providing verified digital company identities, a

[1] *Catena-X, https://www.cofinity-x.com/catena-x*

governance framework with common rules maintained by a non-profit entity founded for that purpose, and open-source standards.

Key benefits:

- **Data sovereignty:** Companies retain full control and ownership of their data and can selectively share it with others without transferring it to an intermediary third party.

Challenges:

- **Governance overhead:** Data spaces require a governance structure to be set up so that rules can be defined and enforced, as well as to provide infrastructure services like identity verification. A simple federation approach without such a governance layer suffers from a lack of these rules, making it harder to build trust relationships.

- **Limited maturity:** While some data spaces like Catena-X and reusable infrastructure components like Eclipse Data Connectors (EDC) are being used in production already, they are relatively young and standardization is still ongoing.

3.1.3. Centralized Ledger

Ledger technologies provide a tamper-evident, immutable record of transactions managed by a single trusted authority. Unlike a standard database, a ledger maintains a cryptographically verifiable history of every change ever made.

These ledgers function by maintaining an append-only journal that tracks every data addition or change. When a new transaction is recorded, the system generates a cryptographic hash of that data and "chains" it to the hash of the previous entry. This creates a Merkle Tree or a linear hash chain. Because each block contains the fingerprint of the one before it, any attempt to alter a historical record would break the chain, alerting the system and the tenants to the tampering.

Examples of centralized ledger technologies are the Ledger feature of Microsoft SQL Server (also available in Microsoft's Azure cloud) and immudb, and open-source immutable database.

Key benefits:

- **Immutability:** Every entry is cryptographically linked, making it impossible to alter history without detection, including by the platform provider.

- **Verifiability:** Provides a transparent, tamper-evident audit trail that can be independently verified by any party with access to the ledger.

Challenges:

- **Lower performance:** Every write to a ledger requires cryptographic operations to link the new data records to the existing chain, increasing computing resource usage and latency compared to a classic database.

- **Trust requirement remains:** While the platform provider cannot tamper with the tenants' data, it can still access and potentially leak it, so tenants must still trust the platform provider if the data is confidential.

3.1.4. Distributed Ledger / Blockchain

A distributed ledger is a consensus-based database shared across multiple sites or institutions. Unlike a centralized database or ledger, no single party has complete control over the data; instead, participants maintain and validate their own copies of the ledger.

When a tenant proposes a transaction (such as the addition of a data record) to a distributed ledger, it is broadcast to all participating nodes, which use a consensus algorithm to agree on the validity of the transaction and its order to relative to other transactions. Once consensus is reached, the transaction is added to every participant's local copy.

Blockchains are the most well-known type of distributed ledgers, characterized by their grouping of transactions into a linear chain of cryptographically linked "blocks." Examples are the Bitcoin and Ethereum networks. In such public permissionless networks, any user can join the network, view the entire ledger, and participate in the consensus process, which is usually based on cryptocurrency-based mechanisms like proof-of-work ("mining", as in Blockchain) or proof-of-stake (vouching for the validity of transactions by tasking cryptocurrency on it, as in modern Ethereum).

More suitable for use in industrial settings are private permissioned blockchain systems like Hyperledger Fabric, where a ledger can only be accessed by authorized members and "channels" allow particular transactions to only be shared between certain participants. Other private permissioned systems like R3 Corda are based on Directed Acylic Graphs (DAGs) rather than blockchains; in these, transactions only need to be shared with the participants that should know about them, rather than the entire network. These technologies can provide data confidentiality without workarounds such as encryption.

Key benefits:

- **Decentralized trust:** Distributed ledgers allow participants collaborate in data storage and sharing to without a single central authority.
- **Tamper resistance:** In addition to allowing data tampering to be detected, as is already possible with a centralized ledger, distributed ledgers usually prevent such tampering from even being committed as consensus will not be reached on faulty transactions.

Downsides:

- **Infrastructure heavy:** Even with selective sharing, participants must still maintain complex node infrastructure to participate in the protocol.
- **Limited data scope:** Distributed ledger technologies are typically optimized for "states" (balances, contract statuses, ownership) rather than large data volumes.

3.2. Data Streaming and Messaging Protocols

Efficient and reliable data streaming and messaging are essential for real-time communication and control within the industrial setting. This chapter explores various data streaming and messaging protocol standards, each offering unique features and capabilities tailored to different communication needs.

3.2.1. MQTT (Message Queuing Telemetry Transport)

MQTT is a lightweight, publish-subscribe messaging protocol designed for resource-constrained devices and low-bandwidth networks. It is widely used in Internet of Things (IoT) applications, enabling efficient communication between devices and cloud platforms.

Key features of MQTT include:

- **Lightweight:** MQTT's small message overhead and efficient encoding minimize bandwidth consumption, making it suitable for low-bandwidth networks and resource-constrained devices.
- **Publish-subscribe:** MQTT's publish-subscribe pattern enables efficient one-to-many communication, allowing devices to publish data to topics and other devices to subscribe to those topics to receive the data.
- **Quality of Service (QoS):** MQTT offers different QoS levels, allowing applications to choose the level of message delivery guarantee based on their needs.

MQTT could be utilized for tasks such as collecting sensor data from edge devices, transmitting real-time updates, or enabling remote monitoring and control of equipment. Its lightweight nature and publish-subscribe pattern make it well-suited for connecting many devices and efficiently distributing data within the platform.

3.2.2. AMQP (Advanced Message Queuing Protocol)

AMQP is an open standard application layer protocol for message-oriented middleware. It provides a standardized framework for reliable message exchange between applications, ensuring message delivery and guaranteeing message order.

Key features of AMQP include:

- **Reliability:** AMQP offers robust message delivery guarantees, including persistent message storage and acknowledgment mechanisms, ensuring that messages are not lost in transit.
- **Flexibility:** AMQP supports various messaging patterns, including point-to-point, publish-subscribe, and request-response, providing flexibility for different communication needs.
- **Interoperability:** AMQP is an open standard, promoting interoperability between different messaging systems and vendors.

AMQP can be employed for tasks such as exchanging messages between different components, ensuring reliable delivery of critical data, or integrating with existing messaging infrastructure. Its reliability and flexibility make it well-suited for scenarios where message delivery guarantees and support for various messaging patterns are essential.

3.2.3. Kafka (Apache Kafka)

Kafka is a high-throughput, distributed streaming platform designed for handling real-time data feeds. It provides a scalable and fault-tolerant infrastructure for ingesting, processing, and storing streams of data.

Key features of Kafka include:

- **High throughput:** Kafka's distributed architecture and efficient data storage mechanisms enable it to handle high volumes of data with low latency.
- **Scalability:** Kafka can be easily scaled horizontally to accommodate growing data volumes and traffic.
- **Fault-tolerance:** Kafka's replication and failover mechanisms ensure data durability and high availability.

Kafka could be utilized for tasks such as ingesting large volumes of sensor data, processing real-time events, or building data pipelines for analytics and machine learning. Its high throughput, scalability, and fault-tolerance make it well-suited for handling the demanding data streaming requirements.

3.2.4. NATS & NATS JetStream

NATS is a high-performance, cloud-native messaging system designed for modern distributed systems. It acts as a lightweight "always-on" dial tone for passing messages between services, providing a highly resilient and simple infrastructure for service discovery, request-reply patterns, and event-driven architectures.

Key features of NATS include:

- **Extreme Performance:** Written in Go, NATS is designed for speed and efficiency, capable of handling millions of messages per second with incredibly low latency.
- **Simplicity and Security:** NATS features a "zero-config" philosophy and a robust security model that supports decentralized authorization, making it easy to deploy across diverse environments.

- **Adaptive Topologies:** It supports various communication patterns, including Pub/Sub, Request-Reply, and Load-balanced queues, allowing it to adapt to complex microservices requirements.

NATS JetStream is the built-in persistence layer for NATS, extending its core capabilities to include data streaming and "at-least-once" delivery guarantees. It transforms NATS from a pure messaging system into a powerful distributed log and stream processing engine.

Key features of NATS JetStream include:

- **Persistence and replay:** JetStream allows messages to be stored on disk or in memory, enabling consumers to replay historical data or catch up after a disconnect.
- **Horizontal scalability:** Much like Kafka, JetStream can be scaled horizontally through clustering, distributing stream data across multiple nodes to handle massive traffic.
- **Pull and push consumers:** It offers flexible consumption models, allowing services to either have data pushed to them in real-time or pull data at their own pace to manage backpressure effectively.

NATS and JetStream are frequently utilized for edge computing, IoT device communication, and microservices orchestration. Their lightweight footprint and built-in self-healing capabilities make them an excellent alternative to heavier streaming platforms when operational simplicity and resource efficiency are priorities.

3.3. Industrial Data Interoperability Standards

A key success factor for the Lasers4MaaS platform is interoperability and integration with the industrial digital landscape, including devices, sensors, and possibly Manufacturing Execution Systems (MES). This section explores relevant standards in this area.

3.3.1. ISA-95 (International Society of Automation - 95)

The ISA-95 standard, also known as ANSI/ISA-95 or IEC 62264, defines an abstract model for the integration of enterprise logistics and control systems. It provides a framework for organizing and exchanging information between different levels of a manufacturing organization.

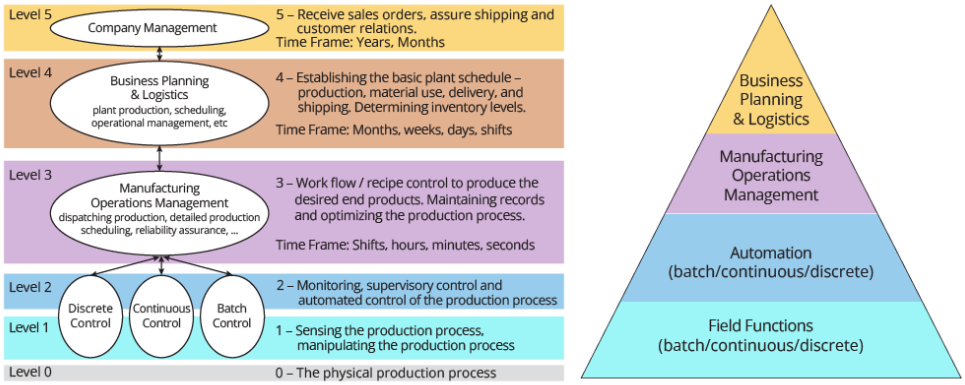

Figure 1: Levels of the ISA-95 Model[2]

ISA-95 organizes industrial activities into a functional hierarchy comprised of five distinct levels. Level 0 represents the actual physical production process and equipment on the plant floor, which monitored and manipulated by the sensors and actuators residing at Level 1. Level 2 encompasses the automated control systems, such as PLCs and SCADA, that orchestrate real-time operations, while Level 3 serves as the Manufacturing Operations Management (MOM) layer, handling work-order execution, detailed scheduling, and quality management over hours or shifts; this is the domain of Manufacturing Execution Systems (MES). Level 4 represents the Business Planning and Logistics

[2] Source: https://www.isa.org/standards-and-publications/isa-standards/isa-95-standard

tier, where Enterprise Resource Planning (ERP) systems manage long-term strategic functions like inventory tracking, procurement, and high-level production scheduling.

Most of ISA-95 is focused on the integration of Level 3 and 4, i.e. of MES and ERP systems. For this purpose, it defines several object models in areas such as areas such as personnel, equipment, physical assets, and material resources, which collectively provide a standardized language for describing manufacturing operations.

3.3.2. OPC UA (Open Platform Communications Unified Architecture)

OPC UA is a platform-independent, service-oriented architecture for industrial communication. It provides a standardized framework for data exchange between various devices and systems, enabling seamless interoperability across different vendors and platforms. On top of this framework, it defines multiple transport protocols, including application-level ones based AMQP, MQTT, and HTTPS.

OPC UA integrates with ISA-95 by providing a companion specification that maps several ISA-95 object models to the OPC UA address space. This allows OPC UA to be used for data transfer between Level 3 and 4 of the IA-95 level architecture. Some MES platforms like Siemens Opcenter leverage this, exposing their data to external system through OPC UA to maximize interoperability.

Key features of OPC UA include:

- **Platform independence:** OPC UA can be implemented on various operating systems and hardware platforms, promoting interoperability across diverse devices and systems.
- **Security:** OPC UA incorporates robust security mechanisms, including authentication, authorization, and encryption, ensuring secure communication and data protection.
- **Information modelling:** OPC UA allows for the definition of information models, enabling the representation of complex data structures and relationships.
- **Wide industrial adoption:** OPC UA has been adopted by a wide range of manufacturers of smart devices, making it a good choice for machines which natively support it.

OPC UA can be employed for tasks such as connecting to industrial equipment, exchanging sensor data, or controlling manufacturing processes. Its platform independence and security features make it well-suited for integrating diverse devices and systems within the platform.

3.3.3. Asset Administrative Shell (Industrie 4.0)

The Asset Administrative Shell (AAS) specifications define a standardized framework for the interoperable digital representation of industrial assets, enabling the interoperable exchange of asset information within or across industrial organizations. Developed by the Industrial Digital Twin Association (IDTA) and partly adopted as international standard by the International Electrotechnical Commission (IEC), it is a key part of the Industrie 4.0 vision for digital transformation in the manufacturing industry.

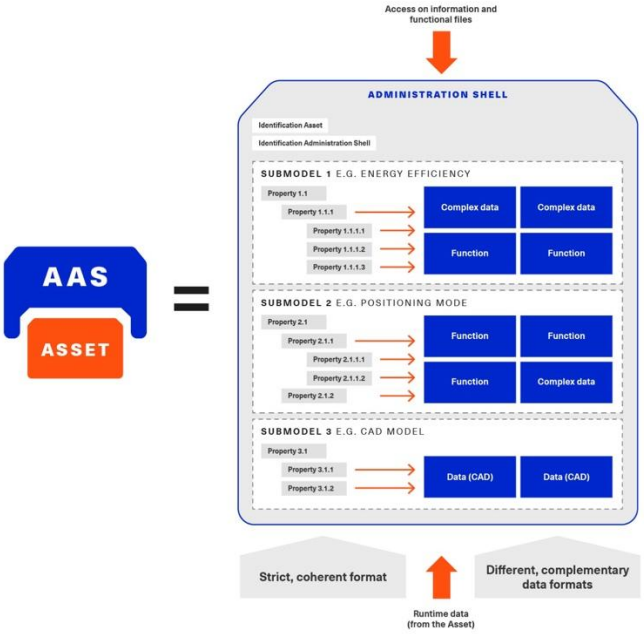

Figure 2: Basic Structure of the Asset Administration Shell[3]

The AAS specifications define a *metamodel* describing the structure of an Administrative Shell, which consists of a unique identifier for the asset and the shell itself, as well as a set of *submodels*, each representing a specific domain or functional aspect of the asset, such as technical properties, documentation, or energy consumption. Several types of submodels (called *submodel templates*) have already been standardized, and new ones can be defined using XML, JSON, or RDF (Turtle).

In addition, APIs for communicating with an AAS server are also specified, with mappings to HTTP and OPC UA, which allows for interoperable exchange with and between Administrative Shells.

4. Platform Database Architecture

This section examines the design and architecture of the Lasers4MaaS platform database and data architecture.

4.1. Choice of Database Technology

We have chosen PostgreSQL as the basis for the Lasers4MaaS platform database. PostgreSQL is a mature, versatile, and extensible open-source database management system widely known and used in the software industry. It closely adheres to the Structured Query Language (SQL) standard and fully supports its relational data model, but also supports a range of non-relational data types such as JSON documents and geometric shapes. It also features vibrant ecosystem of extensions for various use cases, such as PostGIS for geographical data, pgvector for vector similarity search, and TimescaleDB for high-volume time series.

We have chosen PostgreSQL as the basis for the Lasers4MaaS platform database. Its flexibility allows us to use it both for typical data records – such as process development documentation – and for time series like sensor data streamed from the shop floor, avoiding the architectural and operational complexity of maintaining multiple database systems. In addition, the system's open-source nature and vendor-neutral governance model fits the Lasers4MaaS platform's goal of avoiding vendor lock-in.

Ledger technologies were rejected for the platform database, for multiple reasons. Firstly, the main benefit of these systems – tamper resistance – are not very relevant for the types of data and use cases targeted by the platform, especially if a later version of the platform moves to a federated model where each company controls their own data. Secondly, the cryptographic operations needed for writes to a ledger limit throughput, making it a bad fit for ingesting sensor data, which needs to be written frequently. Lastly, most ledger systems have extremely limited querying

[3] Source: https://industrialdigitaltwin.io/aas-specifications/IDTA-01001/v3.1.2/annex/concepts-aas.html

abilities, so most use cases would necessitate streaming to a second database system, increasing architectural complexity and introducing eventual consistency issues.

4.2. Data Location & Ownership

In its initial version, the platform and its database follow the centralized, multi-tenant model described in section 3.1.1. Multi-tenancy is achieved by associating each platform user account with an organization and ensuring that all database queries filter results by the requesting user's organization ID, as is standard practice for many Software-as-a-Service (SaaS) applications.

We believe that the Lasers4MaaS platform should eventually move towards data federation and align with the European Data Spaces initiative, due to its data sovereignty benefits. However, federation also significantly complicates the architecture and implementation of the platform. As the impact of the Lasers4MaaS platform can also be demonstrated with a centralized database, we decided that this approach is sufficient for the initial platform version.

Nevertheless, even in the centralized model, the platform database strictly segments the data storage by company, giving each company exclusive control over their data and data sharing. Data exchange across company borders happens through well-defined data formats and protocols through import / export interfaces or APIs, an approach that transfers well to a future federated deployment model.

Nevertheless, even in the centralized model, the platform database strictly segments the data storage by company, giving each company exclusive control over their data and data sharing. Data exchange across company borders must be explicitly initiated by the involved parties and happens through well-defined data formats and protocols through import / export interfaces or APIs, an approach that transfers well to a future federated Data Spaces model.

4.3. Data Model

The platform database's logical data model defines the database's data *entities*, which represents the types of objects represented in the database; the *attributes* defining the data points recorded for each entity instance; and the *relationships* between the entities. These three things correspond roughly to SQL tables, columns, and foreign keys in the database's physical data model.

The data model is split into multiple sub-models, an overview of which can be seen in Figure 3. They are detailed in the following sub-sections.

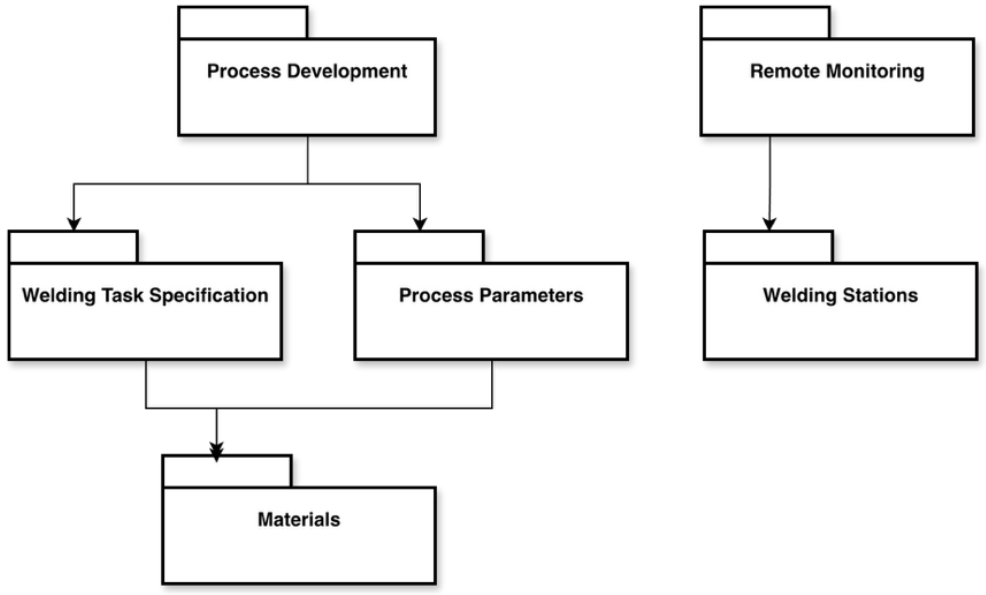

Figure 3: Submodels of the Lasers4MaaS Platform Database data model and their relationships

Note: The details of the data model are expected to still evolve based on the experience gained during implementation and testing. The descriptions below represent a snapshot of the current status.

4.3.1. Materials

Several parts of the data model require a way to reference metals and alloys. In the data model, these are represented as instances of the *Material* entity. For unambiguous identification, we reuse existing designation systems from European and international standards, such as EN 10027-2 steel numbers, ISO 2325-1 designations for Titanium and Titanium alloys, and the Unified Numbering System (UNS) used in North America.

4.3.2. Welding Task Specification

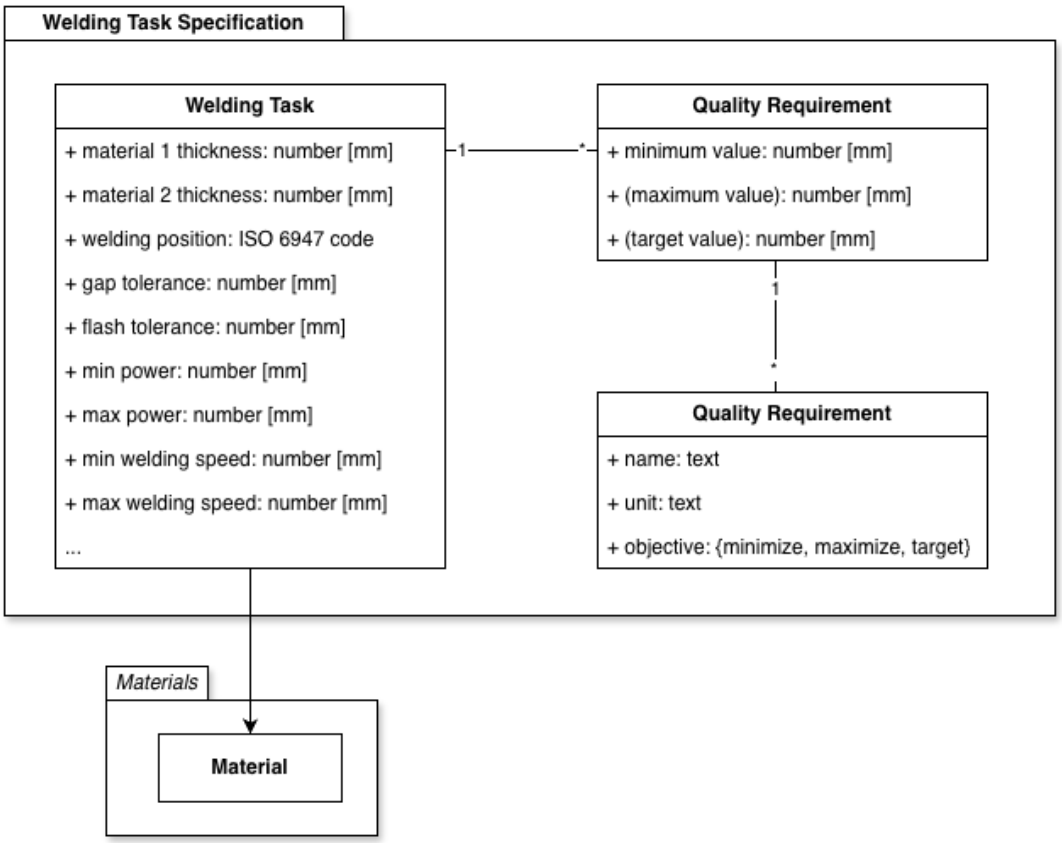

Figure 4: UML class diagram for the *Welding Task Specification* data submodel

The *Welding Task Specification* sub-domain defines the *Welding Task* entity, which represents the specification of a laser welding task to be performed at a particular weld seam. It's attributes and relationships include information about the parts to be welded (such as their materials and thickness), welding position details (based on the ISO 6947[4] standard), the required part-to-part gap and flash tolerance, and constraints on the process parameter space like maximum laser power (owing to the laser system intended to be used in production, for instance).

In addition, the *Quality Requirement* entity represents the expected quality outcomes for a process implementing the welding task. Each requirement is defined in terms of acceptable value range for a *Quality Indicator*, which stands for a particular quantified quality aspect. Each indicator is characterized by a name, its unit, and an "objective" specifying if the indicator value is to be minimized, maximum, or brought as close as possible to a task-specific target value. A set of quality indicators is predefined by the platform for convenience and to add data interoperability; users will be able to also define custom ones for their use cases

[4] *ISO 6947:2019, Welding and allied processes — Welding positions, https://www.iso.org/standard/72741.html*

4.3.3. Process Parameters

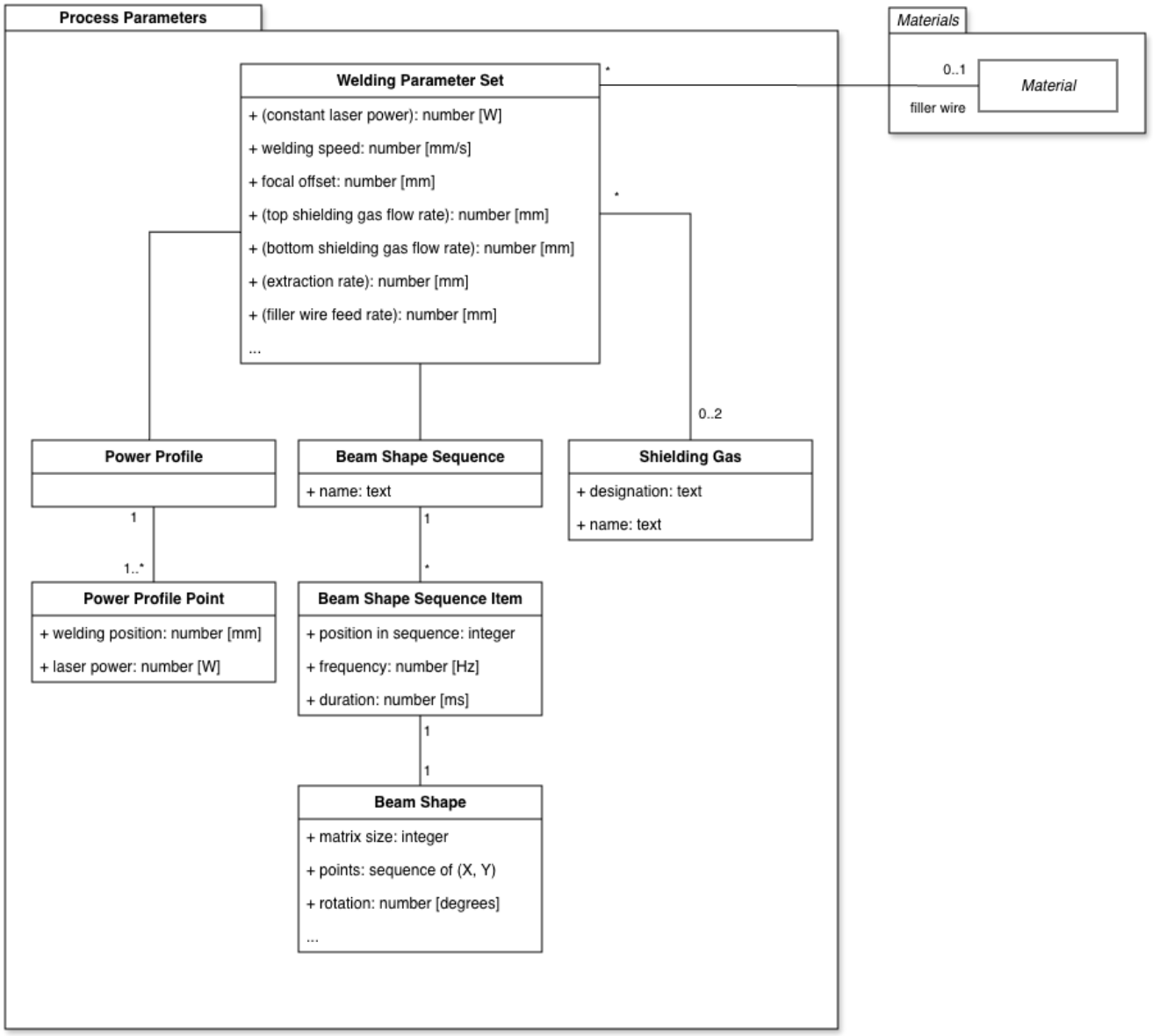

Figure 5: UML class diagram for the *Process Parameters* data submodel

The Welding Parameter Set entity is at the center of the Process Parameters sub-model. An instance of this entity represents a set point in the laser welding process parameter space, including parameter values related to the laser, motion system, fume extraction, and other aspects.

The laser power can be specified as a constant value or as a *power profile*. The latter is a definition of how the laser power should be adjusted upwards and downwards along the welding path through a set of *(welding position, laser power)* points. (Between the points, a linear change of power is assumed.)

As the Lasers4MaaS project explores dynamic beam shaping for reconfigurability, the beam shape is part of the data model in the form of the *Beam Shape* and *Beam Shape Sequence* entity. These conceptualize a beam shape as a list of points on a square grid representing the welding surface underneath the laser, with each grid coordinate corresponding to a surface point which the laser is able to target. The laser is assumed to generate the shape through Coherent Beam Combining (CBC), jumping between the shape's selected points at a given frequency. A Beam Shape Sequence may consist of multiple beam shapes, allowing changes to the shapes and their frequencies over time to be specified as well. This model reflects the capabilities of the CIVAN lasers used in the Lasers4MaaS research and demonstrations, but is expected to generalize to other CBC-based dynamic beam shaping lasers.

Gas and gas mixtures used for shielding are represented by the *Shielding Gas* entity. Instances are referenced using ISO 14175[5] designations.

4.3.4. Process Development

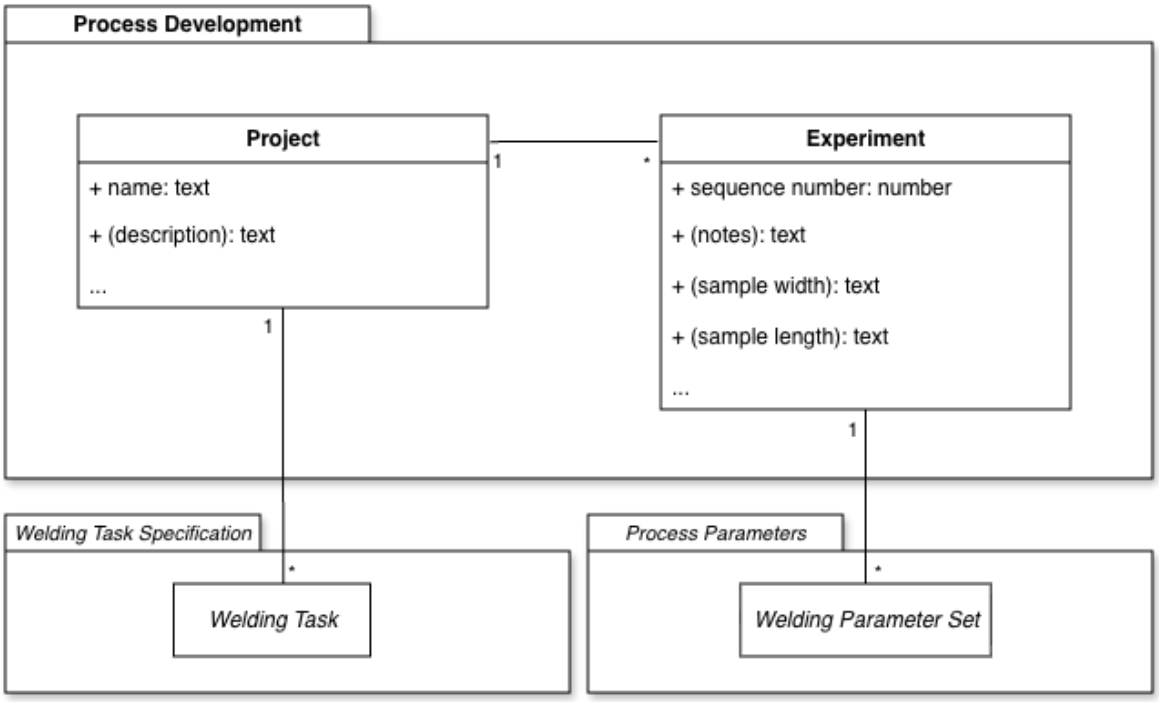

Figure 6: UML class diagram for the *Process Development* data submodel

The *Process Development* sub-model is a layer on top of *Welding Task Specification* and *Process Parameters* for tracking welding process development projects. Its main entities are *Project* and *Experiment*. A *Project* is an effort to find a suitable *Welding Parameter Set* for a *Welding Task,* whereas an *Experiment* records the details and outcomes of testing a particular parameter set. In addition to the parameters themselves, an Experiment can include free-text setup details and file attachments for documentation purposes.

The quality outcomes of an experiment are represented by the *Quality Result* entity, each instance of which records the value measured for a particular *Quality Indicator* associated with the welding task through its *Quality Requirements*.

[5] *ISO 14175:2008, Welding consumables — Gases and gas mixtures for fusion welding and allied processes,* https://www.iso.org/standard/39569.html

4.3.5. Welding Stations

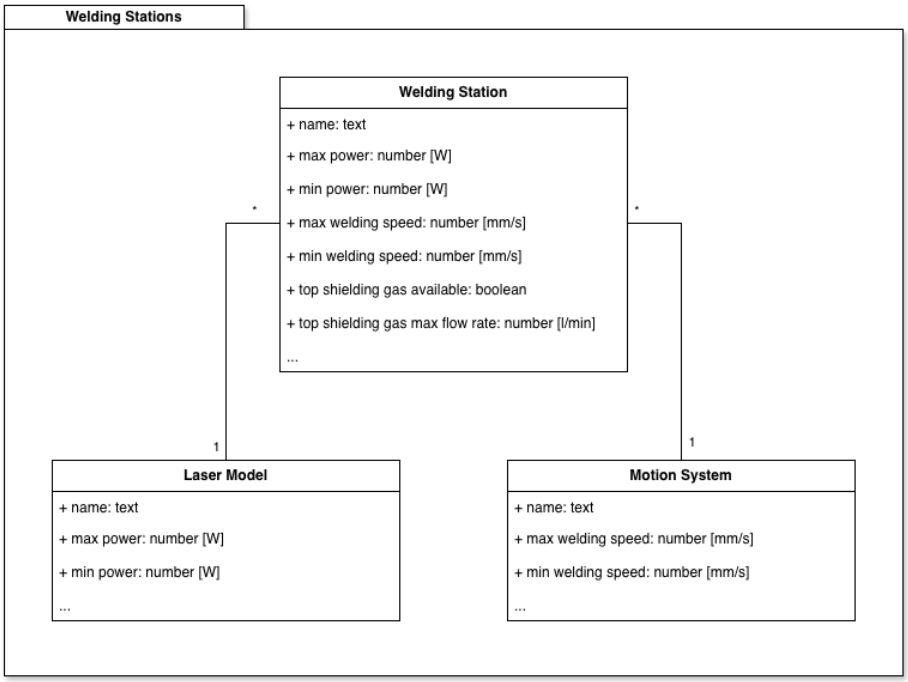

Figure 7: UML class diagram for the *Welding Stations* submodel

The *Welding Station* entity records information about the laser welding stations available for process development and production. It has attributes for identifying the welding station, as well as for recording its capabilities, such as the minimum and maximum supported laser power. In addition, that *Laser Model* and *Motion System* entities capture information about the type of laser and motion hardware that is part the station.

4.3.6. Remote Monitoring

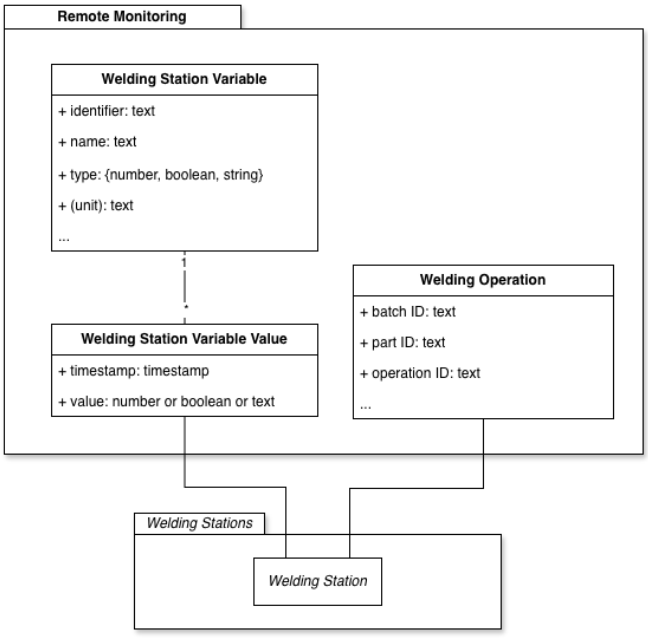

Figure 8: UML class diagram depicting the Remote Monitoring sub-model

The *Remote Monitoring* sub-model records sensor and welding trace data collected on the shop floor (see section 5.4) and associates it with it with the originating *Welding Stations*. This is modeled as a set of *Welding Station Variables* associated with each station, with time series data stored as *Welding Station Variable Values*. The

production traces are represented as sequences of *Welding Operations*. Using the timestamps, the sensor data can be correlated to the welding operations for analysis.

5. Platform Data Interoperability & Integrations

In this section, the Laser4MaaS platform's data architecture is examined from the lens of interoperable data exchange, focusing on data formats, protocols and integrations.

5.1. Interoperability Approach

Trusted, secure and interoperable cross-company data exchange is a central goal of the Lasers4MaaS platform. The platform and data architecture addresses this by giving companies full control over their data and data sharing (see section 4.2); securing communication with encryption and authentication using TLS and OpenID Connect (see Deliverable 5.1, section 6.3); and using well-defined data models, formats and protocols for data exchange, defining new ones where necessary (the topic of this section).

Table 1 summarizes the data formats and protocols used for the various types of interoperable data exchange in the platform. These formats and integrations are further discussed in the sub-sections below.

Table 1: Data formats and protocols used for interoperability

Use Case	**Data Format(s) and protocols**	**Discussed in**
Data exchange between platform components, data exports from the platform database	JSON (with JSON Schema specification), REST APIs	Section 5.2, D5.1
Import of process development experiment series from Design of Experiment (DoE) tools	CSV, Office Open XML (Microsoft Excel), OpenDocument Spreadsheet	Section 5.3
Ingestion of monitoring data from the shop floor	OPC UA, MQTT, JSON	Section 5.5, D11.1
Data exchange between simulation platforms (but not handled by the platform because it works with data to train models, before deployment to the actual Lasers4MaaS platform)	VTK	See D9.1 and D11.1

For integration use cases where there was no established standard that could be relied on, we introduced an abstraction layer to the platform to encapsulate vendor-specific formats and protocols. For instance, beam shape definitions are stored in a vendor-neutral form in the platform database but can be converted a format specific to CIVAN lasers thanks to an import / export mechanism (see section 5.4). This approach minimizes the effort to interoperate with additional systems, which can be done by creating more implementations of the abstraction layer (e.g., a beam shape importer and exporter for another laser system) without changes to the rest of the platform.

5.2. JSON-Based Data Exchange Formats

JSON (JavaScript Object Notation) is the default base format for data exchange within and beyond the Lasers4MaaS platform. Thanks to its simplicity, human readability, and built-in support for reading and generating it in most popular programming languages, its use is extremely wide-spread, especially for HTTP-based APIs, which made it a natural choice.

We use JSON Schema[6] to define the structure of the platform's JSON-based data formats in a formal, machine-readable way. This ensures that the data formats are both well-documented for humans and can automatically checkable by machines.

5.2.1. Welding Tasks & Process Parameters

Canonical JSON representations for the platform data model's *Welding Task Specification* and *Process Parameters* submodels (see sections 4.3.2 and 4.3.3). At the time of writing, these are used in the APIs between the core platform and the Process Selector and Process Optimizer services (described in Deliverable 5.1), but have the potential to be more widely applicable, for instance for exporting process parameters to other systems.

Figure 9: Excerpt from an example welding task represented as JSON

5.2.2. Platform Service APIs

The REST APIs of the platform's services (see Deliverable 5.1, section 6) use JSON for all inputs and outputs. Each API is specified using the OpenAPI standard, which allows using JSON Schema for defining request and response contents.

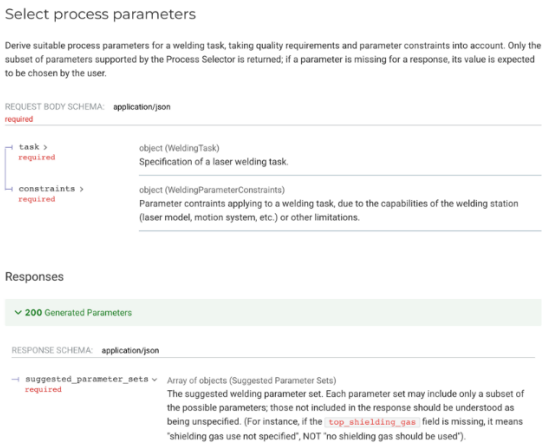

Figure 10: Excerpt from the Process Selector API documentation, generated from the OpenAPI specification

[6] *JSON Schema, https://json-schema.org/*

5.3. Experiment Series Import Spreadsheet Format

For interoperability with Design of Experiments (DoE) tools, the platform allows series of process development experiments to be imported from CSV, Office Open XML (Microsoft Excel), or OpenDocument Spreadsheet files. The sheet is expected to have a particular format, with one row per experiment and column names corresponding to the attributes of the *Welding Task* data model entity.

laser_power	welding_speed	focal_offset	beam_shape_sequence	top_shielding_gas	top_shielding_gas_flow_rate
5000	30	500	Example		
5500	30	500	Example		
6000	30	500	Example	Ar	5
6000	30	500	Example	Ar	10

Figure 11: Example of an experiment series spreadsheet that can be imported into the platform

5.4. Beam Shape Import & Export Format

The initial version of the platform allows users to import and export laser beam shape definitions in CIVAN's vendor-specific format, allowing developed beam shapes to be transferred across welding stations. Knowledge of the format has been limited to a single module in the core platform codebase, making it easy to extend the feature to other vendors' data formats by writing additional module implementations.

```
sizeMatrix:32
16,0
16,-2
15,-3
14,-5
14,-7
13,-8
11,-9
10,-11
9,-12
7,-13
5,-13
3,-14
1,-14
0,-14
-2,-13
-4,-13
-5,-12
-7,-11
-8,-10
-10,-8
```

Figure 12: Example of the CIVAN beam shape definition format

Direct configuration of the laser system parameters through OPC UA or similar interfaces was considered as an alternative or addition to export, but deemed infeasible during the project's runtime, due to current limitations of the CIVAN laser's configurability through such interfaces.

5.5. Monitoring Data Ingestion

Figure 13: Data acquisition flow from local sensors to the Lasers4MaaS platform

For the platform's remote monitoring and maintenance prediction features, it is necessary to stream sensor and production trace data from the shop floor to the platform database. To do so, the data is acquired by a PC on the shop floor. In the two demonstration areas set up for this project, the data acquisition is performed using the standard OPC-UA and MQTT protocols. This is described in more detail in Deliverable 11.1 (*IIoT sensors selection and characterisation*).

Once acquired, the sensor data is transferred from the shop-floor PC to the Lasers4MaaS platform's cloud-hosted database by the platform's *Remote Monitoring Pipeline*, pictured in Figure 14. It does so in two steps:

- A Python application running on the local workstation, called the *Monitoring Data Uploader*, regularly sends the latest sensor data to a message queue (specifically, a NATS[7] server) running in the cloud. Each welding station is represented as a separate queue ("subject" in NATS' terminology), making it possible for the platform to determine which station the data originates from.

- Within the cloud platform, the *Remote Monitoring Worker* platform component subscribes to the message queues and writes the sensor data into the platform database.

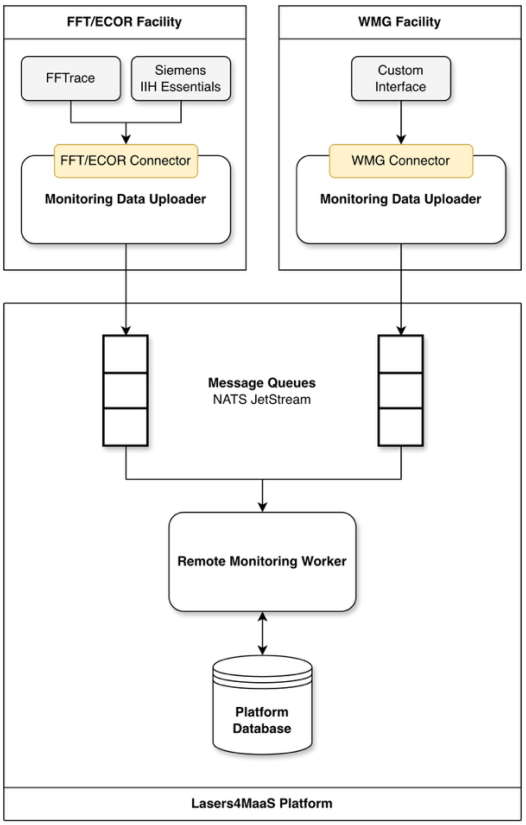

Figure 14: Overview of the platform's Remote Monitoring Pipeline

The intermediate message queue decouples the local application from the database, and also makes it possible for the platform to batch and/or throttle writes to the database to limit the load on it.

Each Monitoring Data Uploader installation has unique NATS credentials, with permissions limited to writing to the message queues corresponding to the facility's welding stations. Furthermore, the application is deployed and run as an OCI (Docker) container, limiting its access to the host system. These measures minimize the application's attack surface.

To abstract away the differences between the FFT/ECOR and WMG facility setups, the Monitoring Data Uploader application has the notion of *connectors*, which encapsulate the process of retrieving the sensor data and translating it to the JSON format expected by the platform. For instance, the connector for FFT/ECOR fetches data from a Siemens IIH Essentials[8] instance, while the one for WMG retrieves them through a custom-built interface. This design allows the application to be easily adapted to new deployment scenarios as the required code changes are limited to developing a new connector or extending an existing one.

[7] https://nats.io/
[8] https://www.siemens.com/en-us/products/industrial-edge/iih-essentials/

6. Impact

From the perspective of this database architectural deliverable, the platform's data models and JSON-based communication schemas are designed to reduce the standardized connectivity onboarding time from its historical baseline of 168 hours to a target of under 4 hours. Because the project is currently in its active development phase, this target has not yet been implemented in live production environments, and we are currently validating JSON-based schemas and API end-points within single-session setups to ensure schema stability. However, by treating data exchange formats as universally standard JSON Schemas, the database ensures that newly onboarded manufacturing nodes and R&D services can exchange welding task definitions, constraints, and process parameters without requiring custom database schema modifications. This standard format decouples the core storage layer from vendor-specific machine languages, transforming traditional days of ad-hoc database engineering into a rapid, automated software-defined configuration.

To establish the foundation for trusted B2B data exchanges, the database is architected to transition the transactional success rate from an unverified baseline to a resilient target of 100% utilizing multi-tenant schema isolation and persistent message queuing. At this stage in the project timeline, the transaction success rate for public, inter-company exchanges remains at a 0% baseline because we are operating in isolated, local-sandbox environments to ensure complete data isolation and prevent intellectual property leakage to competitors before deploying the federated network. The technology selection of PostgreSQL as a highly mature relational database is crucial to reaching this target, as it provides strict logical isolation of tenant data while remaining architecturally ready for future Data Space federation such as Catena-X. Furthermore, by pairing this database with a persistent NATS JetStream messaging layer, the architecture ensures that high-frequency shop-floor telemetry is buffered and safely committed, preventing transactional loss during temporary network disconnects.

To meet the demanding maintenance requirements of highly non-standard laser welding operations, the database structure is designed to raise the remote resolution rate of system anomalies from a 5% baseline to a target exceeding 20%. While the complete integration of these systems remains an active work in progress, we are currently testing and characterizing the database tables required to store high-frequency telemetry metrics such as laser analogue power outputs, pneumatical air consumption, and chilling water temperatures. This structured, time-series data strategy is designed to decouple local workstation processing from cloud storage, allowing the platform's anomaly detection services to analyze machine health trends remotely. Once fully deployed, this persistent, historical log of device metrics will enable remote experts to identify early indicators of component degradation or optical contamination, allowing them to troubleshoot and resolve system faults remotely without requiring on-site travel.

Finally, the platform's database is designed to compress the mean-time to detect physical weld defects by moving the industry away from slow post-weld destructive testing toward near-instantaneous, in-process anomaly detection. This transition is currently being built around a standardized data representation utilizing the open-source Visualization Toolkit format, which allows the database to store and interpolate both high-fidelity synthetic CFD simulation data and real-time physical sensor signals. While the real-time closed-loop control database pipelines are still undergoing active testing and have not yet been deployed to physical shop floors, our offline database validation has verified that spatial-temporal indexing of coordinate points, timestamps, and field variables successfully supports sub-second queries. This validation demonstrates that our current relational schema is on the right track to provide the rapid, low-latency data access required for Reduced-Order Models to flag process deviations in real time during the upcoming demonstration phases.

7. Exploitation opportunities

FUTURICE: Futurice's exploitation strategy for the Welding Process Analytics Database centers on commercializing the sovereign, multi-tenant data architecture and standardized data schemas as highly scalable assets for the broader manufacturing industry. By establishing a production-ready, open-source relational blueprint using PostgreSQL, Futurice avoids vendor lock-in and can directly license or deploy this data architecture as a foundational product layer for specialized industrial Software-as-a-Service (SaaS) and Manufacturing-as-a-Service (MaaS) solutions. The logical data model itself, including the decoupled sub-models for Materials, Welding Task Specifications, Process Parameters, and Remote Monitoring, represents a reusable, standardized industry language that minimizes structural software re-engineering when onboarding new clients or extending platform functionality to alternative laser processing techniques. A core technical asset ready for commercial exploitation is the plug-and-play Remote Monitoring Pipeline architecture developed under task 5.3. By utilizing a stateless edge-to-cloud data ingestion framework via NATS JetStream, Futurice can package a low-latency, "always-on" messaging engine

capable of processing millions of messages per second with minimal compute overhead. This high-performance pipeline can be commercialized as a standalone containerized solution, allowing industrial users to securely buffer, stream, and log telemetry data without introducing eventual consistency errors or putting proprietary process intellectual property at risk. Furthermore, the specific "connector" design pattern implemented to abstract floor-level variations, such as Siemens IIH Essentials or custom interfaces, enables Futurice to provide cost-effective integration services for customers looking to onboard legacy machinery into cloud-based predictive maintenance and real-time monitoring infrastructures. Looking long-term, Futurice positions itself to capture the strategic transition from centralized SaaS deployments to decentralized, peer-to-peer data spaces. Because the database schema strictly enforces multi-tenancy without database-level dependencies or cross-company data sharing, Futurice is uniquely positioned to exploit the market demand for European Data Space integration. This design allows Futurice to sell standardized translation and connection modules that seamlessly migrate a company's centralized repository into a federated node compliant with Gaia-X and Catena-X governance models. Ultimately, this data strategy enables manufacturing networks to securely automate regulatory compliance for the DPP, unlocking a premium service market where supply chain sustainability and life-cycle inventory metrics are instantly computed and cryptographically signed within the unified workflow.

ECOR: Integrating a digital platform which allows data collection focused on AI driven and monitored analysis and optimization of the production processes offers welding component manufacturers, such as ECOR INTERNATIONAL, a major competitive edge. By minimizing defects, saving on consumables, and optimizing cycle times, AI transforms production from a reactive process into a highly predictable, data-driven operation. Key advantages for ECOR INTERNATIONAL are related to:

- **Predictive maintenance**: AI continuously analyses data such as voltage, current, temperature to detect instant anomalies and comparing data with previously collected ones in order to reduce risk of breakdown of the hardware
- **Flawless Quality Assurance**: Real time monitoring of process parameters allows to ensure parameters stay within the established tolerances. AI instantly corrects deviations reducing scrap rates and, at the same time, it ensures every component to meet the required standards
- **Resource Optimization**: AI analyses every variable to fix the best set of process parameters just to optimize the use of consumables such as shielding gas, power, filler metals if required minimizing operational costs and positively impacting on sustainability
- **Automated Traceability**: the digitalization of the process allows easy access to quality documentation which is a valuable instrument both for suppliers and for customers
- **Shorter setup and R&D Times**: digitalization and AI driven processes allow to drastically reduce the time to market for new products and a higher engineering efficiency

Other aspects which are directly reflected on customers include:

- **Unmatched Precision and Quality**: digitalization and AI driven processes reduces human errors, defect rates and increases repeatability of the welding process
- **Faster production and delivery**: Digital integration and AI production scheduling optimizes phases such as cutting, bending, welding, grinding, passivation and so on, accelerating the overall manufacturing lead time
- **Lower Operational Costs**: Digitalization and automation minimizes raw material waste through optimized process design
- **Total Traceability and Transparency**: Digitalization allows real time monitoring of customers' orders, allowing direct access for data collection regarding all the parameters for the welding process and material/component certifications

FFT: From FFT's perspective, the Architecture of the Lasers4MaaS Platform Database makes collected process and machine data systematically available and usable with the focus on laser applications. By structuring and storing sensor, process, and quality-related data, the database enables the analysis of welding processes and plant behavior. By using these solutions, FFT could improve internal process understanding and accelerate plant optimization. Standardized data models and interfaces could also reduce integration efforts.

8. Outlook

The architectural roadmap for the Lasers4MaaS database is designed to evolve alongside the maturation of the European industrial data ecosystem, transitioning from its current centralized implementation toward a fully federated and sovereign coordination system. While the current centralized structure provides the efficiency and low adoption cost necessary for the project's initial demonstration phase, the long-term vision centers on a decentralized data

architecture aligned with Gaia-X and Catena-X principles. This evolution will ensure that manufacturers can selectively share critical process parameters without exposing the proprietary raw sensor data that constitutes their competitive advantage, thereby resolving the "Capital Trap" where high-value assets remain rigid and underutilized due to data privacy concerns.

As the platform scales toward higher technology readiness levels, the database will move beyond serving as a simple repository to becoming the core engine for an automated "Digital Handshake." This meso-level functionality will leverage the database's multi-tenancy capabilities to securely store and transfer validated process recipes between network partners, condensing weeks of manual physical re-certification into a digital validation timeframe of hours to days. By decoupling the digital "recipe" from the physical machine, the data architecture will enable a "Laser Operating System" that treats hardware as a modular endpoint, allowing manufacturing capacity to be routed dynamically across a resilient, federated service network.

Furthermore, the data model is strategically positioned to serve as a macro-level compliance engine for the upcoming ESPR. Future iterations will automate the population of the DPP by capturing and processing life-cycle inventory data, such as energy consumption and material origins, directly within the manufacturing workflow. This integration will transform the database from a technical log into a strategic tool for circularity, allowing end-customers to automatically verify the sustainability footprint of their components. Ultimately, this data strategy aims to stabilize the industrial grid by facilitating Demand Side Management, where the platform routes high-power laser loads to manufacturing sites with the highest availability of renewable energy, establishing a truly sustainable and decentralised "Factory of the Future."