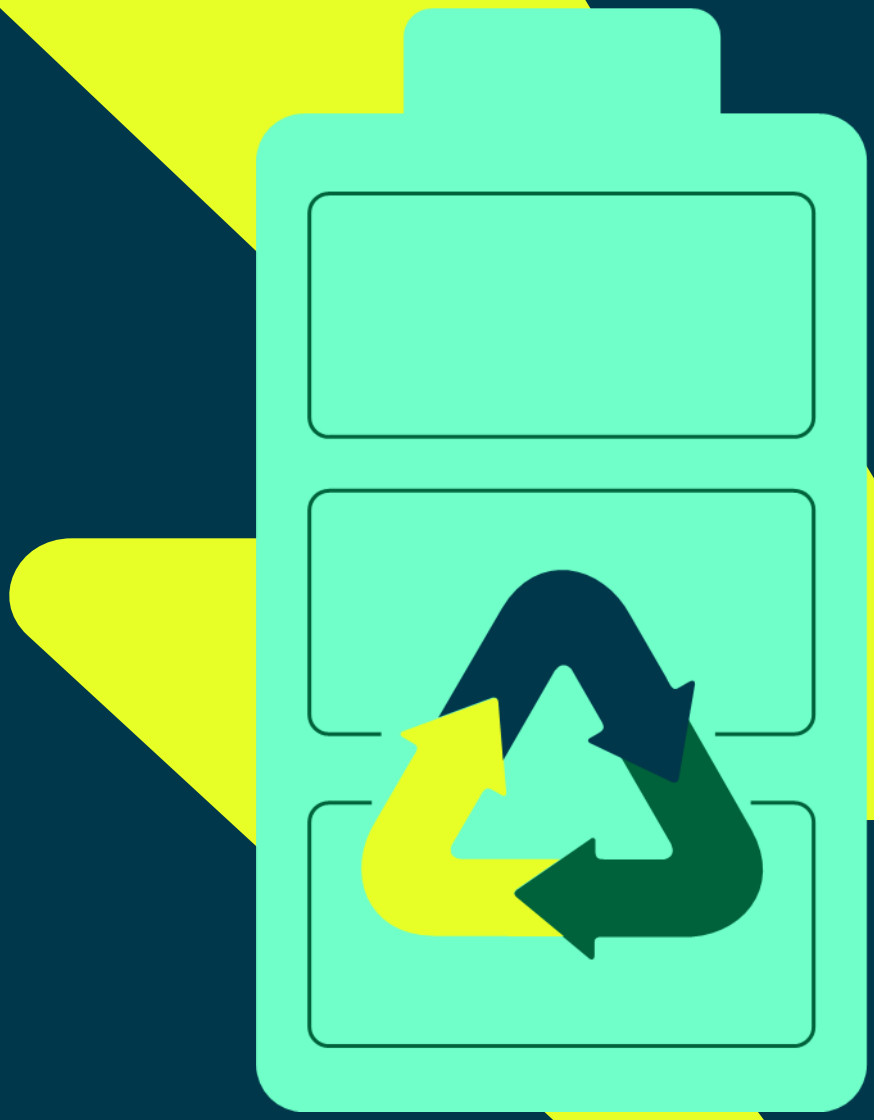
Deliverable D2.10

Battery Passport Platform (Version III)



Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

Feasible Recovery of critical raw materials through a new circular Ecosystem for a Li-Ion Battery cross-value chain in Europe

WP2 - End-of-life LIBs Collection and Characterisation, Digital Tools and Battery Passports deployment

D2.10 - Battery Passport Platform (version III)

Due date of deliverable	**Actual submission date:**	Organisation name responsible for this deliverable: EURECAT
31/08/2026	31/08/2026	

Dissemination Level

SEN	Sensitive	
PU	Public	X

Project Acronym
FREE4LIB

Grant Agreement No.
101069890

Project Start Date
01-09-2022

Project End Date
31-08-2026

Duration
48 months

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

Disclaimer

Free4lib is an EU-funded project that has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no. 101069890. The sole responsibility for the content of this report lies with the authors. It does not necessarily reflect the opinion of the European Union. The European Commission is not responsible for any use that may be made of the information contained therein.

Versions

DATE	VERSION	AUTHOR	COMMENT
30/06/2026	1	EURECAT	first draft of the document
22/07/2026	2	EURECAT, UNIGRAZ	Review
20/08/2026	3	EURECAT	final document
31/08/2026	4	CARTIF	Final review and version

Content

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

List of Abbreviations

ACRONYMS	DESCRIPTION
ABI	Application Binary Interface
API	Application Programming Interface
BOL	Beginning of Life
BP	Battery Passport
BPP	Battery Passport Platform
CRM	Critical Raw Material
CSS	Cascading Style Sheets
DBP	Digital Battery Passport
DTO	Data Transfer Object
EOA	Externally Owned Account
EOL	End of Life
EIP	Ethereum Improvement Proposal
ERP	Enterprise Resource Planning
EU	European Union
EV	Electric Vehicle
EVM	Ethereum Virtual Machine
FR	Functional Requirement
HTTP	Hypertext Transfer Protocol
IPR	Intellectual Property Rights
JSON	JavaScript Object Notation
JWT	JSON Web Token
KMA	Key Management Agent
LIB	Lithium-Ion Battery
M2M	Machine to Machine
MES	Manufacturing Execution System
NMC	Nickel Manganese Cobalt

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

OBJ	Objective
PoC	Proof of Concept
QR	Quick Response
REST	Representational State Transfer
SDK	Software Development Kit
SoH	State of Health
SLA	Service-Level Agreement
TRL	Technology Readiness Level
TS	TypeScript
TX	Transaction
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience
WP	Work Package

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

Executive summary

This deliverable, D2.10 — Battery Passport Platform (Version III), is the third and final report on the design, development and validation of the FREE4LIB Battery Passport Platform (BPP). It closes the development cycle started in D2.8 (Version I, M24) and continued in D2.9 (Version II, M30) and corresponds to the final milestone of Task T2.4, Battery Passport deployment, led by Eurecat with the support of AVL, UNIGRAZ, CARTIF, ERION and IREC-CERCA.

Across the three iterations, T2.4 has focused on a single research question: whether blockchain technology genuinely adds value to a battery passport platform once usability, key management, complexity and operational cost are taken seriously into account. Version I built the foundations — a Solidity smart contract on an Ethereum Virtual Machine (EVM)-compatible network, an API, and a minimal web interface — and validated that the basic mechanics of a blockchain-anchored passport were technically feasible. Version II tackled the most visible barrier to adoption by introducing custom meta-transactions, allowing users to interact with the chain without holding cryptocurrency, redesigning the frontend around role-based dashboards, and adding a hybrid backend with MongoDB and a blockchain oracle. Version III, documented in this deliverable, completes the architecture with a Key Management Agent — a system-to-system component that enables organisations to integrate the BPP without changing their internal applications and without ever exposing their cryptographic keys to the platform — and consolidates the full stack as a publicly accessible deployment at https://free4lib.eurecatsecurity.org/.

Beyond the agent, Version III brings significant refinements to the data model, particularly in the areas of material composition (now structured to align with the categories of the EU Battery Regulation, including critical raw materials, hazardous substances, recycled content and carbon footprint indicators) and battery State of Health (SoH). The smart contracts have been hardened with EIP-712 typed-data signatures, expanded lifecycle state machines that explicitly model the SecondLife stage, and a security tooling pipeline based on Slither (a static analyser for Solidity source) and Mythril (a symbolic-execution engine for EVM bytecode), together with property-based testing. The backend has been hardened for production with rotating refresh tokens, login lockouts, trace identifiers, observability metrics and strict environment validation. The frontend has received a redesigned material composition view and an improved battery passport visualizer.

The platform has been deployed in a real production environment and seeded with a comprehensive synthetic dataset that exercises every code path: cell creation by manufacturers, module and pack assembly by assemblers, SoH recording, recycling and second-life assignment by recyclers, public passport visualization through QR codes, and machine-to-machine integration through the agent. In addition, a real assessment dataset of eleven battery packs

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

defined by AVL was ingested into the live platform and used to validate the analytical assessment of ST2.4.3 end to end (Section 5.6). What was not onboarded within the project lifetime is continuous field telemetry from a specific industrial partner fleet; that remains the one open item of the real-environment validation.

The analytical assessment of battery passport data foreseen in subtask ST2.4.3 has been completed by AVL and is reported in a dedicated chapter of this deliverable (Chapter 4). It consolidates the SoH health parameters and the acceptance-criteria framework agreed between AVL and Eurecat, it has been implemented inside the live platform by Eurecat, and it has been validated against a real AVL dataset of eleven battery packs, as described in Section 5.6.

The most distinctive contribution of this deliverable, however, is its final critical assessment. Rather than presenting the platform as a self-evident success, this document closes with a frank, fact-based evaluation of whether the blockchain-based architecture pays its complexity tax. Each mitigation introduced across the three prototypes — custodial wallets with client-side decryption, meta-transactions, off-chain oracles, and now agents — is analysed in terms of the problem it solves and the problem it creates. The conclusion is deliberately uncomfortable: in the closed scope of FREE4LIB, with a single trusted custodian, no adversarial parties and no live data exchange with external systems, the value added by blockchain over a well-designed conventional database with cryptographic audit trails is marginal. The technology becomes meaningfully advantageous only in scenarios that the project did not, and could not, reach: a multi-party network operated under shared governance — for example, a registry coordinated by the European Union across member states — with adversarial verification between independent organisations and automated cross-organisation workflows enforced by smart contracts.

This deliverable should therefore be read in two complementary ways. As a technical report, it documents the final state of a working blockchain-anchored battery passport platform, the engineering decisions behind it, and the new components delivered in Version III. As a research output, it provides an honest answer to the research question of T2.4 and a set of recommendations for any follow-on work, EU initiative, or industrial actor considering the same technological path.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

1. Introduction

1.1 Background

The FREE4LIB project (Feasible Recovery of critical raw materials through a new circular Ecosystem for a Li-Ion Battery cross-value chain in Europe) is a Horizon Europe research and innovation action funded by the European Union under grant agreement number 101069890. The project addresses one of the most strategically important challenges in the European energy and mobility transition: how to ensure that the lithium-ion batteries (LIBs) that power electric vehicles, stationary storage and industrial applications can be recovered, reused and recycled in a way that is environmentally sound, economically viable and aligned with the regulatory framework that the European Union is now putting in place around batteries.

Within FREE4LIB, Work Package 2 focuses on End-of-life LIBs collection and characterisation, digital tools and battery passports deployment. Inside this work package, Task T2.4 — Battery Passport deployment, led by Eurecat with the participation of AVL, UNIGRAZ, CARTIF, ERION and IREC-CERCA, is responsible for designing, developing and validating a blockchain-based Battery Passport Platform (BPP). The task is structured around three subtasks: ST2.4.1 covers the design and development of the platform; ST2.4.2 covers integration with FREE4LIB processes and deployment in a distributed environment; ST2.4.3 covers the analytical assessment of the data managed by the platform, with a particular focus on State of Health, second-life decisions and lifecycle optimization, and is led by AVL.

T2.4 is documented through three deliverables. D2.8 (Version I, due M24) documented the foundational platform and the design choices behind it. D2.9 (Version II, due M30) documented the introduction of meta-transactions, the role-based frontend and the hybrid on-chain/off-chain backend. The present document, D2.10 (Version III, due M48), is the final report on the platform and on the task as a whole. It documents the components added in this third iteration, consolidates the architectural picture, presents the validation results and — most importantly — provides a final, evidence-based assessment of the role of blockchain technology in a battery passport context.

The Battery Passport itself is conceptually defined within FREE4LIB by Task T5.4. As a digital tool, a Battery Passport is intended to act as the unique digital identity of a battery, carrying information about its composition, manufacture, lifecycle events, condition and end-of-life handling. In its more ambitious form,

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

it is described as a digital twin, capable of dynamic updates throughout the battery's life. In its more practical regulatory form, it is the data record that the EU Battery Regulation and forthcoming digital product passport schemes require manufacturers, importers and recyclers to make available to specific stakeholders. The platform developed within T2.4 is designed to support both readings.

1.2 Motivation

The motivation behind T2.4 has remained unchanged across all three iterations and is rooted in the same combination of environmental, economic and regulatory pressures that motivates the FREE4LIB project as a whole. Lithium-ion batteries are at the centre of Europe's transition away from fossil fuels in mobility and stationary applications, but the linear take-make-dispose model that has characterised most of their first-generation deployment is no longer sustainable. Critical raw materials such as lithium, cobalt, nickel and natural graphite are concentrated in a small number of jurisdictions, the recycling technologies currently in widespread use recover only a fraction of these materials, and the carbon footprint of battery manufacturing remains high. A circular economy for batteries is therefore not optional; it is a strategic and regulatory requirement.

In this context, the role of a Battery Passport Platform is to make the data necessary for circular decisions visible, trustworthy and actionable. Visibility means that all relevant stakeholders — manufacturers, assemblers, second-life integrators, recyclers, regulators and, in some cases, end users — can access the information they need without depending on bilateral data exchanges or proprietary databases. Trustworthiness means that the data cannot be silently altered after the fact and that its provenance can be verified. Actionability means that the data is structured, normalised and accessible through APIs and user interfaces that support concrete operational decisions, such as whether a given battery should be sent to recycling or evaluated for a second life.

Blockchain technology was selected at the start of T2.4 as a candidate substrate for these properties because it offers, in principle, an immutable and verifiable record of events that does not require any single party to be trusted unconditionally. The three iterations of the platform have served to test, in practice, whether this principle translates into measurable benefit once usability, key management, integration with conventional systems and operational cost are factored in. This deliverable presents the conclusion of that test.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

1.3 Objectives of Deliverable D2.10

Deliverable D2.10 has four main objectives:

- **Document the third and final version of the Battery Passport Platform**, including all the components added or refined since D2.9, with sufficient technical depth to make the design decisions reproducible and auditable.
- **Present the Key Management Agent** as the headline new development of this iteration, explaining the problem it addresses, its architecture, its security model and its implications for the integration of the platform with company systems and for the operational scalability of the BPP.
- **Validate the platform** against the functional requirements defined for Version III and report on its deployment in a real production environment, including the limitations and the data that has not been possible to onboard during the project's lifetime.
- **Provide a critical, fact-based assessment** of the role of blockchain technology in a battery passport context, taking into account the full complexity introduced by every mitigation built across the three prototypes, and producing a recommendation for any follow-on work or EU initiative considering the same architectural path.

Unlike D2.8 and D2.9, which were primarily focused on documenting incremental development progress, D2.10 also has an explicit synthesis objective. As the final deliverable of T2.4, it must close the loop opened at the start of the task, summarise what was promised and what was delivered, and leave a record that is useful both inside and outside the FREE4LIB consortium.

1.4 Position of Version III within Task T2.4

Task T2.4 was designed as a three-step development process aligned with the three deliverables of the task. Version I was a controlled-environment proof of concept, intended to validate the most basic technical premises: that a blockchain-anchored data model for batteries was feasible, that smart contracts could enforce role-based access and lifecycle rules, that an API and a user interface could be built around the chain, and that QR codes could be used as a public entry point for battery information. Version II was a usability- and architecture-focused iteration, intended to remove the most visible adoption barriers — gas fees, blockchain account management, the lack of structured dashboards — and to introduce a cleaner separation between on-

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

chain and off-chain concerns through the introduction of an oracle and a MongoDB read model.

Version III, by contrast, is the production iteration. Its primary goal is no longer to prove that a particular feature can be built, but to bring the platform to a state in which it can be deployed, used continuously, integrated by third parties and assessed honestly. Three classes of work follow from that goal. The first class is the Key Management Agent, which addresses the last remaining usability barrier: the requirement for human users to interact with the platform through a web interface in order to register data, which is incompatible with the way real industrial environments produce battery data. The second class is the set of refinements that bring the data model and the user-facing components into alignment with the categories of the EU Battery Regulation. The third class is operational hardening: production deployment, security tooling, observability, and the rigorous validation of the full platform end to end.

Once these three classes of work have been described, the deliverable turns to the assessment that gives the task its meaning. T2.4 did not aim to deploy a battery passport for industrial production within the FREE4LIB consortium; that was beyond both the scope and the resources of the project. T2.4 was intended to explore, deploy and validate a blockchain-based platform in order to answer the same research question in plainer terms: does the technology pay for itself in this context? This deliverable answers that question.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

2. From Version II to Version III

This chapter provides a brief, deliberately compact recap of the previous iterations of the platform and the architectural picture they left at the end of D2.9. It then identifies the limitations that motivated the development of Version III and the design drivers that shaped this third iteration. The intent is to give a reader who is approaching D2.10 without having read the previous deliverables enough context to follow the rest of the document, while avoiding the duplication of material that has already been published in D2.8 and D2.9. The overall trajectory is summarised in Figure 2.1.

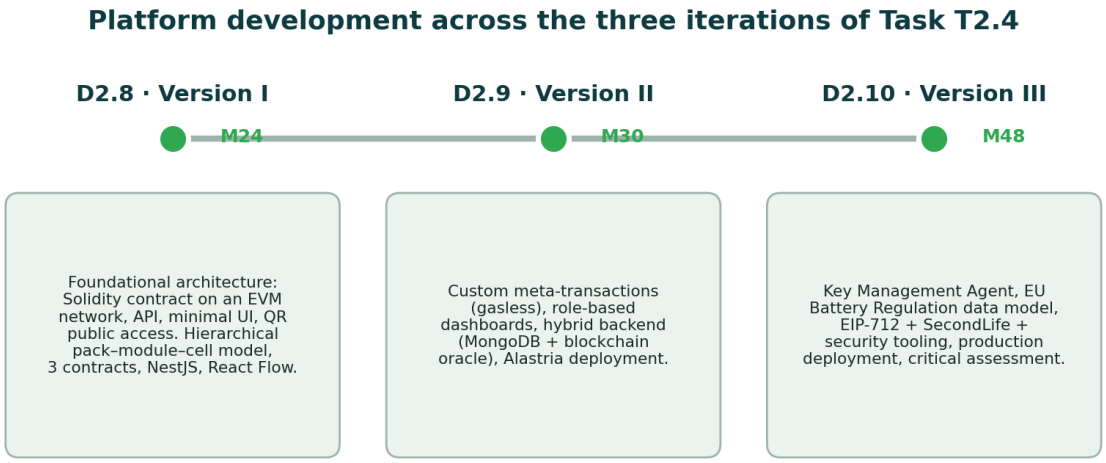


Figure 2.1. Development timeline of the platform across the three iterations of Task T2.4 (D2.8 → D2.9 → D2.10).

2.1 Recap of D2.8 and D2.9

D2.8 documented the development of two prototypes. The first prototype was a foundational version built on Polygon, with a single Solidity smart contract handling a flat data model and a minimal API written in TypeScript and Express, exposed through Swagger and accompanied by a basic HTML interface with QR code generation. This prototype validated the core technical premises of the task: that battery lifecycle events could be modelled as state transitions on a smart contract, that role-based access could be enforced on-chain, and that QR codes could be used to make passport data publicly accessible without requiring any blockchain-specific knowledge from the consumer of the data.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

The second prototype, also documented in D2.8, was the result of an internal feedback cycle involving a workshop facilitated by UNIGRAZ. It introduced the hierarchical pack-module-cell data model that has remained at the core of the platform ever since, replaced the monolithic smart contract with three distinct contracts (CellBatteryPassport, ModuleBatteryPassport and PackBatteryPassport), migrated the API from Express to NestJS for better structure and scalability, and replaced the original HTML interface with a React Flow-based visual frontend capable of representing the hierarchical battery structure interactively. By the end of D2.8, the platform was a functional research prototype but still suffered from significant usability barriers and was not yet suitable for any form of deployment beyond demonstration.

D2.9 was the iteration that turned the prototype into something close to a usable product. It introduced four major changes. First, a custom MetaTransaction smart contract that allows users to sign their intent without paying gas, with a relayer wallet covering the transaction fee on the blockchain side. The decision to develop a custom contract rather than rely on the Gas Station Network was justified by the abandonment of the GSN repository and by the desire to keep the implementation under tight control. Second, a redesigned frontend with a role-based dashboard structure (Admin, Manufacturer, Assembler, Recycler and a Public viewer for QR-based access) and a wallet management workflow in which the user's encrypted wallet is generated at registration, stored on the backend and decrypted locally on the user's device before signing transactions. Third, a hybrid backend that uses MongoDB to store off-chain metadata and a blockchain oracle to listen for on-chain events and synchronise the database with the on-chain state, allowing the platform to serve queries efficiently without overloading the chain. Fourth, a deployment on Alastria, a semi-public blockchain used as a stand-in for a hypothetical EU-managed network.

By the end of D2.9, the platform had a functional architecture covering all the key components of a Battery Passport Platform: smart contracts as the source of truth for lifecycle events, a relayer-based meta-transaction model removing the need for users to manage cryptocurrency, a hybrid backend providing both blockchain immutability and database performance, and a frontend providing role-specific workflows and a public visualization. What it did not yet have was a way for organisations to integrate it into their own systems without using the web interface, a data model fully aligned with the EU Battery Regulation, a production deployment, or a real, end-to-end critical assessment of the architecture as a whole.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

2.2 Outstanding limitations carried over from Version II

The validation chapter of D2.9 explicitly identified several limitations that Version III was expected to address. The most important ones, which set the agenda for the work documented in this deliverable, are summarised below. Each of these limitations is addressed in Version III; the mapping from each limitation to the corresponding solution is made explicit in the next section on the drivers for Version III.

- **Lack of a system-to-system integration path.** Version II assumed that data would be entered into the platform through the web interface by human operators with role-specific dashboards. While this is appropriate for demonstration and for low-volume scenarios, it does not match how real battery data is generated in industrial environments, where manufacturing execution systems, traceability software and laboratory information systems produce events at a rate and a frequency that no human workflow can sustain. A way to bridge this gap was the most important architectural gap left by D2.9.
- **Custodial wallet model with implicit trust in the platform.** Although in Version II the encrypted wallet is decrypted client-side and the backend never holds the plaintext private key, the encrypted keystore itself is stored on the platform. For demonstration purposes this is acceptable, but it places the platform operator in a position of implicit trust that some industrial actors are unwilling to grant. Version III needed to provide an alternative for actors that want to retain complete custody of their cryptographic material.
- **Data model not fully aligned with the EU Battery Regulation.** The Version II data model captured material composition at a relatively coarse granularity. The EU Battery Regulation, in its current and forthcoming forms, requires significantly more structure: critical raw materials must be identified explicitly, hazardous substances must be reported with concentration and hazard class information, recycled content must be reported per material, and per-battery carbon footprint metrics must be available. The Version II model was extensible enough to support these refinements, but the refinements themselves had not yet been made.
- **Limited end-to-end validation in a deployed environment.** Version II was validated through guided demonstrations of the user flows. It had

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

not been deployed continuously in a public environment, it had not been exercised by users outside the development team, and it had not been subjected to any form of operational hardening (rate limiting, login lockouts, observability, production environment validation, security tooling on the smart contracts beyond unit tests). Version III needed to close these gaps before any honest assessment of the architecture could be made.

- **Open question on the value of blockchain itself.** The most important limitation of Version II — and of D2.9 as a deliverable — was that it did not yet allow the project to answer the fundamental research question of T2.4. As long as the platform was a feature-incomplete prototype on a controlled environment, any assessment of the value of blockchain would have been premature and unfair. Version III was expected to bring the platform to a state in which the question could be answered honestly, with the cost of complexity and the practical benefit both visible.

2.3 Drivers for Version III

The work documented in the rest of this deliverable was shaped by these limitations and by a deliberate choice not to expand the scope of the platform any further than necessary. Where Version II added significant new functionality (meta-transactions, oracles, role-based dashboards), Version III concentrates on consolidation and on filling specific architectural gaps. Three drivers can be identified in retrospect.

The first driver was the recognition that the most important missing element of the architecture was a machine-to-machine integration path. This drove the design and development of the Key Management Agent, described at length in Chapter 3 of this document. The agent is a system-to-system gateway that allows organisations to register and submit battery passport events from their own infrastructure, holding their own cryptographic keys and exposing a simplified company-facing API. It is the headline new development of Version III and the main technical contribution of this deliverable.

The second driver was alignment with the EU Battery Regulation. This drove the redesign of the material composition data model for cells, modules and packs, the addition of new SoH data points, the extension of the lifecycle state machines to include the SecondLife stage, the dual identifier strategy that allows both legacy string identifiers and bytes32 canonical identifiers to coexist, and the introduction of an explicit schema versioning field. These

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

changes are technically modest in scope but they bring the platform much closer to the data structures that any future production-grade Battery Passport will need to handle.

The third driver was operational maturity. This drove the production deployment of the platform at https://free4lib.eurecatsecurity.org/, the seeding of a comprehensive synthetic dataset, the introduction of refresh tokens, login lockouts, trace identifiers, ops metrics and production environment validation in the backend, the introduction of a security tooling pipeline based on Slither, Mythril and property-based testing in the smart contract workspace, and the documentation of the operational model in a way that can be reproduced and audited. This driver also produced the conditions under which a critical assessment of blockchain in a battery passport context could finally be carried out without the result being distorted by unfinished engineering.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

3. Development of the Third Prototype (Version III of the Platform)

This chapter documents the technical work performed during Version III of the Battery Passport Platform. It is structured to mirror, where possible, the equivalent chapter of D2.9, so that a reader familiar with the previous deliverable can compare the two versions directly. It begins with the functional requirements that were defined for Version III, organised by component, and continues with the updated architectural overview. It then describes the headline new development of this iteration — the Key Management Agent — in dedicated detail, before turning to the refinements made to the data model, the smart contracts, the frontend and the backend, and ending with the live deployment of the platform at https://free4lib.eurecatsecurity.org/.

The chapter is deliberately concrete. Where the introduction of a component justified abstract discussion in earlier deliverables, the same component in Version III is described in terms of what it does, how it is built and what concrete files it lives in, with code excerpts and JSON examples drawn from the actual repository. This is consistent with the production-readiness focus of this iteration and with the requirement that the deliverable supports both audit and reproducibility.

3.1 Functional Requirements

The functional requirements for Version III are organised in four groups corresponding to the four main components of the platform: backend, frontend, blockchain layer, and the new Key Management Agent. Each requirement is identified by a code of the form FR-XX-nn, where FR denotes a functional requirement and XX the component: BE for backend, FE for frontend, BC for the blockchain layer, and AG for the agent. Requirements that are unchanged from D2.9 are implicitly carried over and are not repeated here. The tables below list only the requirements that are either new in Version III or have been substantially modified.

3.1.1 Backend Functional Requirements

The backend has been hardened for production deployment and extended with new endpoints to support the agent integration path, the refined data model and the additional observability concerns. The new and modified backend functional requirements are summarised in Table 1.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

ID	Requirement Name	Description
FR-BE-08	Refresh Token Rotation	The backend must issue refresh tokens that are randomly generated, hashed before storage, rotated on each refresh and invalidated on logout, ensuring that long-lived sessions do not become a single point of compromise.
FR-BE-09	Login Rate Limiting and Lockouts	The backend must enforce per-account rate limits on login attempts and lock accounts that exceed the configured threshold, recording the failure reasons through the ops metrics service for monitoring and forensics.
FR-BE-10	Trace Identifier Propagation	The backend must accept or generate a trace identifier for every incoming request, propagate it through the request pipeline, return it in response headers and reference it in off-chain metadata associated with relayed transactions, so that frontend, backend and blockchain operations can be correlated.
FR-BE-11	Production Environment Validation	The backend must validate its environment configuration on startup and refuse to start in production unless secrets are strong, CORS allowlists are populated, the trust-proxy flag is set, and blockchain addresses and private keys conform to the expected formats.
FR-BE-12	Role Status Endpoint for Agents	The backend must expose a role-status endpoint that compares the role recorded in MongoDB with the roles granted on-chain through RoleManager, allowing agents to poll until they are fully authorised before proceeding with operations.
FR-BE-13	Refined Material Composition Storage	The backend must persist the refined material composition objects for cells, modules and packs, including critical raw materials, hazardous substances, recycled content and (where applicable) carbon footprint indicators, while keeping a compact representation suitable for on-chain anchoring.
FR-BE-14	Dual Identifier Lookup	The backend must resolve cells, modules and packs by either their legacy string identifier or their canonical bytes32 V2 identifier, using a unified lookup helper that hides the dual representation from controller code.
FR-BE-15	Operational Metrics Endpoint	The backend must expose an ops metrics endpoint in a Prometheus-compatible text format, reporting authentication failures, transaction outcomes, meta-transaction failure reasons and uptime, intended for live monitoring by an external system.

Table 1: New and modified backend functional requirements for Version III

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

3.1.2 Frontend Functional Requirements

The frontend has been updated to surface the refined data model in the user-facing forms and visualisations, to consolidate the wallet decryption pattern around an explicit password modal, and to ensure that the agent operations are correctly reflected in the dashboards used by human users. The new and modified frontend functional requirements are summarised in Table 2.

ID	Requirement Name	Description
FR-FE-08	Refined Material Composition Forms	The frontend must present the refined material composition fields in the cell, module and pack creation forms, including critical raw materials, hazardous substances and recycled content, with validation that prevents the submission of incomplete or inconsistent records.
FR-FE-09	Redesigned Battery Passport Visualizer	The frontend must visualise the hierarchical battery structure as an interactive graph, with a detail drawer for each selected node showing lifecycle state, party information, timestamps, dimensions, composition summaries and SoH history.
FR-FE-10	Wallet Password Modal	The frontend must require explicit re-entry of the wallet password before any operation that produces a meta-transaction signature, treating the password modal as the practical boundary between authenticated session and authorised blockchain action.
FR-FE-11	Refined SoH Recorder	The frontend must allow assemblers and maintenance roles to record SoH measurements with the full set of new SoH data points and to update the assessment metadata, with the records appearing immediately in the relevant dashboards once the meta-transaction has been processed.
FR-FE-12	Recycler Second-Life Workflow	The frontend must allow recyclers to mark a cell as recycled or as assigned to second life, with a confirmation modal that prevents accidental state transitions and reflects the new SecondLife state in the cell lifecycle.
FR-FE-13	Transaction History and Detail Pages	The frontend must allow users to browse transaction history and inspect the on-chain and off-chain metadata of any single transaction in a dedicated detail page.

Table 2: New and modified frontend functional requirements for Version III

3.1.3 Blockchain Functional Requirements

The blockchain layer has been extended to support EIP-712 typed-data signatures, the SecondLife state in the cell state machine and a security tooling

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

pipeline that brings the smart contract codebase to a quality bar appropriate for production deployment. The new and modified blockchain functional requirements are summarised in Table 3.

ID	Requirement Name	Description
FR-BC-07	EIP-712 Typed Meta-Transactions	The MetaTransaction contract must verify signatures produced according to EIP-712 typed-data structures, using a domain separator that includes the contract address, chain identifier, name and version, so that signatures cannot be replayed across contracts or networks.
FR-BC-08	SecondLife Cell State	The CellBatteryPassport contract must support the SecondLife state in the cell lifecycle and enforce the corresponding transitions (InUse to SecondLife, SecondLife to Recycled), aligning the on-chain state machine with the recycler workflow in the frontend.
FR-BC-09	Dual Identifier Support	The Battery Passport contracts must accept bytes32 canonical identifiers in addition to the legacy string identifiers, with referential integrity checks that validate the relationship between cells, modules and packs at write time.
FR-BC-10	Security Tooling Pipeline	The smart contract workspace must include automated security analysis with Slither and Mythril, property-based and fuzz tests in addition to unit tests, and a rollout verification script that confirms the deployed bytecode matches the expected manifest before the platform is considered operational.
FR-BC-11	Deadline-bound Meta-Transactions	Meta-transactions must include an explicit deadline timestamp, after which the signed payload is no longer accepted by the contract, providing a bounded validity window in addition to the per-user nonce.

Table 3: New and modified blockchain functional requirements for Version III

3.1.4 Agent Functional Requirements

The Key Management Agent is the most significant new component of Version III. It is designed as an autonomous, self-custodial gateway that allows an organisation to integrate the Battery Passport Platform into its own information systems without exposing private keys to the platform and without requiring human interaction. Its functional requirements are summarised in Table 4.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

ID	Requirement Name	Description
FR-AG-01	Persistent Encrypted Wallet	The agent must generate or load a persistent encrypted wallet on its own filesystem, encrypted with a password held only by the operating organisation, ensuring that the platform never has access to the agent's private key in any form.
FR-AG-02	Self Registration	The agent must register itself with the Battery Passport Platform on first startup, providing its wallet address and encrypted keystore through the standard registration endpoint, and must idempotently handle the case where it is already registered.
FR-AG-03	Role Activation Polling	The agent must poll the role-status endpoint until an administrator has assigned it the configured role and the role is reflected both in the platform database and on the blockchain through RoleManager, before considering itself ready to operate.
FR-AG-04	Local EIP-712 Signing	The agent must sign every meta-transaction locally, using EIP-712 typed-data structures, with the wallet decrypted only in memory for the duration of the signing operation.
FR-AG-05	Company-Facing Simplified API	The agent must expose a simplified, company-facing API mapped to the BPP operations relevant to the agent's role, hiding the blockchain-specific concerns of nonce retrieval, encoding, signing and submission from the calling application.
FR-AG-06	Token Management	The agent must manage its own JWT access and refresh token lifecycle, obtaining new tokens transparently and recovering from session expiration without external intervention.
FR-AG-07	Operational Logging	The agent must produce structured operational logs that allow the operating organisation to audit its own activity, including registration, role activation, signing and submission events, without leaking sensitive material.

Table 4: Functional requirements for the Key Management Agent

3.2 Updated Architecture Overview

The architecture of the platform in Version III preserves the core ideas introduced in D2.9 — smart contracts as the source of truth for lifecycle events, a relayer-based meta-transaction model, a hybrid backend with MongoDB and a blockchain oracle, and a role-based frontend — and extends them with the addition of the agent as a first-class participant. The result is an architecture in which both human users and external systems can act as authors of battery

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

passport events, in a uniform and verifiable way. The overall structure is shown in Figure 3.1.

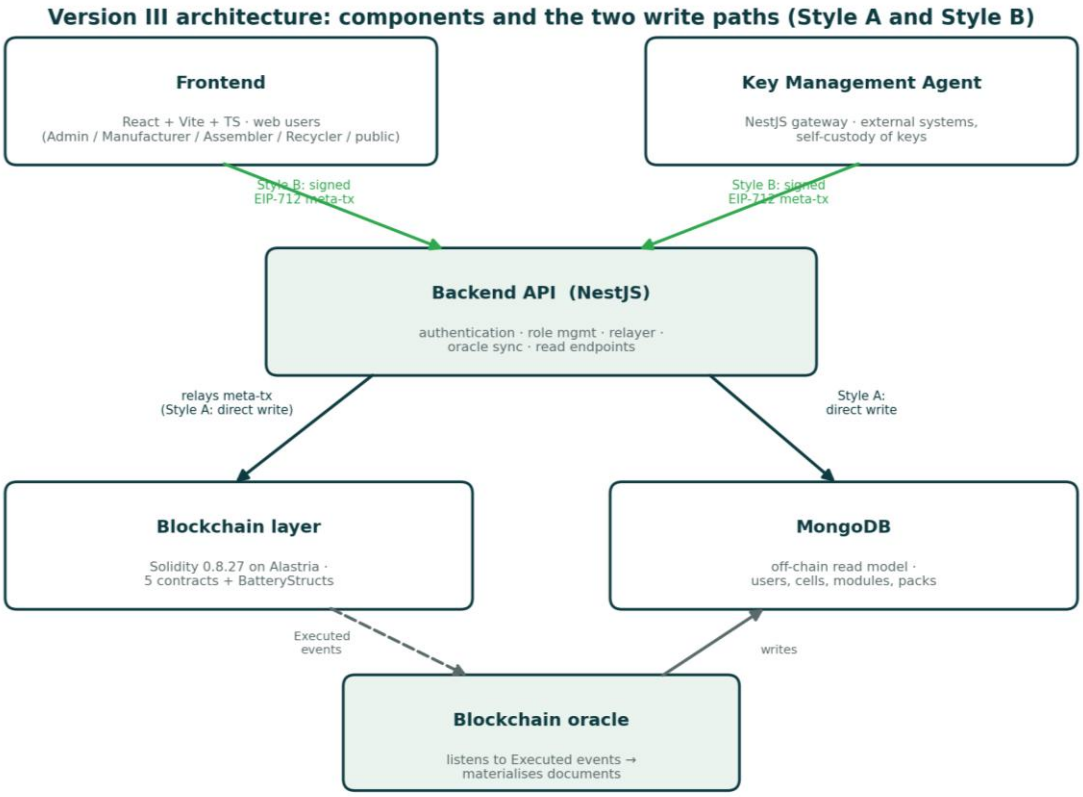


Figure 3.1. Version III architecture: the main components and the two coexisting write paths (Style A backend writes and Style B signed meta-transactions).

3.2.1 Components

The core components of the platform in Version III are the following:

- **Frontend** (React + Vite + TypeScript + Ant Design + React Flow): the user-facing application used by administrators, manufacturers, assemblers and recyclers, as well as by public viewers accessing battery passports through QR codes.
- **Backend API** (NestJS + TypeScript + MongoDB through Mongoose): the central service that handles authentication, role management, transaction processing, off-chain storage, oracle synchronization and the exposure of read endpoints. It is also the relayer that submits meta-transactions on behalf of users and agents.
- **Blockchain layer** (Solidity 0.8.27 contracts on Alastria): the source of truth for lifecycle events, with five contracts (RoleManager,

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

MetaTransaction, CellBatteryPassport, ModuleBatteryPassport, PackBatteryPassport) and a shared BatteryStructs library.

- **Off-chain database** (MongoDB): the persistent store for users, cells, modules, packs and off-chain metadata, populated either by direct backend service writes or by the oracle from blockchain events.
- **Blockchain oracle** (NestJS event listeners): the subsystem that subscribes to Executed events on the smart contracts, decodes the original function calls and materialises or updates the corresponding documents in MongoDB, ensuring that the off-chain read model stays aligned with the on-chain state.
- **Key Management Agent** (NestJS gateway): the new system-to-system component, deployable inside an organisation's own infrastructure, holding its own encrypted wallet and signing meta-transactions locally before submitting them to the platform's relayer endpoint. The agent is described in detail in Section 3.3.

To make the differences between the two iterations explicit, Table 5 summarises the platform's core components, contrasting their state in Version II (D2.9) with Version III (D2.10) and indicating which components are new and which were carried over.

Component	Version II (D2.9)	Version III (D2.10)	Status
Frontend	Role-based React / React Flow dashboards (Admin, Manufacturer, Assembler, Recycler, and a Public viewer for QR access); encrypted wallet generated at registration, stored on the backend and decrypted locally before signing.	React + Vite + TypeScript + Ant Design + React Flow. Refined EU-Battery-Regulation material-composition forms, a redesigned battery passport visualizer, and an explicit wallet-password modal / signing boundary.	Carried over, refined
Backend API	NestJS service handling authentication, transaction processing, off-chain storage and oracle synchronisation; also acts as the relayer.	NestJS + TypeScript + MongoDB (Mongoose): same responsibilities plus refresh tokens, login lockouts, trace identifiers, ops metrics and production-environment validation; still the relayer for both users and the agent.	Carried over, hardened
Blockchain layer	Solidity contracts on Alastria: MetaTransaction plus the Cell/Module/Pack passport contracts (extending MetaTransaction) and RoleManager.	Solidity 0.8.27 on Alastria: five contracts (RoleManager, MetaTransaction, Cell/Module/Pack BatteryPassport) and a shared BatteryStructs library; EIP-712 typed meta-transactions, a SecondLife lifecycle state, and a Slither / Mythril security pipeline.	Carried over, extended

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

Meta-transaction / relayer model	Custom MetaTransaction.sol with a relayer wallet covering gas, enabling gasless user transactions.	EIP-712 typed MetaTransaction; the relayer is folded into the Backend API; the same gasless model now serves both frontend users and the agent.	Carried over, upgraded
Off-chain database (MongoDB)	Stores off-chain metadata, transaction history and data not required on-chain.	Persistent store for users, cells, modules and packs; EU-Battery-Regulation-aligned material model, a dual identifier strategy (string + bytes32) and explicit schema versioning.	Carried over, extended
Blockchain oracle	Listens to on-chain events and updates the off-chain database accordingly.	NestJS event listeners subscribing to Executed events, decoding the original calls and materialising documents in MongoDB; synchronisation improvements.	Carried over, improved
Key Management Agent	— (not present)	NestJS system-to-system gateway, deployable inside an organisation's own infrastructure, holding its own encrypted wallet and signing meta-transactions locally before submitting them to the relayer endpoint.	New in Version III
Hybrid write model (Style A / B)	Present only implicitly (backend service writes alongside signed transactions).	Explicit dual write paths — Style A backend service writes and Style B frontend/agent-signed EIP-712 meta-transactions — documented as a deliberate design choice.	New (made explicit)
Deployment	Alastria test deployment, exercised through guided demonstrations; not continuously deployed.	Live production deployment at free4lib.eurecatsecurity.org with a seeded synthetic dataset and operational hardening.	Carried over, productionised

Table 5: Evolution of the platform's core components from Version II (D2.9) to Version III (D2.10).

3.2.2 Hybrid Write Model

One of the most important architectural traits of the platform in Version III is that it explicitly supports two coexisting write paths, referred to throughout this document as Style A and Style B. This is not a transitional accident but a deliberate design choice.

Style A — Backend service writes. Some operations are performed entirely from the backend. The relevant service (CellsService, ModulesService, PacksService) calls the blockchain directly with the relayer key, writes the corresponding MongoDB documents in the same flow, and creates an off-chain audit entry. This style is used for administrative operations, for state updates

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

that do not require user signatures, and for some legacy paths preserved for compatibility.

Style B — Frontend or agent signed meta-transactions. Most user-driven operations (creating cells, assembling modules and packs, recording SoH, recycling) are performed through signed meta-transactions. The frontend or the agent obtains the current nonce from the backend, encodes the desired contract function call, signs the EIP-712 MetaTransaction payload with the user's or agent's wallet, and submits the signed payload to the backend. The backend relays the transaction on-chain, the contracts verify the signature and the nonce, the contracts emit Executed events, and the oracle materialises the resulting documents in MongoDB.

The coexistence of the two styles is what makes the architecture practical. Style B preserves cryptographic accountability — the user or agent is the one who signed the operation, not the platform — but at the cost of an indirection between the moment when the user acts and the moment when the read model is updated. Style A is more direct and is convenient for cases in which the strong cryptographic guarantee is either unnecessary or actively in the way. The hybrid write model is one of the recurring topics of the critical assessment in Chapter 6, because it captures a tension that runs through the entire platform: the closer the architecture gets to a usable system, the more it needs the conventional database paths that blockchain was supposed to make redundant.

3.2.3 Updated Transaction Flow

The transaction flow in Version III generalises the flow described in D2.9 to account for the agent. From the point of view of the platform, a meta-transaction is a meta-transaction regardless of whether it originates from a browser, from a mobile device or from an agent running inside a company's infrastructure. The relayer logic, the oracle, the database and the contracts behave identically. The differences lie entirely in how the signing wallet is held.

For a frontend user, the wallet is created at registration in the browser, encrypted with the user's password, sent to the backend in encrypted form, stored in MongoDB, returned at login and decrypted locally by the browser when an action is about to be signed. For an agent, the wallet is created at first startup on the agent's filesystem, encrypted with a password held only by the operating organisation, never transmitted to the platform, and decrypted only in memory for the duration of each signing operation.

The full, generalised transaction flow for a Style B operation in Version III is the following: the client (frontend or agent) requests the current nonce and EIP-

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

712 domain data from the backend; it encodes the desired contract function call using the locally held ABI; it signs the resulting MetaTransaction payload with its wallet; it submits the signed payload to the backend's meta-transaction endpoint; the backend's BlockchainBPService loads the correct contract ABI and address, calls executeMetaTransaction on the contract using the relayer key, and waits for the receipt; the contract verifies the signature, the nonce and the deadline, executes the wrapped function, and emits the Executed event; the corresponding oracle service decodes the event payload and creates or updates the relevant cell, module or pack document in MongoDB; the client retrieves the updated state through the standard read endpoints.

3.3 The Key Management Agent

The Key Management Agent (KMA) is the most significant new component delivered in Version III. It is a NestJS gateway, distributed as a containerised application, that runs inside an organisation's own infrastructure and acts on its behalf with the Battery Passport Platform. From the platform's point of view, the agent is just another authenticated user that signs meta-transactions; from the organisation's point of view, the agent is a thin, opinionated bridge that hides the blockchain-specific concerns of the platform behind a simple HTTP API mapped to the operations the organisation actually needs to perform.

This section explains the motivation for the agent, its architecture and design principles, the way it handles its cryptographic material, the registration and role activation flow, the local signing mechanism, the implications for automation and large-scale data onboarding, and the security considerations associated with running it in production.

3.3.1 Motivation

Versions I and II of the Battery Passport Platform were designed around the assumption that data would be entered through the web interface by human operators with role-specific dashboards. This assumption was reasonable for a research prototype and for demonstration scenarios, but it does not survive contact with the way real industrial environments produce battery data. Manufacturers operate manufacturing execution systems and traceability software that emit cell-level events at high frequency. Assemblers run modular production lines that produce module and pack records as part of their normal operations. Recyclers run laboratory information systems that produce State of Health measurements and lifecycle decisions. None of these systems will, in any realistic scenario, ask a human operator to log in to an external web platform and re-type events that have already been recorded internally.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

Three solutions to this problem can be considered. The first is to require each organisation to integrate directly with the blockchain, which means embedding ethers or web3 libraries inside their internal applications, managing private keys, dealing with EIP-712 typed data structures and signing payloads themselves. This is technically possible but it places a very high integration cost on every participating organisation and concentrates blockchain expertise in places where it does not naturally belong. The second is to make the platform's REST API directly callable from the organisation's systems and let the platform sign on the organisation's behalf using credentials shared at registration. This is operationally simple but it makes the cryptographic accountability of the platform meaningless, because all signatures end up being produced by a single platform-side wallet rather than by the organisation that is supposedly responsible for the data. The third option, the one taken in Version III, is to provide a thin component that the organisation runs inside its own infrastructure, that holds its own cryptographic material, and that exposes a simplified interface to the organisation's internal systems.

The third option is the one consistent with the spirit of a blockchain-anchored system. It preserves the property that the signature attached to a battery passport event was produced by the actor responsible for that event, not by the platform operator. It removes the need for organisations to embed blockchain-specific code into their core systems. And it concentrates all the operational complexity of dealing with the platform into a single, well-defined gateway that can be deployed, monitored, updated and audited independently of the organisation's other applications. This component is what the Key Management Agent is.

3.3.2 Architecture and Design Principles

The agent is implemented in TypeScript on top of NestJS, the same framework used for the platform's backend, which keeps the development and operational profile consistent across the project. It is structured around two layers. The core layer handles authentication, wallet management, signing and communication with the platform. The endpoints layer exposes a company-facing HTTP API mapped to the BPP operations the agent is allowed to perform, taking simplified payloads in the company's vocabulary and translating them into the encoded contract calls and signed meta-transactions the platform expects. The relevant directories in the agent workspace are agent/src/core (with auth, wallet and signer subdirectories) and agent/src/endpoints.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

Three design principles have shaped the agent and are worth making explicit because they recur throughout the rest of this chapter. These are not ad-hoc choices: they instantiate well-established security-engineering principles — least privilege, non-custodial key management and economy of mechanism — in the specific context of the agent.[1]

Self custody. The agent's private key never leaves the agent's host. The encrypted keystore is stored on the agent's filesystem in a directory created with restrictive permissions, encrypted with a password that is configured exclusively through the agent's environment and is never transmitted to the platform. The platform receives only the public address and the encrypted keystore is sent only to the registration endpoint as an artefact of the standard registration model, but the platform has no way of decrypting it. This is the property that distinguishes the agent from the platform-assisted custodial model offered to human users in the frontend.

Single-purpose simplicity. The agent is not a general-purpose blockchain client. It implements exactly the operations that an organisation in a given role needs to perform, and nothing else. A manufacturer agent exposes endpoints for cell creation; an assembler agent exposes endpoints for module and pack assembly and for SoH recording; a recycler agent exposes endpoints for recycling and second-life assignment. This restriction makes the agent significantly easier to audit and operate than a general-purpose tool, and it protects the integrating organisation from accidentally calling operations it is not authorised to perform.

Operational autonomy. Once configured, the agent is expected to operate without human intervention for as long as the integrating organisation wants. It manages its own session lifetime (access and refresh tokens), recovers from session expiration transparently, polls the platform for role activation when it is first deployed, and produces structured operational logs that allow the operating organisation to audit its activity locally without depending on the platform for visibility.

3.3.3 Wallet Handling and Key Custody

Wallet handling is the most security-sensitive aspect of the agent and is concentrated in a single service in agent/src/core/wallet. On startup, the agent looks for an existing encrypted keystore at the configured wallet path. If one is

[1] On least privilege and economy of mechanism see J. H. Saltzer and M. D. Schroeder, “The Protection of Information in Computer Systems,” Proceedings of the IEEE 63(9), 1975; on non-custodial, zero-trust key handling see NIST SP 800-207, Zero Trust Architecture, 2020.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

found, it loads the keystore and decrypts it with the configured password using ethers.Wallet.fromEncryptedJson, which is the standard ethers function for handling Web3 secret storage definition (V3) keystores. If no keystore is found, the agent generates a new random wallet using ethers.Wallet.createRandom, encrypts it with the configured password, and writes the resulting keystore to disk with file mode 0600 so that only the owning Unix user can read it. In both cases, the resulting Wallet instance is held only in memory for the lifetime of the agent process and is never written to disk in plaintext form.

The relevant excerpt from the wallet service is reproduced below (Listing 1) to make the mechanism explicit. It corresponds to agent */src/core/wallet/wallet.service.ts*.

```
private async loadOrCreateWallet(): Promise<void> {
  const walletPath = this.configService.get<string>('agent.walletPath')!;
  const password = this.configService.get<string>('agent.password')!;

  if (!password) {
    throw new Error(
      'AGENT_PASSWORD must be set — it is used to encrypt/decrypt the wallet keystore.'
    );
  }

  const dir = path.dirname(walletPath);
  if (!fs.existsSync(dir)) {
    fs.mkdirSync(dir, { recursive: true });
  }

  if (fs.existsSync(walletPath)) {
    this.logger.log(`Loading existing wallet from ${walletPath}`);
    const keystore = fs.readFileSync(walletPath, 'utf-8');
    this.wallet = await ethers.Wallet.fromEncryptedJson(keystore, password);
    this.logger.log(`Wallet loaded — address: ${this.wallet.address}`);
  } else {
    this.logger.log('No wallet found — generating new wallet...');
    this.wallet = ethers.Wallet.createRandom();
```

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

```
    const keystore = await this.wallet.encrypt(password);
    fs.writeFileSync(walletPath, keystore, { mode: 0o600 });
    this.logger.log(`New wallet created and saved to ${walletPath}`);
    this.logger.log(`=== AGENT WALLET ADDRESS: ${this.wallet.address} ===`);
    this.logger.log(`Share this address with the platform admin to assign the role.`);
  }
}
```

Listing 1: The agent's load-or-create wallet routine, from the core wallet service.

Several choices in this excerpt are worth highlighting. The password is mandatory and the agent refuses to start if it is not provided, which prevents an accidental deployment with an unencrypted keystore. The wallet directory is created on demand with the standard permissions of the user under which the agent runs, which is typically a dedicated service account in production. The keystore file itself is created with mode 0600, which means that only that service account can read it; if the agent is running inside a container, this corresponds to the container's user, and the volume in which the keystore lives must be mounted with permissions that preserve this restriction. Finally, the address of the newly created wallet is logged at registration time so that the operating organisation knows what to communicate to the platform administrator for role activation, but the private key is never logged.

This wallet handling model has direct consequences for the security posture of the integration. Compromising the platform does not compromise the agent's keys, because the platform has no copy of them. Compromising the host on which the agent runs does compromise the encrypted keystore, but only an attacker with both the keystore and the password can use the keys; the password is held in environment configuration, in a secrets manager, or in whatever mechanism the operating organisation already uses for its other service credentials. The threat model is therefore equivalent to the threat model of any other authenticated service running inside the organisation's infrastructure, which is exactly the property the design intends to achieve.

3.3.4 Registration and Role Activation

On first startup, the agent must be made known to the Battery Passport Platform before it can perform any operation. This is done through the same registration endpoint used by human users, with the agent providing its username, password, public address and encrypted keystore. The platform creates a user record with the role no_role, exactly as it would for a human

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

user, and returns access and refresh tokens. At this point the agent is registered but not yet authorised to do anything; it needs an administrator to assign it a role through the admin dashboard or through the corresponding API endpoint.

The agent does not assume that role assignment will happen immediately. Instead, it enters a polling loop against the role-status endpoint of the platform, which compares the role recorded in MongoDB with the roles granted on-chain through the RoleManager contract. The agent waits until both sides agree and the on-chain roles include the expected role configured in the agent's environment. Only then does it consider itself ready to operate. This polling pattern is what makes the agent suitable for unattended deployments: an organisation can deploy its agent at any time, and the activation step can happen later, asynchronously, when the platform administrator gets to it.

The relevant excerpts from the authentication client are reproduced below (Listings 2 and 3). They correspond to agent/src/core/auth/bpp-auth.client.ts.

```
private async register(): Promise<void> {
  const username = this.configService.get<string>('agent.username')!;
  const password = this.configService.get<string>('agent.password')!;
  const address = this.walletService.getAddress();
  const encryptedWallet = await this.walletService.getEncryptedKeystore();

  this.logger.log(`Registering agent as '${username}' (${address})...`);

  try {
    await this.http.post('/auth/register', {
      username, password, address, encryptedWallet,
    });
    this.logger.log('Registration successful.');
  } catch (err: any) {
    const status = err?.response?.status;
    if (status === 409) {
      this.logger.log('Agent already registered — skipping.');
    } else {
      this.logger.error(`Registration failed: ${err?.response?.data?.message ??
err.message}`);
```

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

```
      throw err;
    }
  }
}
```

Listing 2: The agent's registration routine, from the authentication client.

```
private async waitUntilAuthorized(): Promise<void> {
  const address = this.walletService.getAddress();
  const expectedRole = this.configService.get<string>('agent.type')!;
  const intervalMs = this.configService.get<number>('agent.rolePollIntervalMs')!;

  this.logger.log(
    `Waiting for admin to assign role '${expectedRole}' to ${address}. ` +
    `Polling every ${intervalMs / 1000}s...`,
  );

  while (true) {
    const { data } = await this.http.get('/auth/role-status', {
      params: { address },
    });

    if (data.isFullyAuthorized && data.onChainRoles.includes(expectedRole)) {
      this.logger.log(`Role '${expectedRole}' confirmed on platform and on-chain.
Proceeding.`);
      return;
    }

    await this.sleep(intervalMs);
  }
}
```

Listing 3: The agent's role-activation polling routine.

Two design choices in this excerpt deserve attention. The first is the explicit handling of the 409 Conflict response, which makes the registration step

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

idempotent: an agent that has already been registered can be restarted without producing a hard error. The second is the polling cadence, which is configurable through agent.rolePollIntervalMs and is logged on every iteration so that the operating organisation knows whether the agent is making progress. In production deployments, the polling interval is typically set to a value that balances responsiveness against load on the platform's role-status endpoint.

The role-status endpoint deserves a brief comment of its own, because it is the mechanism that bridges the off-chain authorization model in MongoDB with the on-chain authorization model in the RoleManager contract. The platform exposes this endpoint precisely because it is possible, in principle, for the two layers to disagree: an administrator might assign a role in MongoDB before the corresponding on-chain transaction has been confirmed, or a transient blockchain failure might leave the on-chain role unassigned. The endpoint reports both states explicitly and the agent only considers itself authorised when they agree. This is the kind of operational detail that does not appear in the architecture diagram of D2.9 but that turns out to be essential when the platform is actually deployed and used.

3.3.5 Local Signing

Once the agent is registered and authorised, it can produce meta-transaction signatures on demand. The signing is performed locally, in the agent's process, using the wallet decrypted from the keystore at startup. The platform is involved only to provide the current nonce and the EIP-712 domain data and, later, to relay the signed payload on-chain. At no point does the platform see, hold or manipulate the agent's private key.

The signing flow is implemented in agent/src/core/signer/signer.service.ts and follows the EIP-712 typed data signing protocol that the MetaTransaction smart contract expects. The agent constructs an EIP-712 domain object containing the contract name, version, chain identifier and verifying contract address, and an EIP-712 value object containing the signer address, the encoded function call (as a hex string), the nonce and a deadline timestamp. It then calls signTypedData on its wallet, which produces a signature according to EIP-712. The relevant excerpt is reproduced below. The routine is shown in Listing 4.

```
private async sign(nonceData: NonceResponse, encodedData: string):
Promise<SignedMetaTx> {

  const wallet = this.walletService.getWallet();

  const deadline = BigInt(Math.floor(Date.now() / 1000) + DEADLINE_OFFSET_SECONDS);
```

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

```
  const nonce = BigInt(nonceData.nonce);


  const domain = {
    name: nonceData.domainName,
    version: nonceData.domainVersion,
    chainId: BigInt(nonceData.chainId),
    verifyingContract: nonceData.verifyingContract,
  };


  const value = {
    user: wallet.address,
    data: encodedData,
    nonce,
    deadline,
  };


  this.logger.debug(
    `Signing MetaTransaction: nonce=${nonce}, deadline=${deadline}`,
  );


  const signature = await wallet.signTypedData(domain, META_TX_TYPES, value);


  return {
    fromAddress: wallet.address,
    signature,
    deadline: deadline.toString(),
  };
}
```

Listing 4: The agent's EIP-712 meta-transaction signing routine, from the signer service.

This excerpt makes the security model concrete. The deadline is computed at signing time and is short enough that even if a signed payload is intercepted in transit, the window during which it can be replayed is bounded. The nonce is taken from the platform's nonce endpoint, and the corresponding contract

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

maintains a per-user nonce counter that is incremented on every successful execution, which prevents the same payload from being executed twice. The domain object includes the verifying contract address and the chain identifier, which makes the signature specific to a particular contract on a particular network and prevents cross-contract or cross-chain replay. And the wallet object used to sign is the one decrypted at startup from the local keystore, which is held only in the agent's process memory.

Once the signature has been produced, the agent submits the signed payload to the platform's meta-transaction endpoint along with the encoded function call and the deadline. From that point on, the flow is identical to a meta-transaction submitted from the frontend: the platform's relayer calls executeMetaTransaction on the contract, the contract verifies the signature, the nonce and the deadline, the wrapped function is executed, the corresponding event is emitted, and the oracle materialises the resulting documents in MongoDB. The agent does not need to know any of this; it only needs to know that its submission has been accepted and that the corresponding on-chain operation has succeeded.

3.3.6 Implications for Automation and Scalability

The introduction of the agent has consequences that go beyond the technical mechanism described in the previous subsections. It changes what it is realistic to do with the platform.

Without the agent, registering a single battery in the platform requires a human operator to log in, navigate to the appropriate dashboard, fill in a form, decrypt the wallet by entering the password and confirm the operation. This is acceptable for a handful of demonstrations but it does not scale. Registering a thousand cells through this process would require thousands of explicit human interactions and would still leave every operation tied to the password fatigue of the operator. Registering a million cells, which is the order of magnitude that any realistic battery manufacturing line will produce, is simply not possible through a human workflow.

With the agent, the same cells can be registered as the manufacturing line emits them. The internal application that records cells in the company's existing systems can be configured to additionally call the agent's company-facing API for each cell, the agent constructs the corresponding meta-transaction, signs it and submits it, and the platform processes the result asynchronously. The throughput limit is no longer the human workflow but the throughput of the underlying blockchain and the rate at which the relayer can submit transactions. On Alastria, with the contract code and the relayer

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

configuration used in the production deployment, this throughput is high enough to keep up with realistic battery manufacturing rates for the moderate-volume scenarios the project is concerned with. For higher volumes, batching primitives like the addCellsBatch function in the CellBatteryPassport contract allow the agent to register many cells in a single signed operation, amortising the cost of signing and the cost of the on-chain transaction across an entire batch.

The agent also changes what kind of organisations can integrate with the platform. An organisation that already has its own information systems can deploy the agent as a sidecar to its existing infrastructure, point it at the platform, and start using the BPP without modifying any of its core applications. The agent is a bounded, well-defined integration surface, which is exactly what enterprise integration teams need in order to take a project like this seriously. From the point of view of the platform operator, the agent is also more predictable than a human user: it does not forget passwords, does not click the wrong button and does not generate support requests that depend on personal availability.

3.3.7 Security Considerations

The security considerations associated with running the agent in production are deliberately more conservative than those associated with running the frontend, because the agent is operating without human supervision and on behalf of an organisation, not an individual.

- **Password handling.** The agent's password must be provided through the environment of the agent's process and never written to a configuration file checked into source control. In containerised deployments this is typically handled through a secrets manager or through the orchestration platform's secret injection mechanism. The password should be rotated according to the operating organisation's normal rotation policies, and the agent supports rotation by re-encrypting the keystore against a new password when restarted with both the old and the new password configured.
- **Filesystem permissions.** The directory containing the keystore must be readable only by the user under which the agent runs. The agent enforces this when it creates the keystore but it cannot enforce it after the fact, so deployments must ensure that the volume in which the keystore lives is mounted with the appropriate permissions and that backups of the keystore are subject to the same restrictions as backups of any other secret.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

- **Network exposure.** The agent's company-facing API is intended to be reachable only from the organisation's internal network, not from the public internet. It does not implement its own authentication for the company-facing API, on the assumption that the network boundary is the relevant trust boundary. Organisations that need a stronger model can place the agent behind their own authentication gateway or extend it to verify per-call credentials.
- **Audit trail.** The agent produces structured operational logs for every registration, role activation, signing and submission event. These logs are intended for the operating organisation, not for the platform, and they should be ingested by the organisation's normal log management infrastructure. Combined with the trace identifiers propagated by the platform's backend, they allow an end-to-end audit of every operation initiated by the agent.
- **Disaster recovery.** Losing the agent's keystore means losing the agent's identity on the platform, because the public address that was registered with the platform corresponds to a private key that no longer exists. The operating organisation must therefore back up the encrypted keystore using the same procedures that it applies to other cryptographic secrets. Re-creating an agent with a new wallet is possible but requires re-registering and re-activating the new agent, which is an explicit administrative operation rather than a transparent recovery.

These considerations do not introduce concerns that are unusual for a service that holds private cryptographic material; they are essentially the same considerations that apply to any service signing payments, certificates or tokens on behalf of an organisation. The relevant point is that they apply to the agent, and not to the platform, which is precisely the separation the design is intended to produce.

3.4 Refined Data Model

The data model in Version III preserves the hierarchical pack-module-cell structure that has been at the core of the platform since the second prototype of D2.8, but it refines several aspects of the model to bring it into closer alignment with the categories of the EU Battery Regulation, to add the data points that were missing from the previous SoH representation, and to support the migration toward canonical bytes32 identifiers that the smart contracts have been moving toward across the iterations. None of these changes alter the fundamental model — a pack contains modules, a module contains cells, every

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

entity has a lifecycle state and a set of party-related metadata fields — but together they bring the structure of the persisted documents close enough to a regulation-compliant Battery Passport to make the platform credible as a reference implementation. The hierarchical structure that this refined model builds on originated in the concept work and the workshop facilitated by UNIGRAZ during the early iterations of the platform, and it has been retained and extended in every version since.

This section describes the three main areas of refinement: the material composition model, the SoH data model, and the dual identifier strategy together with the explicit schema versioning that was introduced in Version III.

3.4.1 Material Composition Aligned with the EU Battery Regulation

The material composition model has been refined separately for cells, for modules and for packs, because the relevant material categories are different at each level of the hierarchy. At cell level, the relevant categories are the chemistry of the cathode, anode and electrolyte, the active materials with their CAS numbers and recycled content percentages, the hazardous substances with their hazard classification and concentration, the critical raw materials with their country of origin, the per-material recycled content figures and the per-energy carbon footprint. At module level, the relevant categories are the housing, busbar, thermal management and potting materials. At pack level, the relevant categories are the casing, thermal management and structural materials. In all three cases, an aggregate recycled content object is included so that the platform can report the total recycled content figures required by the regulation without having to recompute them from the individual material entries on every read.

The cell-level material composition object is the richest of the three because it carries the chemistry information that drives most of the regulatory and operational decisions. A representative example, generated from the seed presets used to populate the production deployment with synthetic data (chemistry family NMC811, EV grade), is reproduced below.

```
{
  "cathodeChemistry": "NMC811",
  "anodeChemistry": "Graphite",
  "electrolyte": "Liquid carbonate (LP57)",
  "activeMaterials": [
```

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

```
    {
      "name": "Lithium Nickel Manganese Cobalt Oxide",
      "casNumber": "346417-97-8",
      "percentageByWeight": 34.8,
      "recycledPct": 8
    },
    {
      "name": "Graphite (synthetic)",
      "casNumber": "7782-42-5",
      "percentageByWeight": 20.7,
      "recycledPct": 0
    },
    {
      "name": "Carbon Black",
      "casNumber": "1333-86-4",
      "percentageByWeight": 3.7
    },
    {
      "name": "PVDF Binder",
      "casNumber": "24937-79-9",
      "percentageByWeight": 2.3
    }
  ],
  "hazardousSubstances": [
    {
      "name": "Cobalt compounds",
      "casNumber": "7440-48-4",
      "concentration": "< 5%",
      "hazardClass": "Carc. 1B",
      "location": "Cathode"
    },
    {
      "name": "Nickel compounds",
      "casNumber": "7440-02-0",
      "concentration": "< 8%",
```


Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

```
      "hazardClass": "Carc. 2",
      "location": "Cathode"
    },
    {
      "name": "Lithium hexafluorophosphate",
      "casNumber": "21324-40-3",
      "concentration": "< 12%",
      "hazardClass": "Acute Tox. 3",
      "location": "Electrolyte"
    }
  ],
  "criticalRawMaterials": [
    {
      "name": "Cobalt",
      "casNumber": "7440-48-4",
      "percentageByWeight": 3.5,
      "originCountry": "DRC / Finland"
    },
    {
      "name": "Lithium",
      "casNumber": "7439-93-2",
      "percentageByWeight": 3.1,
      "originCountry": "Chile / Australia"
    },
    {
      "name": "Natural graphite",
      "casNumber": "7782-42-5",
      "percentageByWeight": 6.4,
      "originCountry": "Mozambique / Brazil"
    }
  ],
  "recycledContent": {
    "cobaltRecycledPct": 12,
    "nickelRecycledPct": 6,
    "lithiumRecycledPct": 4,
```


Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

```
    "totalActiveMatRecycledPct": 8
  },
  "carbonFootprintKgCO2ePerKWh": 65.6
}
```

Listing 5: Refined cell material composition object — NMC811 EV preset

A few aspects of this structure are worth pointing out. The chemistry fields at the top of the object identify the cathode, anode and electrolyte explicitly, which is what allows the platform to differentiate between batteries that share an active material but use different chemistries overall (for example, NMC811 versus NMC622, or the various variants of the LFP family). The activeMaterials array contains the materials that participate in the electrochemical reaction with their CAS numbers and percentages by weight; the recycledPct field is optional and is populated only when the manufacturer is able to attest to a specific recycled content figure. The hazardousSubstances array follows the structure required by the regulation for hazard reporting, with concentration ranges and hazard class strings drawn from the Globally Harmonized System nomenclature. The criticalRawMaterials array carries the CRM-specific information, including the country of origin, which is the information that makes a battery passport useful for due diligence on conflict minerals and on geopolitical supply concentration. The recycledContent object carries the per-material recycled content figures that the EU Battery Regulation requires to be reported separately for cobalt, nickel and lithium, with an aggregate field for the total active material recycled percentage. Finally, the carbonFootprintKgCO2ePerKWh field carries the energy-normalised carbon footprint indicator, which is the metric the regulation uses to compare the climate impact of different batteries.

The module-level material composition object is structurally simpler, because the materials of interest at module level are the housing, the busbars that connect the cells, the thermal management interface and the potting materials that mechanically and thermally bind the assembly together. A representative example is reproduced below.

```
{
  "housingMaterials": [
    { "name": "Aluminum alloy 6061", "percentageByWeight": 39.2, "recycledPct": 34 },
    { "name": "Galvanized steel",    "percentageByWeight": 11.9, "recycledPct": 40 }
  ],
```

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

```
  "busbarMaterials": [
    { "name": "Copper", "casNumber": "7440-50-8", "percentageByWeight": 15.0,
"recycledPct": 28 },
    { "name": "Aluminum", "percentageByWeight": 5.8, "recycledPct": 32 }
  ],
  "thermalManagementMaterials": [
    { "name": "Graphite thermal pad", "percentageByWeight": 9.0 },
    { "name": "Silicone thermal interface", "percentageByWeight": 3.6 }
  ],
  "pottingMaterials": [
    { "name": "Epoxy resin", "percentageByWeight": 7.8 },
    { "name": "Polyurethane", "percentageByWeight": 4.0 }
  ],
  "recycledContent": {
    "aluminumRecycledPct": 34,
    "steelRecycledPct": 40,
    "copperRecycledPct": 28,
    "totalRecycledPct": 23
  }
}
```

Listing 6: Refined module material composition object

At pack level, the relevant materials are the casing of the pack, the thermal management subsystem (which at this level includes the coolant and the cooling plate, in addition to the interface materials), and the structural components such as the high-voltage harness, the crash protection and the fasteners. The pack-level material composition object follows the same pattern as the module-level object but with the categories adapted to the pack scope. A representative example is reproduced below.

```
{
  "casingMaterials": [
    { "name": "Aluminum alloy 6000", "percentageByWeight": 33.1, "recycledPct": 35
},
    { "name": "High-strength steel", "percentageByWeight": 18.4, "recycledPct": 38
},
    { "name": "Polypropylene cover", "percentageByWeight": 4.0, "recycledPct": 15
}
```

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

```
  ],
  "thermalManagementMaterials": [
    { "name": "Ethylene glycol coolant", "percentageByWeight": 8.9 },
    { "name": "Aluminum cooling plate",  "percentageByWeight": 8.3, "recycledPct": 35
},
    { "name": "Silicone TIM",            "percentageByWeight": 3.1 }
  ],
  "structuralComponents": [
    { "name": "Copper HV harness",       "percentageByWeight": 8.4, "recycledPct": 28
},
    { "name": "Composite crash shield",  "percentageByWeight": 4.1 },
    { "name": "Stainless fasteners",     "percentageByWeight": 4.1, "recycledPct": 44
}
  ],
  "recycledContent": {
    "aluminumRecycledPct": 35,
    "steelRecycledPct": 38,
    "copperRecycledPct": 28,
    "plasticRecycledPct": 15,
    "totalRecycledPct": 24
  }
}
```

Listing 7: Refined pack material composition object

The pattern that emerges from the three listings is intentional. Every level of the hierarchy uses the same set of conventions: arrays of named materials with optional CAS numbers, percentages by weight and recycled content percentages, plus an aggregate recycled content object. This consistency makes the data model easy to query and easy to render in the user interface, where the same component can be reused to display the materials of a cell, of a module or of a pack with only minor adjustments to the column headers. It also makes the model easy to extend if the regulation introduces new categories in the future, since adding a new array of materials at any level is a matter of adding a new field rather than restructuring the document.

3.4.2 State of Health Data Model Refinements

The State of Health representation in Version II was a single numeric value associated with a cell or a pack. This was sufficient for the demonstration scenarios of D2.9 but it lacked the structure needed to support the analytical

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

assessment foreseen in subtask ST2.4.3 and to comply with the more sophisticated SoH reporting expected by the EU Battery Regulation. Version III introduces a two-part SoH representation associated with packs.

The first part is a soHAssessment object that carries the metadata of the most recent assessment: the timestamp of the assessment, the assessor identity, the methodology applied (typically a reference to a SoH measurement protocol), and any contextual information that the assessor wants to record alongside the numerical result. The second part is a soHRecords array that carries the historical sequence of measurements, each with its own timestamp, value, and assessor identity. This allows the platform to represent both the latest measurement and the trajectory over time, which is the data that any meaningful second-life or lifecycle decision will need.

At cell level, the SoH information is carried by the lifecycle stage and by hash fields that anchor the corresponding off-chain SoH data. Cells inherit the SoH context of the pack to which they belong through the pack-level soHRecords. This denormalised representation reflects the practical observation that SoH measurements in industrial settings are typically carried out at pack level, not at cell level, even when the underlying physical degradation happens at the cell level.

The new SoH model is also what made it possible to introduce the SecondLife state in the cell lifecycle. The cell state machine in Version III explicitly distinguishes between InUse, SecondLife and Recycled, with the corresponding transitions enforced both in the smart contract and in the backend service code. A cell can move from InUse to SecondLife when a recycler determines, on the basis of its SoH history, that the cell is suitable for reuse rather than for material recovery, and from SecondLife back to Recycled at the end of its second life. This explicit state representation is a precondition for the kind of analytical assessment that AVL is carrying out in parallel with this deliverable, because it makes the second-life pathway visible in the platform rather than hiding it inside the recycler workflow.

3.4.3 Dual Identifier Strategy and Schema Versioning

The Battery Passport contracts have been moving across the iterations from string-based identifiers, which were convenient for early prototypes, toward bytes32 canonical identifiers, which are the natural representation of identifiers on EVM-compatible chains. Version III is the iteration in which both representations coexist explicitly, in order to preserve compatibility with data created during D2.8 and D2.9 while allowing new entities to be created with the canonical representation.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

In practice, this means that every cell, module and pack in the database has a primary string identifier (cellId, moduleId, packId) and an optional bytes32 canonical identifier (cellIdV2, moduleIdV2, packIdV2). The backend exposes a unified lookup helper, buildDualIdLookup, that constructs a Mongo query matching either representation, so that controller code does not need to know which form was used to create the entity. Canonicalisation between the two forms is performed by toCanonicalIdHex, which produces a deterministic bytes32 value from the legacy string identifier when migration is required.

Alongside the dual identifier strategy, Version III introduces an explicit schemaVersion field on every entity. This field is used both to differentiate between V1 and V2 documents in the database and to enable forward-compatible deserialisation: when the backend reads a document, it inspects the schemaVersion field and applies the appropriate decoding logic. This pattern is conventional in production-grade systems but it is the kind of detail that is easy to leave out of a research prototype, and its presence in Version III is one of the markers of the operational maturity that this iteration is meant to achieve.

Persisted indexes have been added to support the dual identifier model. Cells, modules and packs all carry compound indexes on (schemaVersion, identifierV2), which makes the canonical lookups efficient even as the dataset grows. Cells additionally carry an index on the manufacturer blockchain address to support the manufacturer-scoped listings used by the manufacturer dashboard.

It is worth noting that the data model documented in this section corresponds exactly to the persisted data model of the production deployment. The full description of the model — collections, fields, indexes, lifecycle state machines, integrity rules — is maintained as a living artefact in the project repository under docs/DATA_MODEL.md, and any reader interested in the full schema is encouraged to consult that document. The intent of this deliverable is to describe the design choices and the alignment with the regulation, not to duplicate the full reference.

3.5 Smart Contract Evolution

The smart contract workspace, located in backend/web3 in the project repository, has been refined in three directions during Version III: the meta-transaction protocol has been migrated from the custom payload format introduced in D2.9 to a fully EIP-712-compliant typed-data structure, the cell lifecycle state machine has been extended to make the SecondLife stage

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

explicit, and a security tooling pipeline based on Slither, Mythril and property-based testing has been put in place to bring the smart contract codebase to the quality bar that a production deployment requires. Solidity 0.8.27 is used throughout, and the five contracts of the platform — RoleManager, MetaTransaction, CellBatteryPassport, ModuleBatteryPassport, PackBatteryPassport — together with the BatteryStructs library, retain the structure introduced in D2.9.

3.5.1 EIP-712 Typed Meta-Transactions

D2.9 documented the decision to develop a custom MetaTransaction smart contract rather than rely on the abandoned Gas Station Network, and described a payload format in which the user signed a hash of the user address, the encoded function call and the nonce. Version III preserves the decision but upgrades the payload format to EIP-712 typed data. The motivation for this change is twofold.

The first motivation is security. EIP-712 typed data signing produces a signature that is bound to a specific domain — identified by a contract name, version, chain identifier and verifying contract address — and to a specific type definition. A signature produced for one domain cannot be replayed against a different contract, a different chain or even a different version of the same contract. The custom payload format used in D2.9, while functional, did not include the domain separator and was therefore in principle replayable across deployments of the same contract, even if the project's choice to deploy on a single network made this a theoretical concern in practice. Migrating to EIP-712 closes the theoretical gap and brings the meta-transaction protocol in line with the standard practice of the broader Ethereum ecosystem.

The second motivation is interoperability. EIP-712 is the de facto standard for typed-data signing in Ethereum-compatible ecosystems and is supported natively by the major wallet libraries (ethers, viem, web3.js) and by hardware wallets (Ledger, Trezor). By migrating to EIP-712, the platform makes it possible to use any of these tools to sign meta-transactions, which is a precondition for the agent to use ethers.signTypedData and for any future integration with a hardware-wallet-based signing flow. The custom payload format from D2.9 would have required every signing client to reimplement the same custom hashing logic, which is exactly the kind of fragility that a production system should avoid.

The MetaTransaction contract verifies signatures by reconstructing the typed-data hash from the signed fields and recovering the signer address from the signature using ecrecover. The recovered address is then compared to the user

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

field of the payload, and the signature is rejected if they do not match. The contract also enforces a per-user nonce that is incremented after every successful execution and a deadline timestamp after which the signed payload is no longer accepted. The combination of nonce, deadline and EIP-712 domain provides the standard set of guarantees against replay attacks: a payload can be executed at most once, only within a bounded time window, and only against the specific contract for which it was signed.

Inside the Battery Passport contracts (CellBatteryPassport, ModuleBatteryPassport, PackBatteryPassport), all access control checks rely on _msgSender() rather than msg.sender. This is the same pattern used in D2.9 but with the EIP-712 verification underneath. _msgSender() returns the user address recovered from the signature when the call comes through the meta-transaction path, and the actual transaction sender otherwise. This makes role-based access control work transparently for both direct calls and relayed meta-transactions, which is the property that allows the same contracts to be used by the frontend, the agent and the backend's direct write paths without any code-level branching.

3.5.2 Lifecycle State Machines and the SecondLife State

The cell lifecycle state machine in D2.9 had four states: Created, Assembled, InUse and Recycled. Version III adds a fifth state, SecondLife, between InUse and Recycled. The motivation is to make the second-life pathway explicit in the platform rather than hiding it inside the recycler workflow. A cell can now move from InUse to SecondLife when a recycler determines, on the basis of its SoH history, that the cell is suitable for reuse, and from SecondLife to Recycled at the end of its second life. The transition from InUse to Recycled directly is also preserved, for cells that are recycled without passing through a second-life stage.

The module and pack state machines retain the four states they had in D2.9 (Created, Assembled, InUse, Recycled), because in current industrial practice the second-life decision is taken at the cell level, after disassembly of the original module or pack. A module or pack that contains second-life cells is itself a new module or pack, with its own identifier and its own assembly history, rather than a previous module or pack in a new state. This modelling choice is consistent with the way second-life integrators actually work and with the way the EU Battery Regulation describes the responsibility of second-life operators.

State transitions are enforced both in the smart contract and in the corresponding backend service. The contracts perform the authoritative check,

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

because they are the source of truth, but the backend services also perform the same check before submitting the transaction. This is not a redundancy: it allows the backend to provide clear error messages to the frontend or to the agent without waiting for the on-chain transaction to fail, and it prevents accidental gas consumption on transactions that would have been rejected anyway. The corresponding state transition tables are documented in the project repository under docs/DATA_MODEL.md and are reproduced in the Validation chapter of this document where they are exercised by the validation cases.

3.5.3 Security Tooling and Quality Pipeline

D2.9 noted that the smart contracts were tested with unit tests but did not document any structured security analysis. Version III brings the smart contract codebase to a quality bar that is more representative of a production deployment, with three layers of automated quality checks running on the contracts.

- **Static analysis with Slither.** Slither is a Solidity static analyser that detects a wide range of common vulnerabilities and code-quality issues. It is run as part of the smart contract test pipeline and reports issues such as reentrancy patterns, uninitialised storage, unprotected functions, missing zero-address checks and non-standard ERC token usage. The smart contract codebase has been iterated until the Slither output contains no high-severity findings and only the medium- and low-severity findings that are intentional design choices, each documented in the contract source.
- **Symbolic execution with Mythril.** Mythril is a symbolic execution engine for Ethereum smart contracts. It is run on the deployed contracts to detect issues that are not visible to a static analyser, such as integer overflows under specific input conditions, exception ordering issues and call-graph reachability problems. As with Slither, the contract code has been iterated until the Mythril findings are either resolved or explicitly justified.
- **Property-based and fuzz testing.** In addition to the unit tests that exercise specific functions with specific inputs, the contract test suite includes property-based and fuzz tests that generate random inputs and verify that the relevant invariants hold across the random space. Examples of the invariants tested include: a cell that has been added by a manufacturer can never be removed by a non-manufacturer; a meta-transaction with an invalid signature is always rejected; the per-user

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

nonce monotonically increases on every successful execution; the relationship between cells, modules and packs is always preserved, in the sense that a module always references existing cells and a pack always references existing modules.

Beyond these three automated layers, a rollout verification script confirms that the bytecode of the contracts deployed to the Alastria network matches the expected manifest before the platform is considered operational. This script is run as part of the deployment workflow and prevents the platform from starting up against contracts that do not match the expected source. It is the kind of operational hygiene that becomes essential as soon as a system is deployed continuously rather than re-deployed from scratch on every change.

3.6 Frontend Refinements

The frontend in Version III preserves the architecture of D2.9 — a React 18 application built with Vite and TypeScript, using Ant Design as the component library, Axios as the HTTP client, React Router for navigation and React Flow with Dagre for the passport graph visualization — and refines it in three areas: the material composition forms and visualisations, the battery passport visualizer, and the wallet decryption pattern around the password modal. The role-based dashboard structure introduced in D2.9 (Admin, Manufacturer, Assembler, Recycler, Public viewer) is unchanged.

3.6.1 Refined Material Composition Forms and Visualisation

The forms used to create cells, modules and packs have been redesigned to expose the refined material composition fields described in Section 3.4.1. For cells, this means that the manufacturer dashboard now allows the operator to specify the chemistry of the cathode, anode and electrolyte, to enter the active materials with their CAS numbers and percentages, to declare the hazardous substances with their concentration and hazard class, to identify the critical raw materials with their country of origin, to report the per-material recycled content figures and to enter the carbon footprint indicator. For modules and packs, the corresponding forms expose the housing, busbar, thermal management, structural and casing materials according to the level of the hierarchy.

These forms are the most data-rich part of the frontend and they are also the most error-prone if no validation is in place. Version III adds client-side validation that enforces the basic structural rules — material percentages must sum to a plausible value, CAS numbers must follow the correct format, hazard

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

classes must be drawn from a controlled vocabulary, recycled content figures must be in the [0, 100] range — and surfaces validation errors inline as the operator types. Server-side validation continues to be enforced by the backend's NestJS validation pipe with strict DTOs, so that the platform never persists a malformed material composition object even if the frontend validation is bypassed.

The frontend also includes a dedicated material composition viewer that renders the persisted material composition objects in a structured, human-readable format. The viewer groups the materials by category, sorts them by percentage by weight, highlights the critical raw materials and the hazardous substances with appropriate visual cues, and renders the recycled content figures both as numerical values and as proportional visual indicators. This component is reused both in the cell, module and pack detail pages and in the battery passport visualizer described in the next subsection.

3.6.2 Redesigned Battery Passport Visualizer

The battery passport visualizer is the most prominent user-facing component of the platform and the clearest manifestation of its goal. Version III preserves the React Flow-based visual structure introduced in D2.9 — a graph in which packs, modules and cells are represented as nodes and the parent-child relationships are represented as edges — and refines it with a richer detail drawer, a more compact layout produced by the Dagre layout algorithm, and a clearer visual encoding of the lifecycle state of each entity.

When the user opens a battery passport from a QR code, from the URL /battery-passport/{packId} or from a list in one of the dashboards, the frontend requests the aggregate read model from the backend's /battery-passport/{packId} endpoint, which returns the pack together with all the modules referenced by the pack and all the cells referenced by each module. The frontend builds the corresponding React Flow graph, applies the Dagre layout to position the nodes, and renders the result inside an Ant Design layout that includes a header with pack-level information and a side drawer that opens whenever the user clicks on a node.

The detail drawer is the component that has received the most attention in Version III. For a pack, it shows the lifecycle state, the manufacturer and assembler information, the dimensions, the material composition (rendered with the dedicated viewer described above), the SoH history with both the latest assessment and the historical records, and the safety information. For a module, it shows the lifecycle state, the assembler information, the repair history, the dimensions and the material composition. For a cell, it shows the

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

lifecycle state, the manufacturer information, the material composition and any cell-specific lifecycle data. The drawer is dismissable and supports keyboard navigation, which makes it usable both for mouse-driven exploration and for accessibility-aware operation.

This visualisation is the closest the platform comes to delivering on the core promise of a battery passport: not just storing battery data, but making it inspectable, navigable and trustworthy. It is also the component that most directly demonstrates the value of the hierarchical data model, because the same graph naturally represents the structural composition of the battery, the location of every cell within the structure, and the lifecycle state of every entity at every level.

3.6.3 Wallet Password Modal and the Sign Boundary

The wallet decryption pattern that was introduced in D2.9 has been consolidated in Version III into an explicit password modal component, used uniformly across the application. The pattern reflects a deliberate boundary in the trust model of the platform: being logged in is not the same as being authorised to sign blockchain operations. The modal makes this boundary visible to the user.

When a user logs in, the encrypted wallet is retrieved from the backend and stored in browser local storage along with the JWT access and refresh tokens. The wallet is not decrypted at this stage. When the user later attempts to perform a blockchain-write operation — creating a cell, assembling a module, recording an SoH measurement, transitioning a cell to recycled or to second life — the application opens the password modal and asks the user to re-enter the wallet password. The wallet is then decrypted in memory, the meta-transaction is signed, and the decrypted wallet is discarded. The next blockchain-write operation will require a new password entry.

This pattern has two practical benefits. The first is security: the decrypted wallet is never persisted, never written to local storage and never held in memory longer than necessary, which limits the exposure window in the event of a cross-site scripting compromise. The second is psychological: the explicit password prompt forces the user to deliberately authorise each blockchain action, which is consistent with the way users in the broader Ethereum ecosystem are accustomed to interacting with their wallets and which prevents accidental transactions in a way that a single click does not.

The cost of the pattern, which Chapter 6 will return to, is that it makes high-volume operations through the frontend painful. The agent is the answer to that

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

pain: any high-volume scenario should be handled by an agent, not by a human user clicking through password prompts.

3.7 Backend Hardening for Production

The backend, implemented as a NestJS application using TypeScript and MongoDB through Mongoose, has been hardened for production deployment in Version III. The hardening covers four areas: authentication and session management, observability and operational metrics, production environment validation, and the oracle synchronisation between blockchain events and the off-chain read model. None of these changes are visible to a casual user of the platform, but together they are what makes the difference between a research prototype and a service that can be deployed continuously and operated in a public environment.

3.7.1 Authentication and Session Management

Authentication in Version III remains JWT-based, with access tokens signed by the backend and validated on every request. The session management around these tokens has been considerably hardened compared to D2.9. Refresh tokens are now generated as random opaque values, hashed with bcrypt before being stored in the user record, rotated on every refresh and invalidated on logout. This means that a refresh token captured from the wire or from the client can only be used once before becoming invalid, and that the database never holds the refresh token in a form that would allow an attacker to use it directly.

Login endpoints are rate-limited per account and per source address. The AuthSecurityService and OpsMetricsService components in the backend track failed login attempts, lock accounts that exceed the configured threshold, and record the failure reasons for monitoring and forensics. The lockout duration is configurable and the lockout is automatically released after the configured period. This is conventional for any production authentication endpoint, but it was not present in D2.9.

The role status endpoint (GET /auth/role-status), already mentioned in Section 3.3.4 in the context of agent activation, is also used by the frontend during normal operation to verify that the off-chain and on-chain roles agree. If they do not, the user is shown a notification that explains the discrepancy. This endpoint is the practical bridge between the blockchain authorisation model in the RoleManager contract and the conventional authorisation model in the database, and its presence is what makes the dual-write pattern of role

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

assignment (Mongo write followed by on-chain grantRole call) operationally usable.

3.7.2 Observability and Operational Metrics

Three observability features have been added in Version III. The first is trace identifier propagation: every incoming request either carries a trace identifier in the x-trace-id header or has one generated for it by the TraceIdMiddleware, and the same identifier is propagated through the request pipeline, returned in the response headers and referenced in the off-chain metadata associated with relayed transactions. This makes it possible to correlate a frontend or agent request with the corresponding backend log entries, the relayed blockchain transaction and the resulting MongoDB document, which is the kind of correlation that becomes essential as soon as a system is operated continuously.

The second is the operational metrics endpoint at /ops/metrics, which exposes a Prometheus-compatible text format containing authentication failure counts grouped by reason, transaction success and failure counts grouped by contract type, meta-transaction failure reasons, and uptime. These metrics are intended to be scraped by an external monitoring system and are not persisted by the backend itself: a process restart resets the counters. The intent is to provide live operational visibility, not historical analytics; the latter is the responsibility of an external monitoring stack.

The third is the dashboard endpoint at /ops/dashboard, which returns a small JSON document summarising the live operational state of the platform. This endpoint is used by the operations dashboard rendered in the admin role, where it provides a quick visual indication of whether the platform is healthy.

3.7.3 Production Environment Validation

The backend validates its environment on startup using a custom validation function in backend/src/config/env.validation.ts. This function checks that the required environment variables are present, that the JWT and admin secrets are strong and not placeholder values, that the CORS allowlist is non-empty when the platform is running in production mode, that the trust-proxy flag is set when the platform is running behind a reverse proxy, that blockchain addresses conform to the expected hex format, and that the relayer private key is a valid Ethereum private key. If any of these checks fails, the backend refuses to start. This is the kind of fail-fast pattern that prevents the most common class of production incidents — the silent misconfiguration that produces a partially functional service — and that is essential for any platform deployed continuously in a public environment.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

The relevant environment variables are documented in the backend's README and in the project architecture document. They include the database URI, the JWT and admin secrets, the frontend origin, the CORS allowlist, the trust-proxy flag, the blockchain provider URL, the relayer private key and address, the addresses of the deployed contracts and the deployment manifest path.

3.7.4 Oracle Synchronisation Improvements

The blockchain oracle, introduced in D2.9, is the subsystem that subscribes to Executed events on the smart contracts and materialises the corresponding cell, module and pack documents in MongoDB. In Version III, the oracle has been extended to handle the new operations introduced in this iteration (addCellsBatch, addPackSoHRecord, the second-life transitions) and to deal more robustly with the operational realities of a continuously deployed system.

The most important refinement is the handling of partial failures. When the oracle materialises a complex operation such as the creation of a module, it must update multiple documents — the new module, all the cells now referenced by the module, the parent pack if any — in a sequence of MongoDB writes. Version III ensures that each of these writes is performed in an idempotent manner, so that an oracle restart does not produce duplicate or inconsistent documents, and that errors during the materialisation are logged in a way that allows them to be diagnosed and corrected without losing the underlying blockchain event. There is no distributed transaction across the blockchain and the database, and the consistency between the two layers is therefore eventual rather than strict, but the eventual consistency is now reliable in the practical sense that it actually converges and that it does so in bounded time.

A second refinement is the addition of batch handling. The addCellsBatch function in the CellBatteryPassport contract registers a batch of cells in a single transaction, which is essential for the high-volume scenarios that the agent enables. The corresponding oracle service decodes the event, iterates over the batch, and materialises one cell document per cell in the batch, with the same idempotency guarantees as the single-cell path. This is the implementation that allows the agent to register thousands of cells at industrial throughput rates without overloading either the chain or the off-chain read model.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

3.8 Live Deployment at free4lib.eurecatsecurity.org

Version III of the Battery Passport Platform has been deployed in a continuously operating production environment at https://free4lib.eurecatsecurity.org/. The deployment is the first time in the history of T2.4 that the platform has been made available outside the development environment of the team, with a stable URL, a stable set of demo credentials and a stable seeded dataset, in a way that allows any consortium partner or external stakeholder to access it without any kind of installation or configuration on their side.

The deployment runs the production-oriented Docker compose stack defined in docker-compose.yaml, with the following services: a MongoDB instance for the off-chain database, the NestJS backend, the Vite-built frontend served behind Nginx, and an optional one-shot seeding service that populates the database with the synthetic dataset described below. The blockchain layer is the Alastria network, used since D2.9 as a stand-in for a hypothetical EU-managed blockchain. The relayer wallet, the contract addresses and the deployment manifest are injected through environment variables, and the production environment validation described in Section 3.7.3 ensures that the stack does not start with placeholder values.

The seeded dataset deserves particular attention because it is what gives any visitor to the platform something to look at without requiring them to first create their own data. The seed service uses the realistic chemistry presets described in Section 3.4.1 (NMC811 EV grade and a small set of related variants), generates packs, modules and cells with deterministic pseudo-random parameters drawn from a fixed seed, and populates the platform with a comprehensive set of records that exercises every major code path: cells of multiple chemistries, modules and packs of varying composition, SoH history records over time, second-life transitions and recycled cells. The dataset is large enough to make the dashboards meaningful and small enough to remain navigable in the visualisation. It is explicitly synthetic, and the platform is configured to make this clear to anyone who interacts with it.

Demo credentials for the four authenticated roles are documented in the platform itself, on the home page, and were communicated to the consortium during the 36M consortium meeting. The credentials provide access to the Admin, Manufacturer, Assembler and Recycler dashboards. A public viewer is available without credentials through the QR-code-based URL pattern. The

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

home page itself includes quick-login buttons that bypass the manual entry of demo credentials, which makes the platform usable as a guided tour of the role-based workflows.

The deployment is operated by Eurecat using the same operational procedures as other Eurecat-hosted services. It is reachable from the public internet, served over HTTPS with a valid certificate, and protected by the rate limiting and authentication hardening described in Section 3.7.1. The production stack does not currently include automated alerting on top of the metrics endpoint, but the metrics are available for any operator who wants to integrate them with an external monitoring stack.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

4. ST2.4.3 – Analytical Assessment of Battery Platform Data (AVL)

Subtask ST2.4.3 of Task T2.4 is dedicated to the analytical assessment of the data managed by the Battery Passport Platform, with a focus on the assessment of the individual State of Health (SoH) of a battery, the analysis of SoH prediction, the determination of the optimum point in time for second-life applicability or recycling, and the evaluation of the trade-off between optimal design, operation strategy and performance criteria for the optimization of the energy invested over the entire battery lifecycle. This subtask is led by AVL with the support of Eurecat as the platform integrator.

Eurecat and AVL have established a collaboration during the second half of the project to define the framework of this assessment. AVL has helped Eurecat refine the SoH-related data points captured by the platform's data model — the additions described in Section 3.4.2 of this deliverable are a direct consequence of that work — and the two partners have agreed on the scope of the analytical work to be carried out on the data once the platform has been seeded with battery records.

The analytical assessment has now been completed. AVL carried out the assessment on the data managed by the platform, and the results were presented at the 48-month consortium meeting (3 June 2026). This chapter documents the outcome of that assessment: the health parameters and the methodology agreed between AVL and Eurecat, the way the assessment has been implemented inside the live Battery Passport Platform by Eurecat, and the validation of the assessment against a real AVL dataset. The structure follows the subsections agreed for ST2.4.3.

4.1 Objective and Scope of the Analytical Assessment

The objective of the analytical assessment is to determine, from the data captured by the Battery Passport Platform, the individual State of Health (SoH) of a battery and the optimum point in time at which it should be routed to a second-life application or to recycling. The assessment operates on the hierarchical pack-module-cell data model of the platform and, in particular, on

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

the SoH data points recorded for each entity over its lifetime. Its scope covers four questions: (i) the assessment of the current health state of cells, modules and packs; (ii) the prediction of the SoH trajectory from the recorded history; (iii) the second-life-versus-recycling decision; and (iv) the lifecycle-optimization perspective, in which usage and load profiles are considered as levers to extend useful life. The assessment was defined by AVL, drawing on its domain expertise in battery health assessment, and implemented together with Eurecat on top of the platform documented in the rest of this deliverable.

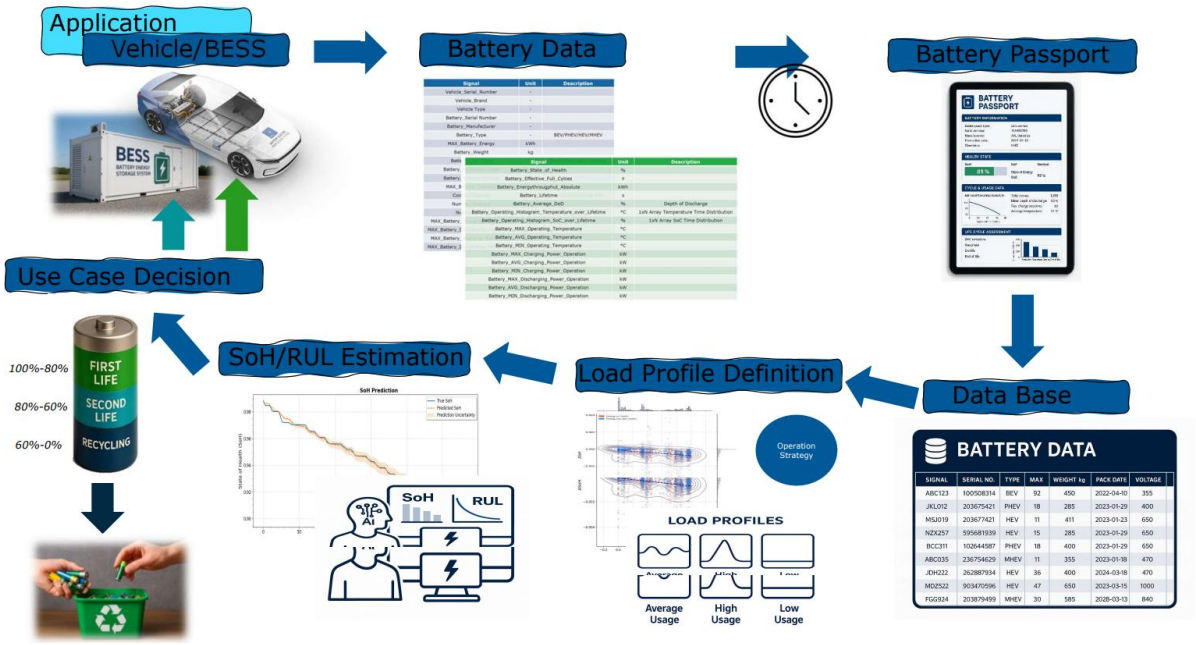


Figure 4.1. End-to-end analytical assessment workflow: from application and battery data to the second-life or recycling decision written back to the passport.

Figure 4.1 shows the end-to-end workflow of the assessment. Application and battery data, both static and dynamic, flow from the vehicle or stationary battery energy storage system (BESS) into the platform's database and battery passport. From there the load profile is characterised, the SoH and the remaining useful life are estimated, and a use-case decision (first life, second life or recycling) is produced and written back to the passport.

4.2 Health Parameters and SoH Data Model

AVL identified the set of health parameters that the platform must capture in order to support the assessment. For each entity (pack, module and cell) these comprise the nominal battery information (energy, nominal, minimum and maximum voltage, cooling concept, and the continuous and peak charge and discharge power) and, crucially, a time series of State-of-Health records.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

Each SoH record carries the date, the State of Health by capacity (SoH-C, %), the number of equivalent full cycles (EFC), the remaining capacity, the internal-resistance increase (SoH-R, expressed as the ratio R_act/R_nominal in percent above 100 %), the nominal energy, the self-discharge (%/month) and the age (days). It also carries the pack or module coulombic efficiency (Ah_discharge/Ah_charge over a 20-80 % SoC window at 0.5C, 25 °C), the round-trip efficiency (Wh_discharge/Wh_charge over the same window) and the voltage spread (the maximum minus minimum cell voltage measured at rest, after more than 3 h, at approximately 50 % SoC and 25 °C; a value above 100 mV is flagged as an issue). The capacity and resistance degradation rates are derived from the two most recent records, as the change in SoH-C (respectively the resistance ratio) divided by the change in EFC, and expressed in %/EFC.

These parameters are stored as an ordered SoH history, sorted with the highest cycle count first, so that the most recent assessment is always available. The refinements of the platform's SoH data model in Version III (Section 3.4.2) are a direct consequence of this framing: the data points listed above were added to the model precisely so that the assessment could be performed on platform data without recourse to any external information.

16	**Battery information**			
17	Energy	kWh		
18	Nominal voltage	V		
19	Minimum voltage	V		
20	Maximum voltage	V		
21	Cooling concept	text		Passive, Air, liquid indirect), direct, refrigerant
22	P_ch_cont	kW		MAX_Battery_Charging_Power_Cont
23	P_dch_cont	kW		MAX_Battery_Discharging_Power_Cont
24	P_ch_peak	kW		MAX_Battery_Charging_Power 30s
25	P_dch_peak	kW		MAX_Battery_Discharging_Power 30s
26	**State of Health**			
27	date			
28	State of Health	SoH [%]		State of Health capacity (SoH_C)
29	Cyles	#		
30	Capacity	%		Change unit: kWh (SoH in percoent is given above)
31	Int. Resistance delta	% (percent above 100%)		State of Health Resistance (SoH_R), Ratio R_act/R_nominal
32	Energy	kWh		Nominal energy
33	Self-discharge	%/month		
34	Age	days		
35	Battery pack coulombic efficency	%		(=Ah_discharge/Ah_charge) describes the retained Ah discharge to charge per cycle 20% to 80%soc, 0.5C, 25°C)
36	Round Trip Efficency (RTE)	%		(=Wh_discharge/Wh_charge) equvivalent cycle 20% to 80%soc, 0.5C, 25°C
37	Degradation rate capcity	%/EFC		Calculation from SoH_C vector = (difference last two entrance SoH capcity)/(difference last two entrance Equivalent Full Cycles(EFC))
38	Degradation rate resistance	%/EFC		Calculation from SoH_R vector = (difference last two entrance Ratio resistance increase)/(difference last two entrance Equivalent Full Cycles(EFC))
39				
40	Date | SoH | Cycles | Cap. Ret.	date, %, #, % date1, %, #, % ...		Reorder based on cycles: and sort with highest cycles first. SEE Tab: Table Cycles | date | % | %

Figure 4.2. Health parameters and SoH data points defined for each pack. The module and cell levels mirror this structure.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

1					
2	SoH Table				
3					
4	Following the recommendation for the SoH table over time				
5	One entry each 3 Month				
6					
7	**Equivalent full cycles**	**Date**	**Remaining Capacity in % (So**	**Ration resistance increase in %**	**Voltage spread in V**
8	13	01/01/2024	99%	101%	0.018
9	100	01/03/2024	97%	108%	0.010
10	200	01/06/2024	97%	109%	0.008
11					
12	Can be rounded to 0 digit		State of Health capacity	State of Health resistance	Pack/Module cell maximum voltage - cell minimum voltage. Resting voltage spread before balancing after >3 h rest at ~50% SOC and 25°C. >100mV issue
13					
14	May add a diagramm with selective x-axis: cycles/time and both values				
15					
16					
17					
18					
19	Battery Risk				
20					
21	Risks battery gone through, yellow or red CCM related.				
22					
23	**Date**	**Error occured**			
24	01/06/2024	Undervoltage detected		High temperature, insulation monitoring, ...	
25					
26					
27					

Figure 4.3. Example SoH history table recorded in the platform, following the recommended quarterly cadence, together with the associated battery-risk log.

4.3 Methodology

A method was chosen that translates the recorded health parameters into a reuse-or-recycle decision through a set of acceptance checks applied to the most recent SoH record of each entity. The starting point is the battery-lifecycle view in Figure 4.4: a battery in its first life (for example in a vehicle) is characterised through its static and dynamic data; when it reaches the end of its first life, the recorded data is used to decide whether it can enter a second life (for example in a stationary BESS) or whether it should be recycled.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

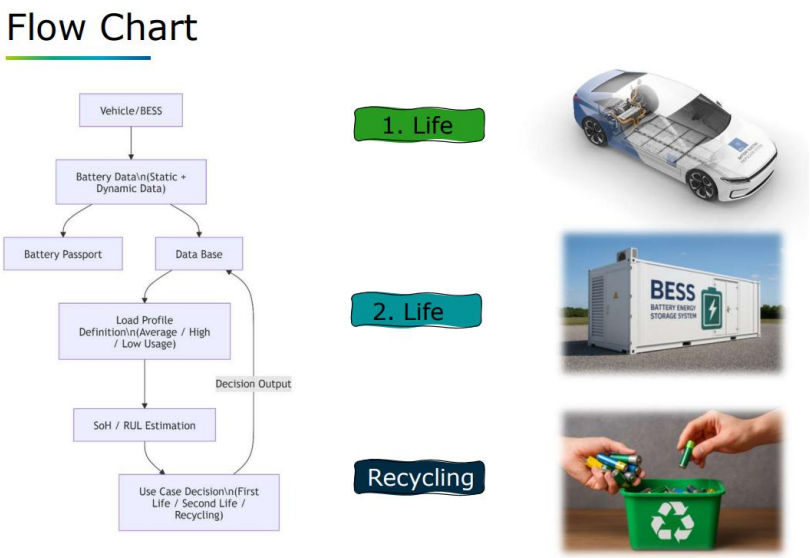


Figure 4.4. Battery lifecycle flow chart: first life, second life and recycling, driven by the recorded battery data.

The decision itself is taken by the digital logic illustrated in Figure 4.5. A battery is highlighted as a second-life candidate when its passport is complete on the major parameters, when no damage has occurred, and when the SoH values are above the defined thresholds. Two of the checks act as mandatory gates - State of Health by capacity and State of Health by resistance - while the remaining checks act as supporting evidence and confidence indicators.

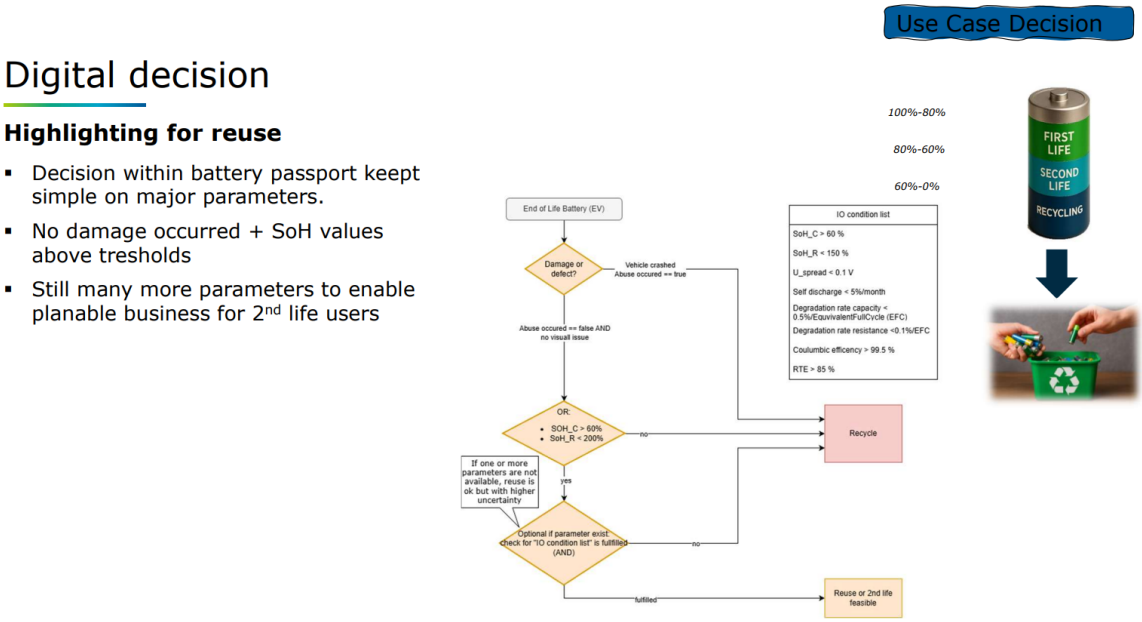


Figure 4.5. Digital decision logic for highlighting a battery for reuse versus routing it to recycling.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

The acceptance criteria agreed with AVL are summarised in Table 4.1.

Assessment check	Gate	Acceptance criterion
State of health by capacity (SoH-C)	Mandatory	> 60 %
State of health by resistance (SoH-R)	Mandatory	< 200 %
State of health by resistance (preferred range)	Preferred	< 150 %
Voltage spread	Supporting	< 0.10 V
Self-discharge	Supporting	< 5 %/month
Coulombic efficiency	Supporting	> 99.5 %
Round-trip efficiency	Supporting	> 85 %
Capacity degradation rate	Supporting	< 0.5 %/EFC
Resistance degradation rate	Supporting	< 0.1 %/EFC

Table 4.1. AVL acceptance criteria for the second-life-versus-recycling decision.

When the two mandatory checks pass, the pack is routed towards second life; the platform additionally reports a confidence level (high, medium or low) and, where supporting evidence is missing or out of range, marks the outcome as "second life possible (uncertainty)" and requires a manual inspection before the decision is saved. When a mandatory check fails - for example when SoH-C falls below 60 % - the pack is routed to recycling. The same logic is applied at module level, so that a single degraded or replaced module can be detected even when the aggregated pack values remain within limits.

4.4 Application to the Battery Passport Platform Data

Eurecat implemented the AVL assessment directly inside the Battery Passport Platform, so that the checks described above are computed automatically from the SoH records held in the platform's data model and surfaced in the user interface. For every pack, module and cell, the platform exposes a State-of-Health view that shows the latest SoH-C and SoH-R against their thresholds, together with the full set of derived indicators - cycles, capacity retention, energy throughput, self-discharge, voltage spread, coulombic and round-trip efficiency and the two degradation rates - and the complete SoH history (Figure 4.6).

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

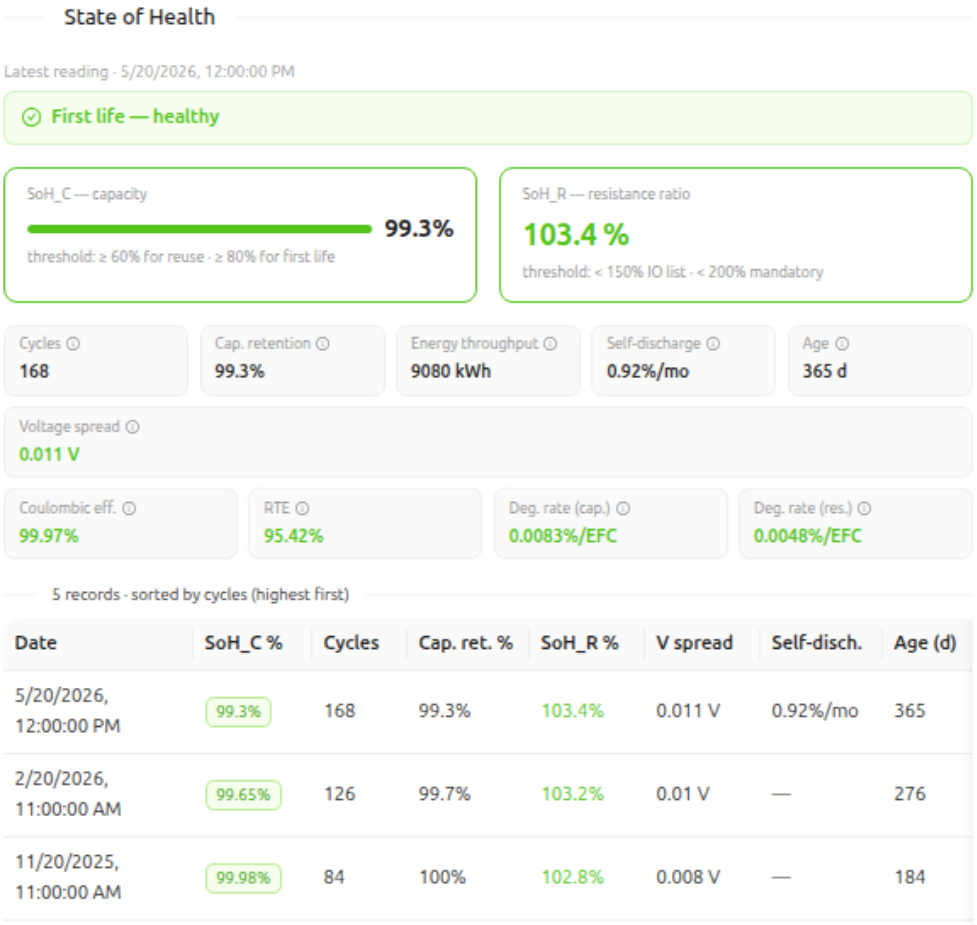


Figure 4.6. State-of-Health view in the platform for a real AVL pack (PACK-REAL-AVL-01), showing the SoH-C and SoH-R gauges, the derived indicators and the recorded SoH history.

From those values the platform produces the AVL assessment table (Figure 4.7): each check is shown with its measured value, its acceptance criterion and a pass/fail result, the two mandatory checks being explicitly flagged as required. At pack level the table also records the source of each value - taken from the worst module, averaged across modules, or calculated from pack history - which makes the aggregation auditable.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

Assessment check	Result	Measured value	Acceptance criterion
State of health by capacity Required	Passed	99.30 %	> 60.00 %
State of health by resistance Required	Passed	103.40 %	< 200.00 %
State of health by resistance (preferred range)	Passed	103.40 %	< 150.00 %
Voltage spread	Passed	0.01 V	< 0.10 V
Self-discharge	Passed	0.92 %/month	< 5.00 %/month
Coulombic efficiency	Passed	99.97 %	> 99.50 %
Round-trip efficiency	Passed	95.42 %	> 85.00 %
Capacity degradation rate	Passed	0.01 %/EFC	< 0.50 %/EFC
Resistance degradation rate	Passed	0.00 %/EFC	< 0.10 %/EFC

Second Life Recycle Pack

Figure 4.7. Automated AVL assessment checks computed by the platform, each with its measured value, acceptance criterion and pass/fail result.

The platform then condenses the checks into an assessment outcome (Figure 4.8): a headline decision (second life feasible, second life possible with uncertainty, or recycling recommended), a confidence level, the count of automated checks passed, the measurement context and · for second-life candidates · a list of recommended applications (secondary electric vehicle, stationary BESS, telecommunications backup, uninterruptible power supply) and, where required, a prompt for manual inspection. The decision is recorded on-chain through the SecondLife and Recycled cell state transitions described in Section 3.5.2, so that the analytical outcome becomes part of the immutable battery passport.

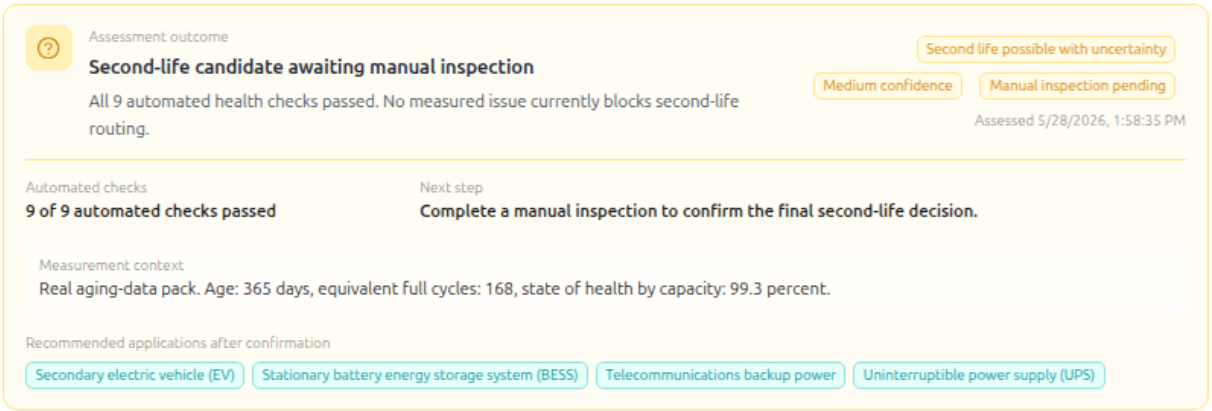


Figure 4.8. Assessment outcome for a real AVL pack: a second-life candidate awaiting manual inspection, with confidence level, measurement context and recommended applications.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

4.5 Lifecycle Optimization

The lifecycle-optimization perspective uses the same data over time rather than at a single instant. Because the platform stores an ordered SoH history for every entity, it can show the evolution of remaining capacity (SoH-C) and resistance increase (SoH-R) against equivalent full cycles and against calendar time (Figure 4.9). The slope of these curves is exactly the capacity and resistance degradation rate used in the assessment, so the same data that supports the instantaneous decision also characterises the trend.

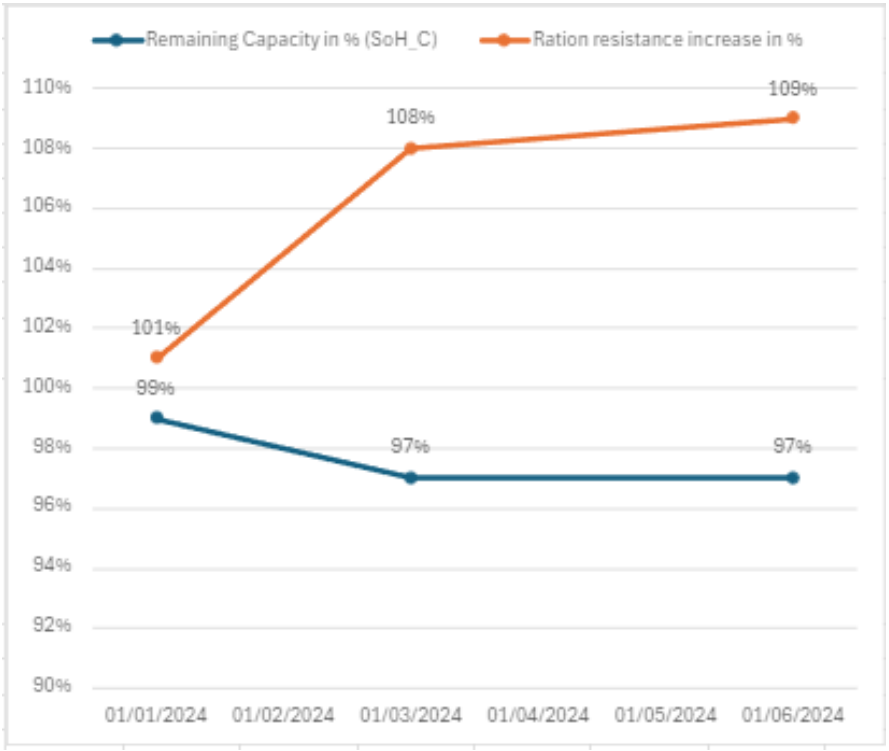


Figure 4.9. Evolution of remaining capacity (SoH-C) and resistance increase (SoH-R) over time, as recorded and plotted by the platform.

This trend is what enables the optimization argument of ST2.4.3: by relating degradation to the recorded usage and load profiles (average, high and low usage, and the SoC windows under which the battery is cycled), it becomes possible to identify the operating strategies that minimise degradation per cycle and therefore extend useful life. The platform does not prescribe an operating strategy, but by capturing the load-profile and SoH data in a structured, comparable form it provides the evidence base on which such an optimization - and the trade-off between performance and longevity - can be made. The real AVL dataset used to exercise this capability is described in Section 5.6.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

4.6 Conclusions of the Analytical Assessment

The analytical assessment confirms that a Battery Passport Platform, when its data model captures the right health parameters, can support data-driven decisions on second-life applicability, recycling timing and lifecycle optimization without recourse to any information held outside the passport. The collaboration between AVL and Eurecat produced three concrete results: a consolidated set of SoH health parameters and their definitions, now part of the platform's data model; an acceptance-criteria framework with two mandatory gates and a set of supporting checks, together with a confidence model and a manual-inspection step; and a working implementation of that framework inside the live platform, in which the checks are computed automatically and the outcome is written back to the immutable passport.

The main limitation of the assessment is the nature of the data available within the project's lifetime. As discussed in Section 5.6, the assessment was exercised against a dedicated, realistic dataset defined by AVL rather than against continuous field telemetry from a specific industrial battery fleet; SoH prediction in particular would benefit from longer real histories than were available. The recommendation for follow-on work is therefore to feed the platform with real operational histories from partner fleets, which would allow the degradation-trend and remaining-useful-life models to be calibrated and validated at scale.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

5. Validation of the Prototype

This chapter documents the validation of the third prototype against the functional requirements introduced in Section 3.1 and against the broader objectives of Version III. It follows the same overall structure as the validation chapter of D2.9, with adjustments for the new components and the new operational reality of a continuously deployed platform. It begins with the validation methodology, continues with the validation of each major component of the platform, includes the validation of the platform against a real AVL assessment dataset (Section 5.6), and ends with a summary of the validation results, including the requirements that are fully validated, those that are partially validated and the limitations identified during validation.

5.1 Validation Methodology

The validation methodology of Version III is based on three complementary activities. The first is end-to-end exercise of the production deployment at https://free4lib.eurecatsecurity.org/, performed by the development team and by selected consortium partners during the second half of the project, using the demo credentials documented on the platform's home page and exercising every role and every workflow that the platform supports. The second is the unit, integration and end-to-end test suites that run as part of the development and deployment pipeline, including Jest unit tests for the backend, Vitest tests for the frontend, Hardhat tests for the smart contracts (with the property-based and fuzz tests described in Section 3.5.3), and the Slither and Mythril security analysis. The third is the validation of the agent integration through a manufacturer-role agent deployed in the development environment, exercising the registration, role activation, signing and submission flow against the same backend that serves the production deployment.

The validation activities are organised by component and, within each component, by functional requirement. For each requirement, the validation establishes a test case description, the expected outcome, the observed outcome and the conclusion (validated, partially validated or not validated). The summary of these conclusions is presented in Section 5.7 of this chapter.

5.2 Deployment Validation

The validation of the production deployment is the most basic and at the same time the most consequential validation activity, because it confirms that the

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

platform can run continuously in a public environment without manual intervention. The deployment has been operational at https://free4lib.eurecatsecurity.org/ since the start of the second half of the project. During this period, the platform has been accessible from the public internet, has served the seeded synthetic dataset to any visitor, has accepted login attempts using the demo credentials, has processed meta-transactions submitted by frontend users and by the test agent, and has maintained the consistency of the off-chain read model with the on-chain state through the oracle subsystem.

The deployment validation also confirmed that the production environment validation described in Section 3.7.3 catches the most common configuration mistakes. During the early phase of the deployment, several iterations were required to bring all the environment variables into a configuration that the validation function would accept, and the validation prevented the platform from starting on multiple occasions until the configuration was corrected. This is the kind of behaviour that is invisible to a casual user but that is critical for any continuously deployed system.

The deployment is verified through the end-to-end validation cases described in the following subsections. Each subsection corresponds to one of the major workflows of the platform and references the functional requirements that the workflow exercises.

5.3 Agent Integration Validation

The validation of the Key Management Agent is the new validation activity introduced in Version III, and it covers the entire lifecycle of an agent from its first startup to its routine operation. The validation was performed with a manufacturer-role agent deployed in the development environment, configured to point at the same backend that serves the production deployment. The following test cases correspond to the agent functional requirements introduced in Section 3.1.4.

5.3.1 Persistent Encrypted Wallet

Test case: Start the agent with no existing keystore and verify that it generates a new wallet, encrypts it with the configured password, writes it to the configured path with mode 0600, and logs the new wallet address. Restart the agent and verify that it loads the existing keystore from the same path, decrypts it with the same password, and reports the same wallet address. Restart the agent with the wrong password and verify that it refuses to start with a clear

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

error message. Outcome: validated. The wallet is generated, persisted and reloaded as expected, and the password mismatch case produces the expected fail-fast behaviour. This validates FR-AG-01.

5.3.2 Self Registration and Idempotency

Test case: Start the agent with a fresh wallet against a backend that does not yet know the agent. Verify that the registration call succeeds, that the user record is created in MongoDB with role no_role, and that the agent receives access and refresh tokens. Restart the agent and verify that the registration call returns 409 Conflict and that the agent handles the conflict gracefully and continues to the role activation step. Outcome: validated. The registration is performed on first startup and is correctly idempotent on subsequent startups. This validates FR-AG-02.

5.3.3 Role Activation Polling

Test case: Start the agent with a fresh wallet and let it enter the role activation polling loop. Verify that it polls the role-status endpoint at the configured interval and that it does not proceed to operational mode until an administrator has assigned the corresponding role both in MongoDB and on-chain. Assign the role through the admin dashboard and verify that the agent detects the change on the next polling iteration and proceeds. Outcome: validated. The polling loop behaves as expected and the agent correctly detects the synchronised authorisation state. This validates FR-AG-03 and indirectly FR-BE-12.

5.3.4 Local EIP-712 Signing and Meta-Transaction Submission

Test case: With the agent fully authorised, instruct it to register a batch of cells through its company-facing API. Verify that it requests a nonce from the backend, encodes the addCellsBatch contract function call with the cell data, signs the EIP-712 MetaTransaction payload locally with the agent wallet, submits the signed payload to the backend's meta-transaction endpoint, and that the backend successfully relays the transaction on-chain. Verify that the corresponding cell documents appear in MongoDB shortly after the transaction is mined. Outcome: validated. The full signing and submission flow works end to end, the resulting cells appear in the off-chain read model with the correct manufacturer attribution, and the transaction can be inspected through the platform's transaction detail page. This validates FR-AG-04, FR-AG-05 and FR-BC-07.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

5.3.5 Token Management

Test case: Allow the agent to operate continuously for a period long enough that its access token expires. Verify that it transparently refreshes the token using the refresh token and that the next operation completes without requiring any external intervention. Outcome: validated. The agent recovers from session expiration without manual recovery. This validates FR-AG-06.

5.4 Data Model and Visualization Validation

The refined data model and the redesigned visualization were validated through manual exercise of the relevant frontend forms and viewers, supported by the strict Data Transfer Object (DTO) validation in the backend. The following test cases correspond to the data-model and visualization functional requirements introduced in Section 3.1.2.

5.4.1 Refined Material Composition Forms

Test case: As a manufacturer, open the cell creation form and enter a complete material composition object covering the chemistry, active materials, hazardous substances, critical raw materials, recycled content figures and carbon footprint indicator. Submit the form and verify that the resulting cell document contains the full material composition object as entered. Repeat with intentionally invalid input (negative percentages, out-of-range recycled content figures, malformed CAS numbers) and verify that the form prevents submission and surfaces the validation errors inline. Repeat the equivalent test for the module and pack creation forms with their respective material composition fields. Outcome: validated. The forms expose the refined material composition fields, the client-side validation prevents the most common errors, and the backend validation catches anything that the client-side validation misses. This validates FR-FE-08 and FR-BE-13.

5.4.2 Redesigned Battery Passport Visualizer

Test case: Navigate to a battery passport through its public URL /battery-passport/{packId}, verify that the page renders the hierarchical structure of the pack as a graph with the pack at the top, the modules below it and the cells below the modules. Click on the pack node and verify that the detail drawer opens and shows the lifecycle state, the manufacturer and assembler information, the dimensions, the material composition rendered with the dedicated viewer, the SoH history and the safety information. Click on a module node and verify that the corresponding details are shown, and similarly for a cell node. Test the QR-code-based access path by scanning a QR code

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

corresponding to a seeded pack and verifying that it opens the same visualization. Outcome: validated. The visualization renders correctly across the seeded dataset, the detail drawer presents the expected information at every level of the hierarchy, and the QR-code-based access works as expected. This validates FR-FE-09.

5.4.3 SoH Recording and Second-Life Workflow

Test case: As an assembler, record a SoH measurement on a pack with the new SoH data points. Verify that the measurement is reflected in the pack's soHRecords array and that the soHAssessment metadata is updated. As a recycler, navigate to a cell that is currently in the InUse state, open the recycle modal, choose the second-life option and verify that the cell transitions to the SecondLife state. Repeat with the recycle option and verify that the cell transitions to the Recycled state. Verify that the corresponding state transitions are also reflected on-chain by inspecting the contract events. Outcome: validated. The SoH recording flow works as expected, the new SoH data points are persisted, the second-life and recycling transitions are correctly enforced both off-chain and on-chain. This validates FR-FE-11, FR-FE-12, FR-BC-08 and the relevant aspects of FR-BC-05.

5.5 Meta-Transaction Validation

The meta-transaction protocol was validated against the EIP-712 typed data signing requirements (FR-BC-07), against the deadline-bounded validity requirement (FR-BC-11) and against the per-user nonce replay protection requirement (FR-BC-04, carried over from D2.9). The validation was performed both through the frontend signing path and through the agent signing path, in order to verify that the protocol is consistent across the two.

Test case: Sign a meta-transaction from the frontend, submit it to the backend, observe the relayed on-chain transaction succeeding and the corresponding event being processed by the oracle. Repeat the same operation immediately, with the same nonce, and verify that the contract rejects the second submission as a replay. Sign a meta-transaction with a deadline in the past and verify that the contract rejects it. Sign a meta-transaction for a contract on a different (test) network and verify that it cannot be replayed against the production deployment. Repeat all the above with the agent as the signer instead of the frontend, verifying that the protocol behaves identically. Outcome: validated. EIP-712 typed data signing is correctly verified by the contract, the per-user nonce prevents replay, the deadline limits the validity window, and the domain

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

separator prevents cross-deployment replay. This validates FR-BC-07, FR-BC-08, FR-BC-11 and the relevant aspects of FR-BC-04.

5.6 Real-Environment Data Validation

The grant agreement description of D2.10 specifies that the final version of the platform should be validated in a real environment. The platform has been deployed in a real production environment at https://free4lib.eurecatsecurity.org/ as documented in Section 3.8 of this deliverable. Beyond the comprehensive synthetic dataset that seeds the public demo, the final version of the platform was also exercised against a real assessment dataset provided by AVL, so that the analytical assessment of ST2.4.3 could be validated live, on actual records, rather than only on generic seed data. The status of the data used for validation is documented honestly in this section.

During the second half of the project, contact was established with one of the consortium partners (Watt4Ever) with the intention of mapping the platform's data model to the fields available in their existing battery datasets and ingesting a subset of their data into the platform. A preliminary data mapping exercise was performed and reported during the 36M consortium meeting. The activity, however, did not progress to actual ingestion: the partner's focus shifted to other priorities, the bilateral effort required to bridge the two data models was higher than initially expected, and the timing did not align with the development schedule of Version III. No other partner came forward with a real battery dataset that could be onboarded within the project's remaining lifetime.

What was nonetheless achieved is a live validation of the platform and of the AVL assessment against a real, purpose-built dataset. AVL defined a set of eleven battery packs (PACK-REAL-AVL-01 to PACK-REAL-AVL-11) representing realistic ageing scenarios: fresh standard packs, fast-aged packs, packs cycled in different state-of-charge windows, a pack with a replaced module, and heavily aged packs at end of life. Unlike the generic synthetic seed, these records carry coherent, realistic SoH histories at pack, module and cell level, and they were ingested into the live platform and assessed through exactly the same automated AVL checks that a real fleet would trigger. This made it possible to test the assessment in operation, on real data, rather than in the abstract.

Table 5.1 reports the dataset together with the outcome produced by the platform for each pack.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

#	Scenario	Platform pack ID	SoH-C %	EFC	SoH-R %	V spread (V)	Self-disch %/mo	Coul. eff %	RTE %	Age (d)	Platform outcome
1	standard	PACK-REAL-AVL-01	99.30	168	103.40	0.011	0.92	99.97	95.42	365	Second life feasible
2	standard	PACK-REAL-AVL-02	100.52	72	101.70	0.007	0.59	99.99	96.47	365	Second life feasible
3	new, fast aged	PACK-REAL-AVL-03	96.90	360	109.20	0.032	1.12	99.94	94.07	730	Second life possible (uncertainty)
4	standard	PACK-REAL-AVL-04	92.25	1008	122.50	0.060	1.77	99.88	91.85	1096	Second life possible (uncertainty)
5	aged 70-80% SoC	PACK-REAL-AVL-05	72.00	1584	172.00	0.185	3.10	99.57	85.20	4748	Second life possible (uncertainty)
6	module replaced	PACK-REAL-AVL-06	87.80	900	132.50	0.092	1.78	99.83	91.28	2192	Second life possible (uncertainty)
7	heavy aged <60% SoC	PACK-REAL-AVL-07	58.00	2220	181.00	0.315	5.02	99.14	78.25	5844	Recycling recommended
8	standard, voltage spread	PACK-REAL-AVL-08	98.36	221	108.80	0.180	0.81	99.96	95.27	546	Second life possible (uncertainty)
9	healthy	PACK-REAL-AVL-09	93.20	640	118.80	0.045	1.38	99.91	92.77	1461	Second life possible (uncertainty)
10	standard 90-10% SoC	PACK-REAL-AVL-10	90.10	760	127.20	0.074	1.86	99.84	90.75	1461	Second life possible (uncertainty)
11	standard 80-90% SoC	PACK-REAL-AVL-11	87.40	920	134.80	0.095	2.10	99.78	89.25	1461	Second life possible (uncertainty)

Table 5.1. The real AVL assessment dataset (PACK-REAL-AVL-01 to 11) ingested into the live platform, with the assessment outcome produced by the platform for each pack.

The eleven cases were chosen to exercise the full range of the decision logic. Examples 1 and 2 are fresh standard packs that pass every check with margin and are routed to second life with high confidence (Figure 5.10). Examples 3, 4, 8, 9, 10 and 11 pass the mandatory gates but trip a supporting check or have incomplete module-level evidence, and are therefore returned as “second life possible (uncertainty)” pending manual inspection. Example 5 is an aged pack cycled in a 70-80 % SoC window whose voltage spread exceeds the 0.10 V limit (Figure 5.11). Example 6 is a pack in which a single module has been replaced, producing a large module-level SoH/EFC mismatch that the module-level checks detect even though the aggregated pack values stay within limits (Figure 5.12). Example 7 is a heavily aged pack whose SoH-C has fallen to 58 %, below the 60 % mandatory gate, and is correctly routed to recycling (Figure 5.13). Health-summary figures are shown only for a representative subset of the packs (Examples 2, 5, 6 and 7); the remaining packs are covered by their row-level outcomes in Table 5.1. Figures 5.9 and 5.14 are context figures — a

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

battery-passport graph and the module inspection form — rather than per-example health summaries.

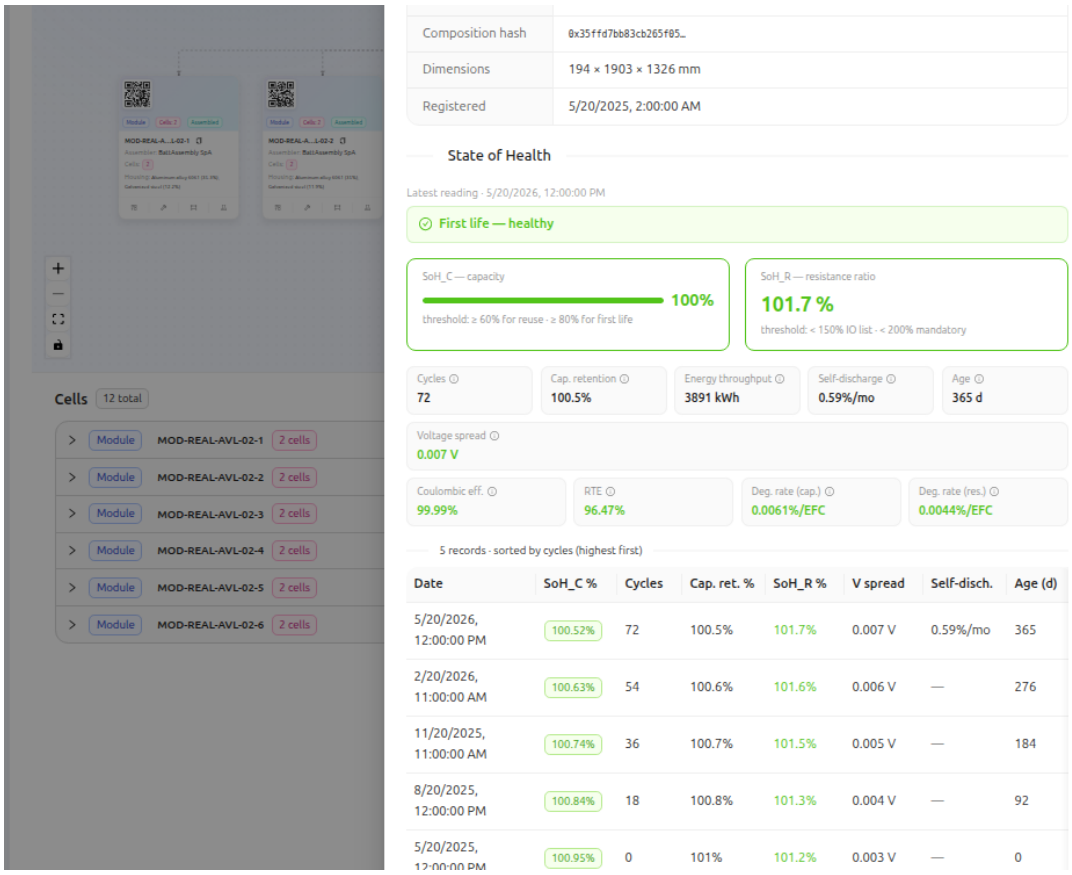
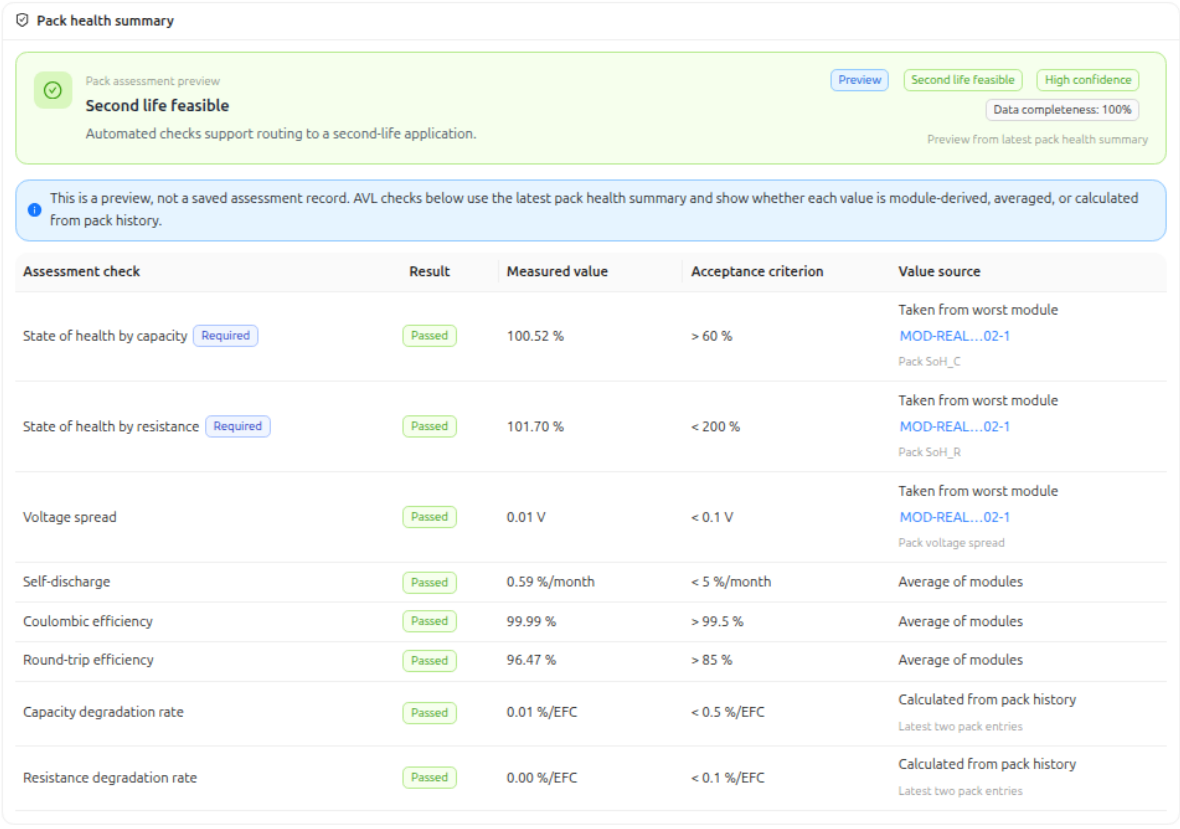


Figure 5.9. Battery passport graph and SoH history for a real AVL pack (PACK-REAL-AVL-02) in the live platform, with its modules and cells.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890



Pack health summary

Pack assessment preview

Second life feasible

Automated checks support routing to a second-life application.

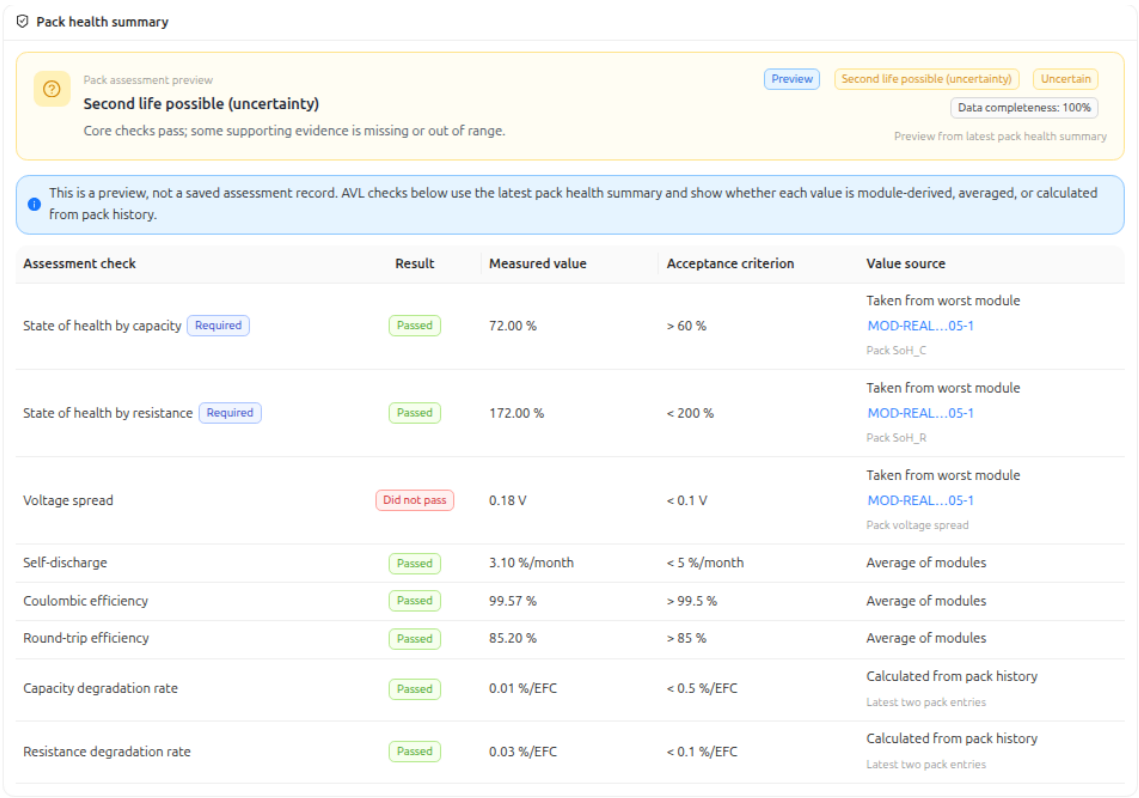
Preview | Second life feasible | High confidence | Data completeness: 100%

Preview from latest pack health summary

This is a preview, not a saved assessment record. AVL checks below use the latest pack health summary and show whether each value is module-derived, averaged, or calculated from pack history.

Assessment check	Result	Measured value	Acceptance criterion	Value source
State of health by capacity Required	Passed	100.52 %	> 60 %	Taken from worst module MOD-REAL...02-1 Pack SoH_C
State of health by resistance Required	Passed	101.70 %	< 200 %	Taken from worst module MOD-REAL...02-1 Pack SoH_R
Voltage spread	Passed	0.01 V	< 0.1 V	Taken from worst module MOD-REAL...02-1 Pack voltage spread
Self-discharge	Passed	0.59 %/month	< 5 %/month	Average of modules
Coulombic efficiency	Passed	99.99 %	> 99.5 %	Average of modules
Round-trip efficiency	Passed	96.47 %	> 85 %	Average of modules
Capacity degradation rate	Passed	0.01 %/EFC	< 0.5 %/EFC	Calculated from pack history Latest two pack entries
Resistance degradation rate	Passed	0.00 %/EFC	< 0.1 %/EFC	Calculated from pack history Latest two pack entries

Figure 5.10. Example 2 - pack health summary: second life feasible, all checks pass with margin.



Pack health summary

Pack assessment preview

Second life possible (uncertainty)

Core checks pass; some supporting evidence is missing or out of range.

Preview | Second life possible (uncertainty) | Uncertain | Data completeness: 100%

Preview from latest pack health summary

This is a preview, not a saved assessment record. AVL checks below use the latest pack health summary and show whether each value is module-derived, averaged, or calculated from pack history.

Assessment check	Result	Measured value	Acceptance criterion	Value source
State of health by capacity Required	Passed	72.00 %	> 60 %	Taken from worst module MOD-REAL...05-1 Pack SoH_C
State of health by resistance Required	Passed	172.00 %	< 200 %	Taken from worst module MOD-REAL...05-1 Pack SoH_R
Voltage spread	Did not pass	0.18 V	< 0.1 V	Taken from worst module MOD-REAL...05-1 Pack voltage spread
Self-discharge	Passed	3.10 %/month	< 5 %/month	Average of modules
Coulombic efficiency	Passed	99.57 %	> 99.5 %	Average of modules
Round-trip efficiency	Passed	85.20 %	> 85 %	Average of modules
Capacity degradation rate	Passed	0.01 %/EFC	< 0.5 %/EFC	Calculated from pack history Latest two pack entries
Resistance degradation rate	Passed	0.03 %/EFC	< 0.1 %/EFC	Calculated from pack history Latest two pack entries

Figure 5.11. Example 5 - pack health summary: the voltage spread (0.18 V) exceeds the 0.10 V limit.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

Pack health summary

Pack assessment preview

Second life possible (uncertainty)

Core checks pass; some supporting evidence is missing or out of range.

Preview | Second life possible (uncertainty) | Uncertain

Data completeness: 100%

Preview from latest pack health summary

This is a preview, not a saved assessment record. AVL checks below use the latest pack health summary and show whether each value is module-derived, averaged, or calculated from pack history.

Assessment check	Result	Measured value	Acceptance criterion	Value source
State of health by capacity Required	Passed	87.80 %	> 60 %	Taken from worst module MOD-REAL...06-1 Pack SoH_C
State of health by resistance Required	Passed	132.50 %	< 200 %	Taken from worst module MOD-REAL...06-1 Pack SoH_R
Voltage spread	Passed	0.09 V	< 0.1 V	Taken from worst module MOD-REAL...06-1 Pack voltage spread
Self-discharge	Passed	1.78 %/month	< 5 %/month	Average of modules
Coulombic efficiency	Passed	99.83 %	> 99.5 %	Average of modules
Round-trip efficiency	Passed	91.28 %	> 85 %	Average of modules
Capacity degradation rate	Passed	0.01 %/EFC	< 0.5 %/EFC	Calculated from pack history Latest two pack entries
Resistance degradation rate	Passed	0.01 %/EFC	< 0.1 %/EFC	Calculated from pack history Latest two pack entries

Figure 5.12. Example 6 - pack health summary: a module has been replaced; the aggregated pack checks remain within limits while module-level checks flag the mismatch.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

Pack health summary

Pack assessment preview
Recycling recommended
One or more required checks do not support automated second-life reuse.

Preview | Recycling recommended | Uncertain
Data completeness: 100%
Preview from latest pack health summary

This is a preview, not a saved assessment record. AVL checks below use the latest pack health summary and show whether each value is module-derived, averaged, or calculated from pack history.

Assessment check	Result	Measured value	Acceptance criterion	Value source
State of health by capacity Required	Did not pass	58.00 %	> 60 %	Taken from worst module MOD-REAL...07-1 Pack SoH_C
State of health by resistance Required	Passed	181.00 %	< 200 %	Taken from worst module MOD-REAL...07-1 Pack SoH_R
Voltage spread	Did not pass	0.32 V	< 0.1 V	Taken from worst module MOD-REAL...07-1 Pack voltage spread
Self-discharge	Did not pass	5.02 %/month	< 5 %/month	Average of modules
Coulombic efficiency	Did not pass	99.14 %	> 99.5 %	Average of modules
Round-trip efficiency	Did not pass	78.25 %	> 85 %	Average of modules
Capacity degradation rate	Passed	0.01 %/EFC	< 0.5 %/EFC	Calculated from pack history Latest two pack entries
Resistance degradation rate	Passed	0.02 %/EFC	< 0.1 %/EFC	Calculated from pack history Latest two pack entries

Figure 5.13. Example 7 - pack health summary: SoH-C at 58 % fails the mandatory gate and recycling is recommended.

Record inspection and save module assessments

This records inspection evidence and saves a module assessment for each selected module.
Failed inspection, damage, abuse, or visual issue also creates a module risk event and blocks second-life routing on the affected modules only.

Affected modules
All modules

* Inspection result
Passed — no blocking condition found

Inspection notes
Record observations, checks performed, or evidence of any defect...

Damage or defect | Abuse or crash | Visual safety issue

Recorded by: 0x1164697bb17EaB2DA95266b380d691A853a9ac08

Record inspection and save assessments

Figure 5.14. Module inspection record form used to confirm or block second-life routing for a real AVL pack.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

Two qualifications are recorded for completeness. First, this dataset, although real in the sense of representing coherent and realistic battery histories defined by the domain partner, is not continuous field telemetry harvested from a specific industrial battery fleet; a validation against such telemetry remains desirable and is recommended as follow-on work. Second, the generic synthetic seed described in Section 3.8 continues to be used for the public demo and for exercising the non-analytical code paths. With these qualifications, the real-environment data validation foreseen for D2.10 has been carried out: the platform and the AVL analytical assessment were validated live, against a real dataset, and produced the correct reuse-or-recycle decision for every one of the eleven packs.

5.7 Summary of Validation Results

The validation activities described in the previous subsections produce the following picture of the state of Version III against its functional requirements.

5.7.1 Fully Validated Requirements

The following requirements have been fully validated against the production deployment and the test agent integration.

- FR-BE-08 (Refresh Token Rotation), FR-BE-09 (Login Rate Limiting and Lockouts), FR-BE-10 (Trace Identifier Propagation), FR-BE-11 (Production Environment Validation), FR-BE-12 (Role Status Endpoint for Agents), FR-BE-13 (Refined Material Composition Storage), FR-BE-14 (Dual Identifier Lookup), FR-BE-15 (Operational Metrics Endpoint).
- FR-FE-08 (Refined Material Composition Forms), FR-FE-09 (Redesigned Battery Passport Visualizer), FR-FE-10 (Wallet Password Modal), FR-FE-11 (Refined SoH Recorder), FR-FE-12 (Recycler Second-Life Workflow), FR-FE-13 (Transaction History and Detail Pages).
- FR-BC-07 (EIP-712 Typed Meta-Transactions), FR-BC-08 (SecondLife Cell State), FR-BC-09 (Dual Identifier Support), FR-BC-10 (Security Tooling Pipeline), FR-BC-11 (Deadline-bound Meta-Transactions).
- FR-AG-01 (Persistent Encrypted Wallet), FR-AG-02 (Self Registration), FR-AG-03 (Role Activation Polling), FR-AG-04 (Local EIP-712 Signing), FR-AG-05 (Company-Facing Simplified API), FR-AG-06 (Token Management), FR-AG-07 (Operational Logging).

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

5.7.2 Partially Validated and Pending Aspects

The following aspects of the validation are not in a fully validated state at the time of writing this deliverable, and are documented here.

- **Real-environment data validation.** As described in Section 5.6, the platform and the AVL assessment were validated live against a real dataset of eleven packs defined by AVL, producing the correct outcome in every case. What remains outstanding is a validation against continuous field telemetry from a specific industrial partner fleet, which could not be onboarded within the project's lifetime.
- **Long-term operational behaviour.** The deployment has been operational continuously since the start of the second half of the project, but the long-term behaviour of the platform under sustained external load has not been characterised. The throughput and latency observed during normal operation are well within the limits required by the project, but no formal load test has been conducted.
- **Multi-agent operation.** The agent integration has been validated with a single manufacturer-role agent. The simultaneous operation of multiple agents in different roles, and the resulting concurrency on the relayer side of the backend, has not been the subject of a dedicated validation activity. There is no architectural reason to expect a problem, but a formal validation would require additional time.
- **AVL analytical assessment.** The analytical assessment of subtask ST2.4.3 has been completed by AVL and is reported in Chapter 4; it has been implemented inside the live platform and validated against the real AVL dataset (Section 5.6). The one aspect that remains for future work is the calibration of the SoH-prediction and degradation-trend models against long real operational histories, which were not available within the project's lifetime.

5.7.3 Limitations Identified During Validation

The validation activities also identified a small number of limitations that do not violate any functional requirement but that are worth documenting because they shape the recommendations in the final chapter of this deliverable.

- **Eventual consistency latency between the chain and the read model.** When a meta-transaction is submitted from the frontend or from the agent, the corresponding update to the off-chain read model is not immediate: it depends on the oracle picking up the Executed event from the chain, decoding it and writing the resulting documents to MongoDB.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

In normal operation, this latency is in the order of seconds and is acceptable. Under adverse conditions, however, the latency can grow, and the user-facing experience can be confusing if a freshly submitted operation is not immediately reflected in the dashboard. The frontend mitigates this by showing a transaction-pending state, but the underlying latency is a property of the architecture and cannot be fully eliminated as long as the read model is built from on-chain events.

- **Operational complexity of running both Style A and Style B write paths.** The hybrid write model described in Section 3.2.2 is what makes the architecture practical, but it also creates a non-trivial operational burden: a maintainer of the platform needs to know which write path is in use for each operation in order to reason about its behaviour. This is not a problem in the current scope, where the development team is also the operations team, but it would become a real burden in any wider operational model.
- **Lack of automated end-to-end test coverage for the agent integration.** The agent functional requirements are validated through manual end-to-end exercise, not through an automated end-to-end test suite. Adding an automated suite would be a relatively modest piece of additional work and is recommended in the final chapter.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

6. Critical Assessment of Blockchain Technology in a Battery Passport Context

The previous chapter established that the platform works: that it can be deployed, integrated and exercised against real data, and that it produces correct results. This chapter turns from whether the platform works to whether it should have been built on a blockchain at all — the question that gives Task T2.4 its research character.[2]

This chapter is the intellectual core of the deliverable. It is also the chapter that justifies the existence of T2.4 as a research task rather than as a development project. T2.4 was set up to explore, deploy and validate a blockchain-based Battery Passport Platform with the explicit purpose of producing an evidence-based answer to a single question: does blockchain technology pay for itself in this context, once the cost of complexity is honestly accounted for? Three iterations of the platform, three deliverables and four years of development provide a substantial basis on which to address that question. This chapter presents that assessment.

The chapter is organised around the question itself rather than around the architecture of the platform. It begins by framing the question in the terms that matter for an evidence-based answer. It then examines, one by one, each of the mitigations that the project introduced across the three prototypes to make blockchain usable in a battery passport context, asking in each case what problem the mitigation solved and what problem it created. It then identifies the cases in which blockchain technology genuinely helps in this context, and the cases in which simpler approaches achieve the same outcome at a fraction of the cost. It examines the multi-party network argument that is most often cited in favour of blockchain in supply-chain settings, and the limits of that argument as observed in practice. It examines the single-authority counter-argument under which a future EU-managed registry could be the most natural home for a battery passport platform. And it ends with a verdict that is intended to be useful to anyone considering the same architectural path, whether inside the FREE4LIB consortium, in a follow-on project or in a wider EU initiative. The

[2] The same question — whether a blockchain is warranted for digital product passports — has also been examined in the EU policy literature; see European Union Blockchain Observatory and Forum, “Digital Product Passport: A Blockchain-Based Perspective,” https://blockchain-observatory.ec.europa.eu/publications/digital-product-passport-blockchain-based-perspective_en.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

verdict is fact-based and is not flattering to the architecture documented in the rest of this deliverable, but the project's own findings are what justify it.

6.1 Framing the Question

The question of whether blockchain technology pays for itself in a Battery Passport context is often answered, in the literature and in marketing material, in terms that are too vague to be useful. Statements like 'blockchain provides immutability and trust' or 'blockchain enables transparency across the value chain' are technically defensible but they do not survive any engagement with the specific properties of the system being built or with the specific properties of the alternatives. They are abstract properties of an idealised technology, not properties of any deployed system. To answer the question honestly, it has to be reformulated in terms that allow it to be tested against evidence.

Three reformulations, each of which can be read as a sub-research question, are useful. The first is functional: what concrete problem does the use of blockchain solve in this context that could not be solved, or could be solved only with significantly more difficulty, by a conventional system architecture? The second is operational: what concrete cost — in development time, in operational complexity, in user experience friction, in integration burden, in security surface — does the use of blockchain impose, and is that cost commensurate with the benefit identified in the answer to the first question? The third is structural: under what conditions, if any, does the answer to the first two questions change in favour of blockchain, and how close is the FREE4LIB context to those conditions?

These three reformulations are the framework used in the rest of this chapter. They are deliberately neutral, in the sense that they do not assume an answer in either direction, and they are deliberately specific, in the sense that they refer to concrete properties of concrete systems rather than to abstract properties of an idealised technology.

It is also worth being explicit about the kind of trust that a Battery Passport Platform needs to provide and the way blockchain is supposed to provide it. The relevant trust property is the assurance that the data carried by a battery passport has not been altered after the fact and that its provenance can be verified. This is an integrity property, not a confidentiality property and not an availability property. Blockchain provides this property by design, in the sense that an entry written to a blockchain cannot be silently altered without leaving a trace and that the provenance of every entry is recorded as part of the chain itself. The relevant question is not whether blockchain provides this property

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

— it does — but whether other architectures can provide an equivalent property at a lower cost, and whether the additional properties that blockchain provides on top (decentralisation, censorship resistance, smart-contract-enforced workflows) are useful in this specific context.

6.2 Mitigations Attempted Across the Three Prototypes

The three iterations of the FREE4LIB Battery Passport Platform are, in retrospect, a sequence of mitigations against the friction that blockchain introduces when it is asked to behave like a conventional information system. Each mitigation solved a real problem and introduced a new one. Examining them in sequence is the most direct way to make the trade-off visible, because it shows where the complexity actually accumulates.

6.2.1 Custodial Wallet with Client-Side Decryption

Problem solved: ordinary users cannot reasonably be asked to manage Ethereum-style private keys. Asking a manufacturer's quality engineer to generate a wallet, back up a seed phrase, install a wallet extension and remember to use it before recording a battery event is a non-starter for any practical adoption. The custodial wallet pattern introduced in D2.9 sidesteps this by generating the wallet at registration, encrypting it with the user's password, storing the encrypted keystore on the platform, and decrypting it client-side when the user wants to sign an operation.

Problem created: the entire purpose of holding a wallet — being the cryptographically accountable owner of a key — is partially undermined by the fact that the encrypted keystore lives on the platform's servers. The platform cannot decrypt the keystore without the password, but the platform is the entity that delivers the password modal to the user, so a malicious or compromised platform could trivially capture the password by modifying the modal it serves. The user's signature, in this model, is therefore as trustworthy as the platform that hosts the modal, which is essentially the trust model of a conventional authenticated system. The cryptographic structure has been preserved but the trust property it was supposed to deliver has been weakened to the point where it is no longer the meaningful contribution of the architecture.

This mitigation is necessary for usability and unavoidable for any deployment that wants to onboard users without specialised knowledge. Its cost is precisely the property that distinguishes blockchain from a conventional system, in the demographic of users that the mitigation is meant to serve.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

6.2.2 Meta-Transactions and the Relayer

Problem solved: ordinary users cannot reasonably be asked to hold cryptocurrency. Even on a low-cost network like Polygon, asking a manufacturer to maintain a balance of MATIC in order to pay gas for battery events is operationally awkward and conceptually strange — the cost has nothing to do with the value of the operation being performed. The meta-transaction pattern introduced in D2.9 sidesteps this by having the user sign their intent and a relayer wallet pay the gas on their behalf. The platform recovers the cost of the relayer through whatever business model it considers appropriate (in the FREE4LIB case, the relayer is funded by the project).

Problem created: the relayer becomes a single point of operational responsibility and, depending on the network, a potential single point of failure. If the relayer wallet runs out of gas, no operations can be performed, even by users who have valid signatures. If the relayer is compromised, the platform can be flooded with transactions that consume the relayer's balance without doing anything useful. If the relayer is taken offline by an external decision (for example, a regulatory action against the operator of the platform), the cryptographic accountability of the system continues to exist on the chain, but the practical ability to write to it disappears.

More importantly, the meta-transaction pattern reintroduces the very property that the original blockchain design was meant to eliminate: a trusted intermediary. The relayer is a centralised actor whose continued operation is required for the system to function. Once that intermediary is in place, the question of whether the underlying chain is decentralised becomes much less relevant, because the practical bottleneck is no longer the chain but the relayer. And once a centralised intermediary is required for the system to function, the conceptual gap between the blockchain-anchored architecture and a conventional database with cryptographic audit trails narrows considerably.

6.2.3 Off-Chain Oracle and Read Model

Problem solved: querying data directly from a blockchain is slow, expensive and structurally limited. Most read patterns required by a usable platform — listing all cells produced by a given manufacturer in a given date range, sorting modules by SoH, paginating through transaction history, joining cell data with module data — are operations that the chain itself cannot perform efficiently, because the chain is structured as an append-only log of state transitions, not as a queryable database. The oracle pattern introduced in D2.9 sidesteps this by listening to events emitted by the contracts and materialising the

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

corresponding documents in MongoDB, which is then queried by the API and the frontend.

Problem created: the read model in MongoDB is a derived projection of the state on the chain. As such, it can drift from the on-chain state if the oracle is delayed, if it processes an event incorrectly or if a new operation is added to the contracts without updating the oracle. The platform now has two sources of truth that have to be kept in sync, and the synchronisation is not trivial because the chain and the database have different consistency models. More importantly, the read model is the part of the system that the user actually sees. The user does not see the chain; the user sees the dashboard, which is rendered from MongoDB. From the user's perspective, the trustworthiness of the data is the trustworthiness of the read model, which is again the trustworthiness of a conventional database. The chain is still there as the underlying source of truth, and it can be consulted in the event of a dispute, but the day-to-day experience of using the platform is the experience of using a conventional system with a blockchain attached.

This is the most significant of the four mitigations because it is the one that exposes the tension between the storage model that blockchain provides and the storage model that a usable system needs. The oracle and the read model are not optional; without them, the platform would be unusable. With them, the architectural argument for blockchain becomes much weaker, because the user-facing experience is now indistinguishable from the experience of a database-backed system.

6.2.4 The Key Management Agent

Problem solved: human users cannot drive the data-entry rate that real industrial environments produce. The agent introduced in Version III sidesteps this by providing a system-to-system gateway that organisations can deploy inside their own infrastructure, holding their own keys and exposing a simplified API to their internal systems. From the platform's point of view, the agent is just another authenticated user signing meta-transactions. From the organisation's point of view, the agent is a thin integration layer that does not require any blockchain expertise inside the organisation's other applications.

Problem created: the agent restores some of the cryptographic accountability that the custodial wallet weakened, because the agent's keys never leave the organisation, but it does so by reintroducing the operational complexity of running a key-holding service. The organisation now has to manage a wallet, back up a keystore, rotate a password, monitor the agent's logs and respond to its failures. These are not unfamiliar problems for any organisation that

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

already runs services holding cryptographic material, but they are real costs that have to be borne in exchange for the cryptographic accountability that the blockchain layer is supposed to provide. Whether they are commensurate with the benefit depends entirely on what the organisation is trying to achieve.

In the specific case of the FREE4LIB platform, where the network is operated by the project and the data is essentially managed by Eurecat as the platform operator, the agent's keys protect the integrating organisation from the platform operator. This is a meaningful protection in a multi-party scenario where the platform and the organisations are independent and possibly adversarial parties. It is a less meaningful protection in the actual scope of the project, where the platform operator and the organisations are all part of the same consortium working toward the same goal. The agent has been designed to be useful in the former case, but it has been deployed in the latter, and the benefit in the latter is correspondingly smaller than the benefit in the former.

6.3 Where Blockchain Genuinely Helps

Having examined the mitigations, the next step is to identify the situations in which the blockchain layer of the platform makes a genuine difference, in the sense that the same outcome would not be achievable, or would be significantly harder to achieve, with a conventional architecture. There are some such situations and they are worth naming explicitly.

The clearest example is the audit-after-the-fact scenario. Suppose that, several years after the events have been recorded, a dispute arises about whether a manufacturer faithfully reported the recycled content of a batch of cells, or whether a recycler correctly reported the SoH of a battery before declaring it suitable for second life. In this scenario, the blockchain layer provides a record of when each event was registered, by which cryptographic identity, and what payload was attached to it. An auditor with access to the chain can verify that the on-chain record is consistent with the off-chain record stored by any party, and any divergence is, in itself, evidence that something has been altered after the fact. This is a property that a conventional database can also provide through cryptographic audit trails (timestamping, hash chains, signed log entries), but it is a property that the blockchain layer provides without any additional engineering, because it is a side effect of the consensus mechanism.

A second example, specific to multi-party scenarios, is the cross-organisation event ordering. If two organisations each maintain their own database of battery events and they need to agree, after the fact, on the order in which events occurred, they have to either trust a single timestamp authority or rely

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

on a more sophisticated coordination mechanism. A shared blockchain provides an inherent ordering of events, with the property that all participants observe the same order and that no participant can rewrite the order without the others noticing. This is a useful property in scenarios where multiple organisations are recording events that interact in ways that depend on their order — for example, a recycler claiming responsibility for a cell that a previous owner has not yet released.

A third example, also specific to multi-party scenarios, is the smart-contract enforcement of cross-organisation workflows. The lifecycle state machines enforced by the Battery Passport contracts are, in principle, a way to ensure that the parties responsible for different stages of a battery's life all agree on the rules of those transitions. A cell cannot move from InUse to SecondLife unless the recycler with the corresponding role calls the function on the contract, the contract enforces the constraint, and the constraint is the same for everyone. In a multi-party scenario, this is a useful property because it removes the need for bilateral agreements about how to handle the boundaries between the responsibilities of different organisations. In practice, however, the value of this property depends on whether the parties involved actually need or want their workflows to be enforced this way, and on whether the contract correctly captures the rules they would have agreed on otherwise.

These three examples are the strongest cases for blockchain in the context of the platform documented in this deliverable. They share a common pattern: they are most valuable when there are multiple independent parties whose relationships are not fully trusted, when the events being recorded are of long-term consequence and when an auditor or an arbitrator needs to be able to reconstruct the history of events without trusting any single party. They are weakest when the parties are part of the same consortium working toward the same goal, when the events being recorded are operational rather than legally consequential, and when the audit, if it ever happens, would be performed by a party that all the participants trust anyway.

6.4 Where Simpler Approaches Achieve the Same Outcome

The cases in which simpler approaches achieve the same outcome as the blockchain-anchored architecture are, in the FREE4LIB scope, considerably more numerous than the cases in which they do not. Several of them deserve to be made explicit because they are not obvious from a distance and because they shape the conclusion of this chapter.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

The first case is the integrity of stored data. The platform stores battery passport data in MongoDB and writes the corresponding events to the chain. The chain is the source of truth, but the day-to-day reads are served from MongoDB. A conventional architecture could achieve the same integrity property by storing the events in any append-only data store (a write-once log file, a hash-chained event table in a relational database, a signed log structure such as the ones used by certificate transparency systems) and signing each event with the writer's key. The signatures, the hash chain and the timestamps would provide the same after-the-fact verification properties as the blockchain layer, with significantly less complexity and at significantly lower operational cost. Software libraries for this kind of cryptographic audit trail are mature, well understood and require no consensus mechanism, no relayer and no network of nodes. The reason this approach is not more visible in the public discourse around supply-chain transparency is essentially marketing rather than technology: 'cryptographic audit trail' is not a phrase that captures the imagination the way 'blockchain' does, even though the underlying property is the same.

The second case is access control. The Battery Passport contracts enforce role-based access control through the RoleManager contract, which records the roles assigned to each address and is consulted by the other contracts before any state-modifying operation. This is a useful property if the trust model requires that no single party can grant or revoke roles unilaterally. It is, however, a property that the platform does not actually need, because the role assignments are made by an administrator with credentials in the platform's admin dashboard, and that administrator could grant the same roles in a conventional access control system with exactly the same effect. The on-chain role enforcement adds a second layer that the platform's backend has to consult, which is the source of the role-status endpoint discussed in Section 3.7.1, but the underlying authority is the same in both cases. A conventional access control system would have produced the same observable behaviour with less code.

The third case is the user experience of the platform. The frontend dashboards, the visualisation of the hierarchical structure, the QR-code-based public viewer, the role-based workflows — none of these depend on the blockchain layer in any meaningful way. They are conventional web application features that read data from a database and write data to a database. They could be wired to a conventional backend without any change to their structure, and the user would not be able to tell the difference. The blockchain is invisible to the user under normal operation, and the cases in which it becomes visible (the eventual

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

consistency latency between the chain and the read model, the password modal that gates blockchain-write operations) are friction points rather than benefits.

The fourth case, which is the most subtle, is the trustworthiness of the data from the perspective of an external consumer. Consider a regulator who wants to verify that the carbon footprint reported by a manufacturer for a given pack is the value that the manufacturer originally registered, not a value that has been adjusted later. In the blockchain-anchored architecture, the regulator can in principle consult the chain and verify the original value. In practice, however, the regulator does not consult the chain directly; the regulator consults the platform's user interface, which reads from MongoDB, which is a derived projection of the chain. For the regulator's trust in the displayed value to be grounded in the chain, the regulator needs to either run their own oracle against the chain or trust the platform to faithfully render the chain's contents. In the first case, the regulator is operating their own integration; in the second case, they are trusting the platform, which is exactly the trust they would have to grant to a conventional system. The blockchain layer provides a fallback in the case where the platform's UI is in dispute, but in normal operation it is not what produces the trust.

6.5 The Multi-Party Network Argument and Its Limits

The strongest argument in favour of blockchain in supply-chain settings is the multi-party network argument: the property that a single shared data substrate, accessible to all parties on equal terms, eliminates the need for bilateral data-exchange agreements between every pair of parties. Instead of N×N integrations, the parties have N integrations with the substrate. Instead of bilateral disputes about the state of the world, the parties have a shared view of it. Instead of trusting each other, the parties trust the substrate. This is a real and meaningful argument, and it is the one that has motivated the deployment of blockchain technology in projects like food safety traceability, container shipping and a range of supply-chain initiatives in the late 2010s.

The argument has limits that are not always discussed honestly. The first limit is that the substrate is only useful if the parties actually use it. Convincing N independent organisations to integrate their internal systems with a shared substrate, to map their data models to a common schema, to manage cryptographic identities, to agree on the governance of the substrate and to absorb the integration cost is a significant adoption challenge, often larger than the engineering challenge of building the substrate itself. Many of the supply-

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

chain blockchain projects of the late 2010s failed not because the technology was insufficient but because the integration cost on the participating organisations was too high relative to the benefit they perceived. The Key Management Agent introduced in Version III of the FREE4LIB platform is precisely an attempt to lower this integration cost, and Section 6.2.4 has already discussed how successful it can be expected to be.

The second limit is that the substrate has to actually be neutral with respect to the participating parties. If the substrate is operated by one of the parties, or by a vendor closely associated with one of the parties, the neutrality argument that motivates the use of a shared substrate evaporates, and the parties might as well use a conventional database operated by the same operator. If the substrate is operated by a neutral third party — for example, a public blockchain like Ethereum — the parties have to accept the operational costs and uncertainties that come with relying on a public chain (gas prices, confirmation times, regulatory exposure of the chain itself), and these costs are not negligible. If the substrate is a consortium chain operated jointly by the participating parties, the governance of the consortium becomes the new problem to solve, and the history of consortium blockchains shows that this governance problem is often the hardest part of the project.

The third limit is that the substrate has to be the right size for the problem. A blockchain that is shared among too few parties is essentially a database with extra steps, because the trust property that the chain provides is only meaningful when the parties are independent and not all controlled by the same entity. A blockchain that is shared among too many parties suffers from coordination overhead, because every change to the contracts, every update to the schema and every operational decision has to be agreed upon by a large number of independent stakeholders. The sweet spot, where the chain is shared among enough parties to be meaningfully decentralised but not so many as to be operationally paralysed, is narrow and specific to each domain.

In the FREE4LIB case, the substrate is the Alastria network, which is a semi-public blockchain operated as a Spanish national consortium and used in the project as a stand-in for a hypothetical future EU network. The platform has been deployed against this substrate, but the data that flows through it is generated by a single party (Eurecat, as the platform operator). The multi-party network property is therefore present in the substrate but not exercised in the deployment. To exercise it, the platform would need to be used by multiple independent organisations writing data through their own agents. The Key Management Agent makes this technically possible, but it is not the same as having it actually happen.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

6.6 The Single-Authority Counter-Argument

The mirror image of the multi-party network argument is the single-authority counter-argument: in a context where the data is ultimately managed by a single entity, the case for blockchain weakens dramatically. If a regulator or a single industrial operator is the ultimate authority on what is recorded and what is not, the integrity properties that blockchain provides can be replicated by that entity using cryptographic audit trails, and the additional properties (decentralisation, censorship resistance, smart-contract enforcement) become moot, because the entity is the one deciding what to record in the first place.

The European Union is the most relevant example of a single authority in this domain. The EU Battery Regulation envisages a registry — a single, regulated, EU-managed point of access for the data that battery passports must carry. If the EU operates this registry, the trust model is that the EU is the trusted authority and the participating organisations submit data to the registry under that authority. In this model, the registry could be a conventional database with cryptographic audit trails, operated by the EU with the same kind of governance and accountability that the EU applies to its other digital infrastructure. The integrity properties would be guaranteed by the operator's accountability rather than by a consensus mechanism, and the participating organisations would interact with the registry through standardised APIs.

Could blockchain still play a role in this model? In principle, yes, in two specific ways. First, as the substrate of the registry itself: the EU could choose to operate the registry on a permissioned blockchain in order to gain the technical properties of immutability, ordering and smart-contract enforcement without having to engineer them from scratch in a conventional database. This is a defensible choice but it is not the only defensible choice, and the engineering reasons to prefer it are not overwhelming once the EU itself is the operator. Second, as a substrate for cross-border collaboration between member states: the EU could choose to operate the registry as a federation of national nodes coordinated by a shared blockchain layer, with each member state running its own node and the blockchain layer providing the coordination between them. This is the scenario in which blockchain technology has the most credible value in a battery passport context, because it combines the multi-party network property with a clear governance model (the EU and its member states) and a clear set of users (the regulated organisations in each member state).

Even in this scenario, however, the engineering case for blockchain is not automatic. A federated database with strong cross-border governance and well-defined APIs could achieve the same operational properties as a federated

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

blockchain with significantly less complexity. The choice between the two is a choice about architectural philosophy and about the political acceptability of different governance models, not a choice that is forced by the underlying requirements.

The honest version of the single-authority counter-argument, then, is that as soon as a single trusted authority is part of the picture, the technological case for blockchain is no longer determined by the requirements; it becomes a matter of preference, of governance and of the operator's appetite for the additional complexity. The FREE4LIB platform is deployed in a context that is, structurally, much closer to the single-authority case than to the multi-party network case. The platform is operated by one entity (Eurecat), the data is generated by parties that all belong to the same consortium, the dispute-resolution authority is the consortium itself, and there is no external regulatory authority in the FREE4LIB context that would behave differently from a conventional system operator. The technological case for blockchain in this specific context is therefore weaker than the case for blockchain in a multi-party adversarial context, and the platform documented in this deliverable should be evaluated against that recognition.

6.7 Verdict: Complexity vs. Value

The verdict that emerges from the previous sections is uncomfortable for anyone who would prefer a more affirmative ending to a four-year development effort, but it is the verdict that the evidence supports.

In the closed scope of the FREE4LIB project, where the platform is operated by a single trusted custodian, the data is generated by parties that all belong to the same consortium, no live data exchange with external systems takes place during the project's lifetime, and no adversarial verification between independent organisations is exercised, the value added by the blockchain layer over a well-designed conventional database with cryptographic audit trails is marginal. Every concrete property that the platform delivers to its users in this scope — integrity of the stored data, accountability of the writers, role-based access control, hierarchical visualisation, public access through QR codes, machine-to-machine integration through agents — could have been delivered by a conventional architecture, with significantly less engineering effort, with significantly less operational complexity and with the same observable behaviour from the user's point of view. The mitigations that the project introduced across the three iterations to make blockchain usable in this context are real and they work, but each of them moves the architecture closer to the

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

conventional one it was supposed to replace, and the cumulative effect of the four mitigations described in Section 6.2 is that the user-facing system is essentially indistinguishable from a database-backed system with a blockchain attached as an audit substrate.

This is not a failure of the engineering. The engineering in Versions I, II and III is solid, the resulting platform is functional and the codebase is clean. It is a failure of the original premise, which was that blockchain technology would naturally and obviously add value in a battery passport context. The premise turns out to be false in the specific scope that the project was able to reach, and the project's most valuable contribution is the fact that it can now say so on the basis of evidence.

There are scenarios in which the verdict would be different. If the platform were used by multiple independent organisations writing data through their own agents, with each organisation holding its own keys and with cross-organisation workflows that depended on the smart-contract enforcement, the multi-party network argument would have force and the cryptographic accountability that the architecture provides would be a meaningful property. If the platform were operated as part of an EU-coordinated cross-border registry with member-state nodes and shared governance, the federated blockchain scenario would offer a credible architectural option. If the platform were facing actively adversarial parties who could not be trusted to operate a conventional database in good faith, the immutability and the consensus-based ordering of the chain would be the most direct way to deny those parties the ability to silently rewrite the history. None of these scenarios is reached by the FREE4LIB project as scoped, but all of them are scenarios that a follow-on initiative could legitimately pursue, and the platform documented in this deliverable provides a starting point that already includes most of the components such an initiative would need.

The verdict, then, has two parts. The first part is that, for the closed scope in which the platform was actually deployed, blockchain was not the technology that the problem demanded. A conventional architecture with cryptographic audit trails would have produced the same outcome with less complexity, less operational cost and less user-facing friction. Anyone considering the same architectural path for a similar closed-scope battery passport project should think very carefully about whether the additional complexity is justified by the specific properties of their context, and should be honest about the fact that the marketing appeal of blockchain is not a substitute for an engineering justification. The second part is that, for the broader scope in which a battery passport could plausibly be used — multi-party networks of independent

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

organisations, EU-coordinated cross-border registries, adversarial verification between parties with conflicting interests — blockchain has a defensible role, and the platform developed in T2.4 is a credible technical foundation for any future work that wants to explore that broader scope. The mitigations developed across the three iterations are not wasted effort: they are exactly the components that any such future work would need, and the lessons learned are exactly the lessons that future work would need to internalise to avoid repeating them.

The deliverable closes with this dual verdict because it is the most useful thing that T2.4 can leave behind. A platform that worked perfectly in its closed scope and hid its limitations would be a less valuable contribution to the FREE4LIB consortium and to the wider community than a platform that worked, that documented its limitations honestly and that provided an evidence-based verdict on the technology it was built on. The first kind of contribution is a development output. The second kind is a research output. T2.4 was always intended to be the second kind, and this chapter is what makes that intention concrete.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

7. Conclusions and Final Remarks

This chapter closes the deliverable and, with it, the documentation of Task T2.4. It synthesises what has been done across the three iterations of the platform, makes explicit what was delivered and what was not, and proposes a set of recommendations for any follow-on work that may build on the foundation laid by FREE4LIB. It is intentionally short. The substantive content of the deliverable is in the previous chapters, and there is no value in restating it here.

7.1 Synthesis of the Three Versions of T2.4

Task T2.4 was set up at the start of the FREE4LIB project as a three-step development effort with a clear research orientation. Version I, documented in D2.8, established the foundational architecture: a Solidity smart contract on an EVM-compatible network, an API, a minimal user interface, QR-code-based public access. The second prototype documented in the same deliverable replaced the monolithic data model with a hierarchical pack-module-cell structure, split the smart contract into three contracts, migrated the backend to NestJS and replaced the original interface with a React Flow-based visual frontend. By the end of M24, the platform was a research prototype that had validated the basic technical premises of the task without yet being usable outside the development environment.

Version II, documented in D2.9, was the iteration that turned the prototype into something close to a usable product. It introduced custom meta-transactions to remove the requirement for users to hold cryptocurrency, redesigned the frontend around role-based dashboards, added a hybrid backend with MongoDB and a blockchain oracle, and deployed the platform on Alastria as a stand-in for a hypothetical EU network. By the end of M30, the platform had a complete architecture covering all the major components of a Battery Passport Platform, but it was not yet integrable by external systems, not yet aligned with the EU Battery Regulation in its data model, not yet deployed in a continuously operating environment, and not yet the subject of an honest assessment of the technology it was built on.

Version III, documented in this deliverable, is the iteration that closes those gaps. It adds the Key Management Agent as the system-to-system integration path, refines the data model to align with the categories of the EU Battery Regulation, hardens the smart contracts and the backend for production deployment, and brings the platform online as a publicly accessible service at

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

https://free4lib.eurecatsecurity.org/. More importantly, it produces the conditions under which an honest assessment of the architecture can be made — and Chapter 6 of this deliverable carries out that assessment without flinching from the conclusions.

Across the three iterations, T2.4 has produced a platform, a methodology, a deployment and an evidence-based answer to a research question. The platform is documented in this deliverable and in its predecessors. The methodology is the iterative cycle of building, deploying, evaluating and revising that took the platform from a controlled-environment prototype to a continuously deployed production service. The deployment is the live service that any consortium partner or external stakeholder can interact with. The answer to the research question is the verdict in Section 6.7. Together, these four outputs are the contribution that T2.4 leaves behind.

The decision to make the platform publicly accessible, rather than restrict it to the consortium, deserves a brief comment. A Battery Passport Platform is by nature an instrument for transparency: a system whose value comes from the fact that the data it carries is accessible to the parties that need it. Restricting the deployment to a closed environment would have made it impossible to test this property in any meaningful way and would have made the validation in Chapter 5 less honest. The decision to deploy publicly accepts the operational cost of running an internet-facing service in exchange for the ability to demonstrate the platform's behaviour in conditions that reflect the way it would actually be used.

7.2 What Was Delivered and What Was Not

It is more useful, in a final deliverable, to be explicit about both sides of the ledger than to dwell on either one alone. The following is the honest accounting of T2.4 against the description in the grant agreement and against the expectations set by D2.8 and D2.9.

7.2.1 What Was Delivered

- **A working blockchain-anchored Battery Passport Platform** with a hierarchical data model, role-based access control, meta-transaction-based user interaction, machine-to-machine integration through agents, and a public access path through QR codes.
- **A continuously deployed production environment** at https://free4lib.eurecatsecurity.org/, with a comprehensive synthetic

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

dataset that exercises every code path of the platform and demo credentials that allow any visitor to explore each role.

- **A completed analytical assessment of battery data (ST2.4.3, led by AVL).** A consolidated set of State-of-Health health parameters, an acceptance-criteria framework with mandatory and supporting checks, and a working implementation inside the live platform, validated against a real AVL dataset of eleven packs (Chapter 4 and Section 5.6).
- **A refined data model** for cells, modules and packs that aligns with the categories of the EU Battery Regulation and supports critical raw materials, hazardous substances, recycled content and carbon footprint indicators.
- **A hardened smart contract codebase** with EIP-712 typed-data meta-transactions, expanded lifecycle state machines including the SecondLife state, and a security tooling pipeline based on Slither, Mythril and property-based testing.
- **A Key Management Agent** that allows organisations to integrate the platform into their own systems while retaining custody of their cryptographic material, with a documented architecture, security model and deployment procedure.
- **An evidence-based critical assessment** of the role of blockchain technology in a battery passport context, drawing on the experience of the three iterations and on the specific behaviour of the deployed platform, and producing a verdict that is honest about both the limits of the architecture in the project's scope and the conditions under which the architecture would be more justified.

7.2.2 What Was Not Delivered

- **A validation against continuous industrial field data.** The platform and the AVL assessment were validated live against a real, purpose-built dataset of eleven packs defined by AVL (Section 5.6), which closes the real-environment validation foreseen for D2.10 at the level of the analytical assessment. What was not achieved within the project's lifetime is the onboarding of continuous field telemetry from a specific industrial partner fleet: a preliminary contact with one consortium partner during the second half of the project did not progress to actual ingestion, and no other partner came forward with such a dataset within the timeline of Version III. This remains the natural next step for a real-scale deployment.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

- **A multi-organisation deployment of the platform.** The platform has the technical components needed to be used by multiple independent organisations writing data through their own agents, but it has not been actually used that way during the project. A multi-organisation deployment would have been the most direct way to test the multi-party network argument that is most often cited in favour of blockchain in supply-chain contexts, and the absence of such a deployment is one of the reasons why the assessment in Chapter 6 is more tentative on the multi-party scenario than on the single-authority scenario.
- **An external integration with a partner system.** The agent has been validated through a development-environment deployment against the same backend that serves the production deployment, but it has not been integrated with any partner's actual production systems. The technical readiness for such an integration has been demonstrated, but the operational and organisational readiness on the partner side was not within the scope of the project's remaining timeline.

7.2.3 Why the Gaps Exist

The gaps identified above are not the result of a failure of execution; they are the result of constraints that were inherent to the scope and the schedule of the project. The platform was always going to be developed and deployed by Eurecat, with the participation of AVL, UNIGRAZ, CARTIF, ERION and IREC-CERCA in supporting roles, and the resources allocated to T2.4 within the broader FREE4LIB project were appropriate for building, deploying and assessing the platform but not for orchestrating a multi-organisation production rollout. Real-world data integration with industrial partners is a project in itself, with its own coordination overhead and its own dependencies on the partners' priorities, and it was not realistic to expect such an integration to happen as a side effect of T2.4. The honest position is to state these constraints openly and to make sure that the deliverable is useful to a future project that has the resources and the mandate to address them.

7.3 Recommendations for Follow-On Work

The following recommendations are addressed to any future project, EU initiative or industrial actor considering work in the same area. They are based on the experience of T2.4 and on the assessment in Chapter 6, and they are intended to be actionable rather than aspirational.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

- **Be honest about the trust model from the start.** The most important question for any battery passport platform is who is supposed to trust whom and why. If the answer is that all the parties involved trust the same operator, the technology choice should reflect that and should not introduce decentralisation properties that the trust model does not require. If the answer is that the parties are independent and possibly adversarial, the technology choice should provide the cryptographic accountability and the cross-party coordination that the trust model demands, and the additional complexity should be accepted as the price of those properties. The mistake is to choose a technology before answering the question.
- **Consider cryptographic audit trails as a serious alternative to blockchain.** For any context in which a single trusted operator is part of the picture, the integrity properties that blockchain provides can be obtained more cheaply through hash-chained event logs, signed log structures or transparency-log architectures. These approaches are mature, well understood and avoid most of the operational complexity of running a blockchain. They should be the default starting point, with blockchain considered only when the trust model genuinely requires the additional properties.
- **If blockchain is chosen, use it to support multi-party scenarios from the beginning.** The value of blockchain in a battery passport context comes almost entirely from the multi-party network property. A platform that uses blockchain but is operated by a single entity captures the cost without capturing the benefit. Any project that chooses blockchain should design from the beginning for multiple independent organisations writing data through their own keys, and should validate the architecture against this scenario rather than against a single-operator scenario. The Key Management Agent developed in Version III is the kind of component that such a project would need from day one.
- **Build the integration tools before building the integration cases.** The biggest practical obstacle to multi-organisation deployment is the integration cost on the participating organisations. A project that wants to build a platform for multiple organisations should invest most of its effort in the components that lower this integration cost — agents, simplified APIs, well-documented integration guides, sandbox environments — before assuming that organisations will be willing to integrate. The agent introduced in Version III is a starting point but it is not enough on its own; a follow-on project would need to wrap it in

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

deployment automation, identity provisioning workflows and monitoring integration in order to make it operationally adoptable.

- **Engage with the EU Battery Regulation as a moving target.** The data model of any battery passport platform has to align with the EU Battery Regulation, and the regulation is itself evolving through delegated acts and implementing decisions. A platform that aligns with the regulation at one point in time will need to be revised as the regulation matures. A follow-on project should plan for this evolution and should keep the data model versioned and extensible, in the way Version III does through the explicit schemaVersion field and the dual identifier strategy.
- **Plan for the analytical use of the data, not just for its storage.** The most valuable thing about a battery passport platform is not that it stores data but that it makes the data usable for decisions — second-life applicability, lifecycle optimization, recycling timing, due diligence on critical raw materials. The collaboration between Eurecat and AVL during the second half of T2.4 has shown that aligning the data model with the analytical needs is essential for the platform to be more than a passive storage system. A follow-on project should treat this alignment as a first-class design concern from the beginning, and should engage the analytical partners early enough to influence the data model rather than having to adapt to it after the fact.
- **Treat real-data validation as a project in itself.** Real industrial data is messy, partial and tied to the priorities of the organisation that produces it. Onboarding it into a new platform is not a side effect of building the platform; it is a substantial activity with its own coordination, mapping, ingestion and validation effort. Any project that intends to validate a battery passport platform against real data should allocate explicit resources to this activity, not assume that it will happen alongside the development. The experience of T2.4 with real-data validation, documented in Section 5.6, is a reminder of how easy it is to underestimate this effort.

7.4 Closing Remarks

The Battery Passport Platform documented in this deliverable is the final output of four years of work by Eurecat and the partners of T2.4. It is a working platform, deployed in a public environment, with a refined data model aligned with the EU Battery Regulation, with a system-to-system integration path through agents, with a hardened backend and a security-tested smart contract

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890

codebase. It is also the substrate of an honest assessment of blockchain technology in a battery passport context, and the assessment has the courage to say that, in the closed scope of the project, the technology did not pay for itself.

This combination — a working platform and a critical assessment of the platform's own foundations — is, to the authors' belief, the most useful contribution that T2.4 can make to the FREE4LIB consortium and to the wider community. A research task is not a marketing exercise; it is supposed to produce knowledge, and the most valuable knowledge is the kind that contradicts the original premise of the work. The premise of T2.4 was that blockchain technology would be a natural fit for a Battery Passport Platform. The evidence collected over three iterations does not support this premise in the closed scope of the project, but it does support it under conditions that the project did not reach. Both findings are useful, both are documented, and both are offered to any reader who wants to make their own choices about the same set of trade-offs.

The platform will remain operational at https://free4lib.eurecatsecurity.org/ for as long as Eurecat continues to operate it, and in any case for at least one year beyond the end of the project — that is, until at least 31 August 2027. The codebase, the documentation, the architecture description and the data model are available within the project repository for any consortium partner who wants to build on them. The agent is designed to be deployable inside any organisation's infrastructure with minimal effort, and the deployment instructions are part of the same documentation. The hope of the authors of this deliverable is that some of these components will outlive the project and find their way into a follow-on initiative, an EU registry or an industrial deployment that operates under conditions where the assessment in Chapter 6 would deliver a different verdict. If that happens, the work documented here will have served its purpose. If it does not, the deliverable will have at least left behind an honest record of what was tried, what was learned and what should be done differently next time.

Funded by the European Union

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from Horizon Europe research and innovation programme under Grant Agreement No.1069890