MAST

Managing Sustainability Tradeoffs

Grant Agreement No.: 22035

D2.3 – Requirements specification and mapping to use cases

Document Number	D2.3
Document Title	Requirements specification and mapping to use cases
Version	1.0
Status	Final
Work Package	WP2
Deliverable Type	Report
Contractual Date of Delivery	M9 – September 2025
M9 – September 2025	M9 – September 2025
Responsible Unit	Wirtek (task leader)
Contributors	WRT, CLW, CPP

Keyword List	Use cases, User Stories, Requirements, Specification
Dissemination level	PU

Amendment History

Version	Date	Author (Unit)	Description
0.1	08/2025	WRT	Initial Draft with Template
0.2	12.09.2025	WRT	Added more content
0.3	19.09.2025	WRT	Major changes in methodology and structure

1 Executive Summary

Deliverable D2.3 consolidates the foundational insights from D2.1 Use Case Descriptions and D2.2 Sustainability State of the Art and Practice into a structured and traceable set of SMART requirements for the MAST project.
The requirements defined span two dimensions: Functional Requirements and Non-Functional Requirements necessary to guide the design, development, and implementation activities in WP3–WP5.

By establishing this requirements foundation, D2.3 enables the design of sustainability-aware architectures (WP3), development of integrated tooling (WP4), and validation through industrial pilots (WP5). It also sets the stage for continuous feedback loops and refinement, ensuring that MAST innovations remain responsive to evolving stakeholder needs and regulatory landscapes.

The requirements are specified for each of the use cases in the Netherlands, Denmark and Portugal (respectively from partners: CPP, WRT, CLW).

2 Table of Contents

3 Introduction

Deliverable D2.3 plays an important role in the MAST project by translating the conceptual and empirical foundations laid out in D2.1 - Use Case Descriptions and D2.2 - Sustainability State of the art and practice into a structured and actionable requirements specification. This document serves as the foundation for the design, development, and validation activities in WP3 (Concept Development), WP4 (Tooling and Integration), and WP5 (Pilots and Validation).
The MAST project addresses the dual challenge of technical and environmental sustainability in software-intensive systems. These two dimensions—maintainability and energy efficiency—are often treated in isolation, leading to suboptimal trade-offs and missed opportunities for synergy (Avgeriou et al., 2016; Deb, 2014). D2.3 aims to bridge this gap by defining a detailed set of SMART requirements that are traceable to industrial use cases, grounded in state-of-the-art research, and aligned with European regulatory frameworks such as the Corporate Sustainability Reporting Directive (CSRD) and the EU Green Deal (European Commission, 2022).
The requirements are categorized into:

Functional Requirements: Derived from user stories, success metrics, and desired system behaviors across the three industrial use cases—Wirtek (UC-WRT), Cleanwatts (UC-CLW), and Canon Production Printing (UC-CPP).
Non-Functional Requirements: Address cross-cutting concerns such as transparency, security, scalability, and interoperability, ensuring alignment with EU directives including the Corporate Sustainability Reporting Directive (CSRD) and the European Green Deal.
By consolidating diverse inputs into a unified specification, D2.3 ensures that the MAST project remains grounded in real-world needs while advancing the scientific and technological frontier of sustainable software and systems engineering.

3.1 Methodology

The methodology adopted for this deliverable is based on a triangulated approach that synthesizes insights from industrial use cases, academic research, and strategic project objectives. This ensures that the requirements are both contextually relevant and scientifically robust, supporting the dual sustainability goals of the MAST project—technical maintainability and environmental impact reduction.
The methodology integrates:

Use case narratives and extensions from D2.1 to define the entire user goal and its detailed steps, including alternatives. They provide the strategic "why" and detailed "how”.

Extended User Stories with focus on the tactical "what" and "for whom" from a user's perspective.

SMART Breakdown to avoid ambiguity and add relevance in the defined scope, timing and way to deliver and validate.

Input Sources

Three primary sources were used to elicit and structure the requirements:

Use Case Narratives and Success Metrics: Derived from D2.1 Use Case Descriptions, these include user stories, current system functionality, desired extensions, and measurable KPIs across the three industrial pilots—Wirtek (UC-WRT), Cleanwatts (UC-CLW), and Canon Production Printing (UC-CPP).
State-of-the-Art Analysis and Industrial Gaps: Based on D2.2 Sustainability state of the art and practice, this input identifies systemic gaps in current sustainability practices, such as fragmented metrics, lack of trade-off analytics, and limited integration of sustainability feedback into DevOps pipelines (Gill & Buyya, 2018; Schmidt et al., 2021).

Quantitative Objectives and Solution Concepts: Extracted from FPP MAST. these include SMART goals (e.g., 15–20% reduction in energy consumption, 20–30% reduction in technical debt), architectural blueprints, and innovation pathways for sustainability-aware tooling and decision support systems.

Synthesis Process

Requirement Formulation:

- Functional and non-functional requirements were written using the “the system shall…” pattern for clarity and testability.
- Each functional requirement was mapped to relevant user stories, with acceptance criteria derived from both user stories and project metrics.
- Where appropriate, Non-Functional requirements were included in the Acceptance criteria.
- Non-functional requirements were similarly structured, focusing on system qualities (e.g., transparency, security, scalability).

SMART Analysis:

- Every requirement was broken down using the SMART framework (Specific, Measurable, Achievable, Relevant, Time-bound) to ensure they are actionable and verifiable.

Traceability and Validation:

- Ensured each requirement is traceable to stakeholder needs and project goals, and that all requirements are complete, non-overlapping, and aligned with the MAST project’s objectives. Each requirement is listing the user stories that are relevant. Validation is insured by the Acceptance criteria.

4 Requirements specification

This section outlines the requirements derived from the three industrial use cases defined in D2.1 Use Case Descriptions and contextualized through the sustainability gaps identified in D2.2 Sustainability state of the art and practice. Each requirement is designed to support the MAST project's dual sustainability goals—technical maintainability and environmental impact reduction—while ensuring traceability to project KPIs and alignment with the European Green Deal and CSRD directives.

4.1 UC-WRT: Clean Code to Green Code

The use case initially described in D2.1 has been further developed in both scope and detail. As a result, this document provides an updated description that expands upon the previous version. Notable enhancements include additional clarifications, a more comprehensive narrative, and the inclusion of new user stories focused on smart scheduling. These updates reflect a natural progression based on internal analyses and the insights gained throughout the project.

4.1.1 Summary

The Wirtek use case, developed in collaboration with Aalborg University, focuses on embedding sustainability trade-off awareness directly into the design and operation of software systems. Centred on the Wappsto IoT/cloud platform, the use case introduces a new capability: allowing software users and developers to see their emission footprint and take action on them, for instance by selecting between alternative implementations of operations — one optimized for performance, another for energy efficiency, scaling down the operations in a not so green timeframe, or dynamically reducing accuracy and performance. These alternatives are accompanied by visualized metrics that expose trade-offs in latency, maintainability, and carbon impact.

Uniquely, the use case applies Wappsto to analyse and improve itself. This closed-loop approach supports Wirtek's strategic Clean Code to Green Code (C2G) ESG initiative, which aims to reduce the carbon footprint of software products during their operational phase. The pilot deployment will take place in Wirtek's own headquarters, where Wappsto is already used in a Tenant Consumption monitoring solution.

Through code instrumentation, trade-off dashboards, and continuous telemetry, this use case will help operational teams and developers better understand the environmental consequences of software design decisions. In doing so, it supports both the technical and environmental sustainability goals of the MAST project.

This use case addresses a crucial sustainability challenge: software design and execution methods typically optimize for speed, maintainability, and/or portability – often at the cost energy efficiency, consequently, carbon emissions. We aim to expose, visualize, and eventually optimize this trade-off by implementing and comparing different software execution strategies within Wappsto itself.
To achieve this, the system will leverage real-time and forecasted energy data from sources such as the ENTSO-E transparency platform and national providers such a as Denmark's Energinet. This allows us to translate energy consumption into a tangible carbon footprint (g CO_2 / kWh) for specific operations, users, or applications, enabling precise and actionable sustainability insights.

4.1.2 Updated User stories

- As a software architect, I want to compare the energy footprint of different function implementations, so I can choose the most sustainable option that meet our performance SLAs.
- As a product manager, I want to offer my customers the choice between high-performance and energy-efficient operations, as well as options to automatically schedule tasks for low-emission periods, so they can align with their own sustainability goals.
- As a system operator, I want real-time dashboards that show the carbon impact of the system's behaviour, so I can make informed configuration changes, such as shifting workloads between data centers to follow the greenest energy.
- As a developer, I want to receive insights into the impact of my code changes on energy consumption and maintainability, By attributing CO_2 emissions to my specific operations, these operations become a tangible, gamified metric, encouraging me to write greener code.
- As a DevOps engineer, I want to integrate carbon footprint analysis into our CI/CD pipeline, so that pull-requests are automatically flagged if they introduce a significant regression in energy efficiency.

- As an ESG officer, I want to generate auditable monthly emission reports from the platform, so I can track our progress against corporate sustainability targets.
- As an end-customer, I want to see a simplified overview of my operations' footprint and understand its carbon savings in relatable terms (e.g "equivalent to X trees planted"), so that I can take actions based on these numbers.

4.1.3 Functional Requirements

FR1. The system shall provide dual-mode software execution modules, allowing users to select between functionally equivalent implementations optimized for performance, energy efficiency, or dynamic accuracy.

Relevant User Stories:

- As a software architect, I want to compare the energy footprint of different function implementations, so I can choose the most sustainable option that meet our performance SLAs.
- As a product manager, I want to offer my customers the choice between high-performance and energy-efficient operations, as well as options to automatically schedule tasks for low-emission periods, so they can align with their own sustainability goals.

Acceptance Criteria:

- Users can select between at least three execution modes for a given operation.
- The system measures and displays the energy, latency, and maintainability trade-offs for each mode.
- The overhead from measurement and telemetry remains under 5% of total resource use.
- ≥15% measurable energy reduction in selected modules within 6 months of deployment.
- Visualized trade-off dashboards available with ≥3 KPIs (energy, latency, maintainability) by M18.

SMART Breakdown:

- **Specific:** Users can choose between performance, energy efficiency, or dynamic accuracy for selected operations.
- **Measurable:** At least three modes are available; ≥15% energy reduction; dashboards with ≥3 KPIs.
- **Achievable:** Built on Wappsto's modular architecture and telemetry integration.
- **Relevant:** Directly supports sustainability and operational flexibility.
- **Time-bound:** Prototype by M18; energy reduction within 6 months.

FR2. The system shall enable carbon-aware scheduling, allowing users to schedule non-urgent computations during periods of lowest grid carbon intensity.
Relevant User Stories:

- As a product manager, I want to offer my customers the choice between high-performance and energy-efficient operations, as well as options to automatically schedule tasks for low-emission periods, so they can align with their own sustainability goals.

Acceptance Criteria:

- Users can schedule tasks based on real-time and forecasted grid carbon intensity.
- The system integrates with external energy data APIs (e.g., ENTSO-E, Energinet).
- ≥25% of new or eligible customers have enabled a "green" feature (e.g., carbon-aware scheduling) by M24.

SMART Breakdown:

- **Specific:** Scheduling is based on grid carbon intensity data.
- **Measurable:** ≥25% customer adoption of green scheduling features.
- **Achievable:** Uses available APIs and Wappsto's scheduling capabilities.
- **Relevant:** Reduces operational carbon footprint.
- **Time-bound:** Customer adoption target by M24.

FR3. The system shall support geographical load shifting, dynamically moving workloads to data centers with the lowest current grid carbon intensity.
Relevant User Stories:

- As a system operator, I want real-time dashboards that show the carbon impact of the system's behaviour, so I can make informed configuration changes, such as shifting workloads between data centers to follow the greenest energy.

Acceptance Criteria:

- The system can shift workloads between at least two regions based on real-time grid data.
- Dashboards reflect the impact of load shifting on carbon emissions.
- All emission calculations are logged for auditability.

SMART Breakdown:

- **Specific:** Workloads can be shifted geographically based on carbon data.
- **Measurable:** At least two regions supported; impact visible in dashboards.
- **Achievable:** Built on Wappsto's distributed architecture.
- **Relevant:** Enables operational carbon optimization.
- **Time-bound:** Feature available by M18.

FR4. The system shall provide trade-off visualization dashboards, showing execution time, maintainability, and energy consumption for each mode.
Relevant User Stories:

- As a system operator, I want real-time dashboards that show the carbon impact of the system's behaviour, so I can make informed configuration changes, such as shifting workloads between data centers to follow the greenest energy.

- As an end-customer, I want to see a simplified overview of my operations' footprint and understand its carbon savings in relatable terms (e.g., "equivalent to X trees planted"), so that I can take actions based on these numbers.

Acceptance Criteria:

- Dashboards display at least three KPIs: energy, latency, maintainability.
- Dashboards are intuitive and require minimal training for non-technical users.
- Data granularity down to microservice/function/API level.

SMART Breakdown:

- **Specific:** Dashboards visualize trade-offs for each execution mode.
- **Measurable:** ≥3 KPIs; user feedback on usability.
- **Achievable:** Uses existing Wappsto visualization and new telemetry.
- **Relevant:** Supports informed decision-making.
- **Time-bound:** Prototype by M18.

FR5. The system shall instrument code and collect telemetry to estimate the energy cost of execution paths, correlating with real-time grid data to attribute CO_2 footprint to each operation.

Relevant User Stories:

- As a developer, I want to receive insights into the impact of my code changes on energy consumption and maintainability. By attributing CO_2 emissions to my specific operations, these operations become a tangible, gamified metric, encouraging me to write greener code.
- As a DevOps engineer, I want to integrate carbon footprint analysis into our CI/CD pipeline, so that pull-requests are automatically flagged if they introduce a significant regression in energy efficiency.

Acceptance Criteria:

- Telemetry framework and grid data API integration achieved by M12.
- CI/CD pipeline flags pull requests with significant energy regressions.
- Emissions attributed to individual microservices, function calls, or API endpoints.

SMART Breakdown:

- **Specific:** Code is instrumented for energy and CO_2 tracking.
- **Measurable:** Telemetry integrated by M12; CI/CD flags regressions.
- **Achievable:** Uses tools like JoularJX and energy APIs.
- **Relevant:** Drives greener development practices.
- **Time-bound:** Telemetry integration by M12.

FR6. The system shall generate auditable monthly emission reports for ESG officers, tracking progress against corporate sustainability targets.

Relevant User Stories:

- As an ESG officer, I want to generate auditable monthly emission reports from the platform, so I can track our progress against corporate sustainability targets.

Acceptance Criteria:

- Monthly reports are generated automatically and are auditable.
- All emission calculations are logged with source data.

- Reports align with corporate sustainability metrics.

SMART Breakdown:

- **Specific:** Monthly emission reports are generated and auditable.
- **Measurable:** 100% of calculations logged; reports generated monthly.
- **Achievable:** Uses system's logging and reporting features.
- **Relevant:** Supports ESG compliance and reporting.
- **Time-bound:** Reporting feature available by M15.

4.1.4 Non-Functional Requirements

NFR1. The system shall ensure transparency by allowing users to trace and understand the source of any sustainability metric shown, including grid intensity data and energy consumption estimates.

Acceptance Criteria:

- All metrics are accompanied by traceable source data.
- Users can access underlying data for any metric displayed.

SMART Breakdown:

- **Specific:** All metrics are traceable to source data.
- **Measurable:** 100% of metrics have accessible source data.
- **Achievable:** Uses system's data logging and UI features.
- **Relevant:** Builds trust and auditability.
- **Time-bound:** Feature available at initial deployment.

NFR2. The system shall protect all data collected for energy analysis under Wirtek's GDPR-compliant platform standards.

Acceptance Criteria:

- All data handling processes comply with GDPR.
- Security audits confirm compliance.

SMART Breakdown:

- **Specific:** Data is handled per GDPR standards.
- **Measurable:** 100% compliance in audits.
- **Achievable:** Uses existing GDPR-compliant infrastructure.
- **Relevant:** Ensures legal and ethical operation.
- **Time-bound:** Compliance confirmed before go-live.

NFR3. The system shall ensure that the overhead from measurement and telemetry remains under 5% of total resource use.

Acceptance Criteria:

- System resource monitoring confirms <5% overhead.
- Performance tests conducted before deployment.

SMART Breakdown:

- **Specific:** Measurement overhead <5%.
- **Measurable:** Overhead measured and reported.
- **Achievable:** Optimized telemetry implementation.
- **Relevant:** Maintains system performance.
- **Time-bound:** Confirmed before production deployment.

NFR4. The system shall be scalable to work across distributed environments with varying deployment sizes.

Acceptance Criteria:

- System tested in multiple deployment scenarios.
- Performance and functionality consistent across scales.

SMART Breakdown:

- **Specific:** Scalable across environments.
- **Measurable:** Successful tests in at least three deployment sizes.
- **Achievable:** Built on modular, distributed architecture.
- **Relevant:** Supports diverse customer needs.
- **Time-bound:** Scalability validated by M18.

NFR5. The system shall be extensible, allowing new trade-off modules and sustainability metrics to be integrated via a documented API or SDK.

Acceptance Criteria:

- API/SDK documentation is available.
- At least one new module integrated post-launch.

SMART Breakdown:

- **Specific:** Extensible via API/SDK.
- **Measurable:** At least one new module added.
- **Achievable:** Modular design and documentation.
- **Relevant:** Supports future growth.
- **Time-bound:** API/SDK available by M18.

NFR6. The system shall attribute emissions down to the level of an individual microservice, function call, or API endpoint.

Acceptance Criteria:

- Emissions data is available at microservice/function/API level.
- Dashboards and reports reflect this granularity.

SMART Breakdown:

- **Specific:** Emissions attributed at fine granularity.
- **Measurable:** Data available for all relevant components.
- **Achievable:** Uses telemetry and logging.
- **Relevant:** Enables actionable insights.

- **Time-bound:** Feature available by M15.

NFR7. The system shall log all emission calculations with their source data to ensure metrics are usable for corporate reporting.
Acceptance Criteria:
- 100% of emission calculations are logged.
- Logs are accessible for audits and reporting.

SMART Breakdown:
- **Specific:** All calculations logged with source data.
- **Measurable:** 100% logging rate.
- **Achievable:** Uses system's logging infrastructure.
- **Relevant:** Supports audit and compliance.
- **Time-bound:** Logging in place before first report.

NFR8. The system shall handle failures gracefully when external grid data APIs are unavailable, using last known data, a regional average, or flagging data as an estimate.
Acceptance Criteria:
- System continues to operate during API outages.
- Users are notified when data is estimated.

SMART Breakdown:
- **Specific:** Graceful degradation on API failure.
- **Measurable:** 100% uptime during outages; user notifications sent.
- **Achievable:** Implements fallback logic.
- **Relevant:** Ensures reliability.
- **Time-bound:** Feature available at launch.

NFR9. The system shall provide intuitive trade-off visualizations requiring minimal training for non-technical users.
Acceptance Criteria:
- User testing confirms ease of use.
- Training materials are minimal and clear.

SMART Breakdown:
- **Specific:** Visualizations are intuitive.
- **Measurable:** Positive user feedback; minimal training required.
- **Achievable:** User-centered design process.
- **Relevant:** Maximizes adoption and impact.
- **Time-bound:** Usability validated by M18.

4.2 UC-CLW: Energy Community Management

The Cleanwatts use case initially described in D2.1 has now been refined and expanded to integrate more speific sustainability-aware decision support into energy community management. Building on the Cleanwatts Living Lab, this updated description incorporates advanced AI forecasting and optimization, support for collective self-consumption (CSC) and flexibility participation, and systematic evaluation of optimization algorithms through sustainability trade-offs.

4.2.1 Summary

This UC focuses on enabling Renewable Energy Communities (RECs) to operate more efficiently, sustainably and transparently using the CleanwattsOS (Cleanwatts operative software). The system aggregates distributed energy resources (PV, BESS, EVs and flexible loads), optimizing their operation through AI-driven forecasting and scheduling.

Uniquely, the UC aims to evaluate not only system-level outcomes (self-consumption, CO_2 reduction, cost savings), but also the sustainability impacts of the AI optimizers themselves. Operators, prosumers, and regulators are supported with dashboards that show trade-offs between economic benefits and carbon impacts, ensuring decisions are both technically and environmentally sound.

The pilot will be validated in the Cleanwatts Living Lab, which integrates real households, PV units, EV chargers, and storage. This real-world testbed makes the evaluation of optimization algorithms both rigorous and practical.

Therefore, this UC addresses the pressing challenge that current optimization systems are cost-driven and rarely account for CO_2 emissions or the computational footprint of AI models. By combining CleanwattsOS with GECAD's infrastructure, the project aims to pioneer a responsible, Green AI approach to energy community management.

4.2.2 Updated User Stories

As a **community operator**, I want accurate forecasts of renewable generation and demand, so I can optimize resources and minimize costs and CO_2 emissions.

As a **prosumer**, I want my PV and storage integrated into the community system, so I can maximize savings and contribute to grid flexibility.

As a **researcher/operator**, I want to assess the trade-offs of different AI optimizers, so I can select methods that balance accuracy with sustainability.

As a **community operator**, I want to visualize the sustainability trade-offs of different AI optimizers, so I can select the most responsible optimization strategy for my community.

4.2.3 Funcional Requirements

FR1. The system shall provide AI-based forecasting and optimization tools to manage energy flows within the community, covering PV generation, storage, EVs, and demand response, while balancing sustainability and cost.

Relevant User Stories:

- As a community operator, I want accurate forecasts of renewable generation and demand, so I can optimize resources and minimize costs and CO_2 emissions.

Acceptance Criteria:

- Forecast accuracy improved ≥25% compared to baseline.
- Day-ahead and intra-day forecasts available.
- Optimization applied in ≥3 scenarios (PV-only, PV+storage, PV+storage+EV).
- Integrated into CleanwattsOS dashboards with <5s response time.

SMART Breakdown:

- Specific: Implement AI-based forecasting for PV, load, EV/storage.
- Measurable: ≥25% accuracy improvement (MAPE/RMSE).
- Achievable: Leverage Living Lab data and GECAD HIL infrastructure.
- Relevant: Improves REC sustainability and cost efficiency.

- Time-bound: Validated by M18.

FR2. The system shall support collective self-consumption and flexibility participation, including automated scheduling of storage and EV charging/discharging.

Relevant User Stories:

- As a prosumer, I want my PV and storage integrated into the community system, so I can maximize savings and contribute to grid flexibility.
- As a community operator, I want to automatically coordinate community assets (PV, BESS, EVs) so that self-consumption is maximized, and the community can participate in flexibility services without requiring manual interventions.

Acceptance Criteria:

- Automated dispatch schedules for storage/EVs.
- Participation in at least one simulated/local flexibility market.
- Reports show ≥10% increase in renewable self-consumption vs. baseline.

SMART Breakdown:

- Specific: Enable coordination of PV, BESS, EVs for CSC and flexibility.
- Measurable: ≥10% increase in renewable self-consumption.
- Achievable: Built on CleanwattsOS VPP (Kiplo software).
- Relevant: Supports EU directives on CSC and flexibility.
- Time-bound: Pilot validated by M24.

FR3. The system shall evaluate the sustainability impacts of different AI optimizers, benchmarking trade-offs between accuracy, cost, and computational energy.

Relevant User Stories:

- As a researcher/operator, I want to assess the trade-offs of different AI optimizers, so I can select methods that balance accuracy with sustainability.
- As a REC operator, I want to understand how different optimizers impact community energy costs and emissions, so I can choose the most suitable strategy for my community.

Acceptance Criteria:

- At least 4 models benchmarked: 2 optimization algorithms (e.g., reinforcement learning, heuristic optimization) and 2 forecasting algorithms (e.g., ANN, ML).
- Reporting of the achieved results.
- Energy consumption measured with telemetry.
- Benchmark reports available in trade-off dashboard.

SMART Breakdown:

- Specific: Compare ANN, ML, and other methods on accuracy vs. energy use.
- Measurable: ≥3 models benchmarked on ≥3 KPIs.
- Achievable: Use GECAD HIL + Green Software Foundation metrics.
- Relevant: Ensures responsible AI deployment.
- Time-bound: Implementation and reporting by M18. Trade-off dashboard completed by M30.

FR4. The system shall provide dashboards that visualize trade-offs of different AI optimizers and their impact on community energy management, showing metrics such as model accuracy, computational/energy cost, CO_2 emissions, and performance outcomes.

Relevant User Stories:

- As a community operator, I want to visualize the sustainability trade-offs of different AI optimizers, so I can select the most responsible optimization strategy for my community.
- As a researcher, I want to compare the performance and sustainability metrics of multiple optimizers, so I can benchmark their effectiveness and publish validated insights on responsible AI in energy systems.

Acceptance Criteria:

- Dashboards show ≥4 KPIs (accuracy, runtime energy use, CO_2 impact, cost).
- Side-by-side comparison of ≥4 AI optimizers.
- Real-time and historical views available.
- Reports exportable for regulatory use.
- GDPR-compliant handling confirmed.

SMART Breakdown:

- Specific: Dashboards must allow visualization of the different optimizers trade-offs, showing at least energy use, CO_2 emissions, model accuracy, and community cost impact, with side-by-side comparison and basic filtering (timeframe, optimizer type).
- Measurable: Dashboards must display at least 4 KPIs and update results within 5 minutes of model execution.
- Achievable: CleanwattsOS dashboards with GECAD's HIL infrastructure, telemetry tools, and Green Software Foundation benchmarking methods.
- Relevant: Empowers operators and researchers to select the most sustainable AI optimizer, supporting responsible AI deployment and Green AI principles.
- Time-bound: Dashboard validated with ≥4 optimizers by M30.

FR5 — The platform shall provide end-users (consumers and prosumers) of the REC with personalized dashboards that translate energy, cost, and CO_2 savings into understandable terms.

Relevant User Stories:

- As a consumer/prosumer, I want to see my energy and CO_2 savings in simple terms, so I can understand my contribution and remain engaged with the community.
- As a community operator, I want to share aggregated sustainability reports with members, so I can strengthen participation and trust.

Acceptance Criteria:

- Dashboard displays at least 3 KPIs
- Results visualized at both household and community levels.
- Comparisons available against baseline and peers.

SMART Breakdown:

- Specific: Provide personalized sustainability insights dashboards for consumers/prosumers.
- Measurable: ≥3 KPIs (e.g., kWh saved, € saved, CO_2 avoided).; household + community levels; peer comparisons.
- Achievable: Uses CleanwattsOS visualization modules for end-users (Kiome).

- Relevant: Boosts engagement and awareness.
- Time-bound: Dashboard validated with pilot users by M30.

4.2.4 Non-Functional Requirements

NFR1 -The system shall provide fast and responsive interactions for both operators and end-users, ensuring that dashboards, forecasts, and optimization results are delivered within acceptable time limits under normal operating conditions.

Acceptance Criteria: 95% of user interactions return results within a few seconds (less than 5).

NFR2 - The system shall ensure secure handling of community and user data, prevent unauthorized access and comply with relevant data protection regulations.

Acceptance Criteria: All data access is authenticated and logged; GDPR compliance verified.

NFR3 - The system shall maintain a high level of service continuity, recovering quickly from faults and minimizing downtime in both single-REC and multi-REC operations.

Acceptance Criteria: System availability ≥99%, with recovery mechanisms in place for critical failures.

NFR4 - The system shall scale to support an increasing number of households and devices as well as growing volumes of time-series data, without significant performance degradation.

Acceptance Criteria: Performance remains stable as new households and devices are added.

NFR5 - The system shall provide intuitive and user-friendly dashboards and tools for different stakeholders (operators, researchers, prosumers), requiring minimal training for effective use.

Acceptance Criteria: Positive feedback in user evaluations; training time kept to a minimum.

NFR6 - The system shall interoperate with external platforms, data sources, and grid systems (e.g., DSOs, market platforms, weather APIs) using standardized interfaces and protocols.

Acceptance Criteria: Demonstrated data exchange with at least one external system without custom integration.

NFR7 - The system shall be designed to accommodate new optimization algorithms, forecasting models, or sustainability metrics without requiring major redesign.
Acceptance Criteria: At least one new algorithm or metric can be integrated post-pilot with minimal effort.

NFR8 - The system shall ensure transparency of AI decisions and optimization results, allowing stakeholders to trace outputs back to data inputs and underlying models.
Acceptance Criteria: All forecasts, optimizations, and KPIs include traceable logs and metadata.

NFR9 — The system shall be maintainable and updatable, enabling fixes, upgrades, and configuration changes with minimal downtime.
Acceptance Criteria: Routine updates can be applied without interrupting normal community operations.

4.3 UC-CPP: Sustainable System Control

Canon Production Printing Netherlands, together with its consortium partners, is driving a transition in the digital printing industry toward more sustainable, long-lifetime systems. To support this shift, a set of functional requirements has been defined that guide the development of software and tooling aimed at improving system maintainability, energy efficiency, and data-driven decision-making. While use cases continue to evolve with CPP and other Dutch consortium partners, functional, non-function, and system and technical requirements remain provisional and may be revised as new insights emerge. Such changes will be reflected upon in future versions of this deliverable.

4.3.1 Summary

The functional requirements outlined for Canon Production Printing's sustainability initiative reflect a strategic shift toward modular, data-driven, and environmentally conscious system design. These requirements are not just technical specifications—they represent a broader ambition to transform how high-volume digital presses are maintained, upgraded, and operated over time.

At a general level, the requirements focus on enabling continuous modular upgrades to existing printing systems. This means that instead of replacing entire machines every few years, customers can implement targeted improvements that enhance sustainability—such as reducing energy consumption or waste—without interrupting operations. Acceptance criteria for this pilot deployment include the ability to deploy upgrades to a significant portion of the installed base within a defined timeframe, and ensuring visibility into upgrade history and system configuration. An additional effect of this new approach is that scope 3 emissions will be reduced as well.

Another major theme is monitoring and optimization of power and waste. Requirements specify that systems should provide real-time and historical data on energy usage, allow for job scheduling that minimizes power draw, and offer customizable algorithms to reduce material waste. These features are validated through dashboards, measurable reductions in waste, and the availability of scheduling tools for multiple common job types.

The system is also expected to include predictive diagnostics that leverage sensor data and machine learning to anticipate issues and guide both customers and R&D teams. These diagnostics are tied to sustainability KPIs, such as energy efficiency and waste reduction, and must generate automated alerts and recommendations. This ensures that both operational and design decisions are informed by real-world data.

Finally, the requirements support early-stage R&D evaluation of sustainability trade-offs. Tools are to be developed that help engineers assess the impact of design choices before products reach the market. These tools must be capable of simulating different scenarios and feeding insights back into the development process, reinforcing a feedback loop between field data and product innovation.

Together, these requirements establish a foundation for a more sustainable, intelligent, and responsive printing ecosystem—one that aligns with both customer needs and broader environmental goals.

4.3.2 Functional Requirements

FR1. The system shall support continuous modular upgrades in the field, enabling sustainability improvements without full hardware replacement cycles.
Relevant User Stories:

- As a field technician, I can view upgrade history and current configuration of a printer, so that I can plan and execute field upgrades effectively.
- As a customer sustainability officer, I can assess the impact of field upgrades on energy efficiency and waste reduction, so that I can report on sustainability improvements.

Acceptance Criteria:

- Once released to Canon's sales regions, modular upgrades can be deployed to at least 25% of the field population within 12 months.
- Upgrade history and configuration are accessible via dashboard.

SMART Breakdown:

- **Specific**: Modular upgrades replace full hardware cycles.
- **Measurable**: ≥25% field deployment; dashboard access.
- **Achievable**: Built on modular architecture.
- **Relevant**: Reduces carbon footprint per application, extends product life.
- **Time-bound**: 12 months for deployment.

FR2. The system shall provide advanced monitoring and optimization of power usage.
Relevant User Stories:

- As a customer job planner, I can schedule jobs together in such a way as to minimize power consumption, so that our operations align with environmental goals.
- As a customer printer operator, I can receive alerts about unusual power consumption patterns, so that I can react to correct suboptimal engine use.

Acceptance Criteria:

- Power management dashboards display real-time and historical data.

- Scheduling feature available for at least 2 types of jobs within 12 months of field deployment.

SMART Breakdown:

- **Specific**: Advanced monitoring and adaptive scheduling.
- **Measurable**: Scheduling for 2+ job types.
- **Achievable**: Uses sensor data and analytics.
- **Relevant**: Reduces operational carbon footprint.
- **Time-bound**: Within 12 months of field deployment.

FR3. The system shall implement smarter, customizable waste rejection algorithms and dashboards to help customers minimize waste while maintaining quality.

Relevant User Stories:

- As a customer job planner, I can adjust rejection thresholds based on job type and customer quality requirements, so that I minimize waste while maintaining acceptable output quality.
- As a customer sustainability officer, I can view waste statistics and trends on a dashboard, so that I can identify opportunities for improvement and track progress.

Acceptance Criteria:

- Customizable rejection algorithms available for at least 2 job types.
- Waste dashboards show trends and actionable insights.
- Demonstrated ≥10% reduction in waste for pilot customers.

SMART Breakdown:

- **Specific**: Customizable waste rejection and dashboards.
- **Measurable**: ≥10% waste reduction; 2+ job types.
- **Achievable**: Uses advanced diagnostics and analytics.
- **Relevant**: Reduces material waste.
- **Time-bound**: 12 months for pilot results following deployment.

FR4. The system shall enable advanced supervisory control, diagnostics, and dashboarding to inform customers and R&D about sustainability impacts and opportunities for improvement.

Relevant User Stories:

- As a customer printer operator, I can access a diagnostics dashboard that explains system behaviour and suggests improvements, so that I can operate the printer more sustainably.
- As an R&D system engineer, I can restructure sensor data into actionable insights, so that customers can make informed decisions about printer usage.

Acceptance Criteria:

- Dashboards provide actionable insights for at least 3 sustainability KPIs.
- Diagnostics available for both customers and R&D.

SMART Breakdown:

- **Specific**: Supervisory control and diagnostics dashboards.
- **Measurable**: 3+ KPIs; dual access (customer/R&D).
- **Achievable**: Built on existing sensor infrastructure.
- **Relevant**: Supports sustainable operation and design.
- **Time-bound**: Prototype by M18.

FR5. The system shall integrate predictive diagnostics using sensor data and analytics to guide both customers and R&D, linked to sustainability KPIs.
Relevant User Stories:

- As an R&D system engineer, I can implement smarter rejection algorithms that consider historical data and sensor feedback, so that waste is reduced without compromising quality.
- As a customer printer operator, I can receive alerts about unusual power consumption patterns, so that I can react to correct suboptimal engine use.

Acceptance Criteria:

- Predictive diagnostics available for at least 2 sustainability KPIs.
- Alerts and recommendations generated automatically.

SMART Breakdown:

- **Specific**: Predictive diagnostics for sustainability.
- **Measurable**: 2+ KPIs; automated alerts.
- **Achievable**: Uses machine learning on sensor data.
- **Relevant**: Reduces waste and energy loss.
- **Time-bound**: Feature available by M18.

FR6. The system shall provide actionable insights and feedback to customers to encourage more sustainable use of printing systems.
Relevant User Stories:

- As a customer sustainability officer, I can generate reports from the dashboard showing customer behaviour trends, so that I can support sustainability initiatives with data.

Acceptance Criteria:

- Feedback mechanisms integrated into dashboards.
- At least 2 types of actionable recommendations provided.

SMART Breakdown:

- **Specific**: Actionable feedback for sustainable use.
- **Measurable**: 2+ recommendation types.
- **Achievable**: Uses analytics and reporting tools.
- **Relevant**: Drives sustainable customer behaviour.
- **Time-bound**: Feature available by M18.

FR7. The system shall provide tools and dashboards to help evaluate sustainability impacts of design decisions early in the R&D process.
Relevant User Stories:

- As an R&D architect, I can define research directions for intelligent scheduling and control systems, so that future products can better balance productivity and sustainability.

Acceptance Criteria:

- Trade-off evaluation tools available for at least 2 design scenarios.
- R&D feedback loop established.

SMART Breakdown:

- **Specific**: Early-stage trade-off evaluation tools.

- **Measurable**: 2+ scenarios; feedback loop.
- **Achievable**: Uses simulation and analytics.
- **Relevant**: Informs sustainable product design.
- **Time-bound**: Tools available by M18.

4.3.3 Non-Functional Requirements

Non-Functional Requirements
NFR1. The system shall provide intuitive interfaces and dashboards requiring minimal training for diverse users (operators, planners, R&D).
Acceptance Criteria: User testing confirms ease of use; training materials are minimal and clear.

NFR2. The system shall ensure secure, role-based access to dashboards and data, protecting customer and operational information.
Acceptance Criteria: Security audits confirm compliance; role-based access implemented.

NFR3. The system shall be scalable to support a large, diverse field population with varying configurations and upgrade histories.
Acceptance Criteria: System tested in at least 3 deployment scenarios; consistent performance.

NFR4. The system shall maintain a modular architecture to support long-term upgrades and minimize technical debt.
Acceptance Criteria: Modular upgrades deployed without major system downtime.

NFR5. The system shall provide high availability and fault tolerance, especially for critical monitoring and control functions.
Acceptance Criteria: ≥99% uptime for dashboards and control systems at ORS connected customers.

NFR6. The system shall adapt to changing customer usage patterns using machine learning.
Acceptance Criteria: Adaptive features validated in at least 2 customer scenarios.

NFR7. The system shall support seamless, minimally disruptive, and backward-compatible field upgrades.
Acceptance Criteria: Upgrades completed with <3 days downtime in pilot tests.

NFR8. The system shall align with CSRD, EU Green Deal, and relevant sustainability standards.
Acceptance Criteria: Compliance confirmed by internal audit.

---

5 Compliance & Alignment

The requirements defined in this deliverable are designed to ensure full alignment with European sustainability directives, relevant research outputs, and industrial best practices. This alignment guarantees that the MAST project not only meets its internal objectives but also contributes to broader regulatory, scientific, and societal goals.

5.1 Corporate Sustainability Reporting Directive (CSRD)

All system components and development practices must support compliance with the CSRD, which mandates transparent reporting of environmental, social, and governance (ESG) metrics for software-intensive systems. This includes:

- Traceable energy and carbon metrics
- Integration of sustainability indicators into CI/CD workflows
- Support for auditability and reporting dashboards

The CSRD is a cornerstone of the EU's sustainability strategy and directly influences the design of MAST's monitoring and reporting tools (European Commission, 2022).

5.2 EU Green Deal Objectives

The MAST project contributes to the EU Green Deal by enabling measurable reductions in carbon emissions and energy consumption across digital systems. Requirements are aligned with the Green Deal's goals of:

- Decarbonization of ICT infrastructure
- Promotion of circular economy principles through modular upgrades

- Empowerment of end-users to make sustainable choices via eco-feedback systems

These objectives are operationalized through use case-specific KPIs and validated in WP4 and WP5 pilots (Gill & Buyya, 2018; Caron et al., 2022).

5.3 Alignment with European Research Outputs

MAST builds upon and extends the findings of prior European research initiatives, including:

- **SDK4ED (H2020)**: Provided foundational tools for technical debt and energy efficiency management but lacked integrated trade-off analysis (Ampatzoglou et al., 2018).. MAST addresses this gap by combining both dimensions in a unified framework.
- **VISDOM (ITEA)**: Advanced software visualization techniques, which MAST adapts for sustainability dashboards and decision support (Fernández-Sánchez & Sepúlveda-Escribano, 2020).
- **SEAS and FUSE-IT (ITEA)**: Focused on energy optimization in smart systems. MAST generalizes these approaches to software-level sustainability analytics and cross-domain applicability (Van Beek et al., 2020).

By leveraging these outputs, MAST ensures methodological rigor, technological continuity, and cross-project interoperability.

5.4 Consortium-Wide Standards and Practices

All requirements are harmonized across the consortium use case partners (WRT, CLW, CPP) to ensure consistency in implementation, validation, and reporting. This includes:

- Use of shared metrics and KPIs
- Adoption of common architectural patterns
- Integration of sustainability tooling into partner-specific DevOps environments

This harmonization supports scalability, replicability, and long-term impact across industrial domains.

6 Conclusion & Next Steps

This deliverable establishes a comprehensive and traceable requirements foundation for the MAST project, synthesizing insights from industrial use cases (D2.1_Use Case Descriptions), state-of-the-art research (D2.2 Sustainability state of the art and practice), and strategic objectives (FPP of MAST). By categorizing requirements into functional, non-functional, and aligning them with European sustainability directives, this specification ensures that MAST innovations are both technically rigorous and policy-compliant.

The requirements defined herein support the project's dual sustainability goals:

- **Technical sustainability**, through maintainability, modularity, and reduced technical debt.
- **Environmental sustainability**, through energy efficiency, carbon footprint reduction, and eco-feedback mechanisms.

These requirements will guide the design and development activities in the following work packages:

- **WP3 – Concept Development**: Translate requirements into architectural designs, sustainability-aware tooling, and optimization frameworks.
- **WP4 – Tooling and Integration**: Implement and integrate the tools into industrial environments, ensuring usability, scalability, and compliance.
- **WP5 – Pilots and Validation**: Deploy and evaluate the tools in real-world settings across the three use cases, validating KPIs and refining solutions based on stakeholder feedback.

6.1 Next Steps:

1. **Architecture Design (D2.4)**: Develop the MAST reference architecture, mapping requirements to system components and interfaces.
2. **Tool Development (WP3)**: Begin implementation of sustainability dashboards, telemetry frameworks, and trade-off analysis tools.
3. **Pilot Planning (WP4)**: Define pilot scenarios and validation methodologies for each use case.
4. **Continuous Refinement**: Update requirements iteratively based on feedback from WP3–WP5 and evolving regulatory landscapes.

By embedding sustainability as a first-class engineering concern, MAST aims to deliver measurable improvements in software maintainability and environmental performance, contributing to the EU Green Deal and setting a precedent for future digital sustainability initiatives.

7 Bibliography

- Ampatzoglou, A., Avgeriou, P., & Chatzigeorgiou, A. (2016). The financial aspect of managing technical debt: A systematic literature review. *IEEE Transactions on Software Engineering, 43*(8), 659–678.
- Ampatzoglou, A., et al. (2018). Technical debt forecasting: A systematic literature review. *Information and Software Technology, 98*, 35–61.
- Avgeriou, P., et al. (2016). Managing technical debt in software engineering. *ACM Computing Surveys, 49*(4), 1–34.
- Beek, D. A. van, et al. (2020). Trade-off analysis in embedded systems design: A model-checking approach. *Journal of Systems and Software, 162*, 110509.
- Caron, E., et al. (2022). Carbon footprint of cloud computing: Measurement challenges and perspectives. *Future Generation Computer Systems, 133*, 145–158.
- Deb, K. (2014). *Multi-objective optimization using evolutionary algorithms*. Wiley.
- European Commission. (2022). *Corporate Sustainability Due Diligence Directive*.
- Fernández-Sánchez, C., & Sepúlveda-Escribano, J. (2020). Software visualization techniques for sustainability metrics: A systematic review. *Empirical Software Engineering, 25*(2), 1234–1258.
- Gill, S., & Buyya, R. (2018). Sustainable cloud computing: Foundations and future directions. *Journal of Cloud Computing, 7*(1), 1–29.
- Izurieta, C., et al. (2017). Organizing Technical Debt Prioritization: A Value-Based Approach. *Journal of Systems and Software, 137*, 188–205.
- Koziolek, H. (2011). Sustainability evaluation of software architectures. *Proceedings of the Joint ACM SIGSOFT Conference*.
- Schmidt, R., et al. (2021). Continuous sustainability assessment in DevOps. *IEEE Software, 38*(6), 45–53.