

# DECENTRALIZED ASSET TRACKING AND MAINTENANCE FOR RAILWAY SYSTEMS USING BLOCKCHAIN TECHNOLOGY

## Executive Summary of Master's Thesis

**Candidate:** Amirhossein Shekari  
**Student ID:** S309094  
**Supervisors:** Prof. Valentina Gatteschi, Prof. Marco Domaneschi, Prof. Valentina Villa  
**Academic Year:** 2024-2025  
**Institution:** Politecnico di Torino

---

## 1. MOTIVATION AND PROBLEM STATEMENT

The railway industry is a safety-critical sector that relies on the management of complex, hierarchical physical assets (trains, wagons, and component equipment). Traditional asset management systems in this domain currently face three fundamental challenges:

1. **Data Silos & Fragmentation:** Stakeholders (operators, maintenance crews, manufacturers, and regulators) maintain isolated, centralized databases. There is no single, interoperable source of truth regarding an asset's lifecycle.
2. **Lack of Trust and Integrity:** Centralized records can be altered, lost, or falsified. An operator cannot cryptographically verify if a maintenance report provided by a third-party crew is authentic, leading to costly redundant audits.
3. **Inefficiency:** Reliance on manual data entry and paper-based logs introduces latency and human error, preventing the industry from moving toward real-time Predictive Maintenance (PdM).

---

## 2. PROPOSED SOLUTION: A 3-TIER HYBRID ARCHITECTURE

This thesis proposes and implements a **Hybrid Blockchain System**. The core innovation is the intelligent separation of data: **Critical Asset Data** is stored on an immutable blockchain to ensure trust, while **User/Application Data** is stored off-chain to ensure performance and privacy.

The system is implemented via a 3-tier architecture:

### Tier 1: The Data Layer (Blockchain Core)

- **Technology:** Solidity Smart Contract deployed on the **Sepolia Testnet**.
- **Function:** Acts as the immutable ledger for asset identity and maintenance history.
- **Data Structure:** Implements a hierarchical relational model (Train → Wagon → Equipment) using optimized mapping strategies.
- **Key Innovation (Gas Optimization):** The contract utilizes **struct packing** (using uint32 instead of uint256 for IDs/timestamps) and **custom errors** to minimize storage costs.

## Tier 2: The Logic Layer (Intelligent Middleware)

- **Technology:** Python **FastAPI**.
- **Function:** Acts as the secure bridge between the user and the blockchain.
- **Hybrid Logic:**
  - **Off-Chain:** Manages user authentication (JWT) and sessions using a local **SQLite** database to ensure GDPR compliance and zero-latency logins.
  - **On-Chain:** Manages private keys and orchestrates transaction signing/submission to the Ethereum network via **Alchemy**.
- **Validation:** Uses **Pydantic** schemas to pre-validate data, preventing invalid transactions from wasting gas.

## Tier 3: The Presentation Layer (User Interface)

- **Technology:** **React** (TypeScript) with TailwindCSS.
- **Function:** A mobile-first interface for maintenance staff.
- **Key Innovation (PWA):** The frontend is built as a **Progressive Web App**. It utilizes Service Workers to cache the application shell, allowing mechanics to view data and draft reports even in **offline environments** (e.g., railway tunnels) where internet connectivity is absent.

## 3. EXPERIMENTAL EVALUATION AND RESULTS

To validate the feasibility of the system, a rigorous experimental evaluation was conducted using a production-grade environment (Hetzner VPS, Dockerized deployment).

### A. Smart Contract Efficiency (Gas Costs)

A comparative analysis was performed between a standard “Naive” implementation and the thesis’s “Optimized” contract.

- **Result:** The optimization strategies (struct packing, custom errors) resulted in a **15% reduction in gas costs** for adding equipment and a **7% reduction** for adding wagons.
- **Economic Implication:** On a Layer 2 network (like Optimism), the cost of recording a maintenance event is projected to be **< \$0.02**, proving the system is economically viable for large-scale fleets.

### B. System Performance (Latency)

The system’s response times were measured to validate the hybrid architecture’s trade-offs.

| Operation Type | Action             | Latency (approx.) | Implication                        |
|----------------|--------------------|-------------------|------------------------------------|
| Off-Chain Read | User Login         | ~ <b>727 ms</b>   | Secure, standard web speed.        |
| On-Chain Read  | View Asset List    | ~ <b>255 ms</b>   | Instant retrieval via API caching. |
| On-Chain Write | Update Maintenance | ~ <b>13.72 s</b>  | Slower due to block confirmation.  |

**Conclusion:** The system provides a fast “Web2-like” experience for 90% of interactions (reading/viewing). The ~14s delay is isolated only to critical write operations, which is an acceptable trade-off to gain global immutability.

### C. Usability and PWA Verification

Google Lighthouse audits confirmed the quality of the frontend engineering:

- **Accessibility: 98/100** (Ensuring usability for diverse staff).
- **PWA Capabilities:** Validated Service Worker registration and Manifest detection, confirming the app is installable and offline-capable.

## 4. CONCLUSION

---

This thesis successfully bridges the gap between theoretical blockchain frameworks and practical industrial application. By designing a **Hybrid Architecture**, the system solves the “Trilemma” of railway asset tracking:

1. **Trust** is guaranteed via the Blockchain.
2. **Privacy/Speed** is maintained via the Off-chain Middleware.
3. **Usability** is ensured via the Offline-First PWA.

The result is a working, full-stack prototype that offers a scalable blueprint for the future of secure, interoperable railway maintenance.