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 D5.1:	Architecture of the Laser4Maas Digital Platform
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	27/05/2026	Final version

LIST OF AUTHORS

Name	Organization	Email
Denis Washington	FUTURICE	denis.washington@futurice.com
Christoph Nützel	FUTURICE	christoph.nuetzel@futurice.com
Thuy Duong Nguyen	FUTURICE	thuy.duong.nguyen@futurice.com
Tan Henry Duc Nguyen	FUTURICE	henry.nguyen@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. Architecture Goals 9
2.2. Architecture Constraints 10
3. User Research 10
3.1. Business Context 10
3.2. User Personas 11
3.3. User Testing 12
4. Platform Features & User Experience 12
4.1. Process Development Documentation 13
4.1.1. User Flow – Persona: Rinaldo, Process Developer 13
4.2. ML-Based Process Selection & Optimization 14
4.2.1. User Flow – AI-assisted parameter generation (Persona: Rinaldo, Process Developer) 15
4.3. Generative-AI-Based Assistance 16
4.3.1. User Flow - Natural language project search (Persona: Rinaldo, Process Developer) 16
4.4. Sustainability & Cost Evaluation 17
4.4.1. User Flow – Persona: Elena, Account Manager 17
4.5. Remote Monitoring & Predictive Maintenance 18
4.5.1. User flow – Pepe, Laser technician 18
4.6. AR Remote Assistance 19
4.6.1. User flow – Pepe, Laser technician (on-site) and Naomi, Laser expert (remote expert) 20
5. Technical Research 21
5.1. Review of Standard Communication Protocols for APIs 21
5.1.1. HTTP & REST APIs 21
5.1.2. GraphQL 21
5.1.3. gRPC 22
5.2. Review of Communication Security & Authentication Standards 22
5.2.1. TLS (Transport Layer Security) 22
5.2.2. OAuth 2.0 & OpenID Connect 22
6. Platform Architecture 23
6.1. Design Principles 23
6.1.1. Microservices Architecture 23
6.1.2. Prefer Standard Protocols & Formats 23
6.1.3. Open-Source 23
6.1.4. Start Centralized, Prepare for Federation 24
6.2. Platform Components 24
6.2.1. Platform Application 24
6.2.2. Platform Database 24
6.2.3. Remote Monitoring Pipeline 25
6.2.4. Process Selector 26

6.2.5. Process Optimizer 26
6.2.6. Process Controller Code Generator 27
6.2.7. Sustainability & Cost Evaluator 27
6.2.8. AR Remote Assistant 28
6.3. Platform Security & Authentication 28
6.4. Platform Deployment Infrastructure for the Demonstration 29
6.4.1. Scaleway 29
6.4.2. Kubernetes 29
6.4.3. Cloud-Managed Data Services 30
6.4.4. Generative AI Infrastructure 30
7. Current Status 30
8. Impact 30
9. Exploitation potentials 31
10. Outlook 32

LIST OF FIGURES

Figure 1: Overview of the process development process 10
Figure 2: High-level user flow for project creation, experiment creation, and experiment result logging 14
Figure 3: Experiment parameter generation 15
Figure 4: Natural language search using Generative AI 16
Figure 5: Mockup for entering production scenario information for sustainability & cost evaluation 18
Figure 6: UI mockup for the remote monitoring overview 19
Figure 7: Components of the Lasers4MaaS Platform 24
Figure 8: Overview of the Remote Monitoring Pipeline 25
Figure 9: Process Selector inputs and outputs 26
Figure 10: Process Optimizer inputs and outputs 26
Figure 11: Visualization of the iterative Process Optimizer workflow 26
Figure 12: Inputs and outputs of the Process Controller Code Generator 27
Figure 13: Inputs and Outputs of the Sustainability & Cost Evaluator 27
Figure 14: Sub-components of the AR Remote Assistant 28
Figure 15: Overview of the infrastructure used for the demo platform deployment 29

LIST OF TABLES

Table 1: User personas for the Lasers4MaaS Platform 11

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
AR	Augmented Reality
DPP	Digital Product Passport
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
ML	Machine Learning
UI	User Interface
TLS	Transport Layer Security
DBS	Dynamic Beam Shaping
VAE	Variational Autoencoder
ROM	Reduced-Order Model
ESPR	Sustainable Products Regulation
DPP	Digital Product Passport

1. Executive summary

This report describes the design and architecture of Laser4Maas digital platform. It covers the user research and user experience (UX) design performed, explains the architecture's principles and components, and elaborates on the technology choices. Evidence is provided for the following work package tasks:

- **T5.1** (*Review of current cloud services, software interfaces and APIs*), with regards to secure data exchange with and within the platform, mainly in section 5. Additional details can be found in Deliverable 1.3 (*Current standards and approaches for interoperability of data streams*).

- **T5.2** (*Design of the service platform enhanced by generative AI for improved customer experience*), primarily in sections 3 (user research) and 4 (UX design).

2. Introduction, Goals & Constraints

To unleash the full servitisation potential of flexible laser beam welding technologies, a digital backbone is required that connects machines, process development activities, production monitoring and predictive maintenance through secure data exchange within and beyond company boundaries. The Lasers4MaaS digital platform aims to provide such a backbone, making it a key stepping stone towards the CONNECT objective set out by the Lasers4MaaS project (see Deliverable 1.1).

To bridge the fundamental disconnect where identical digital process definitions yield varying physical results on different machines, the platform implements a three-layer architecture powered by AI-based decision modules developed in work package 7. At the micro level of the shop floor, the system achieves hardware-agnostic execution by utilizing DBS to neutralize physical hardware variances, effectively treating the machine as a software-defined endpoint. This is supported by a novel VAE methodology that reduces high-dimensional beam shapes into a compact latent space, enabling efficient real-time optimization and structured parameter exploration. At the meso level, the digital platform systematizes trust via an automated digital handshake. Powered by data-driven reduced-order models developed by TUW, the platform can now predict weld pool geometry with median relative accuracies exceeding 97% and solve inverse parameter identification problems near-instantaneously, removing the requirement for traditional weeks-long manual re-certification and physical trials. Finally, at the macro level of the supply chain, the platform ensures industrial resilience by automating compliance with the Ecodesign for Sustainable Products Regulation through the integration of digital product passport data collection into the unified manufacturing workflow.

Note: Some of the terminology used in this report, including the "Physicality Gap," "Digital Handshake," and the "Laser operating System", is grounded in the project's multi-level interoperability framework. This methodology was peer-reviewed 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).

2.1. Architecture Goals

Concretely, the design and architecture of the platform was driven by the following goals:

- **Seamless integration:** Combine disparate capabilities such as process development documentation, AI assistance for parameter selection, and the remote monitoring of laser welding stations into a single well-integrated user interface, providing manufacturers with end-to-end support for servitised laser welding.

- **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 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[1].

- **Modularity:** Decouple services from the platform's core to lay the foundation for a potential decentralized ecosystem of competing service implementation providers.

2.2. Architecture Constraints

In addition, the design of the Lasers4MaaS platform is shaped by the following constraints and concerns:

- **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:** Manufacturers 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 other companies.

3. User Research

In this section, we summarize the results of the user research we performed as input for the design of the Lasers4MaaS platform. This included interviews, work shadowing sessions, and concept feedback sessions.

3.1. Business Context

In the context of the Lasers4MaaS platform, the two major parties involved in laser welding process development and production are the *client* who needs a part to be welded and the *supplier* who develops and executes the welding process. These do not necessarily need to be different companies, as a vertically integrated manufacturer might do its own welding.

The client specifies the welding task to be performed, including the materials of the parts to weld together, the angle in which they need to be joined, and other details. They also specify the quality standards that the weld must fulfil, usually in reference to international, regional and industry-specific standards. The supplier then uses this information to develop a suitable set of welding parameters, using experiments with material samples and / or component prototypes to test different set points. After each experiment, the weld quality is assessed with various methods, and based on the results the parameters are adjusted for the next experiment until a satisfactory result is reached. This can take several weeks.

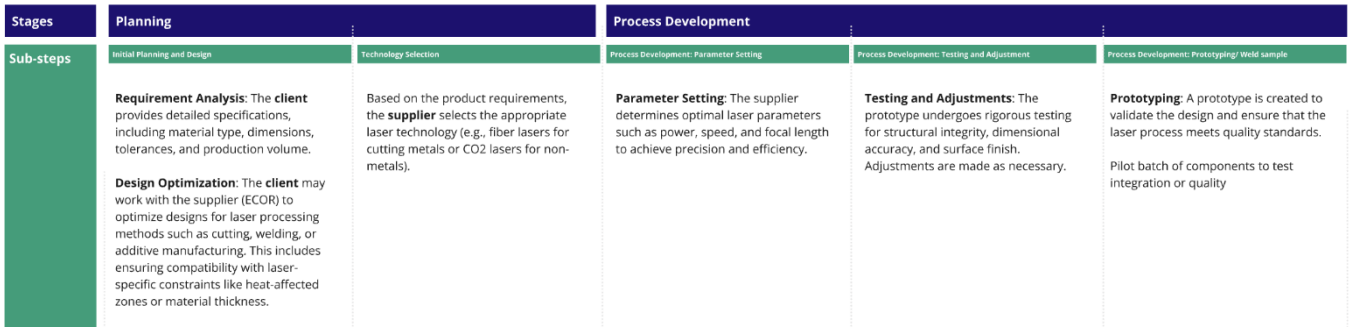


Figure 1: Overview of the process development process

After process development starts, the developed process parameters are used to start production, where the experience gained may lead to further process parameter adjustments.

[1] *Common European data spaces, https://digital-strategy.ec.europa.eu/en/policies/data-spaces*

From the maintenance perspective, the *laser system vendor* is another relevant actor. If the supplier faces issues with a laser, their technicians might need support from the vendor's experts. In non-trivial cases, the laser expert needs to visit the manufacturing site to diagnose the issue, which can significantly delay the laser's repair.

3.2. User Personas

Based on user research and the project members' collective experience, we defined a set of *personas* representing the user groups for the platform. These are detailed in Table *1* below.

Table 1: User personas for the Lasers4MaaS Platform

Party	Name & Role	Description
Supplier	**Rinaldo**, Process Developer	An engineer tasked with developing laser welding processes. Focused on figuring out *how* to make the product. Works primarily in the R&D and setup phases. *Needs:* • Clear requirements from Clients on parts to be produced • Navigating large parameter space and deciding where to go • Starting parameters to begin refining parameters *Pain Points:* • Long process for defining the correct parameters • Time to develop can take up to a year • New laser tech means that we don't have a library of parameters to refer back to *Challenges:* • WPS information must stay highly confidential (company IP) • Costs, delivery time, quality
	Pepe, Laser Technician	Technician working for a supplier. Responsible for **producing parts** and **keeping the machine running**. Works in the execution and operations phase. *Needs:* • Ready-to-use optimized parameters • Clear, early alerts for machine health • Instant support when hardware fails *Pain Points:* • Unexpected machine downtime stops production • Unclear machine errors • Waiting for a machine expert to travel on-site *Challenges:* • Meeting production quotas • Keeping the machine running safely • Maintaining consistent production quality

	Elena, Account Manager	Direct contact to the client. Responsible for consulting, costs, quality, and sales. *Needs:* • Clear requirements from Clients on parts to be produced • Managing costs, quality, and sustainability *Pain Points:* • Lack of insight into how process parameters and other factors affect cost and sustainability metrics *Challenges:* • Costs in process development: running cost for quality • Equipment cost: initial investment
Laser Vendor	**Naomi**, Laser Expert	An expert whose job includes assisting the laser system vendor's customers with technical problems. *Needs:* • Detailed information about the issue experienced by the customer • Direct feedback on whether an action fixed the problem, especially if they cannot be present physically *Pain Points:* • Non-trivial problems often require travelling to the customer site to diagnose and fix the problem Challenges: • Diagnosing problems remotely without being able to see or touch the machine directly • Communicating fixes clearly to someone who may not have deep laser expertise

Note that the supplier's clients are not direct users of the platform in its initial version, and do thus not appear as a user persona above. It is however conceivable that a future version may include them; for instance, the platform might allow receiving generated Digital Product Passports (DPP) or aggregate remote monitoring statistics (such as defect rates) from their suppliers through an API.

3.3. User Testing

Beyond the initial research phase, the platform's user interface has been validated through an ongoing testing throughout the project. New feature sets have been tested and refined in dedicated weekly feedback sessions with representatives of the project consortium. A first consolidated release (v0.5) of the platform was shared with the full project consortium for structured feedback, gathering input from the broader group of domain experts and industrial partners involved in the project. The insights gathered across these sessions have been continuously fed back into the design, driving iterations of the user interface toward the designs presented in section 4. As the platform matures, testing is planned to expand beyond the consortium to include a wider set of industry users, further broadening the range of perspectives informing the final design.

4. Platform Features & User Experience

This section gives an overview of the functionality that will be integrated into the Lasers4MaaS platform within the scope of this project, as well as the platform user experience.

4.1. Process Development Documentation

The platform will allow companies developing laser welding processes (usually as part of producing parts supplied to their clients) to consistently document their process development projects, including welding task specifications, quality requirements, process constraints, and the results of the experiments performed. This database does not only facilitate knowledge sharing and reuse within the organisation, but additionally forms the basis for several other platform features.

4.1.1. User Flow – Persona: Rinaldo, Process Developer

1. Rinaldo opens the platform and navigates to the Projects dashboard, which lists all existing process development projects.

2. He clicks *Create New Project* and fills the welding task specifications, defines quality requirements and sets the welding station and project constraints.

3. The platform creates the project and presents the project details and the experiment list, initially empty.

4. Rinaldo clicks *Create New Experiment* and fills the welding parameters he intends to test (or uses AI-assisted generation – see section 4.2).

5. After running the experiment, Rinaldo logs the measured quality results against each quality indicator previously defined.

6. The platform stores the experiment record and updates the project's experiment history.

7. For subsequent experiments, Rinaldo repeats steps 4-6. The platform pre-fills the form from the previous experiment and optionally generates optimized suggestions (see section 4.2).

8. Once satisfactory quality is achieved, the completed project serves as a documented, searchable record reusable for future tasks with similar specifications.

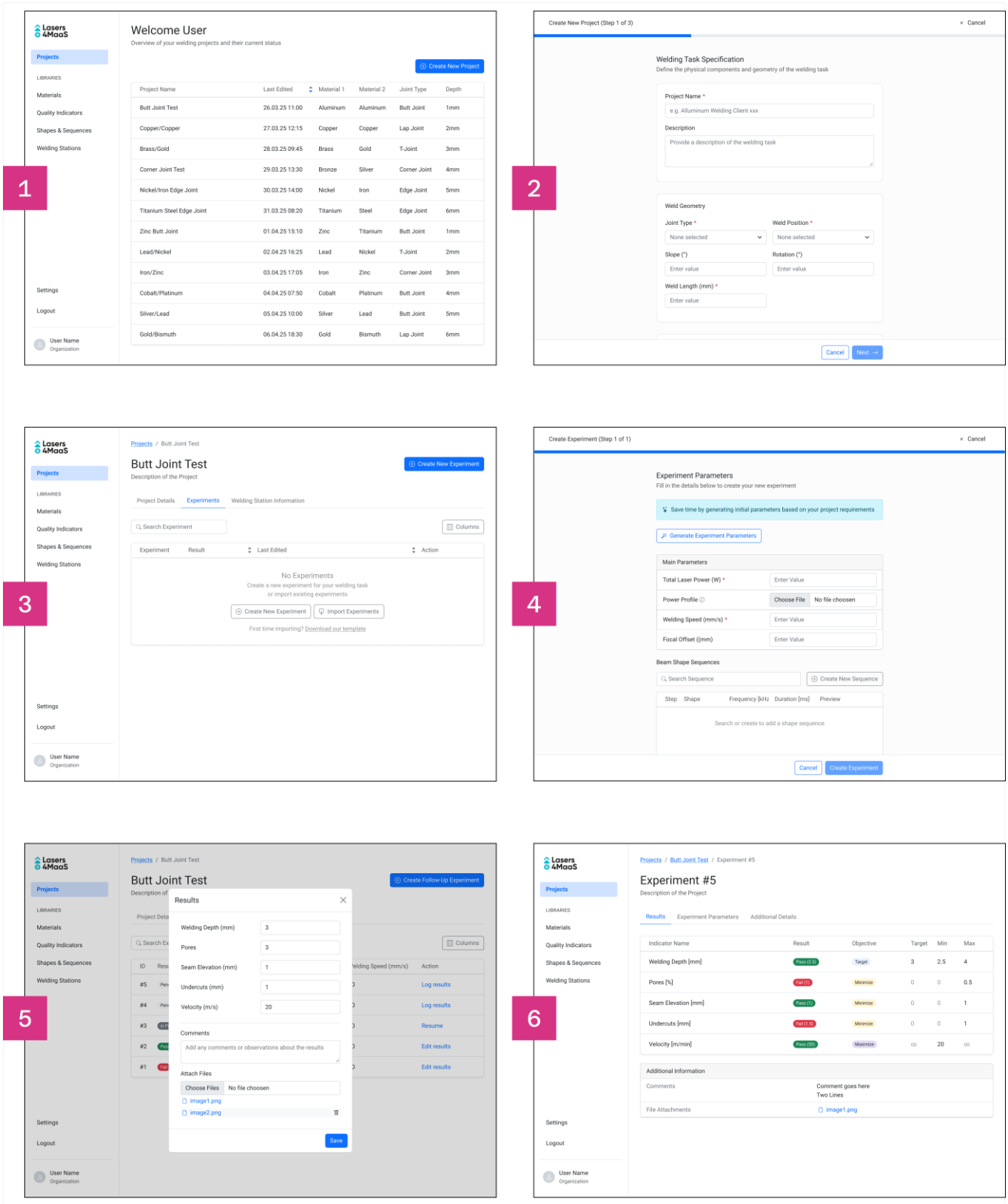


Figure 2: High-level user flow for project creation, experiment creation, and experiment result logging

4.2. ML-Based Process Selection & Optimization

To speed up laser welding process development, the Laser4MaaS platform will provide process developers with the ability to suggest initial process parameters for a given welding task, as well as optimize these parameters based on previous experiments in a project. This Machine-Learning-augmented approach aims to significantly cut the time needed to discover optimal parameters for a given welding task.

4.2.1. User Flow – AI-assisted parameter generation (Persona: Rinaldo, Process Developer)

1. When creating the first experiment in a project, Rinaldo clicks *Generate Experiment Parameters* instead of filling in parameter fields manually.
2. The platform sends the welding task specifications to the Process Selector service.
3. The Process Selector computes a starting parameter set and returns it to the platform.
4. The platform pre-fills the experiment form with the suggested values.
5. Rinaldo reviews the suggestions, makes any adjustments he sees fit, and saves the experiment.
6. After running the experiment and logging the quality results, Rinaldo creates a follow-up experiment.
7. This time, the platform sends the full experiment history (parameter sets and measured quality outcomes) to the Process Optimizer service.
8. The platform pre-fills the follow-up experiment form with the optimized parameter suggestions.
9. Rinaldo repeats steps 6–8 across subsequent experiments, with each round producing progressively better suggestions as more experimental data accumulates.

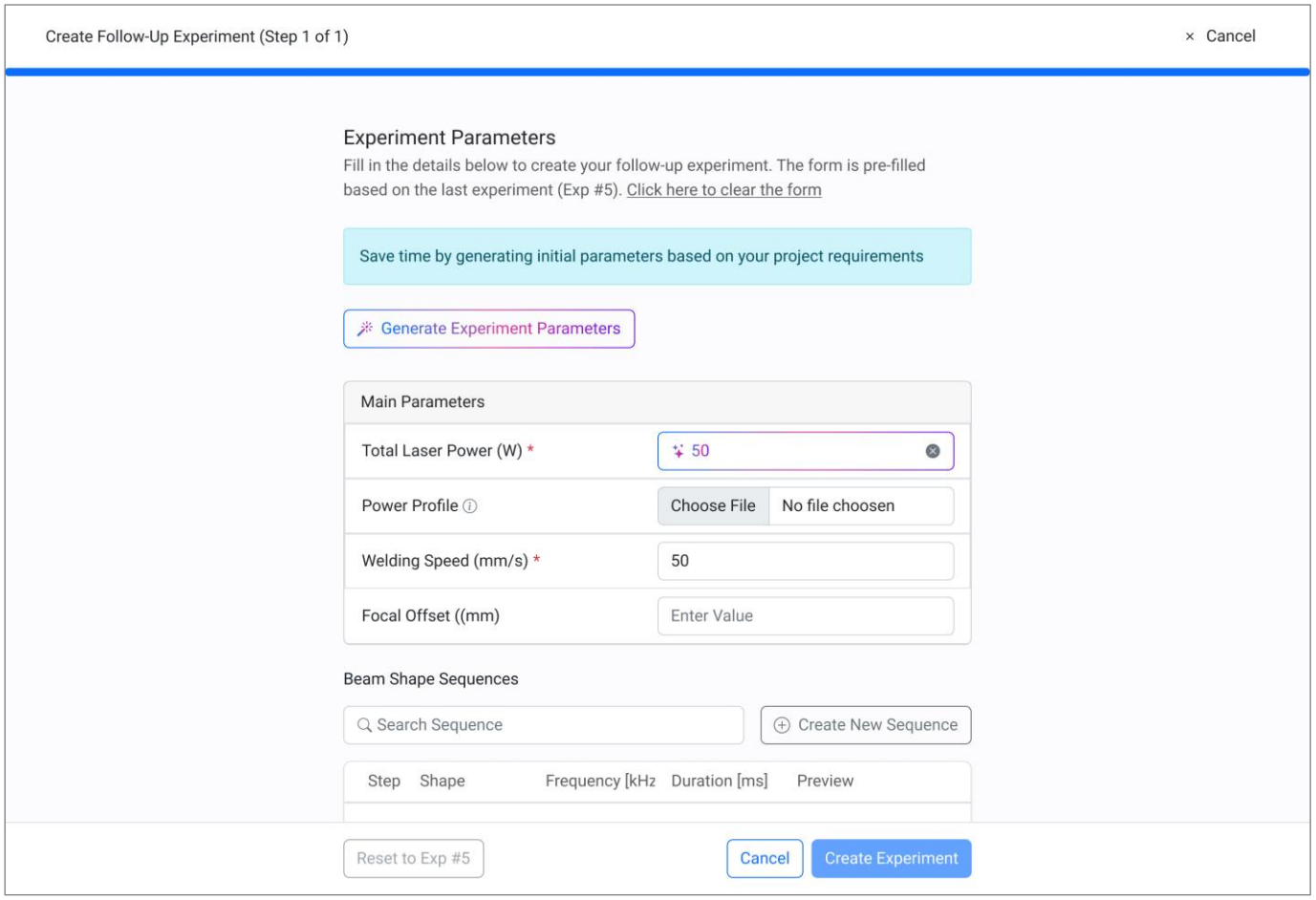


Figure 3: Experiment parameter generation

4.3. Generative-AI-Based Assistance

To further enhance the process development process, the platform will leverage Generative AI (GenAI) models. For the platform's initial version, the focus will be on surfacing knowledge from the company's process database for new welding tasks, for instance by allowing users to search for similar past projects with natural language queries.

4.3.1. User Flow - Natural language project search (Persona: Rinaldo, Process Developer)

1. When starting a new project or experiment, Rinaldo types a natural language query into the platform's AI-powered search bar for example, "show projects from last year where zinc was used".
2. The platform passes the query to the GenAI layer, which interprets the intent and maps it to the structured data in the process database.
3. The platform returns a ranked list of matching projects, displaying key metadata for each result.
4. Rinaldo selects a relevant past project and inspects its experiment history: parameters tested, quality results achieved, and notes recorded.
5. He uses this information to inform his starting point for the current project, avoiding duplication of effort and leveraging accumulated organisational knowledge.

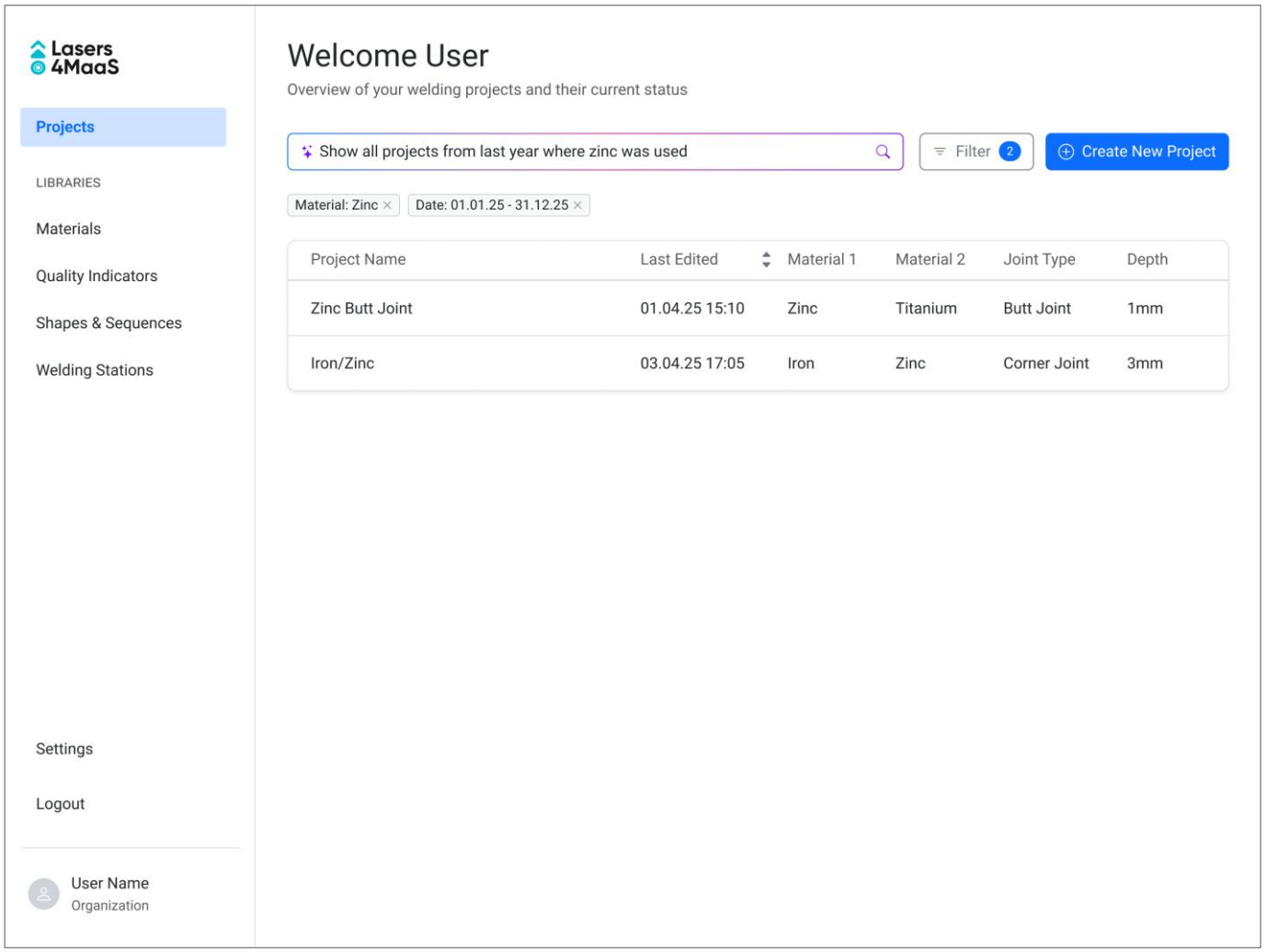


Figure 4: Natural language search using Generative AI

4.4. Sustainability & Cost Evaluation

In addition to weld quality, the Lasers4MaaS Platform will also support organizations in evaluating the cost and ecological impact of the developed welding processes in different scenarios (e.g., choice of manufacturing site). This makes it easier to factor in cost and sustainability considerations when choosing among welding processes. It also provides a stepping stone towards Digital Product Passports (DPPs)[2], which will require companies to make the sustainability footprint of their products transparent.

4.4.1. User Flow – Persona: Elena, Account Manager

1. Elena opens a completed or in-progress process development project and initiates the *Calculate Impact* flow.
2. She defines a production scenario by selecting the production site and specifying the process chain, including pre-processing and post-processing steps relevant to the part being produced.
3. She provides material and logistics information.
4. The platform submits the scenario to the Sustainability & Cost Evaluator service.
5. The service computes estimates for energy consumption, CO_2 emissions, material waste, and unit cost, and returns the results to the platform.
6. The platform presents the results to Elena in a structured summary.
7. Elena defines a second scenario, for example, the same process at a different factory location with a different energy mix and repeats steps 2–6.
8. She compares the scenarios side by side to identify the most cost-effective and sustainable option.
9. The resulting sustainability metrics are stored in the project record and can be used as input toward Digital Product Passport generation.

[2] *EU's Digital Product Passport: Advancing transparency and sustainability*, https://data.europa.eu/en/news-events/news/eus-digital-product-passport-advancing-transparency-and-sustainability

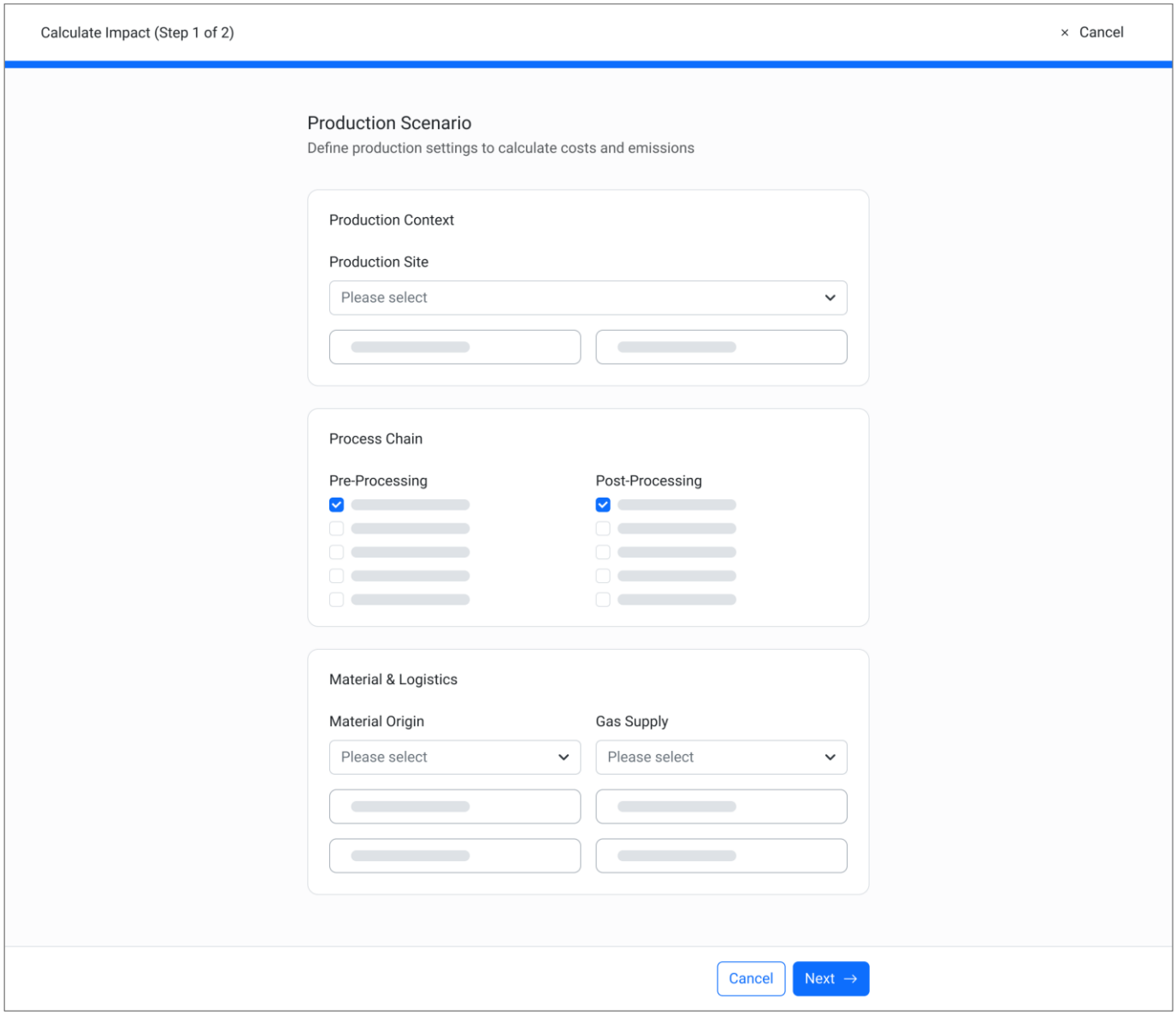


Figure 5: Mockup for entering production scenario information for sustainability & cost evaluation

4.5. Remote Monitoring & Predictive Maintenance

The Lasers4MaaS Digital Platform will give laser technicians insights into the weld quality and device health of their factories' laser welding stations from anywhere. To do so, it will provide the means to continuously ingest sensor and trace data collected at the manufacturing site and transfer it to the platform, where it will be made available to users through a dashboard-style interface.

Additionally, the platform will analyze device-health-related sensor readings to detect anomalies that serve as early warning signs for potential issues, allowing operators to predict maintenance needs and prevent downtime.

4.5.1. User flow – Pepe, Laser technician

1. Pepe opens the platform's Live Monitoring dashboard from any location.
2. The dashboard shows all registered welding stations with their status and any active predictive alerts.

3. The platform continuously ingests sensor and trace data from the shop floor via the Remote Monitoring Pipeline and updates the dashboard in real time.

4. When the anomaly detection model identifies an unusual sensor reading, the platform raises a predictive warning.

5. Pepe clicks on the affected station to open its detail view, where he can inspect remote monitoring data that triggered the alert.

6. Based on the diagnostic information, Pepe schedules a maintenance intervention proactively, before a breakdown occurs.

7. After the intervention, the alert is resolved and the station returns to a normal status in the dashboard.

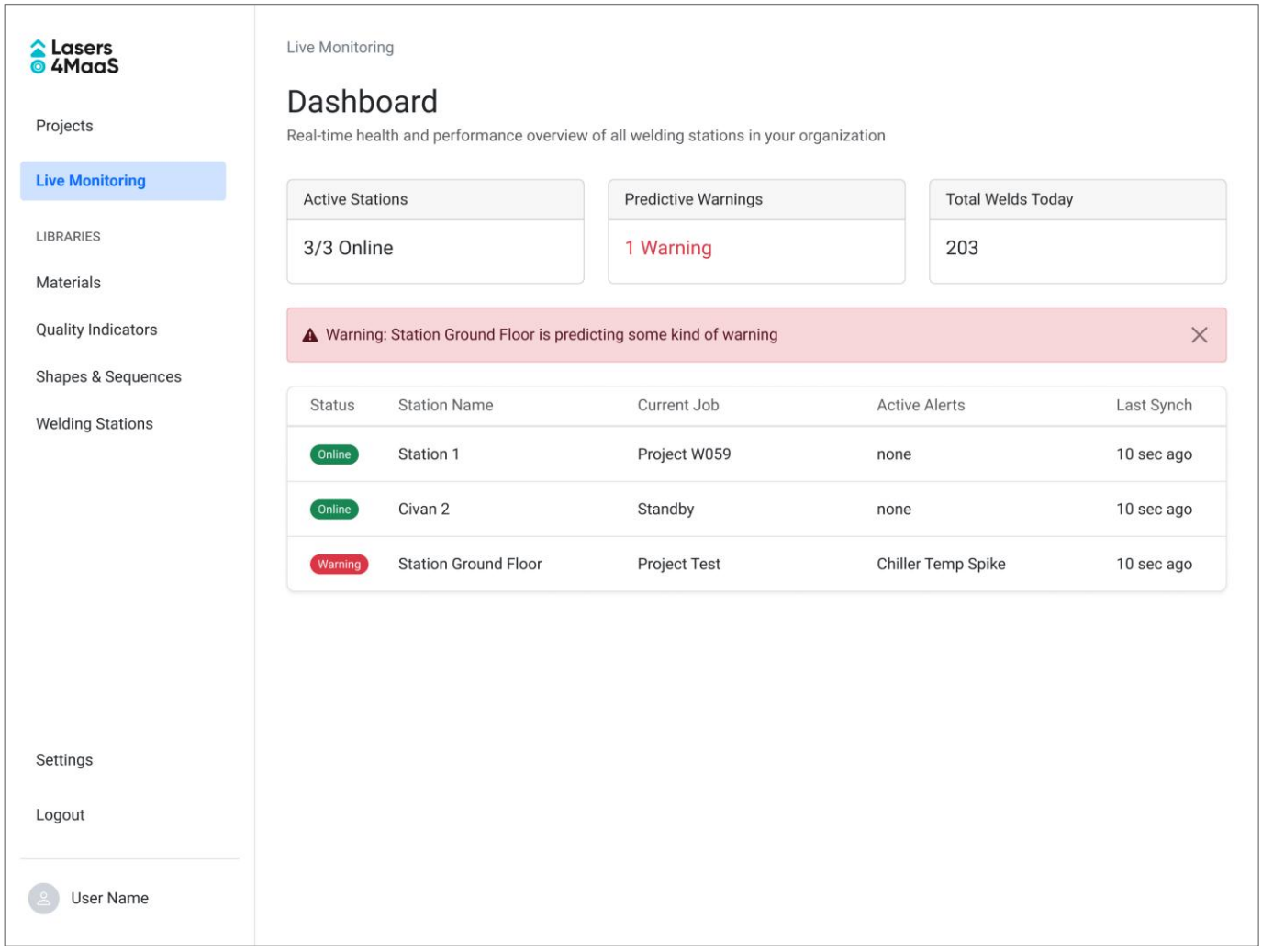


Figure 6: UI mockup for the remote monitoring overview

4.6. AR Remote Assistance

The platform will enable real-time collaboration between on-site laser technicians and remote experts during laser maintenance, repair, and system adjustment tasks. It leverages Augmented Reality (AR) glasses equipped with spatial mapping and video streaming capabilities to bridge the gap between physical operations and remote expertise.

The system focuses on enhancing communication and precision in troubleshooting without relying on full CAD-based visualization and uses the live spatial mapping from the AR device instead, meaning that it does not require complex setup procedures to be used at a particular laser installation.

4.6.1. User flow – Pepe, Laser technician (on-site) and Naomi, Laser expert (remote expert)

1. Pepe encounters a hardware issue he cannot resolve alone and opens the AR Remote Assistance feature in the platform.

2. He puts on the AR glasses and initiates a new remote assistance session.

3. The glasses begin capturing a live video feed and 3D spatial map of the workspace and transmit both to the platform.

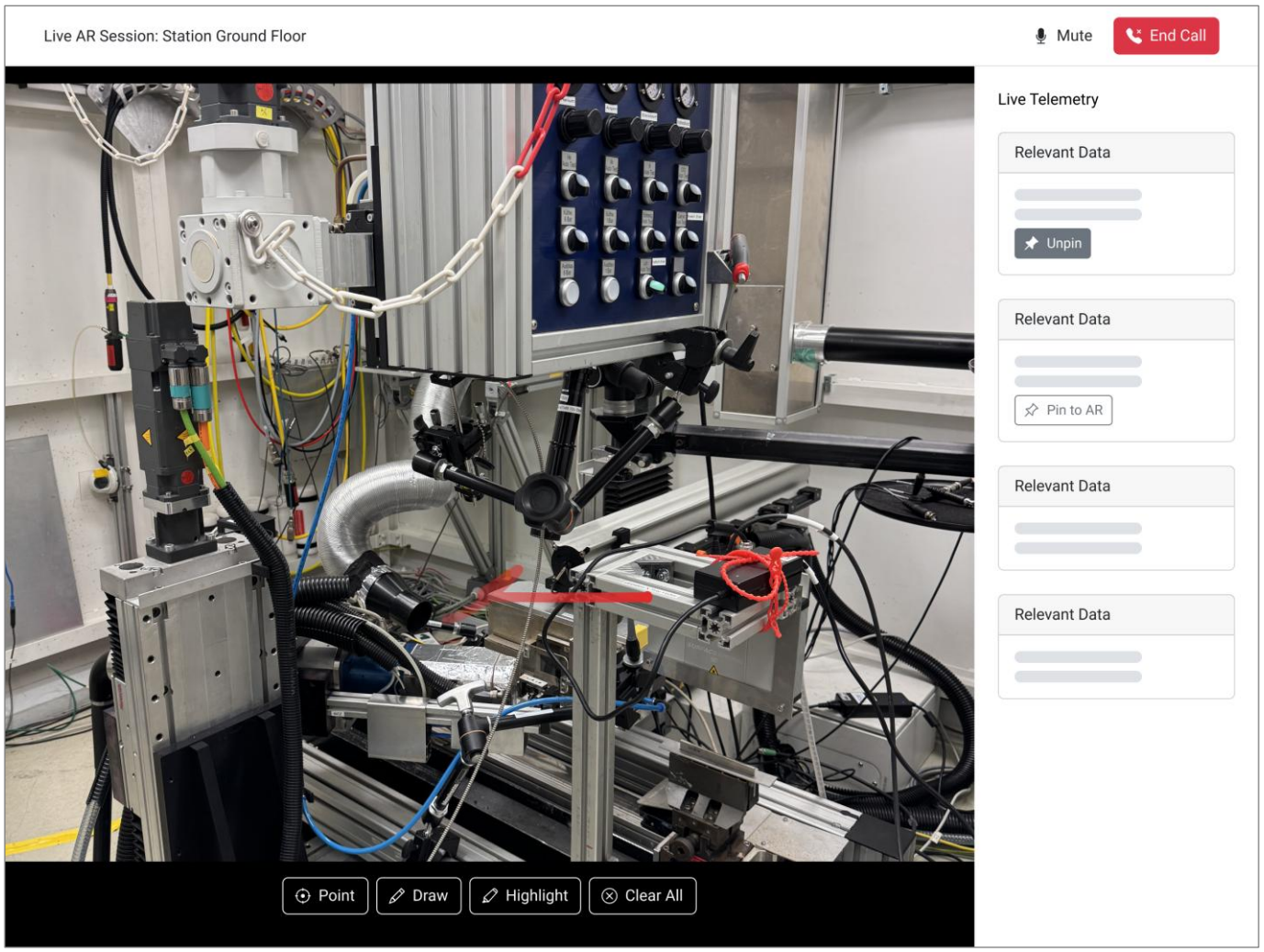
4. The platform notifies Naomi, who joins the session through the platform's web interface on her remote computer.

5. Naomi sees Pepe's live first-person video feed and uses annotation tools to mark specific components she wants Pepe to attend to.

6. Pepe sees the markers in his field of view and follows Naomi's guidance, keeping both hands free for the repair task.

7. Naomi and Pepe communicate over the live audio channel throughout, with Naomi able to update or clear annotations in real time as the repair progresses.

8. Once the issue is resolved, Pepe ends the session from the glasses and both parties are returned to the platform.



5. Technical Research

This section summarizes some of the research done as part of Task 5.1 to prepare the design of the Lasers4MaaS platform architecture. More can be found in Deliverable 5.2, section 3.

5.1. Review of Standard Communication Protocols for APIs

As the Lasers4MaaS platform is meant to be modular, it is critical to ensure data exchange between the platform's components is reliable and happens through clearly defined application programming interfaces (APIs). Also, APIs exposed by the platform to external systems should be based on standard communication protocols that are easy to integrate. This sections explores common options for such protocols.

5.1.1. HTTP & REST APIs

HTTP (Hypertext Transfer Protocol) is the foundational application-layer protocol that powers data communication across the World Wide Web. Operating on a client-server model, it dictates how messages—such as requests for web pages and their subsequent responses—are formatted and transmitted across the internet. Originally designed for fetching simple HTML documents, HTTP has evolved through versions like HTTP/1.1 and the multiplexed HTTP/2 to efficiently transport diverse data payloads, including images, video, and structured text. It is inherently stateless, meaning each request is treated as an independent transaction, though statefulness is often simulated using mechanisms like cookies and tokens.

REST (Representational State Transfer) is an architectural style that leverages the existing capabilities of HTTP to design web services, commonly known as REST APIs. Instead of inventing new standards, REST relies on native HTTP components, utilizing standardized methods like GET, POST, PUT, and DELETE to perform operations on specific "resources" identified by unique URLs. A core principle of REST is that it uses standard, human-readable data formats – most commonly JSON – to transfer the representation of a resource's state between the client and the server.

Because of its simplicity, stateless nature, and seamless alignment with standard web infrastructure, REST has become the dominant convention for building public APIs and connecting software services. Many systems in the industrial space, including modern Manufacturing Execution Systems (MES), also offer REST APIs and integration points.

One downside of REST APIs is that their resource-based approach can make data fetching inefficient. Depending on how the API client's data needs match the way that the API segments data into resources, clients can suffer from "over-fetching" (receiving massive JSON payloads with unnecessary data fields) or "under-fetching" (not receiving enough data in a single API request, forcing the client to make multiple requests to different URLs). Another issue is that HTTP does not allow bidirectional communication or streaming, limiting its applicability.

5.1.2. GraphQL

GraphQL is an open-source data query and manipulation protocol designed to overcome the data-fetching inefficiencies of traditional REST APIs. It avoids over-fetching and under-fetching by doing away with predefined resource endpoints and instead letting clients send precise data queries to the server, using GraphQL's own query language. For each query, the server compiles exactly the requested data and sends it to the client in a single response. This approach makes GraphQL an attractive choice for serving clients with diverse and quickly evolving data needs, such as web applications which combine data in various ways depending on the page.

GraphQL also offers *mutations* to send commands to a GraphQL server, and *subscriptions*, which allow the server to push real-time data updates to the client, solving HTTP's lack of bidirectional communication.

Compared to REST APIs, GraphQL's adoption is low and is mostly concentrated around frontend-to-backend communication – which is what Meta originally designed it for in the context of its Facebook applications – rather than backend service integration, where the protocol's benefits are outweighed by the much larger industry support for REST APIs.

5.1.3. gRPC

gRPC is a modern, open-source, high-performance Remote Procedure Call (RPC) framework that enables client and server applications to communicate transparently. Originally developed by Google, it leverages Protocol Buffers (Protobuf) as its Interface Definition Language (IDL) and underlying message interchange format, providing efficient and type-safe communication.

gRPC's use of Protobuf and HTTP/2 – both based on optimized binary payloads rather than human-readable text like JSON and classic HTTP 1.1 – results in efficient and low-latency communication. The protocol also supports bidirectional streaming, allowing for real-time data exchange and asynchronous communication patterns. Tools for generating client and server code from gRPC interface definitions exist for most popular programming languages, simplifying development and enabling interoperability.

While there are several companies that heavily rely on gRPC for the internal communication between their backend services, its use less common for general externally facing APIs. This is partly explained by the protocol's reliance on HTTP/2, which is much less widely deployed and supported than the dominant HTTP 1.1 protocol. Also, the need to generate client code before accessing an gRPC interface introduces friction and integration challenges that do not exist for REST APIs.

5.2. Review of Communication Security & Authentication Standards

To protect confidentials company data such as process parameters, secure communication within and beyond the platform is essentials. To this end, the following sub-sections surveys relevant security-related standard protocols.

5.2.1. TLS (Transport Layer Security)

TLS is a cryptographic protocol designed to provide secure communication over a computer network, standardized by the Internet Engineering Task Force (IETF). It is widely used to secure web traffic (HTTPS) and other internet-based communications. TLS ensures data confidentiality, integrity, and authenticity through encryption, digital signatures, and certificate-based authentication.

TLS is employed by many application-level communication protocols to provide secure communication channels, including those explored above. By using TLS to encrypting data in transit, they protect sensitive information from unauthorized access and tampering. In addition, they can leverage the TLS certificate infrastructure to provide a first level of authentication, allowing applications to restrict communication to peers they trust, either directly or through a trusted certificate authority. For instance, web browsers ship with a set of "root certificates" from reputable certificate authorities and reject HTTPS connections to websites whose certificate cannot be traced to any of those roots through a "chain of trust" (by virtue of being digitally signed by the root authority or one of its trusted delegates).

5.2.2. OAuth 2.0 & OpenID Connect

OAuth 2.0 is an industry-standard framework designed to grant applications limited access to HTTP resources on behalf of a resource owner. Rather than requiring the owner to expose their credentials, OAuth 2.0 introduces an authorization layer that separates the role of the client from the resource owner. Access is mediated through the issuance of a string-based access token—typically a JSON Web Token (JWT) or an opaque string—by an authorization server. This token encapsulates specific access permissions (scopes) and lifetimes, allowing resource servers to validate permissions independently and securely without exposing sensitive user passwords.

OpenID Connect (OIDC) extends the OAuth 2.0 framework with a formalized identity layer for user authentication, replacing earlier non-interoperable ad-hoc approaches. It introduces an ID token alongside the standard access token, structured as a cryptographically signed JWT. This ID token contains standard claims about the identity of the authenticated user, allowing client applications to securely verify who signed in. OpenID Connect also fills other gaps in the OAuth 2.0 specification, such as a standard mechanism for clients to discover an authorization server's capabilities and the URLs of the endpoints providing them. These additions make OpenID Connect significantly more interoperable than pure OAuth 2.0.

OpenID Connect is implemented by many major identity and access management (IAM) systems, often as the primary integration mechanism. This includes cloud-based single-sign-on services like Okta Identity Cloud and

Microsoft Entra ID (formerly known as Microsoft Azure Active Directory), as well as open-source solutions such as Keycloak.

6. Platform Architecture

This section describes the technical architecture of the Lasers4MaaS platform, including its principles, components, and major technology choices.

6.1. Design Principles

The core principles and decisions guiding the Lasers4MaaS platform's architecture are summarized below. This provides useful context for the details described in the next sections.

6.1.1. Microservices Architecture

The Microservices architectural style promotes the componentization of a system's capabilities into separately deployable services with well-defined interfaces.[3] This approach has several benefits, among them the possibility for services to be independently developed by separate teams using diverse tech stacks. It also allows a service implementation to be, in principle, replaced with another implementation confirming to the same interface.

These attributes make microservices a good fit for the Lasers4MaaS platform's Modularity goal (see section 2). Accordingly, much of the platform's functionality is delegated to a set of services implemented by different project partners (see section 6.2). The platform's core communicates with these services through HTTP-based APIs, coordinating them to create a unified user experience. This approach avoids inter-service dependencies, making the overall system easier to understand and individual service implementations simpler to exchange.

6.1.2. Prefer Standard Protocols & Formats

To maximize interoperability, the platform makes use of industry-standard communication protocols and data formats where available. This includes the following:

- **REST for platform service APIs:** Platform services will expose their capabilities through HTTP APIs with JSON input and outputs, following REST conventions (see section 5.1.1). The APIs will be formally specified using the OpenAPI[4] standard, ensuring they are well-documented and understood by OpenAPI-supporting tools, such as API documentation viewers and client code generators.

- **OpenID Connect for API request authentication:** Instead of inventing custom authentication and authorization mechanisms, we will leverage OpenID Connect access tokens to authenticate API requests made to platform components. This enables interoperability with any standards-compliant identity and access management platform.

In cases where no widely adopted communication standard exists, an abstraction layer is introduced that encapsulates the differences between vendor-specific protocols and formats. This makes it possible to support new integration scenarios by simply adding another implementation of the abstraction layer while leaving the rest of the platform unchanged. An example of this approach is the Remote Monitoring Pipeline, whose "connector" abstraction allows trace and sensor data to be ingested from different data sources (see section 6.2.3).

6.1.3. Open-Source

The Lasers4MaaS platform core is built exclusively on top of open-source programming languages, libraries, frameworks and infrastructure components. This avoids vendor lock-in, enabling deployment to almost any on-premise or cloud environment. The same is true for the service implementations created as part of this project.

[3] *James Lewis and Martin Fowler: Microservices, https://martinfowler.com/articles/microservices.html*
[4] *OpenAPI, https://www.openapis.org/*

6.1.4. Start Centralized, Prepare for Federation

As discussed in Deliverable 5.2, the initial version of the platform is deployed as a centralized, multi-tenant instance shared between the companies, in order to focus our efforts on the unique contributions of the Lasers4MaaS project. That being said, architectural dependencies on this centralized model are avoided where possible. For instance, the platform does not allow data sharing between companies on the database level, but requires data export and import, as if there were a separate Platform Database instance for each company. With this approach, we believe it should be possible to transition the platform to a federated model, such as a Data Space, without major re-architecting of the platform's foundations.

6.2. Platform Components

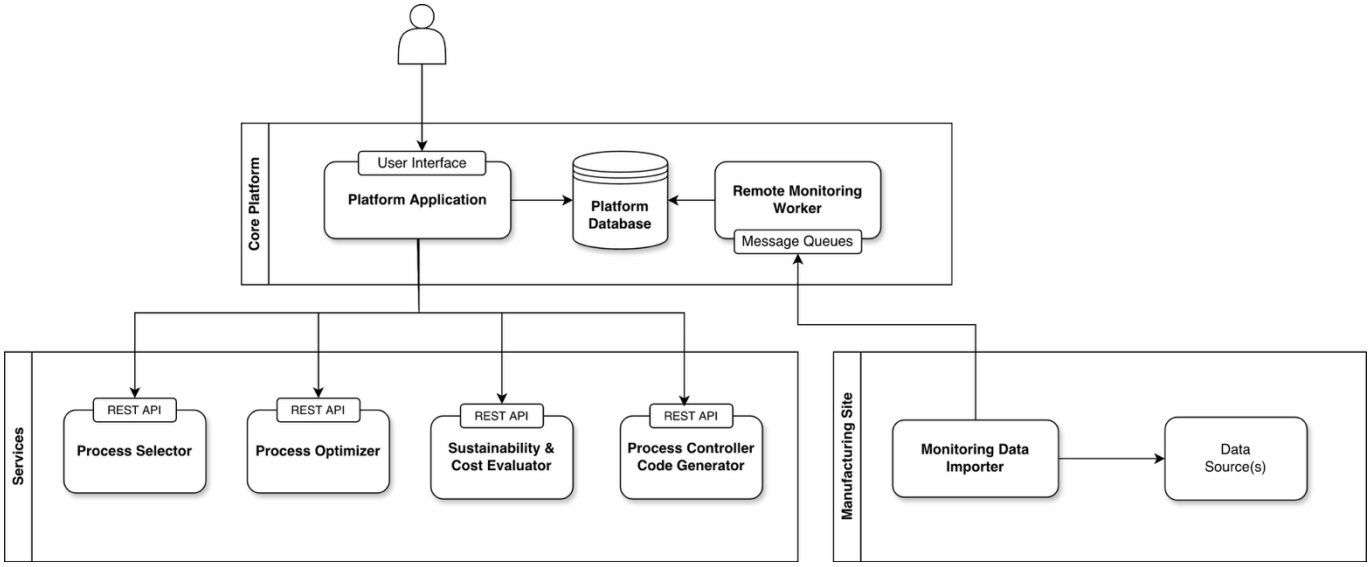


Figure 7: Components of the Lasers4MaaS Platform

The Lasers4MaaS platform consists of a set of core components – summarily called the *core platform* – connected to several services and the platform users' manufacturing sites (see Figure 7). In the core platform sits the *Platform Application*, which integrates the other components into a unified user interface, as well as the *Platform Database* for the platform's data storage needs and the *Remote Monitoring Pipeline* for ingesting monitoring data from the shop floor.

The platform's service layer implements several of the platform's capabilities as a set of independently deployed microservices (see section 6.1.1). The services are not aware of each other or most of the core platform; they simply implement their specified service interfaces, and the Platform Application mediates between them and the Platform Database.

The following sub-sections provide explanations for each of the components.

6.2.1. Platform Application

The Platform Application sits at the heart of the Lasers4MaaS platform. It serves the platform's Web user interface, making it the window to the platform's users. It also integrates the platform services, calling out to them as needed during the platform user flows.

The application is written in Python and uses the Django web framework as its foundation.

6.2.2. Platform Database

The Platform Database is the platform's primary storage layer. It contains all data entered into or collected by the platform, including welding task specifications, process parameters, and monitoring data produced by the connected laser welding stations.

More details on the Platform Database are given in Deliverable 5.2 (*Architecture of the Lasers4MaaS Database*).

6.2.3. Remote Monitoring Pipeline

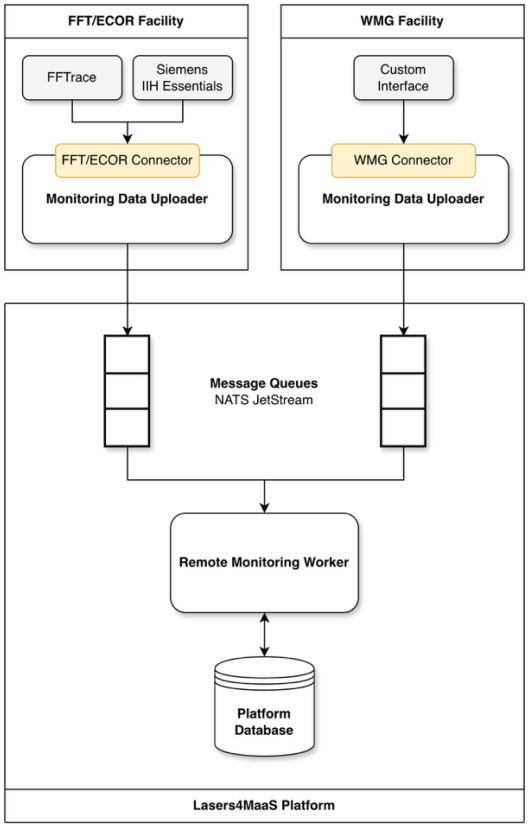


Figure 8: Overview of the Remote Monitoring Pipeline

To support the remote monitoring and predictive maintenance features outlined in section 4.5, monitoring data needs to be transferred from the laser welding station on the shop floor to the Platform Database, and this data needs to be analyzed in real time to detect anomalies that may predict a need for maintenance.

The Remote Monitoring Pipeline fulfils both of these responsibilities. It consists of the *Monitoring Data Uploader*, which is deployed on-premise in the factory, and the *Remote Monitoring Worker*, which is co-located with the Platform Database. The former regularly pushes monitoring data to welding-station-specific message queues provided by the platform, which are implemented as NATS[5] streams; the latter consumes these queues, streaming the data to the Platform Database and performing anomaly detection.

More details on the Monitoring Data Uploader can be found in Deliverables 5.2 (*Architecture of the Lasers4MaaS Platform Database*) and 11.1 (*IIoT sensors selection and characterisation*). The latter also covers how the platform is specifically integrated into the demonstration areas set up as part of this project.

Note that the Platform Application does not directly interact with the Remote Monitoring Pipeline; it simply presents the monitoring data populated by the Remote Monitoring Worker in the background. The Platform Database is thus the implicit integration point, removing the need for an explicit API.

[5] *NATS, https://nats.io/*

6.2.4. Process Selector

Figure 9: Process Selector inputs and outputs

The Process Selector service provides platform users with starting parameters for performing a welding task, given the task specification, quality requirements, and parameter constraints such as the maximum power supported by the target laser system. Its implementation is based on a reduced-order Machine Learning model based on high-fidelity simulations.

The methods and algorithms behind the Process Selector's implementation will be further described in Deliverable 10.1 ("Methods for reduced order modelling with implementation and calibration").

6.2.5. Process Optimizer

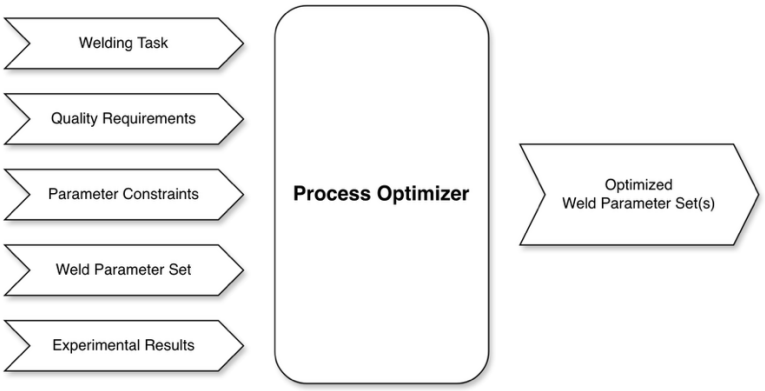


Figure 10: Process Optimizer inputs and outputs

The Process Optimizer service can incrementally optimize welding parameters over a series of process development experiments. Given a welding task specification (including quality requirements and parameter constraints) and an history of tested welding parameter sets with corresponding quality results (measured indicator values), the Process Optimizer one or more suggestions for optimized parameter sets which can be tested in the next experiment. This allows developers to iteratively improve the process in a guided manner.

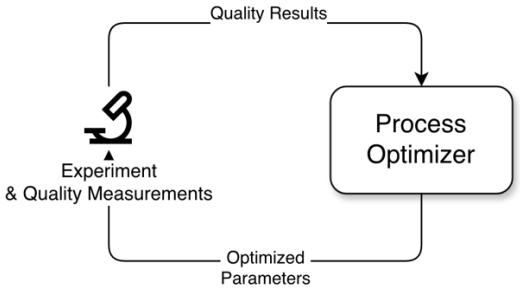


Figure 11: Visualization of the iterative Process Optimizer workflow

Like Process Selector, the Process Optimizer's implementation uses Machine Learning for its suggestions. For each welding task, it uses Bayesian optimization to adjust its suggestions after each reported experiment, guided by a cost function derived from the welding task's quality requirements.

Deliverable 7.1 (*Algorithm for process synthesis, virtual sensors and process control, fast process development*) will explain the Process Optimizer's implementation and the algorithm behind it in detail.

6.2.6. Process Controller Code Generator

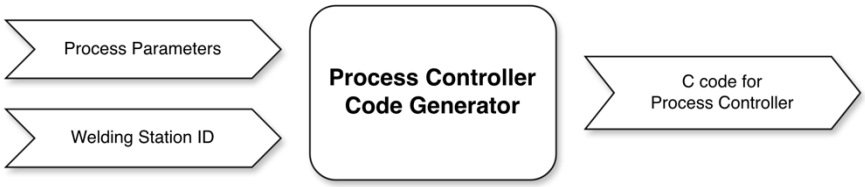


Figure 12: Inputs and outputs of the Process Controller Code Generator

The Process Controller is a programmable hardware device that can dynamically adjust process parameters in during welding in order to compensate variances in part-to-part gap, material imperfections, and other sources of process uncertainty. It does so by translating inputs from sensors installed at the welding station to the parameter adjustments required to maintain the given targets, using a pre-trained simulation model.

The Process Controller Code Generator service provides the code to program the Process Controller hardware. Given target values for the process parameters controlled by the device, as well as the station where the welding is being performed, the service outputs a C code file that can be compiled and flashed on the controller.

A more detailed description of the Process Controller will be given in Deliverable 8.1 (*Trained and calibrated algorithm for process synthesis, virtual sensors and process control, fast process development*).

6.2.7. Sustainability & Cost Evaluator

Figure 13: Inputs and Outputs of the Sustainability & Cost Evaluator

The Sustainability & Cost Evaluator service offers estimates for the resource usage, and environmental impact, and cost of a laser welding process within a particular context, such as the laser system being used, the location of the factory where production will happen, and other contextual information.

Detailed information on the Sustainability & Cost Evaluator's interface and algorithms will be given in Deliverable 14.1 (*Report on LCA and cost evaluation validated in lab with decision support system*).

6.2.8. AR Remote Assistant

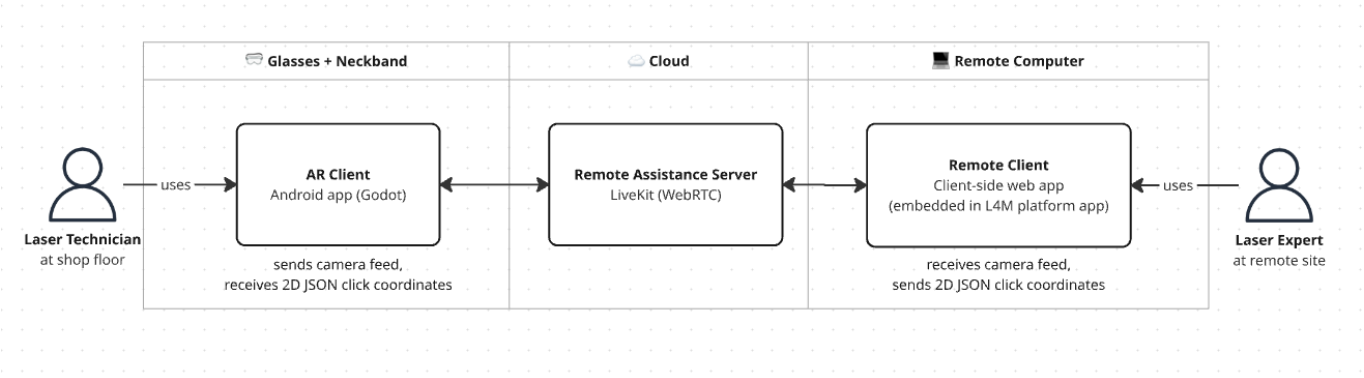


Figure 14: Sub-components of the AR Remote Assistant

The AR Remote Assistant implements the AR Remote Assistance feature (section 4.6). It consists of three collaborating sub-components:

- **AR Client:** The application used by the laser technician at the welding station. It is directly connected to the AR glasses worn by the technician, using it for displaying its user interface and capturing the glasses' camera feed, which is streamed to the Remote Client via the Remote Assistance Server.

- **Remote Client:** The web application used by the laser expert. It will be embedded into the user interface of the Platform Application. It displays the AR camera feed received from the AR Client and allows the laser expert to set markers in it by clicking on a point in the camera image, which results in a corresponding command to be sent to the AR Client.

- **Remote Assistance Server:** The server-side component connecting the AR Client and the Remote Client. It forwards the camera stream sent by the AR Client to the Remote Client, and marker-setting commands in the reverse direction, using the standardized WebRTC protocol.

The AR Client is implemented using Godot, an open-source engine for creating 2D and 3D games and applications. It has built-in support for interacting with AR and VR glasses using the OpenXR standard, and also allows integrating devices not supporting OpenXR through a plugin interface. We implement this interface to integrate the Viture Luma Ultra, the AR device we chose for the demonstration due its lightweightness, see-through glasses, and the complementary Viture Neckband Pro, which allows driving the glasses from a standard Android application (as can be produced with Godot) without a tethered smartphone. It can only be controlled through the proprietary Viture C++ SDK, but thanks to Godot's AR/VR abstraction, the vendor-specific code is limited to the plugin, making the rest of the application vendor-agnostic.

For the Remote Assistance Server, we build on LiveKit, an open-source framework for real-time audio and video applications using the WebRTC protocol. As WebRTC is supported by all major browsers, the LiveKit server can forward the AR camera feed directly to the Remote Client without any proprietary streaming technologies.

The Remote Client will be implemented using the popular open-source React framework.

6.3. Platform Security & Authentication

All network connections in and out of the platform will be encrypted using TLS (see section 5.2.1), ensuring data confidentiality and tamper resistance in transit. The same is the case in the communication between platform components, with the possible exception of communication between components running on the same machine.

For authentication between the Platform Application and the services, we will use OpenID Connect (see section 0). Specifically, each service will act as a resource server and will require each incoming API request to include an access token issued by the platform's authorization server. In the initial version of the platform, this authorization server will be embedded in the Platform Application, which simultaneous acts as its client to obtain access tokens;

in the future, the authorization server will be separate and operated by the platform provider (in a centralized deployment model) or provided by the company hosting the platform instance (in a future federated model where every company has their own platform instance).

6.4. Platform Deployment Infrastructure for the Demonstration

This section describes the technical infrastructure of the Lasers4MaaS platform as deployed for the demonstration at the end of the project, shown below in Figure 15. It is designed to closely resemble a future real-world production deployment in the cloud.

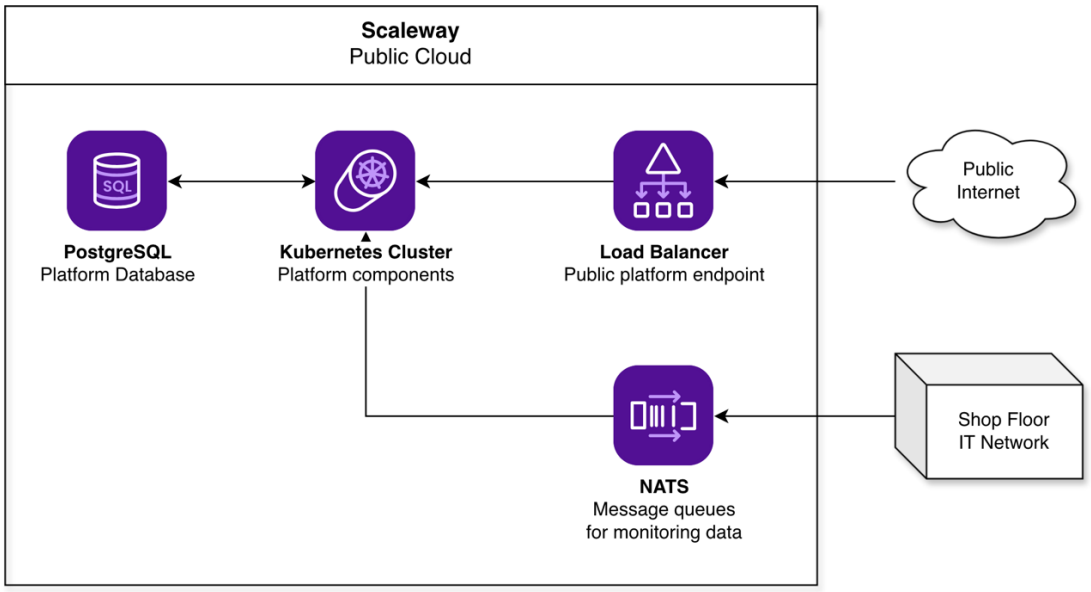


Figure 15: Overview of the infrastructure used for the demo platform deployment

6.4.1. Scaleway

We chose Scaleway[6], a sovereign EU cloud provider, for the demo deployment, for two reasons. First, as Digital Sovereignty is becoming a strategic concern for many EU companies – especially in the industrial sector – we aim to demonstrate that using such a truly sovereign cloud offering for a non-trivial application is feasible. Second, Scaleway's cloud services tend to be managed versions of open-source components, and as such offer us the convenience of low maintenance overhead without creating vendor lock-in, matching our Portability goal (see section 2.1).

6.4.2. Kubernetes

Kubernetes is an open-source, vendor-neutral system for deploying and managing containerized applications.[7] Originally developed by Google, it has become the industry standard for running compute workloads in public and private cloud environments, making it a highly portable deployment target.

For this reason, we use a Kubernetes instance (called a "cluster") to host most core platform components, specifically the Platform Application (section 6.2.1), the Remote Monitoring Worker (section 6.2.3), and the LiveKit server for the AR Remote Assistance feature (section 4.6). In addition, the Kubernetes cluster hosts a HTTP reverse proxy (Traefik) that performs TLS termination and routes incoming requests to the Platform Application, cert-manager to provide the proxy with a TLS certificate from the Let's Encrypt certificate authority, and Grafana Alloy for collecting platform logs and system metrics for monitoring and obversability purposes.

Some of the platform services, such as the Process Selector (section 6.2.4), will be deployed to the Kubernetes instance as well. Others will be hosted externally, such as the Process Controller Code Generator (section 6.2.6), which will run in WMG' data centers. This difference is transparent to the Platform Application, which is simply

[6] *Scaleway, https://www.scaleway.com/en/*
[7] *Kubernetes, https://kubernetes.io/*

configured with the correct service endpoint URLs and communicates with the services the same way in both cases. Thanks to the use of encrypted TLS connections and OIDC for authentication (see section 6.3), data exchange between the core platform and services is secure regardless of deployment location.

6.4.3. Cloud-Managed Data Services

For the Platform Database and the Remote Monitoring Pipeline's messages queues, we use the Scaleway PostgreSQL and Scaleway NATS cloud service. These are cloud-managed versions the open-source PostgreSQL and NATS systems and thus do not create any vendor dependencies for the platform, while reducing operational overhead compared to self-hosting these systems.

6.4.4. Generative AI Infrastructure

For the platform's Generative AI functionality, we will use open-weight models with permissive licenses. Multiple options will be considered, including those published by Mistral AI[8], a French frontier AI lab.

In the demo deployment, we will leverage Scaleway Generative APIs, a cloud-managed service that provides an OpenAI-compatible API for model access. This aligns it with many other open-weight model providers and self-hosting solutions such as vLLM and Ollama, which also provide OpenAI-compatible APIs, thus maximizing interoperability.

7. Current Status

Based on our user research and testing, we have created final or in-progress UX designs for all parts of the platform user interface, which we will continue to refine over the course of the project. We have also implemented and deployed first preview version of the Platform Application and Platform Database implementing the core process development documentation features; testing of this version by the project partners is in progress and will feed back into the design and implementation.

On the service layer, API definitions have been created for the Process Selector and Process Optimizer, and first draft implementations of both are available. They will be deployed and integrated soon. API specification work for the Sustainability & Cost Evaluator and Process Controller Code Generator is ongoing.

For the AR Remote Assistance feature, a proof-of-concept implementation exists of the interaction with the AR glasses, including camera feed acquisition and depth mapping. As a next step, we are working on the

8. Impact

From the perspective of this architectural deliverable, the design of the Lasers4MaaS platform is structured to systematically 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 fully realized on the shop floor, as physical integrations remain confined to isolated local testing. However, the architectural implementation of a decoupled microservices design, which utilizes standardized API wrappers and OCI containerized data uploaders, is explicitly designed to eliminate the manual, custom coding that historically caused these onboarding delays. By treating the physical laser cells as software-defined endpoints, the platform's current design ensures that future integrations of new shop-floor equipment and external digital services can be handled via plug-and-play connector blocks, converting what was once an intensive systems-engineering endeavor into a rapid, ad-hoc digital configuration.

To establish the foundation for trusted B2B data exchanges, the platform's database and middleware architecture have been designed to transition the transactional success rate from an unverified baseline to a resilient target of 100% using validated "Digital Handshakes." At this stage in the project timeline, the transaction success rate remains at a 0% baseline because we are operating in single-session, local-sandbox environments to ensure schema stability before deploying the public, multi-tenant network. The architectural inclusion of a high-performance NATS JetStream messaging broker serves as the critical technical enabler to reach this target. By implementing disk-persistent distributed logs, the platform is designed to handle network latency and connection drops on the shop floor,

[8] *Mistral AI, https://mistral.ai/*

allowing the local uploader's FIFO queue to buffer telemetry payloads and replay them automatically upon connection recovery to ensure that critical process certification records are never lost.

To meet the demanding maintenance requirements of highly non-standard laser systems, the platform 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 is still in progress, we are currently laying the groundwork by testing the laser analogue power monitoring pipelines and evaluating the initial core of our Android-based AR remote assistance application. This setup is specifically designed to allow remote specialists to interactively troubleshoot issues like optical degradation, chiller temperature spikes, or misalignment. By streaming live camera feeds to off-site experts who can then spatially map three-dimensional visual markers onto physical hardware, the platform's architecture is positioned to resolve complex system faults remotely, minimizing the necessity of physical travel and reducing overall operational carbon footprints once the system is fully deployed.

Finally, the platform's unified data architecture 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 common scientific data framework based on the open-source Visualization Toolkit format, which allows the platform to interpolate high-frequency experimental sensor signals with synthetic simulation meshes. While the real-time closed-loop control feedback loop is still undergoing active development and has not yet been deployed to the physical PLC boards on the shop floor, our offline testing has verified that running low-dimensional Reduced-Order Models on the cloud platform can flag process deviations within seconds. This mathematical validation proves that our current software architecture is on the correct track to support real-time state estimation and adaptive quality control during the upcoming demonstration phases.

9. Exploitation potentials

FUTURICE: Futurice's exploitation strategy focuses on the commercialization of the digital orchestration backbone defined in this deliverable, positioning the company as a leading provider of Sovereign Industrial Coordination Systems. By successfully designing and deploying the centralized, multi-tenant core of the Lasers4MaaS platform using a microservices architectural style, Futurice establishes a highly modular software template that can be packaged and commercialized for other high-value, decentralized manufacturing domains. This platform core application, built on mature, open-source frameworks like Python and Django, allows Futurice to market a vendor-lock-in-free "Laser Operating System" that avoids dependency on non-EU hyperscalers and aligns directly with European digital sovereignty objectives. The technical integration of the Remote Monitoring Pipeline using a scalable NATS JetStream message queue infrastructure creates a reusable commercial asset for real-time data ingestion and predictive maintenance analytics across decentralized shop floors. Furthermore, by abstracting vendor-specific machine complexities through a dedicated "connector" framework, Futurice can adapt this data pipeline to new industrial deployment setups by simply scaling the connector abstraction layer, leaving the core infrastructure unchanged and ready for future transition into federated setups like Common European Data Spaces. Finally, Futurice aims to leverage the software solutions built for Generative-AI-based natural language project searching and open-source WebRTC-based AR remote assistance to deliver premium operational service upgrades for on-site troubleshooting, allowing manufacturing networks to shift their competitive advantage from standalone hardware products to fully integrated ecosystem coordination.

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: The Lasers4MaaS digital platform provides a framework to combine data acquisition, analysis, and service orchestration. Due to the integration of process development, monitoring, and AI-based optimization into a unified environment, commissioning and ramp-up of laser applications could be accelerated. The Manufacturing-as-a-Service approach enables FFT to offer flexible, data driven manufacturing services to customers who can use the platform to define and request production processes. A standardized, modular, interoperable and reconfigurable framework ensures that the solutions can be transferred to different applications and production setups, which could strengthen FFT's position as a provider of highly automated production systems.

10. Outlook

The architectural foundation established in Deliverable 5.1 represents the initial phase of a fundamental industrial shift: moving from the rigid "Capital Trap" of machine ownership toward a resilient model of orchestrated manufacturing capacity. The long-term ambition is to establish a Sovereign Industrial Coordination System that organizes demand, supply, data, trust, and economics into a single, cohesive architecture. This "Laser Operating System" will move beyond a simple centralized hub to become a decentralized network where manufacturers pay only for validated, defect-free components rather than idle hardware and capital risk.

Future developments will refine this operating system across its three distinct scopes, beginning with enhanced micro-level autonomy where sovereign edge nodes enable machine safety protocols to autonomously halt defective welds in milliseconds, reacting instantly to process instabilities without the limitations of cloud-induced latency. On the meso-level, the platform will integrate advanced cognitive capabilities through Generative AI models that leverage vast repositories of historical process data to autonomously generate experimental process definitions and predict optimal, waste-free pathways prior to any physical execution. These advancements will be complemented by Augmented Reality interfaces that overlay real-time diagnostic and quality data directly onto physical assets, effectively bridging the expertise gap between on-site technicians and remote specialists.

At the macro-level, the vision extends to the creation of a Sovereign Industrial Grid where production loads are dynamically routed based on renewable energy availability, enabling energy-intensive manufacturing to actively stabilize the grid. By transforming regulatory requirements such as the DPP and ESPR into automated platform features, Lasers4MaaS ensures that European manufacturing remains fundamentally adaptable by design. Ultimately, this architecture addresses the trust gap that currently prevents deep B2B collaboration, replacing technical and legal ambiguity with an automated "Digital Handshake" and a cryptographic "Liability Shield" that protects sensitive intellectual property. This roadmap ensures that the future of European industrial strategy is defined by the strength of its coordinated systems rather than isolated hardware products.