Ref. Ares(2025)1567055 - 27/02/2025

API Specification and Updates

D2.1

HURRICANE

Month 12

Funded by the European Union

HURRICANE is funded under the call HORIZON-CL4-2023-TWIN-TRANSITION-01-37 within Horizon Europe, the European Union's framework programme for research and innovation (grant agreement no.:101138494)

Document Information

Project Title	Sector-coupling hub for circular use of thermal and industrial waste		
Acronym	HURRICANE		
Grant agreement no.	101138494		
Topic ID	HORIZON-CL4-2023-TWIN-TRANSITION-01-37		
EU Contribution	€ 19.090.973,75		
Coordinating entity	ArcelorMittal Belgium		
Start date	01 Jan 2024	End date	31 Dec 2028
Deliverable	D2.1	WP	2
Deliverable type	R- Document, report	Dissemination level	PU
Lead beneficiary	PDMFC		
Authors	C. Marques, N. Pedrosa		
Contributions from	/		

Change Records

Document Revision History			
Issue date	Version	Author/reviewer	Comments
2024/12/30	V0.9	PDMFC (C. Marques / N. Pedrosa)	Release Candidate
2024/12/31	V1.0	PDMFC (C. Marques / N. Pedrosa)	Release Candidate
2025/02/26	V1.1	N. Anand, H. Uijtdebroecks	Reviewed and ready for submission

Disclaimer: This document and its content reflect only the author's view; therefore, the European Commission is not responsible for any use that may be made of the information it contains.

Project Summary

Within HURRICANE a sector-coupling circular hub centred around the ArcelorMittal Ghent site will be created. We will target efficient resource management together with the recovery and utilization of squandered industrial waste heat and water. Together with ArcelorMittal Ghent's ongoing initiatives, this will lead to a reduction of energy, water and raw materials by at least 20%. Thanks to the ongoing projects taking place within and around the Ghent site, the site is already well connected to many other industries like waste suppliers, chemical producers (ethanol offtake & H2 waste gas), renewable power producers, and wastewater treatment. It has become a multi-sectoral hub leading to efficient implementation of industrial symbiosis concepts. The Ghent site has a significant amount of recyclable energy, material and water that allows this symbiosis. These aspects are not only from the steel making processes, but also from other operations taking place in the mentioned "multi-sectoral" hub. This hub can be further enhanced with the integration of waste heat with its ongoing initiatives. Our solution aims at developing and demonstrating novel heat recovery (heat exchanger) and upgrading (heat pumps) solutions from selected operations and then coupling it with the internal and external off takers by means of a heat grid. With digital tools, aspects like broadening the district heating network, and adapting the heat demand profile of the buildings to better match the intermittent of the waste heat, can be optimized. Finally, an integrated software tool for circular hubs that combines the different tools and data produced at the different operations will be developed and validated. Through two virtual demonstrations and circular hubs blueprint the replication potential will be proven.

Acknowledgement: HURRICANE – Sector-coupling hub for circular use of thermal and industrial waste (grant agreement number: 101138494) is funded under the call HORIZON-CL4-2023-TWIN-TRANSITION-01-37 within Horizon Europe, the European Union's framework programme for research and innovation.

HURRICANE Consortium

Name	Short name	Country	Contact
ArcelorMittal Belgium	AMB	Belgium	Project Coordinator: Joke Bauwens Joke.bauwens@arcelormittal.com
Centre De Recherches Metallurgiques ASBL	CRM	Belgium	Participant : Bart Vervaet Bart.vervaet@crmgroup.be
Vlaamse Instelling voor Technologisch Onderzoek N.V.	VITO	Belgium	Participant : Johan Desmedt johan.desmedt@vito.be
DBFZ GERMAN BIOMASS RESEARCH CENTRE	DBFZ	Germany	Participant : Konrad Siegfried konrad.siegfried@dbfz.de
Università Politecnica delle Marche	UPM	Italy	Participant : Filippo E. Ciarapica f.e.ciarapica@univpm.it
PDM E FC PROJECTO DESENVOLVIMENTO MANUTENCAO FORMACAO E CONSULTADORIALDA	PDMFC	Portugal	Participant : Nuno Pedrosa nuno.pedrosa@pdmfc.com
Lokaal Bestuur Zelzate	City Zelzate	Belgium	Participant : Els De Vilder Milieudienst@zelzate.be
Provincie Oost-Vlaanderen	ProvEF	Belgium	Participant : Yanti Ehrentraut yanti.ehrentraut@oost-vlaanderen.be
Finarmit	Finarmit	Belgium	Participant : Filip Keppens filip.keppens@fineg.be
Organisatie Broeders Van Liefde	SJB	Belgium	Participant : Luc Querter luc.querter@janbaptist.broedersvanliefde.be

Table of Contents

Executive Summary

The HURRICANE project focuses on creating a sector-coupling circular hub at the ArcelorMittal Ghent site, targeting efficient resource management by recovering and utilizing industrial waste heat and water. The project aims to achieve a 20% reduction in energy, water, and raw material consumption through innovative solutions, including heat exchangers, heat pumps, and a heat grid that connects internal and external off-takers.

A critical component of this project is the development of an integrated software tool for managing circular hubs, which will combine data and digital tools to optimize energy, water, and heat utilization. This tool will facilitate better decision-making and operational efficiency across multiple sectors.

This document, API Specification and Updates, outlines the specifications and updates related to the software application programming interfaces (APIs) integral to the HURRICANE platform. These APIs serve as the communication layer for integrating various systems, data sources, and stakeholders within the circular hub. The specification includes details on API functionality, data exchange protocols, and security measures, ensuring smooth interoperability between diverse systems across the circular hub.

Additionally, the document provides an overview of the updates made to existing APIs to enhance their functionality, support new features, and improve the user experience. This includes the integration of new data sources, refined data handling capabilities, and optimizations for scalability and security.

The successful implementation of these APIs is crucial for the overall success of the HURRICANE project, supporting the creation of a dynamic, interconnected system that drives resource efficiency and circular economy principles across the industrial hub. This document ensures that the system's architecture is scalable, adaptable, and ready for future integration, paving the way for the replication of this model in other industrial settings.

1 Introduction

The HURRICANE project aims to establish a sector-coupling circular hub centred around the ArcelorMittal Ghent site, with the goal of optimizing resource management, waste heat recovery, and water utilization. As part of this initiative, this document focuses on defining the technical specifications and updates for the APIs that will be used to integrate and streamline various systems, tools, and data sources within the circular hub.

The circular hub concept within HURRICANE is designed to connect multiple sectors and industries operating in the Ghent area, creating a highly efficient industrial symbiosis model. This model leverages recovered waste heat, water, and materials from diverse processes to reduce energy consumption and raw material use. Achieving this requires seamless data exchange between multiple systems and stakeholders to monitor and optimize resource flows, including waste heat and water, across the entire hub.

The API is a critical component of the Hurricane project, enabling seamless integration and communication between the various software modules and system components. The API defines a standardized interface, allowing different systems to interact efficiently, ensuring that data flows smoothly and that components function cohesively within the project.

This document serves as a comprehensive guide for the developers, system integrators, and stakeholders involved in the Hurricane project. It outlines the structure, features, and functionalities of the Hurricane project's API, providing details on the core functionality and recent updates introduced during the current development cycle.

1.1 Objectives:

The main objectives of this document are:

1. To specify the API's endpoints, detailing the functionality, request types, parameters, and expected responses.
2. To highlight the latest updates and changes made to the API, including newly introduced features, enhancements, deprecations, and bug fixes.
3. To guide developers on how to integrate the API with other system components, ensuring smooth interaction and compatibility.
4. To set best practices for API security, error handling, and performance optimization.

1.2 Scope of the Document:

1. API Design and Endpoints: A detailed specification of the available API endpoints, including the necessary request types, parameters, and response structure.
2. Updates and Modifications: A summary of recent changes to the API, including the addition of new functionalities, adjustments to existing endpoints, and deprecated features.
3. Integration Guidelines: Guidance on integrating the API with other components of the Hurricane system, ensuring compatibility and minimizing potential issues.
4. Security Considerations: An overview of the security measures implemented to safeguard data and ensure secure access to the API.
5. Error Handling and Troubleshooting: Best practices for handling errors and interpreting API responses, helping developers troubleshoot integration challenges.

The APIs specified in this document are integral to enabling this level of coordination. They will serve as the communication backbone between various subsystems within the hub, facilitating data transfer and interactions between heat exchangers, heat pumps, energy management systems, industrial processes, and external off-takers. The development and deployment of these APIs will enable the creation of a digital platform for the HURRICANE project, which will enhance monitoring, decision-making, and optimization capabilities.

This document outlines the key design principles, functional specifications, and updates required for the APIs, ensuring that they are robust, scalable, and adaptable to the needs of the HURRICANE ecosystem. Additionally, it will highlight the integration points with other tools and systems within the project, contributing to the broader goal of increasing the circularity and sustainability of industrial processes. Furthermore, it will address ongoing updates to API functionalities in response to evolving requirements, challenges, and technological advancements within the project.

The successful implementation of these APIs will support the HURRICANE project's vision of efficient resource management and the development of a circular hub blueprint, which will be demonstrated and validated through virtual and physical demonstrations.

2 API Overview

This API is designed for managing industrial heat recovery systems, with the goal of capturing and utilizing waste heat from industrial processes. The purpose is to optimize their energy efficiency by identifying, monitoring, and recovering waste heat from various sources.

The key components and functions are:

1. Industrial Processes (*ProcessController*, *ProcessService*)
    - Manage different types of industrial processes (BATCH_PROCESS, CONTINUOUS_PROCESS)
    - Track and organize processes that generate waste heat
    - Handle process creation, updates, and monitoring
2. Heat Sources (*HeatSourceController*, *HeatSourceService*)
    - Manage various types of heat sources (COOLING_WATER, SLAG, TUNNEL, COIL)
    - Track temperature ranges and characteristics
    - Associate heat sources with specific industrial processes
    - Enable searching heat sources by temperature ranges
3. Heat Recovery Technology (*HeatRecoveryTechController*, *HeatRecoveryTechService*)
    - Handle different types of heat recovery technologies:
        - Heat exchangers
        - Heat pumps
        - Organic Rankine Cycle (ORC)
        - Steam generators
        - Thermoelectric generators
    - Manage compatibility between heat sources and heat recovery technologies
    - Track efficiency degradation of recovery systems
4. Sensor Management (*SensorController*, *SensorService*)
    - Provide real-time monitoring capabilities
    - Handle sensor readings and calibration
    - Support different sensor types (temperature, pressure, flow)
    - Manage sensor maintenance schedules
    - Calculate averages, analyze readings over time
5. Heat Source Characteristics (*HeatSourceCharacteristicController*, *HeatSourceCharacteristicService*)
    - Categorizes heat sources by temperature ranges (LOW, MEDIUM, HIGH)
    - Tracks specific characteristics of heat sources
    - Helps in matching heat sources with appropriate recovery systems

As for the ontology, it provides a semantic framework that defines:

1. Relationships between different components (hasHeatSource, implementsRecovery, providesEnergyTo)
2. Classification of processes and equipment
3. Energy utilization pathways
4. Quality control and maintenance activities
5. Performance metrics and monitoring requirements

The API implementation has:

- Clear separation of concerns (controllers, services, repositories)
- Comprehensive documentation (Swagger/OpenAPI)
- Robust error handling
- Strong data validation
- Efficient relationship management
- Auditing capabilities
- Scalable architecture

The API is built using Spring Boot, implementing a RESTful architecture where resources are exposed through HTTPS endpoints with standardized methods (GET, POST, PUT, DELETE). The API uses JSON for data exchange, supported by the extensive use of DTOs (Data Transfer Objects) for request/response.

Security is based on OAuth 2.0 with JWT (JSON Web Tokens) for authentication and authorization. The security configuration includes role-based access control (RBAC) with different permissions for administrators, operators and support for further roles.

The persistence layer uses JPA (Java Persistence API) with Hibernate as the ORM provider. Error handling provides consistent error responses across the API.

2.1 GDPR Compliance

The system primarily deals with industrial process data rather than personal data. There are, however, several areas where personal data might be indirectly processed, particularly in the context of system operators and administrators. Care must be taken to balance system's primary function as an industrial process management tool while ensuring compliance with data protection regulations. While the GDPR has provisions to allow some personal data processing from system operators and administrators, the amount of personal data will always be minimized, ensuring proper system auditing, but protecting users.

3 API Specifications

The API implements OAuth 2.0 Resource Server authentication with JWT (JSON Web Tokens) with a stateless session management policy. In this way, no session state is stored on the server side, making the API more scalable.

Currently, the application uses role-based access control (RBAC) with different authorization levels:

- Public endpoints: GET requests to **/processes/*** and Swagger documentation URLs
- Admin-only endpoints: POST operations on **/processes/*** and all **/recovery-systems/*** endpoints
- Admin/Operator endpoints: **/heat-sources/*** and **/sensors/***
- All other endpoints require authentication
- Bearer token authentication scheme is used to access the API

3.1 API Endpoints

Spring Boot REST API for Industrial Heat Recovery

Version: v1.0.0

Base URL: https://api.hurricane.pdmfc.com

3.2 Energy Utilization Management

3.2.1 PUT /energy-utilizations/{id}

Update an energy utilization configuration

Field	Description
Parameters	id (path) Type: integer (int64) Required: Yes
Request Body	Content-Type: application/json
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ PUT 'https://api.hurricane.pdmfc.com/energy-utilizations/example' \\ -H 'Accept: application/json' \\ -H 'Content-Type: application/json' \\ -d '{}'

3.2.2 DELETE /energy-utilizations/{id}

Delete an energy utilization configuration

Field	Description
Parameters	id (path) Type: integer (int64) Required: Yes
Responses	Status Code: 200 Description: OK
Example	curl -X \\ DELETE 'https://api.hurricane.pdmfc.com/energy-utilizations/example' \\ -H 'Accept: application/json'

3.2.3 POST /energy-utilizations

Create a new energy utilization configuration

Field	Description
Request Body	Content-Type: application/json
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ POST 'https://api.hurricane.pdmfc.com/energy-utilizations' \\ -H 'Accept: application/json' \\ -H 'Content-Type: application/json' \\ -d '{}'

3.2.4 GET /energy-utilizations/type/{type}

Find energy utilizations by type

Field	Description
Parameters	type (path) Type: string (enum: DIRECT_HEAT_APPLICATION, PREHEATING, DRYING_OPERATION, POWER_GENERATION, CHP, PRODUCT_PROCESSING, UTILITY_SYSTEM) Required: Yes
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ GET 'https://api.hurricane.pdmfc.com/energy-utilizations/type/example' \\ -H 'Accept: application/json'

3.2.5 GET /energy-utilizations/optimize/{recoverySystemId}

Find optimal energy utilization configurations for a heat recovery component

Field	Description
Parameters	HeatRecoveryTechId (path) Type: integer (int64) Required: Yes
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ GET 'https://api.hurricane.pdmfc.com/energy-utilizations/optimize/example' \\ -H 'Accept: application/json'

3.2.6 GET /energy-utilizations/cascade/{primaryUtilizationId}

Get the complete energy utilization cascade chain

Field	Description
Parameters	primaryUtilizationId (path) Type: integer (int64) Required: Yes
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ GET 'https://api.hurricane.pdmfc.com/energy-utilizations/cascade/example' \\ -H 'Accept: application/json'

3.3 Heat Source Characteristics

3.3.1 POST /heat-source-characteristics

Create a new heat source characteristic

Field	Description
Request Body	Content-Type: application/json
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ POST 'https://api.hurricane.pdmfc.com/heat-source-characteristics' \\ -H 'Accept: application/json' \\ -H 'Content-Type: application/json' \\ -d '{}'

3.3.2 GET /heat-source-characteristics/temperature-range/{range}

Get characteristics by temperature range

Field	Description
Parameters	range (path) Type: string (enum: LOW_TEMPERATURE, MEDIUM_TEMPERATURE, HIGH_TEMPERATURE) Required: Yes
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ GET 'https://api.hurricane.pdmfc.com/heat-source-characteristics/temperature-range/example' \\ -H 'Accept: application/json'

3.3.3 GET /heat-source-characteristics/heat-source/{heatSourceId}

Get characteristics for a heat source

Field	Description
Parameters	heatSourceId (path) Type: integer (int64) Required: Yes
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ GET 'https://api.hurricane.pdmfc.com/heat-source-characteristics/heat-source/example' \\ -H 'Accept: application/json'

3.4 Heat Source Management

3.4.1 POST /heat-sources

Create a new heat source

Field	Description
Request Body	Content-Type: application/json
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ POST 'https://api.hurricane.pdmfc.com/heat-sources' \\ -H 'Accept: application/json' \\ -H 'Content-Type: application/json' \\ -d '{}'

3.4.2 GET /heat-sources/{id}

Get a heat source by ID

Field	Description
Parameters	id (path) Type: integer (int64) Required: Yes
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ GET 'https://api.hurricane.pdmfc.com/heat-sources/example' \\ -H 'Accept: application/json'

3.4.3 GET /heat-sources/temperature-range

Find heat sources within a temperature range

Field	Description
Parameters	minTemp (query) Type: number (double) Required: Yes maxTemp (query) Type: number (double) Required: Yes
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ GET 'https://api.hurricane.pdmfc.com/heat-sources/temperature-range?minTemp=example&maxTemp=example' \\ -H 'Accept: application/json'

3.4.4 GET /heat-sources/process/{processId}

Get all heat sources for a process

Field	Description
Parameters	processId (path) Type: integer (int64) Required: Yes
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ GET 'https://api.hurricane.pdmfc.com/heat-sources/process/example' \\ -H 'Accept: application/json'

3.5 IndustrialProcess Management

3.5.1 GET /processes/{id}

Get a process by ID

Field	Description
Parameters	id (path) Type: integer (int64) Required: Yes
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ GET 'https://api.hurricane.pdmfc.com/processes/example' \\ -H 'Accept: application/json'

3.5.2 PUT /processes/{id}

Update a process

Field	Description
Parameters	id (path) Type: integer (int64) Required: Yes
Request Body	Content-Type: application/json
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ PUT 'https://api.hurricane.pdmfc.com/processes/example' \\ -H 'Accept: application/json' \\ -H 'Content-Type: application/json' \\ -d '{}'

3.5.3 DELETE /processes/{id}

Delete a process

Field	Description
Parameters	id (path) Type: integer (int64) Required: Yes
Responses	Status Code: 200 Description: OK
Example	curl -X \\ DELETE 'https://api.hurricane.pdmfc.com/processes/example' \\ -H 'Accept: application/json'

3.5.4 GET /processes

Get all processes with optional filtering

Field	Description
Parameters	type (query) Type: string (enum: BATCH_PROCESS, CONTINUOUS_PROCESS) pageable (query) Type: object Required: Yes
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ GET 'https://api.hurricane.pdmfc.com/processes?type=example&pageable=example' \\ -H 'Accept: application/json'

3.5.5 POST /processes

Create a new process

Field	Description
Request Body	Content-Type: application/json
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ POST 'https://api.hurricane.pdmfc.com/processes' \\ -H 'Accept: application/json' \\ -H 'Content-Type: application/json' \\ -d '{}'

3.6 Heat Recovery Technology Management

3.6.1 POST /heat-recovery-tech

Create a new heat recovery technology instance

Field	Description
Request Body	Content-Type: application/json
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \ POST 'https://api.hurricane.pdmfc.com/heat-recovery-tech ' \ -H 'Accept: application/json' \ -H 'Content-Type: application/json' \ -d '{}'

3.6.2 POST /heat-recovery-tech/{heatRecoveryTechId}/heat-sources/{heatSourceId}

Assign a heat recovery tech component to a heat source

Field	Description
Parameters	heatRecoveryTechId (path) Type: integer (int64) Required: Yes heatSourceId (path) Type: integer (int64) Required: Yes
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \ POST 'https://api.hurricane.pdmfc.com/heat-recovery-tech/example/heat-sources/example' \ -H 'Accept: application/json'

3.6.3 GET /heat-recovery-tech/compatible/{heatSourceId}

Find compatible heat recovery components for a heat source

Field	Description
Parameters	heatSourceId (path) Type: integer (int64) Required: Yes
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ GET 'https://api.hurricane.pdmfc.com/heat-recovery-tech/compatible/example' \\ -H 'Accept: application/json'

3.7 Sensor Management

3.7.1 POST /sensors

Create a new sensor

Field	Description
Request Body	Content-Type: application/json
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ POST 'https://api.hurricane.pdmfc.com/sensors' \\ -H 'Accept: application/json' \\ -H 'Content-Type: application/json' \\ -d '{}'

3.7.2 POST /sensors/readings

Record a new sensor reading

Field	Description
Request Body	Content-Type: application/json
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ POST 'https://api.hurricane.pdmfc.com/sensors/readings' \\ -H 'Accept: application/json' \\ -H 'Content-Type: application/json' \\ -d '{}'

3.7.3 GET /sensors/{sensorId}/readings

Get sensor readings for a time range

Field	Description
Parameters	sensorId (path) Type: integer (int64) Required: Yes startTime (query) Type: string (date-time) Required: Yes endTime (query) Type: string (date-time) Required: Yes
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \\ GET \\ 'https://api.hurricane.pdmfc.com/sensors/example/readings?startTime=example&endTime=example' \\ -H 'Accept: application/json'

3.7.4 GET /sensors/{sensorId}/average

Calculate average reading for a time range

Field	Description
Parameters	sensorId (path) Type: integer (int64) Required: Yes startTime (query) Type: string (date-time) Required: Yes endTime (query) Type: string (date-time) Required: Yes
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \ GET \ 'https://api.hurricane.pdmfc.com/sensors/example/average?startTime=example&endTime=example' \ -H 'Accept: application/json'

3.7.5 GET /sensors/calibration-needed

Find sensors needing calibration

Field	Description
Parameters	thresholdDate (query) Type: string (date-time) Required: Yes
Responses	Status Code: 200 Description: OK Content-Type: */*
Example	curl -X \ GET \ 'https://api.hurricane.pdmfc.com/sensors/calibration-needed?thresholdDate=example' \ -H 'Accept: application/json'

4 API Future Updates

The API's future development prioritizes critical enhancements across several key areas to improve its functionality, security, and performance. The most pressing update would be the implementation of further comprehensive security measures, including rate limiting to prevent API abuse, enhanced audit logging for tracking security-relevant operations, and more granular permission levels beyond the current ADMIN/OPERATOR roles.

Performance optimization represents another crucial area for improvement, focusing on the implementation of caching mechanisms for frequently accessed data and pagination for all list endpoints. These optimizations would be particularly beneficial for endpoints handling sensor data and energy utilization metrics, ensuring the API remains responsive even as the volume of data grows. Additionally, query optimization and strategic database indexing would enhance the performance of complex data retrievals.

The implementation of additional comprehensive monitoring capabilities can include health check endpoints, metrics collection for API usage patterns, and enhanced error tracking mechanisms.

4.1 Versioning:

- Current version: v1
- API is versioned using URI path versioning
- Breaking changes will result in new version number
- Multiple versions may be supported simultaneously if data integrity is not compromised.

5 Conclusions

The Heat Recovery API represents a comprehensive approach for managing industrial heat recovery systems, built on modern technologies and best practices. The implementation leverages mature technologies like Spring Boot for RESTful services, OAuth 2.0 with JWT for security, and JPA for data persistence, creating a robust foundation for industrial process management. The API's architecture uses a clear separation of concerns through its layered structure, ensuring maintainability and scalability.

Key features of the API support sophisticated heat source management, sensor data handling, energy utilization optimization, and heat recovery technologies integration. The underlying ontology provides a rich semantic framework for representing industrial processes and their relationships, while the comprehensive error handling system ensures reliable operation. Security is implemented through role-based access control and certain aspects such as rate limiting, will be implemented as future enhancements.

The API specification provides a clear contract between the service and its consumers, enabling seamless integration with various client applications. It establishes a foundation for future development, with well-defined paths for enhancement in areas such as real-time capabilities, performance optimization, and security features.

Looking forward, the API specification will guide the evolution of the heat recovery platform and evolve throughout the project life. Planned enhancements focus on critical areas including security hardening, performance optimization, real-time capabilities, and improved monitoring. These improvements will be implemented while maintaining backward compatibility within the established architectural principles. The specification also highlights the need for additional features such as standardized energy efficiency calculations and integration with third-party systems, setting a clear direction for future development.

This deliverable thus serves not only as documentation of the current system but as a strategic roadmap for the API's evolution, ensuring that future development aligns with both technical best practices and business requirements in the industrial heat recovery domain.