Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Computer Engineering (38^{th}cycle)

Blockchain Technology for the Automotive Sector

By

Alberto Butera

******

Supervisor(s):
Prof. Valentina Gatteschi, Supervisor

Doctoral Examination Committee:
Prof. Claudio Schifanella, Referee, Università di Torino
Prof. Roman Beck, Referee, Bentley University

Politecnico di Torino
2026

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Alberto Butera
2026

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

Abstract

The automotive sector is one of the most data-intensive industrial ecosystems in the global economy, and its value chain depends on information that crosses organizational, jurisdictional, and lifecycle boundaries. The current data infrastructure, organized around isolated proprietary databases under unilateral stakeholder control, leaves multiple persistent trust gaps along the vehicle lifecycle. Among them, this thesis addresses four: opacity in automotive supply-chain sustainability, absence of verifiable digital product passports for regulated vehicle components, information asymmetries in second-hand vehicle markets, and untransparent municipal governance of road infrastructure. The four gaps share a single structural cause: the absence of a decentralized, tamper-evident, and auditable data infrastructure.

This thesis investigates blockchain as a unifying trust infrastructure for the automotive ecosystem and develops four engineering contributions that address the gaps in lifecycle order. The first contribution combines blockchain with fully homomorphic encryption on Zama's fhEVM to enable supplier sustainability indices to be computed on encrypted KPIs without raw data disclosure. The second contribution is a provision-by-provision analysis demonstrating that blockchain jointly satisfies the technical requirements imposed by EU Regulation 2023/1542 on digital battery passports, covering decentralized storage, data authentication, differentiated access control, and ownership-linked data responsibility. The third contribution is a dynamic, role-gated, lockable ERC-721 vehicle NFT with per-field state-machine governance and an integrated decentralized marketplace, supporting multi-stakeholder updates from OEMs, insurers, inspectors, and repair shops. The fourth contribution is a three-contract coordination layer for municipal pothole management, combining ERC-2771 meta-transactions for gasless multi-source reporting, an ERC-20 incentive minted on validation, grid-based spatial duplicate detection, and a multi-factor priority-scoring algorithm.

Beyond the individual contributions, a cross-cutting analysis recovers three architectural patterns that recur across the four systems: hybrid on-chain/off-chain storage with content-addressed bridging, role-based access control with policy-driven asymmetry inversion, and state machines for lifecycle-bound mutable state. The analysis then positions the contributions on a single privacy–transparency spectrum spanning from encrypted-by-default computation on private data, through differentiated regulatory access, to fully public verifiability. The same architectural primitives serve opposite policy goals on this spectrum depending on whether the application protects confidentiality or provenance. Normalized cost analysis, Slither-based security assessment, and direct comparison with federated-data-space alternatives such as Catena-X situate the work within the broader landscape of European industrial digital infrastructure.

The thesis concludes that blockchain, in the architectural configurations developed here, is technically capable and economically viable for the four classes of automotive trust gap, under bounding conditions stated explicitly: trust in institutional anchors at the attestation layer, the maturity of the fhEVM infrastructure for the encrypted-computation regime, and the gap between prototype and production deployment. The architectural vocabulary recovered from the four contributions is positioned as a foundation for the broader project of building verifiable digital infrastructures for the European automotive sector.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Context

The automotive sector is one of the largest and most complex industrial ecosystems in the global economy, with annual revenues exceeding three trillion dollars and over eight million workers in the European Union alone [3]. Its value chain involves an unusually diverse set of interdependent stakeholders, raw-material suppliers, component manufacturers, OEMs, logistics providers, regulatory authorities, insurers, repair networks, vehicle owners, and municipal governments, each generating and consuming data at a different stage of the same product's lifecycle. Four converging trends are reshaping this ecosystem: electrification, which introduces new dependencies on critical-material supply chains and circular-economy obligations [4]; vehicle connectivity, which turns automobiles into networked data platforms; autonomous-driving sensor suites, which make vehicles into mobile perception systems; and new mobility models such as ride-sharing and mobility-as-a-service, which accelerate ownership turnover and raise the importance of verifiable vehicle history [3, 4]. The common consequence is that the modern vehicle is, above all, a data-intensive asset whose value, safety, compliance, and environmental footprint depend on information that crosses organizational boundaries. The data infrastructure has not kept pace. Information about vehicles, components, supply chains, driving patterns, and road conditions remains siloed in proprietary databases operated by individual stakeholders, with limited interoperability and no shared mechanism for independent verification. The result is a set of persistent trust deficits across every lifecycle stage.

1.2 Problem Statement

This section traces four specific trust gaps along the vehicle lifecycle, each of which motivates one of the contributions presented in this thesis. While they manifest at different stages and involve different stakeholders, these gaps share a common structural cause: the absence of a decentralized, tamper-proof, and auditable data infrastructure.

Gap 1 — Supply chain opacity. A vehicle's lifecycle begins in a globally distributed supply chain whose environmental and social practices are now a matter of regulatory urgency. The EU Corporate Sustainability Due Diligence Directive (CSDD), in force since 2024, requires large companies to identify, prevent, and mitigate adverse human-rights and environmental impacts across their entire supply chain [5]. Compliance is impeded by a structural tension: suppliers resist disclosing performance metrics that could reveal proprietary processes, while purchasing companies cannot evaluate them without that data. No existing solution lets suppliers demonstrate sustainability without disclosing raw KPIs, nor lets purchasers evaluate them on verifiable, tamper-proof indices. The supply chain for a single EV battery, cobalt extraction in the DRC, cathode refining in China, cell assembly in South Korea, pack integration in Germany, exemplifies the difficulty; the same tension extends to other CSDD-targeted industries such as textiles.

Gap 2 — Absence of verifiable digital product passports. EU Regulation 2023/1542 [6] requires all batteries placed on the EU market from 2027 to carry a digital battery passport (DBP) based on a decentralized data system, with cryptographic data authentication, differentiated access for the public, regulatory authorities, and parties with legitimate interest, and lifecycle persistence beyond the originating economic operator. No officially recognised implementation standard exists, and no systematic analysis has demonstrated how a specific technology satisfies all the regulation's provisions. The gap between regulatory ambition and technical implementation creates uncertainty before the 2027 deadline and risks fragmented industry solutions.

Gap 3 — Untrusted vehicle history. Once a vehicle is sold, its history (ownership, mileage, maintenance, insurance, inspections, accidents) accumulates across disconnected databases maintained by manufacturers, dealers, insurers, inspection stations, and government registries, with no unified immutable record accessible to all relevant parties. The consequences are severe: a European Parliament study estimates that up to 40% of cross-border second-hand vehicles in Europe have rolled-back odometers, costing buyers approximately €8.9 billion per year [7]. Odometer fraud is one symptom of a broader information asymmetry that buyers cannot resolve independently, whether a vehicle has been in a major accident, whether mandatory inspections were performed, whether insurance ever lapsed [8]. Existing commercial vehicle-history services depend on centralized, incomplete, and non-independently-verifiable databases, with access fees that create additional barriers [9].

Gap 4 — Opaque infrastructure governance and untapped vehicle sensing potential. Modern connected vehicles already carry the sensor suites (cameras, IMUs, GPS receivers) that autonomous-driving functions require, and these same sensors can detect road-surface defects automatically as the vehicle drives [10, 11]. The fleet thus forms a latent distributed sensing network at zero marginal hardware cost. Yet this potential is largely wasted: detected defects flow into proprietary municipal servers with no transparency in prioritization, no accountability for response times, and no mechanism to incentivize sustained vehicle or citizen participation [12]. Conventional computerized maintenance management systems (CMMS) impose enterprise licensing costs in the tens to hundreds of thousands of dollars per deployment without offering vehicles or citizens visibility into how maintenance resources are allocated.

The four gaps differ in stakeholders, data types, and regulatory context, but share a single structural feature: multiple parties must create, share, or verify data on infrastructure that is centralized, fragmented, opaque, and vulnerable to manipulation. They are therefore not isolated problems requiring four unrelated solutions but four manifestations of a single underlying deficiency: the absence of a decentralized trust infrastructure for the automotive ecosystem.

1.3 Blockchain as a Trust Infrastructure for Automotive Systems

Each of the four gaps shares a common requirement: a data infrastructure in which multiple independent parties can create, share, and verify information without relying on a central authority to guarantee its authenticity. The current approach, centralized databases under unilateral stakeholder control, fails not for technical reasons but because centralization is itself the source of the problem. Blockchain technology provides such an infrastructure. A blockchain is a distributed ledger maintained by a network of independent nodes that collectively validate transactions, producing an append-only, tamper-evident history that no single participant can alter retroactively [13, 14]. Its core properties (immutability, transparency, decentralization, and programmability through smart contracts [15]) map directly onto the requirements above. Recent developments, including Layer-2 scaling solutions [16] and fully homomorphic encryption on-chain [17, 18], have extended the design space to applications previously infeasible on cost or privacy grounds. Chapter 2 covers the technical foundations in detail; this section establishes only that the infrastructure exists and that this thesis investigates its application to the four gaps above. Figure 1.1 summarises the conceptual framework: four trust gaps spanning the vehicle lifecycle, addressed by blockchain as a unified decentralized trust infrastructure.

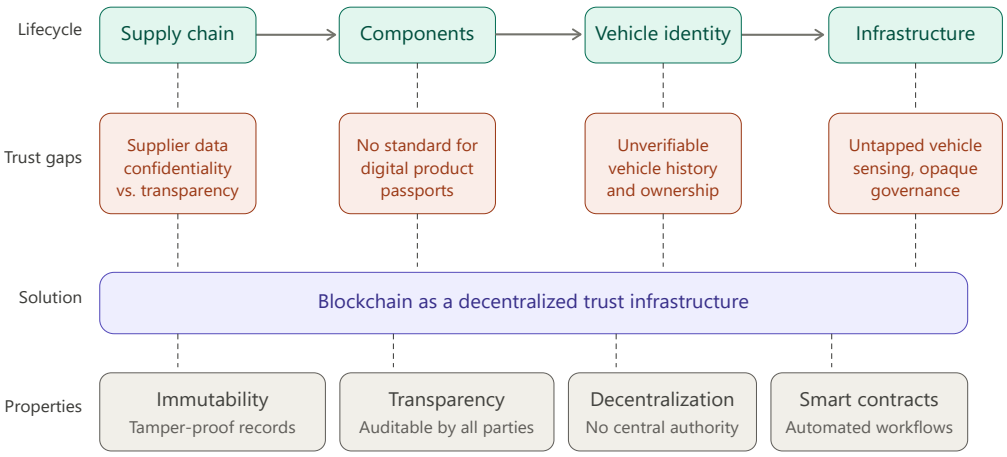

Fig. 1.1 Trust gaps across the vehicle lifecycle and the role of blockchain as a unifying decentralized trust infrastructure. Each lifecycle stage presents a specific trust deficit that this thesis addresses through a dedicated contribution.

1.4 Research Questions and Objectives

The four gaps motivate four research questions, each targeting a specific lifecycle stage. Together they investigate blockchain as a trust infrastructure across the automotive ecosystem.

RQ1: *To what extent can blockchain technology combined with fully homomorphic encryption enable verifiable supplier sustainability evaluation without compromising sensitive data?*

Practical feasibility of such an approach along three dimensions: computational and economic cost of operating on the fhEVM infrastructure, trust assumptions inherent in the threshold key management architecture, and scalability as the number of evaluation criteria increases.

RQ2: *To what extent does blockchain technology satisfy the technical requirements imposed by the EU Battery Regulation for the implementation of digital battery passports?*

The question is whether blockchain's architectural properties align systematically with the regulation's provisions on decentralized storage, data authentication, differentiated access control, and lifecycle management, and what specific design strategies are required for compliance.

RQ3: *Can an NFT-based vehicle identity system provide a complete, tamper-proof lifecycle record and support trustworthy decentralized second-hand vehicle trading at costs competitive with existing centralized alternatives?*

The question examines whether representing a vehicle as an NFT can support multi-stakeholder updates (insurance, inspection, repair) and an integrated marketplace while remaining economically competitive with commercial history and marketplace platforms.

RQ4: *Can a decentralized blockchain-based system effectively coordinate multi-source infrastructure defect reporting with transparent prioritization and token-based incentives, at costs viable for municipal deployment?*

The question examines whether smart contracts can orchestrate the full defect-management lifecycle (multi-source reporting from connected vehicles and citizens, spatial duplicate detection, algorithmic priority scoring, and tokenized reward distri-

bution upon municipal validation) at costs competitive on Layer-2 with traditional CMMS platforms.

The mapping of research questions to objectives, papers, and lifecycle stages is summarised in Table 1.1. The ordering of research questions follows the vehicle

Table 1.1 Mapping of research questions to objectives, contributions, and lifecycle stages.

RQ	Objective	Paper	Lifecycle Stage	Ch.
RQ1	Design and implement a privacy-preserving blockchain framework for supplier sustainability evaluation using homomorphic encryption	P3	Supply chain	4
RQ2	Systematically analyze the EU Battery Regulation and demonstrate blockchain's alignment with all digital passport requirements	P2	Components	5
RQ3	Design, implement, and evaluate an NFT-based vehicle lifecycle tracking and decentralized marketplace system	P1	Vehicle identity	6
RQ4	Design, implement, and evaluate a blockchain-based pothole management system with token incentives, spatial indexing, and priority scoring for connected vehicles	P4	Infrastructure	7

lifecycle: from supply chain (RQ1) through component traceability (RQ2) and vehicle identity (RQ3) to infrastructure interaction (RQ4).

1.5 Research Methodology

The thesis follows the Design Science Research (DSR) paradigm [19, 20], which is concerned with designing and evaluating technological artifacts that address identified problems through iterative cycles of problem identification, solution design, implementation, and evaluation. The paradigm fits these research questions, which span from regulatory analysis (RQ2) to full system design and implementation (RQ1,

RQ3, RQ4). Each contribution follows the same four-step workflow, adapted to its specific question:

1. **Problem identification and requirements analysis:** Each contribution analyses the specific trust gap it addresses, drawing on regulatory documents, academic literature, and, where available, industry data, producing a set of functional and non-functional requirements.

2. **System design:** A system architecture is designed, specifying on-chain and off-chain components, smart-contract structure, data flow among stakeholders, and interaction with external systems (decentralized storage, mobile applications, vehicle sensors). Architectural decisions are justified regarding the requirements.

3. **Prototype implementation:** Three of the four contributions (P1, P3, P4) deliver deployed Solidity smart contracts together with front-end DApps or mobile interfaces. The fourth (P2) produces a systematic regulatory analysis rather than a software artifact; within DSR, it serves as the requirements-analysis phase for the digital-product-passport domain. The technology stack (Solidity, OpenZeppelin libraries, IPFS, Hardhat) is described in Chapter 2.

4. **Evaluation:** Each prototype is evaluated along the dimensions appropriate to its contribution: gas-based cost analysis projected across EVM-compatible networks (P1, P3, P4); security analysis using Slither static analysis with manual review of common vulnerability patterns (P1, P3, P4); functional testing via unit and integration tests (P4); comparative analysis against existing academic and commercial alternatives (P1, P3, P4); and, for P2, regulatory compliance analysis mapping the provisions of the EU Battery Regulation onto blockchain properties.

Although the four contributions are presented in lifecycle order (supply chain in Chapter 4 through infrastructure in Chapter 7), they were developed in a different chronological order: vehicle identity (Chapter 6) was first, in 2023; battery passport analysis (Chapter 5) in 2024; supplier evaluation (Chapter 4) in 2025; and pothole management (Chapter 7) in 2026.

1.6 Contributions

The thesis presents four contributions, each addressing one trust gap in lifecycle order. Although developed independently and published in separate venues, they share a common technology stack and recurring architectural patterns analysed in Chapter 8. The mapping to research questions, papers, and chapters is given in Table 1.1.

Contribution 1 — Privacy-Preserving Blockchain Framework for Supplier Evaluation. A framework that integrates fully homomorphic encryption (via the fhEVM infrastructure) with a public Ethereum-compatible blockchain. Suppliers submit encrypted KPIs; the smart contract computes a composite sustainability index entirely on the ciphertexts; the resulting index is published on-chain. The architecture is KPI-agnostic, so an automotive instantiation evaluating battery-cell suppliers reuses the same contract logic. A Sepolia-testnet prototype shows a per-evaluation workflow cost of $39.30 and no high-severity Slither findings. Published at ACM GoodIT 2025 [21].

Contribution 2 — Blockchain Alignment with EU Battery Regulation for Digital Passports. A provision-by-provision analysis of EU Regulation 2023/1542 demonstrating that the architectural primitives exercised by the engineering contributions (decentralized storage, cryptographic data authentication, role-based differentiated access, and ownership-linked data responsibility) jointly satisfy all the regulation's technical requirements for digital battery passports. Unlike the other three contributions, this is an analytical rather than implementation contribution. Published at IEEE BRAINS 2024 [22].

Contribution 3 — NFT-Based Vehicle Lifecycle Tracking and Decentralized Marketplace. A fully decentralized system representing each vehicle as a dynamic ERC-721 NFT carrying its complete lifecycle record (manufacturing, insurance, inspection, repair). Authorised stakeholders submit updates validated by the OEM; a locking mechanism blocks ownership transfer while validation is pending; an integrated marketplace contract supports listing, bidding, fiat-gateway payment, and automatic ownership transfer. NFT minting costs approximately $5 and routine

marketplace operations stay below $10 at January 2023 mainnet conditions, with no high-severity Slither findings. Published in IEEE Access 2023 [1].

Contribution 4 — Blockchain-Based Pothole Management for Connected Vehicle Networks. A three-contract coordination layer in which connected vehicles and citizens submit road-defect reports through an ERC-2771 meta-transaction forwarder. A grid-based spatial duplicate-detection scheme reduces gas by 75 % on duplicates; a multi-factor priority-scoring algorithm computable from public state replaces opaque municipal prioritization; an ERC-20 incentive is minted on validated reports. Multi-network projections place annual operating cost on Layer-2 in the $14–$30 range: two orders of magnitude below comparable proprietary CMMS platforms. Submitted to Future Generation Computer Systems [23].

Cross-cutting contributions. Beyond the individual contributions, Chapter 8 examines the recurring architectural patterns (hybrid on-chain/off-chain storage, role-based access control, lifecycle-bound state machines, token-based incentives), the cost and scalability profile across all four systems, the privacy–transparency spectrum they jointly span, and the regulatory implications.

1.7 Thesis Outline

The remainder of the thesis is organised as follows. Chapter 2 provides the technical background: blockchain architecture, smart contracts, token standards, decentralized storage, privacy-preserving cryptography, and the automotive regulatory landscape. Chapter 3 surveys the state of the art and identifies the gaps that motivate each contribution. Chapters 4 to 7 present the four contributions in lifecycle order: supplier evaluation (Ch. 4), digital battery passports (Ch. 5), NFT-based vehicle identity (Ch. 6), and pothole management for connected vehicles (Ch. 7). Chapter 8 synthesises the architectural patterns, cost profile, privacy–transparency positioning, and regulatory implications across the four contributions. Chapter 9 answers the research questions and identifies five future research directions.

Chapter 2

Background and Enabling Technologies

This chapter provides a unified technical background covering the foundational concepts, technologies, and domain-specific standards that underpin the four contributions presented in this thesis. The purpose is twofold: first, to equip the reader with the knowledge necessary to understand the design decisions and architectural choices made throughout the subsequent chapters; and second, to avoid the repetition that would arise if each contribution chapter were to re-introduce the same foundational material independently. The topics are organized progressively, beginning with the general principles of distributed ledger technology and moving toward the specific application domain of the automotive industry. Section 2.1 introduces the fundamental properties of blockchain technology and the main categories of distributed ledgers. Section 2.2 focuses on the Ethereum platform and its smart contract execution environment, which serves as the primary infrastructure for all contributions. Section 2.3 describes the token standards employed across multiple chapters. Section 2.4 addresses the problem of data management in blockchain-based systems, presenting decentralized storage solutions and hybrid on-chain/off-chain strategies. Section 2.5 introduces the privacy-preserving cryptographic techniques that are central to Contribution 1. Section 2.6 covers smart contract security, a cross-cutting concern addressed in every implementation chapter. Finally, Section 2.7 provides the necessary context on the automotive ecosystem and the regulatory landscape, establishing the domain-specific motivation for the thesis.

2.1 Blockchain Technology

Blockchain technology, broadly understood as a class of distributed ledger technologies (DLTs), provides the foundational trust infrastructure upon which every contribution in this thesis is built. This section introduces the key concepts, architectural properties, and design trade-offs that inform the technology choices made throughout the subsequent chapters.

2.1.1 Distributed Ledger Fundamentals

A distributed ledger is a data structure replicated, shared, and synchronized across multiple nodes in a geographically distributed network [13]. Unlike traditional centralized databases, where a single entity controls data access and modification, a distributed ledger distributes both data storage and the authority to validate changes across the network participants. A blockchain is a specific type of distributed ledger in which validated transactions are grouped into blocks that are cryptographically linked to form an append-only chain. Each block contains a cryptographic hash of the preceding block, a timestamp, and a set of transactions, creating a tamper-evident data structure in which any modification to a past block would invalidate the hash chain from that point onward [24].

The key properties that make blockchain technology relevant to the applications presented in this thesis are the following. *Immutability* ensures that once data is recorded on the ledger, it cannot be silently altered or deleted without detection. This property is critical for vehicle history records (Chapter 6), supplier sustainability indices (Chapter 4), and infrastructure defect reports (Chapter 7). *Transparency* allows all network participants, or designated subsets of them, to independently verify the state of the ledger without relying on a trusted intermediary. *Decentralization* eliminates single points of control and failure, distributing trust across the network rather than concentrating it in one organization. *Non-repudiation* guarantees that once a participant has submitted a transaction, its authorship is cryptographically verifiable and cannot be denied. Finally, *programmability*, enabled through smart contracts, allows the automation of complex multi-party workflows directly on the ledger.

The data model of a blockchain can be understood at two levels. At the *transaction level*, each transaction represents an atomic operation, such as transferring a token, updating a record, or invoking a smart contract function, that is digitally signed by the initiating party using public-key cryptography. At the *block level*, a set of validated transactions is bundled into a block, which is proposed by a designated node and accepted by the network according to a consensus protocol. Each block header typically contains the hash of the previous block header, a Merkle root summarizing all transactions in the block, a timestamp, and protocol-specific metadata. The Merkle tree structure enables efficient verification of transaction inclusion without downloading the entire block content, a property known as Simplified Payment Verification (SPV) [13].

2.1.2 Consensus Mechanisms

The consensus mechanism is the protocol by which distributed nodes agree on the current state of the ledger in the absence of a central coordinator. The choice of consensus mechanism fundamentally affects a blockchain's security guarantees, throughput, latency, energy consumption, and degree of decentralization. The two most prominent families of consensus mechanisms are described below.

Proof of Work (PoW). In PoW-based systems, such as the original Bitcoin protocol [13], nodes (called miners) compete to solve a computationally intensive cryptographic puzzle. The first miner to find a valid solution earns the right to propose the next block and receives a reward. The difficulty of the puzzle is periodically adjusted to maintain a target block time. PoW provides strong security guarantees under the assumption that no single entity controls more than 50% of the network's computational power, but it comes at a substantial energy cost. Ethereum originally employed PoW but transitioned to Proof of Stake in September 2022 through an event known as "The Merge" [25].

Proof of Stake (PoS). In PoS-based systems, block proposers are selected based on the amount of cryptocurrency they have staked as collateral, rather than on computational work. Validators who behave dishonestly risk having their stake partially or fully confiscated, a mechanism known as *slashing*. PoS achieves consensus with

significantly lower energy consumption than PoW, while preserving the decentralization and security properties necessary for the applications considered in this thesis. Ethereum's PoS implementation requires validators to stake 32 ETH, and the protocol randomly selects a validator to propose each block while committees of validators attest to the validity of proposed blocks [26]. The transition to PoS reduced Ethereum's energy consumption by approximately 99.95% [25], a consideration that is particularly relevant for sustainability-oriented applications such as the supplier evaluation framework (Chapter 4) and the battery passport analysis (Chapter 5).

Other consensus mechanisms, such as Practical Byzantine Fault Tolerance (PBFT), Delegated Proof of Stake (DPoS), and Proof of Authority (PoA), are employed in permissioned or semi-permissioned blockchain networks. While these alternatives offer higher throughput and lower latency, they typically require a known and trusted set of validators, which limits their applicability in scenarios requiring open, permissionless participation.

2.1.3 Public vs. Permissioned Blockchains

Blockchain networks can be broadly classified into three categories based on their governance model and participation rules.

Public (permissionless) blockchains, such as Ethereum, are open to anyone: any participant can read the ledger, submit transactions, and participate in the consensus process without requiring authorization. Public blockchains offer the strongest guarantees of transparency and censorship resistance, but they face scalability challenges due to the overhead of global consensus. All four contributions in this thesis are designed for or evaluated on public blockchain infrastructure, because the applications they address, ranging from vehicle marketplace transparency to municipal infrastructure monitoring, require verifiability by stakeholders who may not know or trust each other a priori.

Private (permissioned) blockchains, such as Hyperledger Fabric [27], restrict participation to a predefined set of authorized nodes. Read and write permissions are controlled by a governance authority, and consensus is typically achieved through lighter-weight protocols such as PBFT or Raft. Private blockchains offer higher throughput and lower latency but sacrifice the open verifiability that characterizes public networks. Several related works reviewed in Chapter 3 employ Hyperledger

Fabric for automotive applications, but this thesis argues that the trust gaps identified in the vehicle lifecycle are best addressed by architectures that do not require trust in a consortium operator.

Hybrid (consortium) blockchains combine elements of both models, allowing controlled access for write operations while maintaining public readability. This approach is sometimes adopted in industry settings where a consortium of known entities cooperates, but the resulting data must be verifiable by external parties.

The choice of a public blockchain for the contributions in this thesis is motivated by the need for maximum transparency and verifiability. However, the cost and privacy implications of this choice are non-trivial and are discussed under each contribution.

2.2 Ethereum and Smart Contracts

Ethereum serves as the primary blockchain platform for all the implementations presented in this thesis. This section provides a technical overview of the Ethereum architecture, its execution environment, its cost model, and the ecosystem of compatible networks that are used for evaluation purposes.

2.2.1 Ethereum Virtual Machine (EVM)

The Ethereum Virtual Machine (EVM) is the runtime environment in which smart contracts are executed on the Ethereum network [26, 28]. The EVM is a stack-based, almost Turing-complete virtual machine that operates deterministically across all nodes in the network: given the same input state and the same transaction, every node produces exactly the same output state. This determinism is essential for maintaining consensus, as all validators must independently arrive at the same result when processing a block of transactions.

Smart contracts are compiled into EVM bytecode and deployed to the blockchain, where they receive a unique address and persist indefinitely unless explicitly designed to be destroyable. Once deployed, the bytecode of a smart contract is immutable: it cannot be modified or patched. This immutability provides strong integrity guarantees, since users can verify that the code governing their interactions will not change

after deployment. However, it also means that bugs in deployed contracts cannot be corrected without deploying a new contract and migrating state, which has motivated the development of upgradeable contract patterns using proxy architectures [29].

The EVM operates on a world state, which is a mapping from account addresses to account states. Two types of accounts exist: *externally owned accounts* (EOAs), controlled by private keys and used by human users to initiate transactions, and *contract accounts*, which contain executable code and are triggered by incoming transactions or messages from other contracts. Each contract account maintains its own persistent storage, a key-value store where state variables are held between transactions.

2.2.2 Solidity and Smart Contract Development

Solidity is the dominant high-level programming language for writing Ethereum smart contracts [30]. It is a statically typed, contract-oriented language with syntax influenced by JavaScript, Python, and C++. Solidity supports inheritance, libraries, user-defined types, and a rich set of data structures including mappings, dynamic arrays, and structs. The language compiles to EVM bytecode through the Solidity compiler (`solc`), which also generates the Application Binary Interface (ABI) used by external applications to interact with deployed contracts.

All smart contracts implemented in this thesis are written in Solidity and leverage the OpenZeppelin library [29], a widely audited collection of reusable contract components that implements standard interfaces such as ERC-20 and ERC-721, access control mechanisms (role-based access control via `AccessControl`, ownership via `Ownable`), and security utilities (reentrancy guards, safe math operations). The use of battle-tested library components significantly reduces the attack surface of custom implementations.

The development, testing, and deployment workflow across all contributions follows a consistent toolchain. The Hardhat framework [31] (used in Contributions 1, 2, and 4) provide local blockchain instances for development, automated testing frameworks, and deployment scripts for testnet and mainnet deployment. Unit tests are written in JavaScript or TypeScript and verify the correctness of individual functions, while integration tests validate multi-step workflows involving multiple stakeholder roles.

2.2.3 Gas Model and Cost Optimization

Every computational step and storage operation performed by the EVM consumes a resource called *gas*. The gas model serves two purposes: it prevents infinite loops and resource exhaustion attacks by bounding computation, and it creates an economic incentive for validators to process transactions. The total cost of a transaction is determined by the amount of gas consumed multiplied by the gas price, which is denominated in gwei (1 gwei = 10^{-9} ETH).

Prior to EIP-1559, transactions (retrospectively Type 0, or legacy) carried a single `gasPrice` field and were included through a first-price auction: users bid, validators selected the highest bidders, and the entire fee accrued to the validator. This mechanism has two well-documented pathologies. First, it is not incentive-compatible, a user has no way to determine the minimum bid that will secure inclusion, and the rational strategy is systematic overbidding, so users routinely paid above the clearing price. Second, fee estimation was unstable: wallets could only extrapolate from recent blocks, and the estimate could be invalidated within a single block.

EIP-1559 [32] introduced Type 2 transactions, which decompose the fee into a protocol-determined base fee, burned rather than paid to the validator, and a user-specified priority fee paid to the validator as an inclusion tip. The user supplies two parameters: `maxPriorityFeePerGas` and `maxFeePerGas`, the latter being an absolute ceiling on the per-gas price the user is willing to pay. The base fee adjusts deterministically from the parent block, rising or falling by at most 12.5% per block as a function of the deviation of the parent block's gas usage from the 15M-gas target, and the difference between `maxFeePerGas` and the realised price is refunded.

The advantages relevant to this thesis are three. Fee predictability: because the base fee of block n+1 is a deterministic function of block n, it is known in advance within a ±12.5% band, which makes ex-ante cost estimation tractable, the cost tables of Chapters 4, 6 and 7 are meaningful precisely because of this property. Overpayment protection: the refund mechanism removes the systematic overbidding of the first-price auction, so a user may set a generous `maxFeePerGas` without expecting to pay it. Bounded sponsor exposure: for the meta-transaction architecture of Chapter 7, in which a relayer pays gas on behalf of citizen reporters, `maxFeePerGas` gives the sponsoring municipality a hard per-transaction cost ceiling,

which is a precondition for budgeting a public-sector deployment. Under the legacy model, a sponsor's exposure during a congestion event would have been unbounded.

The disadvantages are equally concrete and are not always acknowledged. EIP-1559 does not reduce the average cost of transacting; it redistributes it, burning the base fee rather than transferring it to validators, and the long-run price level remains set by demand for blockspace. It smooths within-block variance but not sustained congestion: under persistent demand the base fee compounds at up to 12.5% per block and can rise by an order of magnitude within a few minutes, so the tail of the cost distribution remains heavy. The priority-fee signal has also degraded in informational value since 2021, as a substantial fraction of transaction ordering has migrated to off-protocol block-building and private order flow, meaning the observed priority fee no longer reflects the whole inclusion market. Finally, on Layer-2 rollups (the deployment target that Chapters 6 and 7 conclude is economically viable) the user-facing fee is dominated not by the L2's own EIP-1559 market but by the cost of posting data to L1, which since EIP-4844 is governed by a separate blob fee market with its own independent controller. The three-parameter Type 2 model is therefore a necessary but not sufficient account of what an L2 user actually pays.

All fiat costs reported in this thesis are computed under Type 2 semantics, as gasUsed × (baseFee + priorityFee), at the reference conditions stated in each table.

The gas costs of individual EVM opcodes are specified in the Ethereum Yellow Paper [28]. Storage operations are by far the most expensive: writing a 256-bit word to contract storage for the first time costs 20,000 gas (`SSTORE` from zero to non-zero), while subsequent modifications cost 5,000 gas. By contrast, computational operations such as addition (`ADD`, 3 gas) and multiplication (`MUL`, 5 gas) are comparatively inexpensive. This cost structure has direct implications for the design of all systems presented in this thesis: minimizing on-chain storage is a primary optimization objective.

Several gas optimization strategies are employed across the contributions:

- **Hybrid on-chain/off-chain storage:** Only essential metadata is stored on-chain, while large data objects (images, detailed documents, comprehensive metadata) are stored on IPFS with their content identifiers (CIDs) recorded on the blockchain. This pattern is used in all contributions that handle large datasets (Chapters 6, and 7).

- **Event-based logging:** Non-critical historical data is emitted as events rather than stored in contract state variables. Events are recorded in transaction logs, which are significantly cheaper than contract storage, and can be efficiently queried by off-chain applications. The pothole management system (Chapter 7) extensively uses this pattern to reconstruct system history from event logs.

- **Data packing:** Contract storage is addressed in 32-byte slots, and the compiler assigns consecutive state variables to the same slot when their declared widths permit. Because the dominant costs are per-slot (20,000 gas for a cold write to a zero slot, 2,100 for a cold read) narrowing fields to the smallest sufficient type and declaring them adjacently reduces cost directly. In the vehicle record of Chapter 6, declaring the owner address (20 bytes), a uint40 timestamp (5 bytes), a uint8 state enumeration and a bool lock flag consecutively fits 27 bytes into one slot, where the unnarrowed declaration would occupy four; the saving is roughly 60,000 gas on first write. The technique pays only when the co-located fields are written in the same transaction, since otherwise each write becomes a read-modify-write of the shared slot. It does not transfer to the encrypted regime of Chapter 4, where every euint32 is a full 32-byte ciphertext handle regardless of the magnitude it encodes.

- **Short-circuiting and early reverts:** Input validation is performed at the beginning of functions using `require` statements, so that invalid transactions fail early and consume minimal gas before reverting.

Throughout this thesis, transaction cost is reported in three units of decreasing stability. Gas units are a deterministic property of the compiled bytecode and the EVM semantics, invariant across chains and across time; they are the primary figure. Native-token cost is gas units multiplied by the gas price under Type 2 semantics, and varies with network congestion but not with currency markets. Fiat cost additionally incorporates the native-token exchange rate and is therefore the least durable of the three; it is reported because it is what a prospective adopter must budget, but every fiat figure is stamped with the date and rate at which it was computed. Because the four contributions were measured at different times, over an interval in which both gas prices and the ETH/USD rate moved substantially, the per-chapter fiat figures are not mutually comparable as reported; Section 8.3 re-prices them onto a single set of reference parameters for cross-contribution comparison, and that normalised table, not the per-chapter ones, is the basis for any claim about relative cost.

2.2.4 EVM-Compatible Chains and Layer-2 Solutions

While Ethereum provides the strongest security and decentralization guarantees, its transaction costs can be prohibitive for applications requiring frequent interactions. This has motivated the development of EVM-compatible networks that offer lower costs while maintaining compatibility with the Ethereum toolchain.

Layer-2 (L2) scaling solutions are protocols that execute transactions off the main Ethereum chain (Layer 1) while periodically anchoring their state to it, thereby inheriting its security guarantees. The two primary L2 architectures are:

- **Optimistic rollups** (e.g., Optimism, Arbitrum) execute transactions off-chain and post compressed transaction data to Ethereum L1. They assume transactions are valid by default and rely on a challenge period during which any observer can submit a fraud proof to dispute an invalid state transition. Optimistic rollups can reduce transaction costs by 10–100$\times$ compared to L1 [16].
- **Zero-knowledge (zk) rollups** (e.g., zkSync, StarkNet, Polygon zkEVM) also execute transactions off-chain but generate cryptographic validity proofs (typically zk-SNARKs or zk-STARKs) that are verified on L1. This eliminates the need for a challenge period, enabling faster finality, but imposes higher computational overhead for proof generation.

Several contributions in this thesis (Chapters 4, 6, and 7) project transaction costs across multiple EVM-compatible networks, including Polygon, Arbitrum, Optimism, and Base, to assess the economic viability of deployment on L2 platforms. The EVM compatibility of the Solidity smart contracts ensures that the same codebase can be deployed on any of these networks without modification, a significant practical advantage for future real-world adoption.

2.3 Token Standards and NFTs

Token standards define standardized interfaces for creating and managing digital assets on the Ethereum blockchain. These standards ensure interoperability between contracts, wallets, and decentralized applications. This thesis employs two primary

token standards, ERC-20 and ERC-721, and references additional standards that are relevant to the evolution of the proposed systems.

2.3.1 ERC-20: Fungible Tokens

The ERC-20 standard [33] defines a common interface for fungible tokens, i.e., tokens that are mutually interchangeable and divisible. Each unit of an ERC-20 token is identical in value and functionality to every other unit, analogous to units of a currency. The standard specifies a minimal set of functions, including `transfer` (to send tokens from one account to another), `approve` and `transferFrom` (to delegate spending authority to another account or contract), and `balanceOf` (to query the token balance of an account). Events such as `Transfer` and `Approval` are emitted to enable off-chain applications to track token movements. In this thesis, ERC-20 tokens is used in the pothole management system (Chapter 7) as a programmable incentive mechanism: a custom token (PBC, PotholesCoin) is minted upon municipal validation of defect reports and distributed to reporters as compensation for their civic contributions. The use of a standardized token interface ensures that the reward tokens are compatible with existing wallets and can, in principle, be exchanged or used in other decentralized applications.

2.3.2 ERC-721: Non-Fungible Tokens

The ERC-721 standard [34] defines an interface for non-fungible tokens (NFTs), which are unique, indivisible digital assets. Unlike ERC-20 tokens, each ERC-721 token has a distinct identifier (`tokenId`) and can carry unique metadata. The standard specifies functions for ownership tracking (`ownerOf`), safe transfer (`safeTransferFrom`), and approval management. The `tokenURI` function returns a URI pointing to a JSON metadata object that describes the token's attributes, which is typically hosted on decentralized storage.

ERC-721 tokens are central to the vehicle identity system presented in Chapter 6. In that contribution, each vehicle is represented as a unique NFT whose metadata encodes manufacturing details, ownership history, insurance records, inspection reports, and repair logs. The ownership transfer mechanism of ERC-721 naturally maps to the concept of vehicle sale: when a vehicle is sold, the corresponding NFT

is transferred to the new owner, and the entire lifecycle history remains permanently attached to the token. The metadata is stored on IPFS, with the CID referenced through the `tokenURI` function, ensuring that the vehicle's complete record is both immutable and verifiable.

2.3.3 Other Relevant Standards (ERC-1155, Soulbound Tokens, ERC-2771, , ERC-4337, EIP-7702)

Three additional token standards are relevant to the broader context of this thesis, although they are not directly implemented in the current contributions.

ERC-1155 [35] is a multi-token standard that supports both fungible and non-fungible tokens within a single contract. It enables batch operations (e.g., transferring multiple token types in a single transaction), which can significantly reduce gas costs in systems that manage diverse asset types. An ERC-1155-based architecture could be a natural evolution for a unified vehicle lifecycle system that manages both unique vehicle identities (non-fungible) and reward tokens (fungible) in a single contract.

Soulbound Tokens (SBTs) [36], proposed by Weyl, Ohlhaver, and Buterin, are non-transferable tokens that represent credentials, affiliations, or commitments. Unlike standard ERC-721 tokens, SBTs cannot be sold or transferred, making them suitable for representing certifications, professional qualifications, or verified attributes. In the automotive context, SBTs could represent vehicle inspection certificates, driving licenses, or repair shop accreditations that should be bound to a specific entity and not transferable.

ERC-2771 [37], the *Secure Protocol for Native Meta Transactions*, defines the interface by which a trusted forwarder contract can submit a transaction on behalf of a user who has signed the call payload off-chain. The user's address is appended to the call data and extracted by the target contract, which verifies the forwarder is trusted before treating the call as authentic. This pattern enables gasless onboarding: a user with no ETH holdings can interact with on-chain applications because a third party (the forwarder operator) pays the gas. The pothole management system of Chapter 7 employs ERC-2771 as a constitutive primitive rather than a convenience: gasless reporting is what makes the system reachable by the citizen population whose participation determines its value (Section 7.2.1).

ERC-4337, achieves account abstraction entirely at the application layer, without consensus changes. Users hold smart-contract accounts rather than externally owned accounts (EOAs); they express intent as a UserOperation submitted to a separate mempool; bundlers aggregate these into ordinary transactions routed through a singleton EntryPoint contract; and a paymaster contract may agree to settle the gas, enabling sponsorship. Because validation logic is contract code, ERC-4337 also supports session keys, batched operations, arbitrary signature schemes, and social recovery (none of which ERC-2771 provides). Its costs are a per-operation overhead of roughly 40,000-100,000 gas relative to a direct call, an account-deployment cost on first use, and a dependency on bundler and paymaster infrastructure that is not part of consensus.

EIP-7702, activated on Ethereum mainnet with the Pectra upgrade in May 2025, introduces a set-code transaction type by which an EOA installs a delegation pointer to contract code, acquiring smart-account behaviour at its existing address. This removes the principal migration barrier of ERC-4337, the need for users to abandon an address they already hold and fund, and permits a sponsor to pay for execution authorised by an EOA signature.

Positioning of ERC-2771 in this thesis. The pothole-management system of Chapter 7 uses ERC-2771 rather than either alternative, for reasons that are partly historical and partly architectural, and it is worth separating them.

The historical reason is that the system was designed and implemented before EIP-7702 was available on mainnet, at a point when ERC-4337 bundler infrastructure was maturing but not yet routine.

The architectural reasons remain valid. First, ERC-2771 imposes no requirement on the user's wallet: a citizen reporter signs an EIP-712 typed-data payload, which every mainstream wallet can produce, whereas ERC-4337 requires the client to speak the UserOperation format. For a system whose central design bet is that onboarding friction determines participation, this is not a minor consideration. Second, the trusted-forwarder pattern concentrates the trust delegation in a single, small, auditable contract, which maps cleanly onto the deployment model in which a municipality operates the relayer; ERC-4337 distributes the equivalent trust across an EntryPoint, a bundler market, and a paymaster. Third, the on-chain surface is smaller, which matters for a system whose security argument rests on static analysis.

The costs of the choice should equally be stated. ERC-2771 requires the target contract to be forwarder-aware, overriding _msgSender() and _msgData(), which couples the application contract to the sponsorship mechanism; it offers no batching, no session keys, and no signature-scheme flexibility; and it is being superseded in practice. Migrating the Chapter 7 contracts to an EIP-7702 sponsorship model would remove the ERC2771Context inheritance and restore msg.sender semantics: a small change to the contracts, and a larger one to the relayer infrastructure.

2.4 Decentralized Storage

The tension between the need for comprehensive data storage and the prohibitive cost of on-chain storage is a recurring design challenge across all contributions in this thesis. This section describes the decentralized storage solutions employed and the hybrid strategies adopted to balance data integrity, availability, and cost.

2.4.1 IPFS and Content-Addressable Storage

The InterPlanetary File System (IPFS) is a peer-to-peer distributed file system that uses content-addressable storage [38]. In contrast to location-addressed systems (e.g., HTTP URLs), where content is identified by its location on a specific server, IPFS identifies content by its cryptographic hash. When a file is added to IPFS, it is split into chunks, each chunk is hashed using a cryptographic hash function, and the chunks are organized into a Merkle Directed Acyclic Graph (DAG). The root hash of this DAG, called the Content Identifier (CID), serves as the globally unique, self-certifying address of the file. This architecture provides two important properties: *content integrity*, since any modification to the file would change its CID, enabling immediate detection of tampering; and *deduplication*, since identical content stored by different users maps to the same CID.

IPFS does not inherently guarantee data persistence: content is available only as long as at least one node in the network stores and serves it. In practice, data persistence is ensured through *pinning services* (e.g., Pinata, Infura, nft.storage) that commit to hosting specific content indefinitely. All contributions in this thesis that store data on IPFS rely on pinning services for persistence, and the associated availability risks are discussed as limitations in the respective chapters.

2.4.2 On-Chain vs. Off-Chain Data Strategies

The design of blockchain-based systems requires careful decisions about which data to store on-chain (in smart contract storage or event logs) and which to store off-chain (on IPFS, in traditional databases, or in other external systems). This trade-off is governed by several factors: cost (on-chain storage is orders of magnitude more expensive), privacy (on-chain data on public blockchains is visible to all participants), data size (smart contract storage is optimized for small values, not large files), and verifiability (on-chain data benefits from the blockchain's immutability and consensus guarantees).

The contributions in this thesis adopt a consistent hybrid strategy that can be summarized as follows. Critical operational metadata, including asset identifiers, ownership records, status flags, access control lists, and cryptographic hashes of off-chain content, is stored on-chain. This ensures that the integrity and provenance of all data can be independently verified by any network participant. Large data objects, such as vehicle images, detailed metadata documents, geographic context information, and evidence photographs, are stored on IPFS, with their CIDs recorded on-chain. This approach maintains end-to-end verifiability (any observer can retrieve the off-chain data, hash it, and confirm that the hash matches the on-chain record) while keeping transaction costs manageable.

Table 2.1 summarizes the on-chain and off-chain data partitioning across the four contributions.

The table reveals that Contribution 1 is unique in storing all data on-chain, albeit in encrypted form, because the fhEVM infrastructure requires ciphertexts to reside in contract storage for homomorphic computation. Contributions 3 and 4 follow the IPFS-based hybrid pattern, with content-addressed CIDs anchoring image and metadata payloads to on-chain records.

2.4.3 The Oracle Problem and Real-World Data Ingestion

A fundamental challenge in blockchain-based systems that interact with the physical world is the *oracle problem*: blockchains are deterministic, isolated execution environments that cannot natively access external data sources. Any real-world information, such as sensor readings, market prices, regulatory documents, or phys-

Table 2.1 Summary of on-chain vs. off-chain data allocation across thesis contributions.

Contribution	**On-Chain Data**	**Off-Chain Data**
C1 Supplier Evaluation	Encrypted KPIs, sustainability indices, supplier addresses, access control	— (all data on-chain via fhEVM)
C2 Battery Passport	(Regulatory analysis)	(Regulatory analysis)
C3 Vehicle NFTs	Token IDs, ownership records, metadata CIDs, marketplace listings	Vehicle images, detailed metadata, inspection/repair documents (IPFS)
C4 Pothole Mgmt.	Report IDs, coordinates, CIDs, status, duplicate counts, token balances	Pothole images, geographic context metadata (IPFS)

ical asset attributes, must be introduced into the blockchain through an oracle, a mechanism that bridges the off-chain world with on-chain smart contracts [39].

Oracles can be classified along several dimensions. *Centralized oracles* rely on a single trusted entity to provide data, which introduces a single point of failure that undermines the decentralization guarantees of the underlying blockchain.*Decentralized oracle networks*, such as Chainlink, aggregate data from multiple independent sources and use consensus or reputation mechanisms to ensure data integrity. *Hardware oracles* use trusted execution environments or tamper-proof sensors to provide cryptographically signed data directly from the physical world.

The contributions in this thesis adopt different approaches to this challenge depending on the data source and trust model of each application. In the vehicle identity system (Chapter 6), authorized stakeholders (OEMs, repair shops, insurance companies) act as *institutional oracles*: their real-world authority and accountability provide the trust basis for the data they submit, enforced through role-based access control. In the pothole management system (Chapter 7), the architecture combines two oracle types: connected vehicles act as sensor oracles, with onboard CV models submitting cryptographically signed detection reports to the blockchain under the assumption that the edge software is uncompromised; a subsequent municipal validation step acts as a human oracle, filtering fraudulent or inaccurate reports before rewards are distributed. Photographic evidence is stored on IPFS with on-chain CID anchors.

None of the current contributions employ decentralized oracle networks such as Chainlink Chainlink[1], as the data sources in each case are domain-specific and stakeholder-authenticated rather than publicly available data feeds. However, the reliance on trusted data sources remains a cross-cutting limitation discussed in Chapter 8, and the integration of decentralized oracle infrastructure is identified as a future research direction in Chapter 9.

2.5 Cryptographic Techniques for Privacy

The transparency of public blockchains, while essential for verifiability, creates a fundamental tension with data privacy requirements. Several cryptographic techniques can resolve this tension by enabling computation or verification on data without revealing the underlying plaintext. This section introduces the three main approaches relevant to this thesis.

2.5.1 Homomorphic Encryption

Homomorphic encryption (HE) is a class of encryption schemes that allow computations to be performed directly on ciphertexts, producing encrypted results that, when decrypted, correspond to the result of the same operations performed on the plaintexts [17]. Formally, an encryption scheme $\mathscr{E}$ is homomorphic for an operation $\oplus$ if:

$$\mathscr{E}(m_1) \otimes \mathscr{E}(m_2) = \mathscr{E}(m_1 \oplus m_2) \tag{2.1}$$

where m_1 and m_2 are plaintext values, $\oplus$ is the plaintext operation, and $\otimes$ is the corresponding ciphertext operation.

HE schemes are classified by the operations they support:

- **Partially Homomorphic Encryption (PHE):** Supports either addition or multiplication on ciphertexts, but not both. Examples include RSA (multiplicatively homomorphic) and Paillier (additively homomorphic). PHE schemes are efficient but limited in the computations they can express.

[1] https://chain.link/. Accessed: Friday 31st July, 2026.

- **Somewhat Homomorphic Encryption (SHE):** Supports both addition and multiplication but only for a limited number of operations before noise accumulation renders the ciphertext undecryptable.
- **Fully Homomorphic Encryption (FHE):** Supports arbitrary computations on encrypted data without any limit on the number or type of operations. FHE was first demonstrated as theoretically feasible by Gentry in 2009 [17] using ideal lattice-based constructions. Subsequent generations of FHE schemes (BGV, BFV, CKKS, TFHE) have progressively improved performance, though FHE remains computationally intensive compared to plaintext operations.

A worked example. The utility of the homomorphic property is best seen in a setting where one party holds data it will not disclose and another needs a function of that data. Consider three suppliers reporting a single environmental KPI (i.e. annual water consumption) to a purchasing company that must compute a weighted sector aggregate but has no right to see the individual figures.

Under an additively homomorphic scheme such as Paillier, each supplier i encrypts its value x_i under the evaluator's public key and publishes only the ciphertext $\mathscr{E}(x_i)$. The evaluator, holding no secret material beyond its own decryption key, computes

$$\mathscr{E}(x_1) \otimes \mathscr{E}(x_2) \otimes \mathscr{E}(x_3) \;=\; \mathscr{E}(x_1 + x_2 + x_3), \tag{2.2}$$

where $\otimes$ denotes modular multiplication of ciphertexts. Decrypting the result yields the sum, and nothing else: the individual x_i are never available to the evaluator at any point in the protocol. Weights are accommodated by the same property, since scalar multiplication by a public constant w_i is repeated addition,

$$\mathscr{E}(x_i)^{w_i} \;=\; \mathscr{E}(w_i x_i), \tag{2.3}$$

so the evaluator can compute $\mathscr{E}(\sum_i w_i x_i)$ from the ciphertexts and a public weight vector alone.

Two features of this construction generalise to the framework of 4. First, the *structure* of the computation is public, anyone can see which ciphertexts were combined and with which weights, while the *inputs* remain private; this asymmetry is what makes the result auditable without being disclosive. Second, the evaluator is untrusted with respect to the inputs but trusted with respect to the output, so

the design question becomes not "who may compute?" but "who may decrypt what?". When the evaluator is a smart contract rather than a company, the first property becomes stronger still, because the computation is not merely publishable but replayable by any observer.

The limitation of the partially homomorphic case is equally instructive: Paillier supports addition but not multiplication of two ciphertexts, so a weighted sum is expressible but a product of two private values is not. Fully homomorphic schemes lift this restriction (2.5.2), at a substantial cost in computation and ciphertext size.

The supplier evaluation framework (Chapter 4) employs FHE to compute weighted sustainability indices from encrypted Key Performance Indicators (KPIs), ensuring that the raw supplier data is never exposed to purchasing companies or to blockchain observers.

2.5.2 Fully Homomorphic Encryption on Blockchain (fhEVM)

The fhEVM is an extension of the Ethereum Virtual Machine developed by Zama that adds native support for FHE operations within smart contracts [18]. The fhEVM introduces encrypted data types (e.g., `euint8`, `euint16`, `euint32`) that can be manipulated using standard arithmetic and comparison operations, with the underlying cryptographic operations handled transparently by the modified EVM. The key technical components of the fhEVM infrastructure are the following:

- **TFHE scheme:** The fhEVM uses the TFHE (Torus Fully Homomorphic Encryption) scheme, which operates on encrypted bits and supports fast bootstrapping to reduce noise accumulation, enabling unlimited sequential operations.
- **Key Management Service (KMS):** The fhEVM relies on a threshold key management service that distributes the decryption key among multiple parties, so that no single entity can decrypt ciphertexts unilaterally. Decryption requires the cooperation of a threshold number of key holders.
- **Client-side encryption:** Suppliers encrypt their data using the network's public key before submitting it to the smart contract. The encrypted values are stored in contract storage and can be operated upon by the smart contract logic without decryption.

- **Access control for decryption:** The fhEVM provides access control primitives that determine which addresses are permitted to request decryption of specific ciphertext values, enabling differentiated data visibility.

The supplier evaluation framework (Chapter 4) leverages fhEVM to implement a smart contract that receives encrypted sustainability KPIs from suppliers, computes a weighted sustainability index entirely on the ciphertexts using homomorphic addition and multiplication, and stores the encrypted result on-chain. The sustainability index can then be decrypted by authorized parties (the purchasing company) while the underlying KPI values remain permanently encrypted.

It is important to acknowledge the trust assumptions of the fhEVM architecture. The current implementation relies on a centralized or semi-centralized KMS for key management, introducing a trust dependency that partially offsets the decentralization benefits of the underlying blockchain. This limitation and potential mitigation strategies, such as combining fhEVM with ZK-SNARKs for verifiable decryption, are discussed in Chapter 4.

2.5.3 Zero-Knowledge Proofs

A zero-knowledge proof (ZKP) is a cryptographic protocol in which a prover can convince a verifier that a statement is true without revealing any information beyond the truth of the statement itself [40]. ZKPs satisfy three properties: *completeness* (an honest prover can always convince an honest verifier), *soundness* (a dishonest prover cannot convince an honest verifier of a false statement, except with negligible probability), and *zero-knowledge* (the verifier learns nothing beyond the validity of the statement).

In the blockchain context, ZKPs are used both for privacy (enabling private transactions on public chains, as in Zcash and Tornado Cash) and for scalability (enabling zk-rollups to prove the correctness of batched state transitions without re-executing all transactions on L1). While zero-knowledge proofs are not directly employed in the current implementations of this thesis, they represent a natural complement to the FHE-based approach of Contribution 1 and are discussed as a future research direction in Chapter 9. Specifically, ZK-SNARKs could be combined with FHE to provide end-to-end verifiability of computations on encrypted data, reducing the trust assumptions inherent in the current fhEVM architecture.

Beyond their role in privacy, ZKPs underpin the zk-rollup scaling solutions discussed in Section 2.2.4. Several cost projections in this thesis (Chapters 6, 4, and 7) include Polygon zkEVM among the evaluated deployment targets, making the ZKP scalability paradigm directly relevant to the economic analysis of the proposed systems.

2.6 Smart Contract Security

Smart contract security is a cross-cutting concern that affects every contribution in this thesis. The immutability of deployed contracts means that vulnerabilities cannot be patched after deployment, making pre-deployment security analysis essential. This section surveys the most relevant vulnerability classes and the analysis tools employed throughout the thesis.

2.6.1 Common Vulnerabilities and Attack Vectors

The smart contract security landscape has been shaped by several high-profile incidents that collectively led to the identification and categorization of recurring vulnerability patterns. The most relevant to this thesis are the following:

Reentrancy. A reentrancy attack occurs when a malicious contract exploits the ability to call back into the vulnerable contract before the first invocation has completed, potentially re-executing state-changing logic. The canonical example is the 2016 DAO attack, in which approximately 3.6 million ETH were drained due to a reentrancy vulnerability. The standard mitigation is the Checks-Effects-Interactions pattern, in which state variables are updated before any external calls, complemented by the use of the `ReentrancyGuard` modifier from OpenZeppelin [29].

Access control bypass. Smart contracts that fail to properly restrict sensitive functions to authorized callers can be exploited by unauthorized users. This is particularly critical in the multi-stakeholder systems presented in this thesis, where different roles (OEMs, repair shops, insurance companies, municipal authorities) must have different permissions. All contributions employ role-based access control,

implemented through OpenZeppelin's `AccessControl` or `Ownable` contracts, to enforce strict authorization boundaries.

Integer overflow and underflow. Prior to Solidity version 0.8.0, arithmetic operations on unsigned integers did not revert on overflow or underflow, leading to potentially catastrophic value wrapping. Since Solidity 0.8.0, arithmetic operations include built-in overflow checks that automatically revert on invalid results. All contributions in this thesis use Solidity 0.8.x or later, benefiting from these built-in protections.

Front-running. On public blockchains, pending transactions are visible in the mempool before inclusion in a block, enabling malicious actors to observe a transaction and submit a competing transaction with a higher gas price to be processed first. This is relevant to marketplace operations (Chapter 6), where a front-runner could observe a purchase transaction and attempt to buy the asset first. Commit-reveal schemes and private transaction relays are potential mitigations, though they add complexity and are not implemented in the current prototypes.

Denial of Service. Contracts that iterate over unbounded data structures can be rendered inoperable if an attacker inflates the data to the point where a single transaction exceeds the block gas limit. The smart contracts in this thesis are designed with bounded iterations and pagination mechanisms to mitigate this risk.

2.6.2 Formal Verification and Automated Analysis Tools

Formal verification applies mathematical methods to prove that a smart contract satisfies a formal specification, guaranteeing correctness regarding the specified properties. While full formal verification is resource-intensive, several automated tools provide practical approximations:

Slither: Slither [41] is a static analysis framework for Solidity that runs a suite of vulnerability detectors covering reentrancy, unprotected functions, state variable shadowing, and other patterns. It reports findings categorized by severity (high, medium, low, informational). Slither is the primary security analysis tool used across

all contributions in this thesis. The typical workflow involves running Slither on the compiled contracts, analyzing the findings, and distinguishing true positives from false positives attributable to the use of inherited library code.

Mythril and Manticore: Mythril [42] and Manticore [43] are symbolic execution engines that explore the space of possible execution paths and identify states that violate security invariants. These tools are more computationally expensive than static analysis but can detect deeper vulnerabilities that depend on specific input sequences.

Echidna: Echidna is a property-based fuzzing tool for Solidity that generates random sequences of transactions to test invariants defined by the developer. It is particularly useful for discovering edge cases in complex state machine logic, such as the lifecycle management workflows in Chapters 6 and 7.

2.6.3 Security Best Practices and Design Patterns

Beyond vulnerability-specific mitigations, the smart contracts developed in this thesis adhere to a set of general security design patterns:

- **Checks-Effects-Interactions:** All state modifications are performed before external calls to prevent reentrancy.
- **Principle of least privilege:** Each function is restricted to the minimum set of roles required for its execution.
- **Fail-safe defaults:** Functions default to the most restrictive behavior; permissions must be explicitly granted.
- **Emergency stop (circuit breaker):** Critical contracts include a pause mechanism (`Pausable` from OpenZeppelin) that allows an administrator to halt operations in case of a detected vulnerability.
- **Pull over push:** Where possible, token distributions follow the pull pattern, in which recipients claim their rewards, rather than the push pattern, which iterates over recipient lists and risks gas limit issues.

2.7 The Automotive Ecosystem and Relevant Standards

This section introduces the domain-specific context that motivates the thesis. It covers the current structure of vehicle identification and registration, the emerging paradigm of connected vehicles, and the EU regulatory framework that several contributions address.

2.7.1 Vehicle Identification and Registration

Every motor vehicle produced for sale in regulated markets is assigned a Vehicle Identification Number (VIN), a 17-character alphanumeric code standardized by ISO 3779 and ISO 3780 [44]. The VIN encodes information about the manufacturer, vehicle attributes (model, engine type, body style), and a sequential production number. The VIN serves as the primary identifier for vehicle registration, insurance, recall tracking, and history reporting throughout the vehicle's lifecycle.

Current vehicle registration and history systems are centralized and fragmented. In Europe, each member state maintains its own vehicle registry, with limited cross-border interoperability. Vehicle history reports, provided by commercial services such as Carfax (US) and CarVertical (EU), aggregate data from multiple sources (government registries, insurance databases, inspection records, police reports) but depend on the willingness and accuracy of data providers. This centralized model introduces several trust gaps: registries can be corrupted or manipulated (particularly relevant for odometer fraud), data availability depends on commercial relationships, and cross-border verification is cumbersome. The European Parliament has estimated that odometer fraud affects 5–12% of second-hand vehicle transactions in the EU, causing annual losses of €5.6–9.6 billion [7].

The vehicle identity system (Chapter 6) proposes an alternative model in which each vehicle is represented as a unique on-chain token (ERC-721 NFT) carrying its complete lifecycle record, eliminating the dependence on centralized registries and commercial history providers.

2.7.2 Connected and Autonomous Vehicles

Modern vehicles are increasingly equipped with sensors, connectivity modules, and onboard computing platforms that generate and transmit data continuously. Connected vehicles communicate through Vehicle-to-Everything (V2X) protocols, which encompass Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Network (V2N), and Vehicle-to-Pedestrian (V2P) communication channels. The data generated by connected vehicles, including telemetry, location, speed, acceleration, and environmental sensor readings, creates opportunities for new services but also raises significant challenges related to data ownership, privacy, and trust.

One contribution in this thesis directly leverages connected vehicle data. The pothole management system (Chapter 7) envisions connected vehicles as mobile sensing platforms that detect road surface defects using onboard cameras and inertial measurement units (IMUs), reporting them to the blockchain for municipal maintenance prioritization. The same architectural discussion (Chapter 7, Section 7.3.6) also addresses the complementary role of citizen smartphones as a reporting channel, preserving coverage in regions where connected vehicles have not yet been deployed at scale.

2.7.3 EU Regulatory Framework (Battery Regulation, CSDD)

The European Union has enacted two regulatory instruments that directly motivate contributions in this thesis.

EU Battery Regulation (2023/1542). Adopted in July 2023, the EU Battery Regulation [6] establishes comprehensive requirements for the sustainability, safety, and lifecycle management of batteries placed on the EU market. A central provision is the mandatory introduction of digital battery passports (DBPs) for all electric vehicle batteries starting from February 2027. The regulation specifies that the DBP must be based on a "decentralized data system" that ensures data integrity, authentication, and differentiated access control for multiple stakeholder groups: the general public (access to basic battery information), notified bodies and market surveillance authorities (access to compliance data), and parties with a "legitimate interest" such as recyclers and second-life operators (access to detailed technical data). The regulation further requires that data responsibility transfers between

economic operators as the battery changes hands, and that the passport remains accessible throughout the battery's lifecycle and for a specified period after end-of-life.

Chapter 5 presents a systematic analysis demonstrating that blockchain technology satisfies all technical requirements imposed by this regulation, establishing a direct mapping between regulatory provisions and blockchain architectural properties.

Corporate Sustainability Due Diligence Directive (CSDD). The CSDD, adopted in 2024, requires large companies operating in the EU to implement due diligence measures to identify, prevent, and mitigate adverse human rights and environmental impacts across their supply chains [5]. The directive creates a regulatory obligation for supply chain transparency that is difficult to fulfill with traditional information systems, as suppliers are reluctant to disclose sensitive performance data to their customers. Chapter 4 addresses this tension by proposing a framework that combines blockchain and FHE to enable verifiable sustainability evaluation without exposing confidential supplier data.

2.7.4 Digital Product Passports

The concept of digital product passports (DPPs) extends beyond batteries to encompass a broad range of products. The EU's Ecodesign for Sustainable Products Regulation (ESPR), proposed in 2022, envisions DPPs as standardized digital records that accompany products throughout their lifecycle, containing information about materials, manufacturing processes, repairability, recyclability, and environmental footprint. While the specific data requirements and implementation standards are still under development for most product categories, the battery passport mandated by Regulation 2023/1542 serves as the first concrete instantiation of the DPP concept.

The DPP paradigm aligns directly with the technical capabilities demonstrated across this thesis. The NFT-based vehicle representation (Chapter 6) can be understood as a vehicle digital product passport that tracks the asset from manufacturing through multiple ownership changes to eventual disposal. The battery passport analysis (Chapter 5) demonstrates regulatory compatibility for the specific case of EV batteries. The supplier evaluation framework (Chapter 4) addresses the upstream

data requirements of DPPs by enabling verifiable supply chain assessment. Together, these contributions provide a comprehensive technical foundation for blockchain-based digital product passports across the automotive lifecycle.

This chapter has presented the technical and domain-specific foundations that the subsequent chapters build upon. The reader is now equipped with the necessary background to understand the design decisions, architectural patterns, and evaluation methodologies employed in the four contributions. Chapter 3 surveys the existing literature across the relevant research domains and identifies the specific gaps that motivate each contribution.

Chapter 3

Literature Review

This chapter surveys the state of the art across the research domains addressed by this thesis and identifies the specific gaps that motivate each contribution. Unlike the paper-specific related work sections presented in Chapters 4-7, which are tightly scoped to the individual problems each publication addresses, this chapter provides a panoramic view of the broader research landscape. Its purpose is threefold: first, to establish that the trust gaps identified in Chapter 2 are widely recognized in the literature; second, to demonstrate that existing solutions address these gaps only partially, leaving significant open problems; and third, to show that the gaps, although studied largely in isolation by different research communities, share structural characteristics that call for a unified response.

The review is organized thematically. Section 3.1 examines the application of blockchain technology to the automotive industry, covering vehicle identity management, odometer fraud prevention, and second-hand vehicle markets. Section 3.2 surveys blockchain-based supply chain management, including traceability, supplier evaluation, privacy-preserving approaches, and the emerging domain of digital product passports, which represents the regulatory formalization of supply chain transparency for specific product categories. Section 3.3 reviews blockchain-based incentive mechanisms, including token economics, and crowdsourcing for civic participation. Section 3.4 addresses blockchain applications in smart cities and infrastructure management. Finally, Section 3.5 synthesizes the gaps identified across all domains and positions the contributions of this thesis within the existing research. The ordering of sections moves from domain-specific applications (automotive, sup-

ply chain) to cross-cutting capabilities (incentives, smart cities) before converging on the gap synthesis; this differs intentionally from the lifecycle ordering of Chapters 4-7, as the literature review must first establish each research community's perspective before showing how they connect.

Throughout this chapter, the reader should note a recurring observation: while each domain has produced valuable individual contributions, the literature reveals a persistent fragmentation: systems are designed for isolated use cases, evaluated in narrow contexts, and rarely connected to the broader ecosystem in which they would need to operate. This fragmentation is itself a key finding of the review, and it motivates the integrative perspective that distinguishes this thesis from the individual works it surveys.

3.1 Blockchain in the Automotive Industry

The automotive sector has attracted significant research attention as a target domain for blockchain technology, driven by the industry's inherent multi-stakeholder complexity and the critical importance of data integrity across the vehicle lifecycle. Early survey works by Fraga-Lamas and Fernández-Caramés [3] and Singh [45] identified a broad spectrum of potential applications, spanning vehicle-to-vehicle communication, component provenance tracking, insurance automation, fleet management, and ownership transfer. These surveys established a consensus that blockchain's core properties, e.g. immutability, transparency, decentralization, and non-repudiation, are well suited to resolving the trust deficits that permeate the automotive ecosystem. However, they also noted that the majority of proposed applications remained at the conceptual stage, with few fully implemented and evaluated prototypes.

Subsequent research has concentrated on three interrelated areas, each of which addresses a specific aspect of the vehicle's digital identity: how a vehicle is uniquely represented on-chain (Section 3.1.1), how its operational history is recorded and protected from tampering (Section 3.1.2), and how this information enables trustworthy trading in secondary markets (Section 3.1.3).

3.1.1 Vehicle Identity and Digital Twins

The representation of physical vehicles as digital entities on a blockchain has been approached through two complementary paradigms: registry-based systems, in which a smart contract maintains a mapping between vehicle identifiers (typically VINs) and associated data, and token-based systems, in which each vehicle is represented as a unique cryptographic token.

In the registry paradigm, several early systems proposed storing vehicle-related data on the blockchain by linking records to the Vehicle Identification Number. Commercial initiatives such as VINChain [46] and CarVertical [47] aggregate data from government registries, insurance databases, police records, and paid APIs, storing them on the blockchain indexed by VIN. These platforms demonstrated the feasibility of blockchain-based vehicle data aggregation, but operate as centralized intermediaries that curate and control data access, limiting the decentralization benefits that blockchain could otherwise provide. The Renault Group partnered with Microsoft and VISEO to develop one of the first industrial prototypes of a digital car maintenance book based on blockchain [48], although the system remained a proof-of-concept without public deployment.

Academic contributions have explored more decentralized approaches. Singh et al. [49] proposed V-CARE, a blockchain framework for secure vehicle health records in which car owners, manufacturers, insurance companies, and road authorities participate as consortium members with differentiated write permissions. Brousmiche et al. [50] described a Quorum-based system for digitizing vehicle lifecycle information, in which the manufacturer creates a digital car book at the point of sale and grants selective access to other actors such as insurers. A distinctive feature of their work is the analysis of lessons learned from deploying a consortium blockchain in an industrial context.

The token-based paradigm leverages the ERC-721 Non-Fungible Token standard [34] to create a one-to-one mapping between physical vehicles and unique on-chain assets. This approach has gained traction in adjacent domains: Gebreab et al. [51] demonstrated NFT-based traceability for medical devices, Alnuaimi et al. [52] applied it to fine jewelry, and Elmay et al. [53] to shipping containers. These works established that NFTs can serve as digital passports for physical assets across diverse supply chains, providing provenance tracking and ownership manage-

ment through the token's built-in transfer semantics. However, prior to the vehicle identity system presented in Chapter 6, no work had combined NFT-based vehicle representation with a complete multi-stakeholder update workflow, a decentralized marketplace with price negotiation, and a locking mechanism that prevents transfers when mandatory updates are pending [1].

The concept of digital twins (virtual replicas of physical assets updated in real time) represents a more ambitious form of vehicle identity. Naseri et al. [54] surveyed the use cases, requirements, and platforms for creating digital twins of electric vehicle battery systems, identifying blockchain as one of the enabling technologies for ensuring data integrity across the twin's lifecycle. While the digital twin concept extends beyond what any single contribution in this thesis implements, the NFT-based vehicle representation (Chapter 6) can be understood as a foundational building block toward a full vehicle digital twin Chapter 9.

3.1.2 Vehicle History and Odometer Fraud Prevention

The integrity of vehicle history records has motivated a dedicated stream of research, driven primarily by the economic impact of odometer fraud. The European Parliament estimated that up to 40% of second-hand vehicles traded across EU borders have manipulated odometers, resulting in annual losses of approximately €8.9 billion [7]. This figure has served as a key motivator for blockchain-based solutions that make such manipulation detectable or infeasible.

Chanson et al. [55] presented one of the earliest blockchain-based odometer fraud prevention systems. Their approach records mileage and GPS data on the blockchain, with data acquired from a dongle connected to the vehicle's OBD-II interface. While the prototype was limited (data flows through a laptop before reaching the blockchain) the work was seminal in articulating how blockchain's immutability directly addresses the odometer rollback problem. The authors further discussed privacy considerations, suggesting that raw data could be encrypted before off-chain storage, with only the vehicle owner retaining access.

Vasile and Groza [56] proposed DeMetrA, a system that records odometer readings and tracks vehicle ownership using asymmetric key pairs. Each vehicle is represented by a key pair created at registration, with the private key used for both odometer updates and ownership transfers. The system introduces a layered privacy

model in which different data items have different visibility levels, anticipating the hybrid on-chain/off-chain strategy that several subsequent works would adopt.

Beyond odometer data, Saldamli et al. [57] proposed a more comprehensive vehicle health recording system that acquires data from the OBD-II port, including engine-related parameters such as coolant and oil temperature, to compute a maintenance rating and a driver behavior score. This work is notable for attempting to derive higher-level indices from raw vehicle data, although the scoring algorithm is rudimentary and the system lacks an incentive mechanism to encourage data submission.

Reimers et al. [58] shifted the focus upstream, proposing a system that integrates blockchain and IoT devices to track components through the automotive supply chain. Their work bridges the vehicle identity and supply chain domains, demonstrating that the provenance of individual components can be linked to the vehicle they are eventually installed in. This cross-domain perspective aligns with the thesis-wide view that vehicle identity, supply chain transparency, and component traceability are interconnected rather than independent problems.

A common limitation across these works is the reliance on trusted data sources: the systems assume that the entity submitting data (the OBD-II dongle, the manufacturer, the inspection station) is honest. No existing vehicle history system provides cryptographic guarantees against collusion between a malicious seller and a complicit data provider. This limitation is inherent to any system that bridges physical and digital worlds, an observation discussed further in Chapter 8 as a cross-cutting challenge.

3.1.3 Second-Hand Vehicle Markets

The application of blockchain to second-hand vehicle trading addresses the information asymmetry that characterizes pre-owned vehicle markets. Nigam et al. [8] conducted a qualitative study demonstrating that consumers, particularly younger generations, perceive blockchain as a trust-enhancing technology for used car transactions, while intermediaries view it as both an opportunity and a potential threat to their business model.

Several blockchain-based platforms for used vehicle trading have been proposed, employing different blockchain infrastructures and architectural strategies. Zhang

et al. [59] developed a traceability system for used cars on Hyperledger Fabric, designed for a consortium of manufacturers, logistics enterprises, testing institutions, and dealers. Similarly, Zhang et al. [60] proposed a deposit platform for used-car transactions on the same infrastructure, incorporating test reports, insurance data, and purchase certificates. Both works leverage the permissioned nature of Hyperledger Fabric to manage access control, but this choice limits decentralization and introduces a dependency on the consortium's governance.

Yoo and Ahn [61] combined Ethereum smart contracts with IPFS for decentralized document storage, where the smart contract stores only content hashes. This hybrid on-chain/off-chain approach reduces costs while maintaining data integrity through hash verification, a pattern that has become a de facto standard in the field and that all implementation chapters in this thesis adopt in various forms.

Kumar et al. [62] focused specifically on the bidding process, proposing an Ethereum-based e-auction system in which vehicle documents are stored on IPFS and an inspector certifies the vehicle's condition before the bidding smart contract collects offers. Subramanian and Thampy [63] addressed the specific case of pre-owned electric vehicles, arguing that the battery's condition, affected by charging history and driver behavior, requires dedicated tracking that traditional vehicle history systems do not provide.

Table 3.1 summarizes the key characteristics of existing blockchain-based vehicle systems. A systematic comparison reveals two persistent gaps. First, none of the existing systems provides a complete lifecycle management framework that combines vehicle identity creation, multi-stakeholder history updates, compliance enforcement (through mechanisms such as transfer locking), and integrated marketplace functionality within a single architecture. Second, no prior work employs Non-Fungible Tokens for vehicle representation for second-hand trading, despite the natural fit between the ERC-721 standard's unique ownership semantics and the one-to-one relationship between a physical vehicle and its digital identity. Chapter 6 addresses both gaps by proposing a system in which each vehicle is minted as a dynamic ERC-721 token carrying updatable metadata, with a marketplace smart contract that handles listing, bidding, negotiation, and automatic ownership transfer [1].

Table 3.1 Comparison of blockchain-based vehicle identity and second-hand market systems.

Work	Platform	NFTs	Marketplace	Multi-stakeholder	IPFS
VINChain [46]	Graphene	No	No	Limited	No
CarVertical [47]	Ethereum	No	No	Limited	No
Singh et al. [49]	Ethereum	No	No	Yes	No
Brousmiche et al. [50]	Quorum	No	No	Yes	No
Zhang et al. [59]	Hyp. Fabric	No	No	Yes	No
Yoo and Ahn [61]	Ethereum	No	No	No	Yes
Kumar et al. [62]	Ethereum	No	Auction only	No	Yes
Subramanian [63]	Ethereum	No	No	Yes	Yes
Ch. 6 [1]	**Ethereum**	**Yes**	**Full**	**Yes**	**Yes**

3.2 Blockchain for Supply Chain Management and Digital Product Passports

Supply chain management has become one of the most active application domains for blockchain technology, driven by the fundamental alignment between blockchain's transparency properties and the supply chain's need for verifiable, multi-party data sharing. The literature in this area has evolved along three complementary trajectories: the use of blockchain for general traceability and transparency (Section 3.2.1), the specific problem of evaluating and selecting suppliers (Section 3.2.2), and the emerging recognition that privacy must be preserved even as transparency is enhanced (Section 3.2.3). This section concludes by examining digital product passports (Section 3.2.4), which represent the regulatory culmination of supply chain transparency—the point at which the European Union has mandated that blockchain's traceability promises be realized in concrete, legally binding form.

The progression from traceability through supplier evaluation and privacy to digital passports is not merely a convenient organizational choice: it mirrors the escalating demands placed on supply chain information systems, from voluntary transparency to regulatory compliance, and it reflects the trajectory of the contributions in Chapters 4 and 5.

3.2.1 Traceability and Transparency

Blockchain-based supply chain traceability has been extensively studied across multiple industries. Leng et al. [64] conducted a comprehensive survey of blockchain-empowered sustainable manufacturing and product lifecycle management under Industry 4.0, identifying key research clusters in provenance tracking, circular economy support, and multi-stakeholder coordination. Their analysis confirmed that blockchain's immutability and decentralization can address the opacity inherent in global supply chains, but also highlighted persistent challenges related to scalability, interoperability, and the integration of IoT devices for automated data capture.

In the automotive domain specifically, Reimers et al. [58] proposed a system integrating blockchain and IoT devices to track components through the automotive supply chain, demonstrating that component-level provenance can be linked to the vehicle in which parts are eventually installed. This work, already discussed in Section 3.1.2, bridges the vehicle identity and supply chain traceability domains, illustrating the interconnection that motivates the lifecycle perspective of this thesis.

Ribeiro da Silva et al. [65] examined the potential of blockchain for enabling data sharing and circular economy practices in the electric vehicle battery supply chain. Through a case study approach, the authors demonstrated that blockchain can support the traceability of battery materials from extraction to recycling, but noted that practical adoption is hindered by the reluctance of supply chain actors to share competitively sensitive data: a finding that directly motivates the privacy-preserving approach of Chapter 4.

The textile industry, which serves as the validation domain for Contribution 1, has also received attention. Alves et al. [2] developed a traceability platform for monitoring environmental and social sustainability across the textile and clothing value chain. Their work defined the KPI taxonomy (organized into environmental/circular and economic/social categories) that the supplier evaluation framework in Chapter 4 adopts for its sustainability index computation.

A common architectural pattern across these works is the hybrid on-chain/off-chain storage strategy: storing lightweight references (hashes, identifiers, and cryptographic proofs) on the blockchain while keeping bulk data (documents, images, and sensor readings) in off-chain storage systems such as IPFS or private databases. This pattern, which emerged independently in multiple supply chain implementations, is

also employed across all four contributions in this thesis: a convergence examined in Chapter 8.

3.2.2 Supplier Evaluation and Selection

The specific problem of evaluating and selecting suppliers has generated a rich literature that spans multiple methodological traditions. Traditional approaches rely on multi-criteria decision-making (MCDM) methodologies, which formalize the inherently multi-dimensional nature of supplier evaluation. Ecer et al. [66] proposed an integrated framework combining LMAW for criteria weighting with TOPSIS for supplier ranking, applied to a healthcare supply chain. Hendiani and Walther [67] addressed the uncertainty inherent in human judgment by proposing a novel interval-valued intuitionistic fuzzy approach. In the textile sector specifically, Sithi et al. [68] integrated Triple Bottom Line principles with SWARA and TOPSIS methods, while Vaezi et al. [69] combined the Best-Worst Method for criteria weighting with MULTIMOORA for supplier ranking. Zekhnini et al. [70] introduced the concept of supplier viability, encompassing digitization, resilience, and sustainability, as an evaluation criterion within a fuzzy inference system designed for the automotive industry.

Machine learning approaches have also been explored. Abbasian et al. [71] combined K-means clustering with Random Forest classification, using principal component analysis for dimensionality reduction, to improve the accuracy of supplier assessments. Pap et al. [72] proposed an AI-assisted supplier selection method that they benchmarked against MCDM approaches, though without detailing the specific methodology.

A third stream of research has introduced blockchain technology to the supplier evaluation process, primarily to enhance transparency and auditability. Haffar and Özceylan [73] proposed a blockchain-based system for automating supplier selection in sustainable and lean-agile supply chains, incorporating economic, environmental, social, and leagility criteria alongside a reputation system based on mutual ratings. Vaezi et al. [69] integrated MCDM methodologies with blockchain to leverage its transparency and traceability properties. Modares et al. [74] used blockchain to enhance information sharing among supply chain stakeholders, thereby facilitating more informed supplier selection decisions.

Despite this diversity of approaches, a critical observation emerges: none of the existing works addresses the tension between transparency and data confidentiality. MCDM and ML methods assume that evaluators have access to the raw supplier data, which suppliers may be reluctant to share. Blockchain-based systems enhance the transparency and integrity of the evaluation process, but still require suppliers to disclose their performance metrics in some form. The gap, the absence of a mechanism that enables suppliers to demonstrate their sustainability while keeping their underlying data private, is precisely what Contribution 1 addresses through fully homomorphic encryption (Chapter 4).

3.2.3 Privacy-Preserving Approaches

The tension between supply chain transparency and data confidentiality has been recognized as a fundamental challenge by several researchers. Berger et al. [75] conducted a study on the sensitivity of data included in digital product passports, demonstrating that data science and machine learning techniques can be used to preserve the confidentiality of certain data fields while still enabling meaningful product management. Their work established the conceptual need for privacy-preserving data exchange for DPPs, but did not propose a specific cryptographic implementation.

Sahai et al. [76] proposed a system combining blockchain with zero-knowledge proofs (ZKPs) to enable privacy and traceability in supply chains. Their approach allows participants to prove compliance with supply chain requirements without revealing the underlying data, using ZK-SNARKs for succinct non-interactive proofs. While ZKPs provide strong privacy guarantees, they are best suited to proving binary predicates (e.g., “this supplier’s emissions are below a threshold”) rather than computing quantitative indices from encrypted inputs.

Fang et al. [77] developed a blockchain-cloud framework for privacy-enhanced distributed industrial data trading based on verifiable credentials. Their system separates data ownership from data access through a credential-based model, allowing data holders to selectively disclose attributes. This approach is relevant to the DPP context (Section 3.2.4) but does not address the specific problem of performing computations on encrypted data.

The approach adopted in Chapter 4, fully homomorphic encryption on blockchain via the fhEVM infrastructure [18], occupies a distinct position in this landscape. Unlike ZKP-based approaches, which prove properties of data without revealing it, FHE enables arbitrary computations on encrypted data, producing encrypted results that can be decrypted only by authorized parties. This allows the smart contract to compute a composite sustainability index from encrypted KPIs, a capability that goes beyond binary compliance verification to produce a quantitative, comparable score. Among the surveyed works, no prior system combines FHE with blockchain smart contracts for supplier sustainability evaluation [21]. The trade-off is computational cost: FHE operations are orders of magnitude more expensive than plaintext operations, a limitation that the cost analysis in Chapter 4 quantifies.

The privacy-preserving techniques discussed in this section form a spectrum that is relevant across the entire thesis, not only to the supply chain domain. The vehicle identity system (Chapter 6) operates at the transparent end of this spectrum, with all vehicle data publicly verifiable. The supplier evaluation framework (Chapter 4) pushes furthest toward the private end, with computation performed entirely on encrypted data. This privacy-transparency spectrum is analyzed as a cross-cutting theme in Chapter 8.

3.2.4 Digital Product Passports and EU Regulatory Landscape

The concept of digital product passports (DPPs) represents the regulatory formalization of the traceability and transparency capabilities that the preceding subsections have examined from a technical perspective. The European Union has taken the leading role in this domain, with the EU Battery Regulation (2023/1542) [6] serving as the first regulation to mandate DPPs for a specific product category. From February 2027, all electric vehicle batteries placed on the EU market must carry a digital battery passport providing information on materials, manufacturing, performance, and end-of-life management [78]. The broader Ecodesign for Sustainable Products Regulation (ESPR), proposed in 2022, envisions extending DPPs to a wide range of product categories.

Several international initiatives have contributed to defining the DPP concept. The Global Battery Alliance (GBA), a partnership of over 150 organizations, proposed a proof-of-concept battery passport defining technical parameters and tracking

requirements, including indices for child labor and human rights [79]. The Battery Pass consortium, co-funded by the German Federal Ministry for Economic Affairs, published a comprehensive content guidance document translating regulatory requirements into implementation specifications [80]. Despite these initiatives, no officially recognized standard for implementing DPPs has been established, creating uncertainty for manufacturers approaching the 2027 deadline.

In the scientific literature, practical DPP implementations remain scarce. Siska et al. [81] presented BatWoMan, a system defining the DPP concept with relevant stakeholder interactions and an architecture based on the International Data Spaces and Gaia-X frameworks for data management. Bandini et al. [82] introduced a DBP using Ultra-High-Frequency RFID technology for battery lifecycle traceability. Nowacki et al. [83] proposed a DPP architecture based on IOTA distributed ledger technology and smart contracts, identifying a use case structure common to multiple application areas. These implementations explore different technology stacks, but none provides a systematic analysis of how a specific technology aligns with all provisions of the EU Battery Regulation.

The potential of blockchain technology for DPP implementation has been discussed in several survey works. Leng et al. [64] and Ribeiro da Silva et al. [65] analyzed blockchain's role in battery supply chain traceability and circular economy support. Teng et al. [84] surveyed industrial data-driven approaches to energy savings, identifying blockchain-based digital twins and DPPs as enabling technologies. These works consistently identify blockchain as a promising infrastructure for DPPs, citing its decentralization, immutability, and cryptographic data authentication, but they remain at the level of potential analysis rather than regulatory alignment.

Chapter 5 fills this gap by conducting a systematic, provision-by-provision analysis of the EU Battery Regulation, demonstrating that blockchain technology satisfies all technical requirements for DPP implementation, including data storage on a decentralized system, data authentication and integrity, differentiated access control for multiple stakeholder groups, and data responsibility transfer upon ownership change [22]. Beyond its standalone contribution, this regulatory analysis serves a structural role within the thesis: it establishes that the architectural patterns implemented in the engineering contributions (role-based access control, hybrid on-chain/off-chain storage, ownership-linked data management) are not merely technically sound but aligned with the direction of EU policy. The connection be-

tween regulatory requirements and architectural design is examined systematically in Chapter 8.

The supply chain and DPP literature reviewed in this section reveals a clear progression: from voluntary traceability initiatives, through increasingly sophisticated supplier evaluation methods, to the recognition that privacy preservation is essential, and finally to the regulatory mandates that formalize these requirements. The contributions of this thesis follow the same trajectory, from privacy-preserving supplier evaluation (Chapter 4) through regulatory compliance analysis (Chapter 5) to the vehicle-level identity system that extends these principles downstream (Chapter 6). The next section shifts focus from what a vehicle is made of to what it does, examining blockchain-based incentive mechanisms for civic participation.

3.3 Blockchain-Based Incentive Mechanisms

The preceding sections examined how blockchain can establish trustworthy records of vehicle identity, supply chain provenance, and regulatory compliance. This section shifts attention from data integrity to behavior modification: how blockchain-based incentive mechanisms can motivate desirable actions by individuals and communities. This topic is relevant to the pothole management system (Chapter 7), which adapts token-based incentive patterns from adjacent domains, in particular usage-based insurance and participatory sensing, to the specific challenge of sustaining crowdsourced civic infrastructure reporting. The section proceeds from general principles to specific applications. Section 3.3.1 reviews the broader landscape of blockchain-based token economics and gamification, including the usage-based-insurance literature in which blockchain-mediated reward mechanisms for individual behavioral data have been studied most extensively. Section 3.3.2 discusses crowdsourcing and citizen participation mechanisms, connecting these incentive patterns to the smart city applications reviewed in Section 3.4.

3.3.1 Token Economics and Gamification

Blockchain-based token economies provide a technical infrastructure for designing incentive mechanisms that are transparent, programmable, and resistant to manipula-

tion by a central authority. Unlike traditional reward systems in which a single entity controls the rules of distribution, token-based incentives execute through smart contract logic that is auditable by all participants. The ERC-20 standard [33] provides the technical foundation for creating fungible utility tokens on Ethereum, enabling automated distribution according to predefined rules encoded in smart contracts.

Research on decentralized autonomous organizations (DAOs) has demonstrated that token ownership can effectively incentivize user contributions in decentralized platforms. Khan et al. [85] explored the integration of blockchain and edge computing for participatory smart city applications, showing that token-based rewards can align individual participation incentives with collective platform benefit. Their architecture demonstrated the feasibility of using smart contracts to automate reward distribution in multi-stakeholder environments, but did not address the specific challenges of calibrating incentive levels to sustain long-term engagement.

The application of gamification principles, the use of game-design elements in non-game contexts, to blockchain-based systems represents a relatively unexplored intersection. Hamari et al. [86] conducted a literature review of empirical studies on gamification, concluding that gamification generally produces positive effects on engagement, though outcomes depend heavily on context and user demographics. Bitrián et al. [87] demonstrated that gamification enhances user engagement in mobile applications through mechanisms such as challenges, leaderboards, and achievement systems.

A key design challenge for blockchain-based incentive mechanisms is the sustainability of token value. If tokens lack redemption pathways that participants perceive as valuable, accumulation becomes meaningless and participation declines. The pothole management system (Chapter 7) addresses this challenge by proposing multiple utility channels: municipal service redemption, local business partnerships, and governance participation rights. This multi-channel approach reflects a broader recognition in the literature that purely monetary incentives are insufficient for sustained civic engagement, and that non-monetary motivations, such as community impact, reputation, and recognition, must complement financial rewards.

3.3.2 Crowdsourcing and Citizen Participation

Mobile crowd sensing (MCS) has emerged as a cost-effective paradigm for collecting urban-scale data by leveraging the sensor capabilities of smartphones and connected vehicles carried by ordinary citizens [88]. The key challenge in MCS is sustaining participation: without adequate incentives, contributions decline over time as novelty fades. The literature has explored several approaches to this challenge.

Li et al. [89] developed an anonymous reporting mechanism with dynamic incentives for participatory sensing, combining privacy protection with adaptive reward levels that respond to contribution density. Xiong et al. [90] proposed hybrid mechanisms integrating reputation scores with service-based returns, demonstrating that non-monetary incentives can complement financial rewards to sustain engagement. Chen et al. [91] demonstrated smartphone-based infrastructure inspection through simulation, though their architecture relies on a centralized coordination platform that concentrates control over data quality assessment and reward distribution.

These centralized crowdsourcing approaches share a common limitation: the incentive rules, payment processing, and quality assessment remain under single-entity control, creating the same opacity and trust deficits that motivate blockchain adoption in the other domains reviewed in this chapter. Blockchain-based crowdsourcing addresses these limitations by encoding contribution verification and reward distribution in transparent smart contracts, eliminating the need to trust the platform operator.

The intersection of crowdsourcing and blockchain for civic applications has received limited attention. Knuuti et al. [92] introduced a gamification-based approach for road condition data collection, using a mobile game to incentivize citizens to collect GPS-tagged video data for AI-based pavement classification. However, their system maintains a centralized data pipeline where prioritization and reward distribution remain opaque. The few blockchain-based systems for infrastructure reporting, discussed in detail in Section 3.4, address only partial aspects of the crowdsourcing challenge, typically omitting either the incentive mechanism, the duplicate detection logic, or the integration with municipal workflows.

This gap motivates the design of the pothole management system in Chapter 7, which unifies blockchain-based crowdsourcing with token incentives, spatial duplicate detection, algorithmic prioritization, and municipal workflow coordination

in a single framework. The system adapts the token-incentive patterns reviewed in Section 3.3.1, originally developed in the individualized context of usage-based insurance, to the collective context of civic infrastructure monitoring, where the participation unit is a community rather than a single driver and the reward pathway is municipal service redemption rather than premium calibration.

The incentive mechanisms reviewed in this section reveal the range of domains, from individualized insurance telematics to collective civic reporting, in which blockchain-mediated token rewards have been used to sustain participation that centralized systems failed to motivate. In every case, blockchain provides the transparency and auditability that centralized incentive systems lack, while token economics enable programmable reward distribution. The next section examines the broader smart city context in which these incentive mechanisms operate, reviewing blockchain applications in urban governance and infrastructure management.

3.4 Blockchain for Smart Cities and Infrastructure

This section examines blockchain applications in urban governance, infrastructure management, and connected vehicle networks: the context in which the pothole management system (Chapter 7) operates. While the incentive mechanisms reviewed in Section 3.3 provide the economic layer for motivating participation, the smart city literature addresses the broader question of how blockchain can reshape the relationship between municipalities, citizens, and urban infrastructure. The section is organized around three themes: the general landscape of blockchain-based urban governance (Section 3.4.1), the specific domain of infrastructure monitoring and maintenance (Section 3.4.2), and the emerging role of connected vehicles as civic sensing platforms (Section 3.4.3).

3.4.1 Urban Governance and Public Services

Sahoo et al. [93] conducted a comprehensive scoping review of blockchain technology in smart city contexts, identifying five primary knowledge clusters: urban mobility and transportation, transparent urban governance, decentralized resource management, IoT integration and data security, and digital identity management.

This taxonomic breadth reflects a consensus that blockchain's potential in the urban domain extends well beyond any single application, addressing systemic trust deficits inherent in centralized municipal systems.

In the governance domain specifically, Shan et al. [94] investigated collaborative governance models for smart government based on blockchain, demonstrating through an evolutionary game-theoretic analysis that blockchain-based transparency can shift equilibrium strategies toward greater cooperation among government agencies. Their work is notable for going beyond the purely technical dimension of blockchain adoption to model the strategic behavior of institutional actors, a perspective that is relevant to understanding the adoption dynamics of any blockchain-based municipal system, including the pothole management platform proposed in Chapter 7.

Haque et al. [95] provided a broader conceptualization of smart city applications, identifying requirements, architectural patterns, security considerations, and emerging trends. Their analysis confirmed that the core challenge in smart city systems is coordinating heterogeneous actors, such as citizens, municipal agencies, IoT devices, and third-party service providers, under conditions of limited mutual trust. Blockchain's capacity to provide a shared, immutable coordination layer without requiring a trusted intermediary makes it architecturally well suited to this challenge.

In the transportation domain, blockchain has been applied to vehicle registration, supply chain tracking for automotive parts, and ride-sharing platforms [95]. Vehicular ad-hoc networks (VANETs) have received particular attention, with blockchain-based architectures proposed for secure V2X communication and trusted data exchange among vehicles [96]. These applications demonstrate blockchain's suitability for coordinating distributed mobile actors, a characteristic directly relevant to the vehicle-based reporting pathway in the pothole management system.

The overarching observation from this literature is that blockchain has been widely explored for smart city governance at the conceptual and survey level, but the number of fully implemented, evaluated, and cost-analyzed systems remains limited. Most works propose architectures without deploying prototypes, and those that do deploy rarely provide comprehensive economic analysis. This gap between conceptual promise and practical validation motivates the detailed cost projections across five blockchain networks presented in Chapter 7.

3.4.2 Infrastructure Monitoring and Maintenance

Road infrastructure maintenance has traditionally relied on Pavement Management Systems (PMS), decision-support tools operating at network and project levels for budget allocation and maintenance treatment selection [97, 98]. Despite decades of refinement, these systems face persistent challenges: condition assessment depends on periodic manual inspections or dedicated survey vehicles, creating temporal gaps during which defects remain undetected and deteriorate. More recent intelligent approaches employing neural networks for deterioration prediction [12] and optimization models balancing condition against costs operate on centrally collected data within closed workflows, leaving transparency deficits unaddressed.

Smart contracts extend blockchain's utility beyond passive record-keeping to active workflow automation in the infrastructure domain. Tran et al. [99] developed a Machine-as-a-Service platform combining Ethereum smart contracts with IPFS storage for industrial equipment maintenance. Their system demonstrates how blockchain can orchestrate multi-stakeholder maintenance workflows without central coordination, with the architectural pattern of storing large data objects on IPFS while maintaining integrity references on-chain providing a template that the pothole management system (Chapter 7) adapts for cost-effective image management.

Among the few blockchain-based systems specifically addressing road infrastructure, three works deserve attention. Rumpa et al. [100] proposed InfraChain, a sensor-enabled roadway management application combining blockchain with digital twin concepts. Their system uses embedded acoustic and piezoelectric sensors for automated defect detection, but omits incentive mechanisms and duplicate detection, two capabilities essential for scaling beyond controlled sensor deployments to open crowdsourced reporting. Matzutt et al. [101] demonstrated blockchain storage for citizen-reported street problem images with a priority scheduling mechanism, providing the closest precedent to the pothole management system. However, their work, presented as a poster, does not include token-based incentives or vehicle integration, and the priority algorithm is not described in sufficient detail to assess its multi-factor capabilities. Mubarak et al. [102] combined EfficientDet-based computer vision detection with blockchain storage, but lack prioritization algorithms and mechanisms for handling spatially proximate duplicate reports.

Commercial platforms demonstrate greater deployment maturity but uniformly lack blockchain-based transparency. FixMyStreet [103], pioneered by the UK-based mySociety organization, established the civic reporting paradigm with open-source code and partial transparency, but relies on manual triage for prioritization. SeeClickFix [104] provides municipal workflow integration as a proprietary SaaS platform without transparency guarantees. Waze [105] enables driver hazard reporting with gamified rewards, but neither shares data systematically with municipal repair authorities nor provides algorithmic prioritization.

Table 3.2 summarizes this landscape. The systematic comparison reveals that no existing system, academic or commercial, comprehensively integrates blockchain transparency, ERC-20 token incentives, multi-factor algorithmic prioritization, spatial duplicate detection, and hybrid IPFS/blockchain storage within a unified framework accessible to both connected vehicles and citizen reporters. Chapter 7 addresses this gap [23].

Table 3.2 Comparison of pothole management systems (academic).

System	Arch.	Detection	Vehicle	Tokens	Priority	Dupl.
Rumpa et al. [100]	BC	Sensor	Infra.	No	No	No
Matzutt et al. [101]	BC	Photo	No	No	Yes	No
Mubarak et al. [102]	BC	CV	No	No	No	No
Knuuti et al. [92]	Centr.	AI video	No	Yes	Severity	No
FixMyStreet [103]	Centr.	Manual	No	No	Manual	No
Ch. 7 [23]	**BC**	**CV+Phone**	**Yes**	**ERC-20**	**Multi-f.**	**Grid**

3.4.3 Connected Vehicle Networks for Civic Applications

The final dimension of the smart city literature relevant to this thesis concerns the use of connected vehicles as mobile sensing platforms for civic purposes, a capability that the pothole management system in Chapter 7 is designed to exploit.

Modern connected and autonomous vehicles (CAVs) increasingly incorporate sensor suites, such as cameras, LiDAR, radar, and inertial measurement units, that are primarily designed for autonomous driving functions but can simultaneously serve infrastructure assessment purposes [106]. Soto et al. [96] conducted a comprehensive survey of road safety and traffic efficiency applications based on C-V2X technologies, demonstrating that the communication infrastructure for vehicle-to-infrastructure

data exchange is maturing rapidly. The combination of onboard sensing capabilities with V2X communication creates the technical preconditions for vehicles to function as distributed infrastructure monitors at no marginal hardware cost.

In the specific domain of pothole detection, computer vision approaches have demonstrated the feasibility of automated, vehicle-mounted defect identification. Asad et al. [11] deployed YOLO-family models on edge devices, achieving real-time detection at high accuracy across diverse road conditions. Rubin et al. [10] enhanced detection through attention mechanisms while also estimating pothole area for severity assessment. Mohan et al. [107] demonstrated early smartphone-based approaches using accelerometer data, while more recent works have combined multiple sensing modalities: Bej et al. [108] integrated computer vision with ultrasonic sensors for depth estimation, and Govindan et al. [109] fused accelerometer data with GPS for automatic report generation.

Despite these technological advances, existing detection systems operate within centralized data pipelines where detected defects are reported to proprietary servers controlled by single entities. The detection capability exists; what is missing is a trustless coordination layer that transparently connects detection events to municipal response workflows. The vehicle-based detection pathway specified in Chapter 7 addresses this gap by defining how computer vision models running on edge hardware submit cryptographically signed detection reports to the blockchain via the same meta-transaction pathway used by citizen reporters, integrating vehicle sensing into the decentralized coordination framework.

This subsection completes the review's coverage of the connected vehicle in two complementary roles: as an asset whose identity must be tracked across its lifecycle (Section 3.1), and as a civic actor whose sensing capabilities contribute to public infrastructure management (this subsection). These two roles correspond, respectively, to the vehicle-as-record framing of Chapter 6 and the vehicle-as-sensor framing of Chapter 7. Together they illustrate why the four contributions of this thesis, though addressing different problems, are fundamentally connected.

The smart city and infrastructure literature reviewed in this section confirms that blockchain technology is widely recognized as a promising foundation for transparent urban governance, but that practical implementations, particularly for road infrastructure management, remain scarce. The few existing blockchain-based sys-

tems address only partial aspects of the pothole management challenge, and no prior work integrates the full set of capabilities (token incentives, algorithmic prioritization, duplicate detection, vehicle integration, and IPFS storage) that a comprehensive decentralized platform requires. The next section synthesizes the gaps identified across all four preceding domains and positions the contributions of this thesis within the resulting landscape.

3.4.4 Tokenised Incentives for Civic Participation

The intersection of the two preceding strands (tokens issued by or on behalf of a public authority, earned through civic contribution and redeemable against public services or participation rights) is the body of work that most directly motivates the incentive mechanism of Chapter 7, and is treated separately here.

The theoretical case rests on the observation that a token performs several functions at once. Rozas et al. [110] identify tokenisation as the first of six blockchain affordances for self-governing communities, arguing that a token can encode recognition of contribution, entitlement to a common resource, and a right to participate in collective decisions without these being separable as money and voting rights are. Pazaitis et al. [111] decompose value in commons-based production into its production, its *record*, and its actualisation, and argue that distributed ledgers are principally an innovation in the second: a decomposition that maps onto Chapter 7 directly. Empirical work remains dominated by pilots: Tan and Rodriguez Müller [112] find that institutional and organisational factors shaped the Barcelona participation pilot at least as strongly as the technology did, while Kassen [113] surveys purpose-built participation tokens across several cities. One distinction this literature does not consistently preserve is worth noting: between a credential that *authorises* participation and a token that *rewards* it.

Whether token rewards actually elicit contribution is an empirical question on which the evidence divides. Ballandies [114] reports a randomised controlled trial ($n = 132$, 2×2 factorial) confirming motivational crowding-out for blockchain tokens specifically, rather than by analogy from the broader incentives literature. Raban [115], analysing four years and some 52,000 contributions in an online information market, finds the opposite: economic incentives serve as enticement while social incentives sustain participation and are associated with higher eventual

gains, so that such markets are catalysed by social activity rather than cannibalised by it. Chen et al. [116], in a quasi-experimental study of roughly 98,000 users of a token rewarded platform, find that ownership-conferring tokens increase contribution effort but reduce its novelty, and that the incentive effect diminishes over time. Crowding-out is therefore contingent on how tangible and intangible incentives are combined rather than on the presence of a token as such, and a token incentive requires periodic recalibration rather than being a one-off design decision.

A concern orthogonal to motivation, raised by Kassen [113], is that decentralised participation can deepen existing inequalities: participation presupposes a smartphone, a data connection and discretionary time, all correlated with income. Where reports determine the allocation of repair budgets, the priority queue risks reflecting who reported rather than where the defects are.

3.5 Research Gaps

The preceding four sections have surveyed the state of the art across the research domains relevant to this thesis: blockchain in the automotive industry (Section 3.1), supply chain management and digital product passports (Section 3.2), incentive mechanisms (Section 3.3), and smart cities and infrastructure (Section 3.4). This section synthesizes the findings, first by identifying the specific gap that motivates each contribution, and then, more importantly, by examining the cross-cutting patterns that emerge when these gaps are considered collectively.

Four specific research gaps emerge from the literature reviewed in this chapter, each corresponding to a contribution of this thesis. Table 3.3 maps each gap to the literature section where it was identified, the research question it motivates, and the chapter that addresses it.

Gap 1 - Privacy-preserving supplier evaluation. The supply chain literature (Section 3.2.2) has produced a rich set of methodologies for supplier evaluation, spanning MCDM frameworks, machine learning models, and blockchain-based transparency systems. However, all existing approaches assume that evaluators have access to the raw supplier data, or at best protect it through conventional encryption that prevents computation on encrypted values. No prior work enables a purchasing

Table 3.3 Summary of research gaps, their sources in the literature review, and the corresponding thesis contributions.

Gap	Description	Lit. Section	RQ	Ch.
G1	No mechanism for verifiable supplier evaluation without disclosing sensitive data	3.2.2, 3.2.3	RQ1	4
G2	No systematic analysis of blockchain alignment with EU Battery Regulation for DPPs	3.2.4	RQ2	5
G3	No NFT-based vehicle lifecycle system with integrated marketplace and compliance enforcement	3.1.1, 3.1.3	RQ3	6
G4	No comprehensive decentralized pothole management system with token incentives, prioritization, and vehicle integration	3.4.2, 3.4.3	RQ4	7

company to evaluate a supplier's sustainability from verifiable, tamper-proof indices *without* accessing the underlying KPIs. The privacy-preserving approaches surveyed in Section 3.2.3 include zero-knowledge proofs (suited to binary compliance verification rather than quantitative index computation) and verifiable credentials (which separate ownership from access but do not enable on-chain computation). Fully homomorphic encryption on blockchain, the approach adopted in Chapter 4, occupies a unique and previously unexplored position in this design space.

Gap 2 - Regulatory alignment for digital battery passports. The digital product passport literature (Section 3.2.4) has established blockchain as a promising infrastructure for DPPs, and several prototype implementations have been proposed using different technology stacks (RFID, IOTA, Gaia-X). However, no prior work has conducted a provision-by-provision analysis of the EU Battery Regulation to demonstrate that a specific technology systematically satisfies all regulatory requirements. This gap between regulatory ambition and technical validation creates uncertainty for the industry and risks fragmented implementations. Chapter 5 fills this gap through a systematic mapping that also serves a structural role within the thesis, establishing that the architectural patterns implemented in the engineering contributions are aligned with the direction of EU policy.

Gap 3 - NFT-based vehicle lifecycle management. The vehicle identity literature (Sections 3.1.1–3.1.3) has explored both registry-based and token-based approaches to representing vehicles on the blockchain. However, existing systems address individual aspects of the problem, odometer fraud prevention, history recording, or marketplace functionality, without providing an integrated lifecycle management framework. As the comparison in Table 3.1 demonstrates, no prior work combines NFT-based vehicle representation with a multi-stakeholder update workflow, compliance enforcement through a locking mechanism, and a decentralized marketplace with price negotiation and automatic ownership transfer. Chapter 6 addresses this gap with a complete system.

Gap 4 - Comprehensive decentralized pothole management. The smart city and infrastructure literature (Section 3.4.2) has produced blockchain-based systems that address partial aspects of the pothole management challenge: blockchain storage without incentives (Rumpa et al.), priority scheduling without tokens or vehicle integration (Matzutt et al.), or AI detection without prioritization or duplicate handling (Mubarak et al.). Commercial platforms offer deployment maturity but lack blockchain-based transparency. No existing system integrates all necessary capabilities, such as token incentives, algorithmic prioritization, spatial duplicate detection, vehicle integration, and hybrid IPFS/blockchain storage, within a unified decentralized framework. Chapter 7 addresses this gap with the most comprehensive of the four contributions.

The four gaps are not isolated, but the case that they are manifestations of a single deficiency, that they share a recurring architectural template, and that the four contributions of this thesis traverse a coherent privacy–transparency design space: these are claims that require the engineering contributions of Chapters 5 and 7 as evidence and are developed as the synthesis of Chapter 8. This chapter has done its job: it has surveyed the state of the art across the four research domains and identified the specific gaps that motivate each contribution. The four chapters that follow present the contributions in lifecycle order, supply chain (Chapter 4), component traceability (Chapter 5), vehicle identity (Chapter 6), and infrastructure interaction (Chapter 7), and Chapter 8 returns to the cross-cutting picture they jointly support.

Chapter 4

Privacy-Preserving Supplier Evaluation on Blockchain

This chapter is based on the following publication:

A. Butera, N. Romani, S. Meneguzzo, and V. Gatteschi, "A Privacy-Preserving Blockchain Framework for Sustainable Supplier Evaluation in the Textile Industry," in *Proceedings of the 2025 International Conference on Information Technology for Social Good (GoodIT '25)*, Antwerp, Belgium, Sep. 2025, pp. 24–32. doi: 10.1145/3748699.3749769.

4.1 Chapter Introduction

This chapter addresses the supply chain opacity trust gap (Gap 1, 1.2). The CSDD directive [5] requires purchasing companies to evaluate the sustainability performance of their suppliers, but this conflicts with suppliers' legitimate interest in withholding KPIs that may reveal proprietary processes. Existing approaches to supplier evaluation, such as multi-criteria decision-making, machine learning, and blockchain-based transparency systems alike, assume the evaluator can read the raw KPIs in plaintext (Section 3.2.2). The framework presented here resolves this tension by integrating fully homomorphic encryption (FHE) with a public Ethereum-compatible blockchain. Suppliers submit their sustainability KPIs in encrypted form to a smart contract deployed on an FHE-enabled blockchain (fhEVM); the contract

computes a composite sustainability index entirely on the ciphertexts, without ever decrypting the underlying values; and the resulting index is published on-chain, where any purchasing company, regulator, or auditor can read and compare it. The framework thus produces a public output (the sustainability score) from private inputs (the KPIs), without requiring trust in any intermediary and without relying on Non-Disclosure Agreements. Its architecture is KPI-agnostic, which means that the contract operates on n encrypted numerical values with configurable weights, so an automotive instantiation (battery-cell suppliers evaluated on conflict-mineral, water-consumption, or recycled-content KPIs) reuses the same logic. The textile validation domain that grounds the prototype is justified in Section 4.3.2.

4.2 Related Work

Chapter 3 provides a comprehensive review of the supplier evaluation and privacy-preserving supply chain literature. This section does not replicate that analysis but instead positions this contribution against the three categories of work most directly relevant to it, drawing on the detailed surveys of Sections 3.2.2 and 3.2.3.

The first category comprises supplier evaluation methodologies that assume access to raw supplier data. Multi-criteria decision-making (MCDM) frameworks such as those proposed by Ecer et al. [66] (LMAW–TOPSIS), Hendiani and Walther [67] (interval-valued intuitionistic fuzzy methods), and Sithi et al. [68] (SWARA–TOPSIS for textile suppliers) provide increasingly sophisticated ranking mechanisms, but all require that the evaluating company can read the supplier's performance values in plaintext. Machine learning approaches, including the K-means/Random Forest pipeline of Abbasian et al. [71], face the same prerequisite. The contribution of this chapter is orthogonal to these methodologies: it does not propose a new ranking algorithm but rather a new *data access model* that could, in principle, underpin any of them. The weighted sum used here for the sustainability index computation is deliberately simple; more complex MCDM or ML algorithms could be implemented on encrypted data as the computational capabilities of FHE mature, a direction discussed in Section 4.7.

The second category comprises blockchain-based supplier evaluation systems. Haffar and Özceylan [73] proposed a blockchain system for sustainable and lean-agile supply chains that includes economic, environmental, social, and leagility criteria

alongside a mutual reputation mechanism. Vaezi et al. [69] combined MCDM with blockchain to enhance transparency and traceability. Modares et al. [74] used blockchain to improve information sharing among supply chain stakeholders. These systems leverage blockchain's transparency and auditability, but they still require suppliers to disclose their performance data, either on-chain or to an evaluator who then records a result on-chain. They thus address the *integrity* dimension of the trust gap (ensuring that evaluation data is tamper-proof) without addressing the *confidentiality* dimension (ensuring that sensitive data is not exposed). The framework presented in this chapter addresses both dimensions simultaneously.

The third category comprises privacy-preserving approaches for supply chain data. As reviewed in Section 3.2.3, Sahai et al. [76] combined blockchain with zero-knowledge proofs (ZKPs) to enable privacy and traceability. ZKPs are well suited to proving binary compliance predicates (e.g., "this supplier's carbon emissions are below a regulatory threshold") but are not designed for computing quantitative indices from multiple encrypted inputs. Fang et al. [77] proposed a verifiable credential framework for industrial data trading that separates data ownership from data access, but does not enable computation on the data itself. Berger et al. [75] demonstrated that data science techniques can preserve confidentiality in digital product passport data, but focused on data anonymization rather than on enabling evaluation from encrypted values. this contribution occupies a distinct position in this landscape: fully homomorphic encryption enables the smart contract to perform *arbitrary arithmetic* on encrypted inputs, producing a quantitative sustainability index without ever accessing the plaintext KPIs. Among the surveyed works, no prior system combines FHE with blockchain smart contracts for the specific purpose of supplier sustainability evaluation.

Table 4.1 summarizes the positioning of this contribution along three dimensions: whether the system provides transparency/auditability of the evaluation process, whether it preserves the confidentiality of the supplier's raw data, and whether it enables quantitative index computation from private inputs.

4.3 Background: Textile Industry Use Case

This section provides the domain-specific context that grounds the framework developed in the remainder of the chapter. Section 4.3.1 revisits the regulatory moti-

Table 4.1 Positioning of the proposed framework against related work.

Approach	Transparent evaluation process	Confidential supplier data	Quantitative index from private inputs
MCDM methods [66–68]	No	No	Yes
ML methods [71]	No	No	Yes
Blockchain + evaluation [73, 69]	Yes	No	Yes
ZKP-based [76]	Yes	Yes	No[a]
Verifiable credentials [77]	Partial	Yes	No
This work (FHE + blockchain)	**Yes**	**Yes**	**Yes**

[a] ZKP-based approaches can prove binary compliance statements but do not compute quantitative indices from encrypted inputs.

vation for sustainable supplier evaluation, focusing on the specific obligations that the CSDD directive imposes on purchasing companies. Section 4.3.2 justifies the choice of the textile industry as the validation domain and introduces the structural characteristics of its supply chain that make the tension between transparency and confidentiality particularly acute. Section 4.3.3 presents the sustainability KPI taxonomy adopted in this work, following the two-category decomposition proposed by Alves et al. [2].

4.3.1 The CSDD Directive and Compliance Requirements

The Corporate Sustainability Due Diligence Directive (EU 2024/1760), which entered into force on 25 July 2024, constitutes the most recent and far-reaching regulatory instrument on supply chain sustainability in the European Union [5]. The directive imposes on large companies operating in the EU an obligation to conduct human rights and environmental due diligence across their "chain of activities," which encompasses not only direct suppliers but also, to the extent reasonably knowable, indirect upstream partners. A comprehensive introduction to the CSDD is provided in Section 2.7.3; this section focuses on the three specific obligations that motivate the framework presented in this chapter.

Identification of adverse impacts. Article 8 of the directive requires companies to identify actual and potential adverse human rights and environmental impacts arising from their own operations and from those of their business partners across the

chain of activities. In practice, this obligation translates into a structured evaluation of supplier performance against a set of sustainability indicators, because impacts cannot be identified without measurable data from the upstream actors.

Preventive and corrective measures. Articles 10 and 11 require companies to take appropriate measures to prevent, mitigate, or remedy the identified impacts. The selection of business partners (i.e. supplier selection), the establishment of contractual cascading of sustainability requirements, and the temporary or permanent suspension of relationships with non-compliant suppliers are all explicit instruments envisioned by the directive. Each of these actions requires the evaluating company to possess verifiable information about supplier performance at the moment the decision is made.

Accountability and public reporting. Article 16 requires companies to publish an annual report on their due diligence practices. The credibility of such reports depends on the traceability and tamper-resistance of the underlying data, a property that aligns naturally with blockchain-based record keeping.

The three obligations above create what can be termed a *bidirectional data-access problem*. The purchasing company must access supplier data to fulfill identification (Article 8) and selection (Articles 10–11) duties; the supplier, however, is neither willing nor, under competition and trade-secret law, entirely free to share detailed operational data, particularly with large buyers who are simultaneously suppliers' most valuable customers and their most capable potential competitors through vertical integration. Existing industry practice addresses this conflict through Non-Disclosure Agreements (NDAs) and trusted third-party auditors. Both mechanisms are costly, slow, and ultimately still require that the raw data be disclosed to some party, shifting trust rather than eliminating the need for it. The framework presented in this chapter takes a different approach: by performing the evaluation directly on encrypted data, it makes the disclosure problem disappear at its source.

4.3.2 The Textile Industry as Validation Domain

The validation of the proposed framework was conducted within the MICS (Made in Italy Circular and Sustainable) Extended Partnership, a research program funded

under the Italian National Recovery and Resilience Plan (PNRR, Mission 4 Component 2)[1]. The MICS partnership focuses on sustainable supply chain practices in key Italian manufacturing sectors, with the textile and clothing industry identified as a priority domain. Three considerations motivated the selection of this domain for this work.

First, the textile sector is among the most significant contributors to environmental and social externalities at the global scale. It ranks among the top industries as water consumption, chemical pollution of freshwater bodies, greenhouse gas emissions from energy-intensive processes such as dyeing and finishing, and labor rights concerns in developing-country production hubs. These externalities place the sector at the center of the CSDD's regulatory scope, which means that the framework's motivation directly applies to its validation domain rather than being transplanted from an unrelated context.

Second, the structural characteristics of textile supply chains make them an unusually clear illustration of the transparency–confidentiality tension identified in Section 4.3.1. A typical garment traverses a long and geographically distributed chain: cotton cultivation, ginning, spinning, weaving, dyeing, finishing, cutting, assembly, and finishing operations may each occur in a different country, often separated by multiple tiers of subcontracting. Each intermediate actor possesses detailed performance data that would be highly revealing to both regulators (who need it for due diligence) and competitors (who could use it to reverse-engineer processes or undercut margins). No current mechanism enables sustainability verification without exposing this data.

Third, and from a practical standpoint, the MICS partnership provided access to a validated KPI taxonomy developed specifically for the textile and clothing value chain by Alves et al. [2]. This taxonomy is presented in the following subsection and adopted as the KPI set for the prototype.

It is important to reiterate the point made in Section 4.1: the framework's architecture is KPI-agnostic. The choice of the textile domain determines *which* indicators are measured, but not *how* they are aggregated into an index or *how* the aggregation is performed on encrypted data. An automotive instantiation, targeting,

[1]MICS project, funded by the European Union Next-GenerationEU, D.D. 1551.11-10-2022, PE00000004. The manuscript reflects only the authors' views; neither the European Union nor the European Commission can be considered responsible for them.

for instance, cell-manufacturing suppliers of Tier-1 battery pack integrators, would reuse the entire on-chain and cryptographic infrastructure, substituting only the KPI vector and the weight configuration. The textile validation thus demonstrates the framework in the domain for which the KPI taxonomy was readily available, while preserving applicability to the automotive supply chains that motivate the thesis as a whole.

4.3.3 Sustainability KPIs and the Two-Index Decomposition

The KPI taxonomy adopted in this work follows the structure proposed by Alves et al. [2] for the textile and clothing value chain. The authors of that work partition sustainability indicators into two categories that reflect complementary dimensions of supplier performance:

- **Environmental and Circular Score (ECS).** This category captures indicators associated with the environmental footprint and circular-economy behavior of a specific product or production batch. Relevant activities include logistics (transportation distance and modality) and production processes (spinning, weaving, dyeing, finishing, and garment assembly). The ECS is *batch-specific*: a single supplier may produce multiple batches with different ECS values depending on the resources consumed and the materials recovered for each.
- **Economic and Social Score (ESS).** This category captures company-level economic and social parameters. Relevant indicators include total headcount, gender distribution across functions, salary parity, contract types, working-time arrangements, training activities, financial stability, certifications held by the company, and voluntary social initiatives. The ESS is *company-specific*: it characterizes the supplier as an organization rather than a particular output.

The two-category structure is not merely a taxonomic convenience. It reflects a substantive difference in the temporal and referential granularity of the two families of indicators: ECS values are updated for every new production batch and should therefore be aggregated across batches when a single score per supplier is required, while ESS values are updated on a longer cadence (typically annual) and refer to the organization as a whole. This asymmetry is reflected directly in the design of the smart contract (Section 4.4): the per-supplier Circular Economy & Environmental

Sustainability Index (CEI) is computed as the arithmetic average of the per-batch ECS contributions, while the Social & Economic Sustainability Index (SEI) is updated as a single value per reporting period from the ESS indicators.

From the full taxonomy proposed by Alves et al., a subset of 13 indicators was selected as the benchmark KPI set for the prototype. Table 4.2 summarizes the selected indicators, grouped by category.

Table 4.2 Subset of textile-sector KPIs adopted for the prototype, grouped by category. The taxonomy follows Alves et al. [2].

#	Category	KPI
1	ECS	Water consumption
2	ECS	Quantity of recycled water
3	ECS	Electricity consumption
4	ECS	Self-consumption of renewable energy
5	ECS	Natural gas consumption
6	ECS	Steam consumption
7	ESS	Total number of employees
8	ESS	Total number of women
9	ESS	Total number of men
10	ESS	Number of women in non-top-management functions
11	ESS	Number of men in non-top-management functions
12	ESS	Average salary of women in non-top-management functions
13	ESS	Average salary of men in non-top-management functions

Two properties of this selection deserve explicit mention. First, the indicators are *quantitative* and *numerical*: each KPI reduces to a number that can be represented as an encrypted unsigned integer in the smart contract, which is a precondition for homomorphic aggregation. Qualitative indicators (e.g. presence or absence of a specific certification) are represented as binary-encoded numbers when retained. Second, the selection is deliberately *limited*, as the goal of the prototype is to demonstrate the feasibility of privacy-preserving aggregation rather than to provide an exhaustive sustainability assessment. A production deployment would extend the KPI set to cover the full spectrum of ESG indicators required under the CSDD; the framework's modularity supports this extension without architectural changes, subject to the scalability limits of FHE that are quantified in the evaluation (Section 4.6).

The composite sustainability index (SI) that the smart contract ultimately publishes is defined in Section 4.4.4, after the system architecture and the encryption model have been introduced. At this stage, it suffices to state that the SI is a weighted combination of the supplier-level CEI and SEI, computed entirely on encrypted inputs.

4.4 Proposed Framework

This section presents the proposed framework in four progressively specific steps. Section 4.4.1 introduces the system architecture, identifying the three principal components (web application, smart contract, and fhEVM-enabled blockchain), the actors that interact with them, and the data flows between them. Section 4.4.2 focuses on the cryptographic layer, explaining how the Zama fhEVM infrastructure enables computation on encrypted KPIs and clarifying the trust assumptions that this architecture carries. Section 4.4.3 describes the smart contract design, covering the data model, the main functions, and the access-control policy that distinguishes plaintext from ciphertext data. Section 4.4.4 formalizes the sustainability index computation as a composition of two category-specific indices and makes explicit the mapping between the mathematical definition and its homomorphic realization on-chain.

4.4.1 System Architecture

The framework is organized around three components and four categories of actors. Figure 4.1 depicts the overall architecture; each element is described below.

Actors. Four categories of actors participate in the system.

- **Suppliers.** A supplier is an upstream actor in the supply chain that produces goods or components and is subject to sustainability evaluation. Suppliers register on the platform, upload datasets of sustainability KPIs, and control access to their own raw KPI values.
- **Purchasing companies (brands, retailers, OEMs).** These are downstream actors that use the sustainability indices published on the blockchain to make

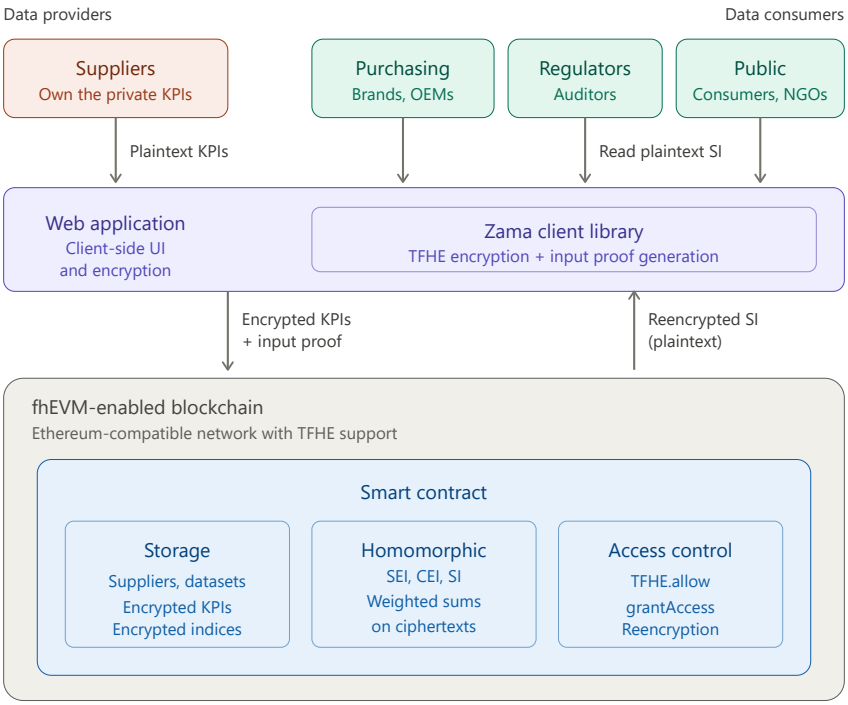

Fig. 4.1 System architecture of the privacy-preserving supplier evaluation framework. Suppliers submit encrypted KPIs through a web application; the smart contract, deployed on an fhEVM-enabled blockchain, computes the sustainability index directly on the ciphertexts. Purchasing companies read the resulting index without accessing the underlying data.

supplier-selection and due-diligence decisions. In CSDD terminology, they are the companies subject to due-diligence obligations.

- **Regulators and auditors.** Although not required to interact actively with the smart contract in the current implementation, regulators and third-party auditors are beneficiaries of the system: they can query the on-chain sustainability indices and the publicly visible computation logic to verify compliance without issuing individual data requests.

- **General public.** Consumers, non-governmental organizations, and journalists can also read the published sustainability indices, supporting the public-accountability function that is increasingly expected of sustainable supply chain reporting.

Components. Three components implement the system.

- **Web application.** The web application is the interaction front-end. It provides a graphical interface for supplier registration, dataset upload, access-grant management, and sustainability-index consultation. Crucially, the web application also performs *client-side encryption*: the numerical KPI values entered by the supplier are encrypted in the browser using the Zama library before being transmitted in any form, so that no plaintext ever leaves the supplier's device. The web application also generates the zero-knowledge input proof that the fhEVM requires to attest to the well-formedness of the ciphertexts submitted to the contract. The centralization implications of this architectural choice are discussed in Section 4.7.
- **Smart contract.** The smart contract is deployed on an fhEVM-enabled Ethereum-compatible blockchain. It stores the supplier registry, the encrypted datasets, and the encrypted category indices, and exposes the functions that implement the framework's business logic (Section 4.4.3). All computations on encrypted values are executed by the fhEVM within the contract's transactional scope; no external computation service is required.
- **fhEVM-enabled blockchain.** The underlying blockchain must support the fhEVM extension (Section 2.5.2), which augments the standard EVM with encrypted data types and homomorphic operations. The Zama-maintained fhEVM deployment on Ethereum's Sepolia testnet was used for the prototype; any fhEVM-compatible network can host the smart contract without modification.

Data flow. A complete evaluation cycle proceeds as follows. (1) The supplier registers on the platform, producing an on-chain `Supplier` record keyed by the supplier's wallet address. (2) The supplier creates one or more datasets: an annual dataset of ESS indicators for the SEI computation, and one dataset per production batch of ECS indicators for the CEI computation. Each dataset is encrypted client-side before submission. (3) Upon receipt, the smart contract computes the per-dataset score (homomorphically) and updates the supplier's SEI or CEI index (the CEI is maintained as a running arithmetic average across all batch datasets submitted by that supplier). (4) The supplier invokes the index-aggregation function, which computes

the overall Sustainability Index (SI) as a weighted combination of SEI and CEI. The aggregate SI is the single value that is exposed in plaintext to external readers; all intermediate values remain encrypted on-chain. (5) Purchasing companies read the SI of any registered supplier by invoking a read function that requires no permissions.

This architecture deliberately collapses the workflow into a small number of transactions per supplier, because on-chain homomorphic operations are orders of magnitude more expensive than plaintext arithmetic. The cost analysis in Section 4.6 quantifies this directly.

4.4.2 Homomorphic Encryption Layer

The cryptographic machinery underlying the framework is Zama's fhEVM, introduced in Section 2.5.2. This subsection focuses on the elements of fhEVM that are material to the contract design.

Encrypted data types. The fhEVM exposes a set of encrypted integer types (`euint8`, `euint16`, `euint32`, and `euint64`) that can be stored in contract storage and manipulated using overloaded arithmetic and comparison operators. Each encrypted value is a ciphertext under the TFHE scheme, encrypted against the network's public key and decryptable only through the threshold Key Management Service (KMS) operated by the fhEVM infrastructure. In this implementation, KPI numerical values and the derived per-category indices are stored as `euint32` values, which provide sufficient range for the KPI magnitudes encountered in the textile use case while minimizing ciphertext storage cost.

Client-side encryption and input proofs. When a supplier submits a dataset, the web application invokes the Zama client-side library to encrypt each numerical KPI. The library returns, for each input, an encrypted value and a zero-knowledge proof attesting that the ciphertext is a well-formed encryption of a value within the declared integer range. The input proof is essential: without it, an attacker could craft malformed ciphertexts whose homomorphic manipulation would yield ill-defined results or trigger panics in the contract. The smart contract verifies the input proofs upon receipt; this verification is transparent to the application developer, handled

by the `TFHE.asEuint32(...)` family of functions provided by the Zama Solidity library.

Homomorphic operations. Once ciphertexts are stored in contract storage, the smart contract can perform addition, subtraction, multiplication, and comparison operations on them using standard Solidity syntax (e.g. `a + b` where `a` and `b` are `euint32` values). These operations execute on the ciphertexts and produce new ciphertexts; no intermediate plaintext is ever materialized on-chain. The TFHE scheme's bootstrapping property ensures that noise does not accumulate to the point of corrupting the final result, regardless of the number of sequential operations.

Access control for decryption. A ciphertext on-chain is useless without the ability to decrypt it, and the fhEVM deliberately separates *storage of ciphertext* from *permission to decrypt*. Zama's Solidity library exposes two primitives that the contract uses to manage this separation: `TFHE.allow(ciphertext, address)`, which grants a specific address the right to request decryption of a specific ciphertext, and `TFHE.allowThis(ciphertext)`, which grants the contract itself the right to operate on the ciphertext in subsequent calls. The framework uses these primitives as follows:

- Upon dataset upload, the contract calls `allowThis` on each encrypted KPI so that subsequent homomorphic operations can use them, and `allow` with the supplier's own address so that the supplier retains the ability to decrypt its own data.
- The final Sustainability Index is published via a dedicated function that performs *reencryption*, a Zama primitive that re-encrypts the stored ciphertext under a one-time public key provided by the requester, yielding a plaintext SI visible to any reader. This allows public consumption of the SI while keeping the underlying ciphertexts inaccessible.
- The `grantAccess` function exposed to suppliers allows them to selectively extend `allow` rights for specific raw KPI ciphertexts to chosen counterparties (e.g. a long-term industrial customer performing enhanced due diligence).

Trust boundaries. The fhEVM architecture introduces three distinct trust assumptions that a rigorous evaluation of the framework must acknowledge. First, the client-side web application is a trusted component for the supplier: if it is compromised, plaintext KPIs could be exfiltrated before encryption. Second, the threshold KMS, which collectively holds the network's secret key, is a distributed but non-decentralized authority: a collusion of a threshold number of key holders could in principle decrypt any on-chain ciphertext. Third, the correctness of homomorphic operations and input-proof verification depends on the correctness of the fhEVM implementation itself, which at the time of writing is a relatively young infrastructure. These trust assumptions are the principal subject of Section 4.6.3 and inform the limitations discussed in Section 4.7.

4.4.3 Smart Contract Design

The smart contract is the locus of the framework's business logic. This subsection describes its data model and its public interface; the deployment details (compiler version, library dependencies, network configuration) are deferred to Section 4.5.

Data model

The contract maintains three categories of state variables, organized along the plaintext/ciphertext distinction.

- **Supplier records.** A mapping from a supplier's wallet address to a `Supplier` struct. The struct stores the overall Sustainability Index (`SI`), the per-category indices (`SEI`, `CEI`), and the arrays of dataset-level scores that feed into those indices. Except for the wallet address itself and a boolean `isSupplier` flag, every field of the `Supplier` struct is encrypted. The overall SI is encrypted at storage but exposed in plaintext via reencryption.
- **ESS datasets.** A mapping from dataset identifier to an `EssDataset` struct containing the supplier ID (plaintext), the KPI names (plaintext), the encrypted array of numerical KPI values, and the encrypted dataset-level SEI contribution.

- **ECS datasets.** A mapping from dataset identifier to an `EcsDataset` struct analogous to the ESS case, with the addition of a plaintext production-batch ID, the field that distinguishes batch-specific CEI contributions from company-level SEI contributions.

Table 4.3 summarizes the plaintext/ciphertext decomposition. The rationale for this split is the tension described in Section 4.3.1: identifiers and KPI names are public so that the *structure* of the computation is fully auditable, while numerical values and derived indices are encrypted so that sensitive operational data is not disclosed. Only the final aggregate SI is exposed in plaintext (via reencryption), because the SI is precisely the quantity the evaluating company needs and the minimum disclosure required to serve the purpose.

Table 4.3 Plaintext/ciphertext decomposition of the smart contract state.

Data cluster	**Field**	**Storage**
Supplier	Wallet address	Plaintext
Supplier	`isSupplier` flag	Plaintext
Supplier	Overall Sustainability Index (SI)	Encrypted[a]
Supplier	SEI index	Encrypted
Supplier	CEI index	Encrypted
Supplier	Array of SEI dataset scores	Encrypted
Supplier	Array of CEI dataset scores	Encrypted
ESS dataset	Supplier ID	Plaintext
ESS dataset	KPI names	Plaintext
ESS dataset	KPI numerical values	Encrypted
ESS dataset	Dataset-level SEI contribution	Encrypted
ECS dataset	Supplier ID	Plaintext
ECS dataset	Production batch ID	Plaintext
ECS dataset	KPI names	Plaintext
ECS dataset	KPI numerical values	Encrypted
ECS dataset	Dataset-level CEI contribution	Encrypted

[a] The overall SI is stored encrypted but exposed in plaintext to any reader via the reencryption primitive.

Public interface

The contract exposes seven public functions that together implement the evaluation workflow. Table 4.4 lists them; the ones marked with an asterisk are restricted by the contract's access-control logic.

Table 4.4 Principal functions of the smart contract.

Function	Caller	Purpose
`addSupplier`	Anyone[b]	Register a new supplier; initialize `isSupplier` flag and empty score arrays.
`addDataset_SEI`	Supplier*	Upload an ESS dataset; compute the dataset-level SEI contribution; update the supplier's SEI.
`addDataset_CEI`	Supplier*	Upload an ECS dataset for a given production batch; compute the dataset-level CEI contribution; update the supplier's CEI as a running average.
`computeSustainabilityIndex`	Supplier*	Aggregate SEI and CEI into the overall SI using the configured weights.
`grantAccess`	Supplier*	Extend `TFHE.allow` rights for the supplier's own encrypted KPI values to a specified address.
`getNumericValues`	Authorized addresses	Retrieve plaintext KPI values via reencryption; succeeds only if the caller has been granted access.
`getSustainabilityIndex`	Anyone	Retrieve the supplier's plaintext SI via reencryption; no access control.

[b] The caller's address is recorded as the supplier's wallet; subsequent supplier-only functions are gated on this binding.

Two design choices deserve emphasis. First, supplier registration is permissionless: any wallet address can become a supplier by calling `addSupplier`. This is deliberate, the framework does not impose gatekeeping at the contract level, because such gatekeeping would reintroduce a centralized trust point. Identity verification, to the extent it is needed, is a policy layer above the contract; a purchasing company that reads an SI from the blockchain can use off-chain evidence to confirm that the wallet address indeed corresponds to a known supplier. Second, `getSustainabilityIndex` is deliberately open to anyone. The SI is the public output of the framework, and restricting its readability would defeat the transparency goal (Section 4.3.1).

4.4.4 Sustainability Index Computation

The overall Sustainability Index published by the contract is the composition of two category-specific indices, defined below. Throughout this subsection, n^{ESS} denotes the number of KPIs in an ESS dataset and n^{ECS} the number of KPIs in an ECS dataset (in the prototype, $n^{\mathrm{ESS}} = 7$ and $n^{\mathrm{ECS}} = 6$, following Table 4.2).

Dataset-level scores. For a given ESS dataset d with encrypted KPI values $x_1^{(d)}, \ldots, x_{n^{\mathrm{ESS}}}^{(d)}$ and per-KPI weights $w_1^{\mathrm{ESS}}, \ldots, w_{n^{\mathrm{ESS}}}^{\mathrm{ESS}}$, the dataset-level SEI contribution is defined as

$$\mathrm{SEI}_d \;=\; \sum_{i=1}^{n^{\mathrm{ESS}}} w_i^{\mathrm{ESS}} \cdot x_i^{(d)}. \tag{4.1}$$

For an ECS dataset b (associated with a specific production batch) with values $y_1^{(b)}, \ldots, y_{n^{\mathrm{ECS}}}^{(b)}$ and per-KPI weights $w_1^{\mathrm{ECS}}, \ldots, w_{n^{\mathrm{ECS}}}^{\mathrm{ECS}}$, the dataset-level CEI contribution is defined analogously:

$$\mathrm{CEI}_b \;=\; \sum_{i=1}^{n^{\mathrm{ECS}}} w_i^{\mathrm{ECS}} \cdot y_i^{(b)}. \tag{4.2}$$

Both sums are computed homomorphically by the smart contract using the fhEVM's `TFHE.add` and `TFHE.mul` operations. The per-KPI weights are stored in plaintext in the contract, reflecting the design choice that the *computation logic* (weights and structure) is public while the *inputs* (KPI values) are private.

Supplier-level indices. Given a supplier who has uploaded D^{ESS} ESS datasets and D^{ECS} ECS datasets, the supplier's category indices are

$$\mathrm{SEI} \;=\; \mathrm{SEI}_{d^\star}, \qquad d^\star = \text{most recent ESS dataset}, \tag{4.3}$$

$$\mathrm{CEI} \;=\; \frac{1}{D^{\mathrm{ECS}}} \sum_{b=1}^{D^{\mathrm{ECS}}} \mathrm{CEI}_b. \tag{4.4}$$

The asymmetry between (4.3) and (4.4) reflects the temporal granularity argument made in Section 4.3.3: the SEI is company-level and is updated as a single value per reporting period, while the CEI is batch-specific and is therefore averaged across batches to produce a single supplier-level value. The averaging in (4.4) is

implemented homomorphically via `TFHE.add` on the stored array of encrypted batch contributions, followed by `TFHE.div` by the (plaintext) number of batches.

Overall Sustainability Index. The overall SI is a weighted combination of the two category indices:

$$\mathrm{SI} \;=\; \alpha \cdot \mathrm{SEI} \;+\; \beta \cdot \mathrm{CEI}, \qquad \alpha + \beta = 1. \tag{4.5}$$

The weights α and β are stored in the contract as plaintext `plaintextSEIFactor` and `plaintextCEIFactor`, respectively. In the prototype, they were set to $\alpha = 0.6$ and $\beta = 0.4$, assigning slightly greater weight to social and economic sustainability. This choice was deliberate for the validation setting but is not intrinsic to the framework: the weights are contract parameters and can be updated by a designated administrator (or, in a more decentralized deployment, by a governance mechanism) without any modification to the computation logic or the encryption workflow.

On the choice of a linear model. The use of a weighted sum as the aggregation function is a deliberate simplification whose limitations are explicit. Weighted linear combinations cannot capture interactions between KPIs (for instance, the synergy between electricity consumption and self-consumption of renewable energy, which should depress the combined environmental impact more than a linear model would predict). More sophisticated MCDM methodologies, such as TOPSIS, VIKOR, PROMETHEE, produce more faithful composite scores but involve operations (notably division and non-polynomial functions) that are either prohibitively expensive or impossible in the current fhEVM. The choice of a linear model is therefore pragmatic: it exercises the FHE machinery at scale, produces an interpretable score, and leaves open the refinement to richer aggregation functions as fhEVM primitives mature. This limitation is revisited in Section 4.7.

4.5 Implementation and Deployment

Section 4.4 presented the framework at the design level: what the components do, how data flows between them, and what the smart contract computes. This section describes how the design was realized as a working prototype, detailing

the technology stack (Section 4.5.1), the contract development and deployment workflow (Section 4.5.2), and the front-end implementation that closes the loop with the supplier (Section 4.5.3). The evaluation of the resulting system, such as cost, security, and trust-model assessment, is the subject of Section 4.6.

4.5.1 fhEVM Integration

The prototype integrates three software layers: the smart contract layer (Solidity, Zama fhEVM library, OpenZeppelin), the client-side cryptographic layer (Zama JavaScript library), and the orchestration layer (Hardhat development environment). Table 4.5 summarizes the exact versions used in the prototype, as of the paper's submission (May 2025).

Table 4.5 Technology stack used for the prototype implementation.

Layer	Component	Version / reference
Language	Solidity	0.8.x[a]
FHE library (on-chain)	Zama `fhevm-solidity`	v0.5
Utility library	OpenZeppelin Contracts	4.9.x[a]
FHE library (client)	Zama `fhevmjs`	v0.5
Development framework	Hardhat	2.x
Target network	Ethereum Sepolia (fhEVM-enabled)	—
Cryptographic scheme	TFHE	(Torus FHE)

[a] The specific minor versions are those active in the Hardhat configuration at deployment time.

On-chain components. The `fhevm-solidity` library exposes the encrypted integer types (`euint8`, `euint16`, `euint32`, `euint64`) and the homomorphic operations used throughout the contract. The contract imports two core modules:

- `TFHE.sol`, which provides the encrypted data types and the operators `TFHE.add`, `TFHE.mul`, `TFHE.sub`, `TFHE.div`, `TFHE.lt`, `TFHE.eq`, and the `asEuint32(einput,bytes)` constructor that verifies an input proof and returns an encrypted integer handle.
- The `GatewayCaller.sol` module, which provides `TFHE.allow(ciphertext, address)` and `TFHE.allowThis(ciphertext)` for managing per-ciphertext access rights, and the `reencrypt` primitive used by the public read functions.

From OpenZeppelin, the contract inherits the `ReentrancyGuard` contract, whose `nonReentrant` modifier is applied to every state-changing function that performs external calls (notably the dataset-upload functions, which interact with the TFHE library). This is a defensive measure: although the external calls in question are to deterministic cryptographic primitives that cannot trigger reentrancy in practice, the standard mitigation is applied to minimize the attack surface and to satisfy the security analysis discussed in Section 4.6.2.

Client-side cryptography. The web application is a single-page JavaScript application that integrates the Zama `fhevmjs` library for client-side encryption. The relevant workflow is the following. First, on page load, the application fetches the network's public key from the fhEVM gateway and initializes the encryption context. Second, when the supplier submits a dataset, the application iterates over the KPI form inputs, invoking `fhevmjs.encrypt32(value)` on each numerical value to obtain the encrypted `euint32` ciphertext and the accompanying zero-knowledge input proof. Third, the application packages the array of ciphertexts and the input proof into a single transaction that invokes the corresponding `addDataset_*` function on the smart contract. The `asEuint32(einput, bytes)` invocation inside the contract verifies the proof; if verification fails, the transaction reverts.

A second cryptographic primitive is used on the read side. When a user invokes `getSustainabilityIndex` or `getNumericValues`, the web application generates a one-time ECDSA keypair, sends the public key to the contract as part of the call, and receives back a reencryption of the requested ciphertext under that public key. The application then decrypts the reencryption locally using the matching private key and displays the plaintext value in the GUI. The one-time keypair is discarded after use. This mechanism ensures that no persistent decryption key is ever exposed beyond the request in which it is used.

Trust assumptions introduced by the toolchain. Two trust dependencies are worth making explicit at this point; both are revisited in Section 4.6.3. First, the Zama KMS, which holds the network's decryption key under threshold secret sharing, must be trusted not to collude beyond the threshold. Second, the `fhevmjs` library executing in the supplier's browser must be trusted to perform encryption correctly and not to exfiltrate plaintext before encryption. The first dependency is inherent

to the current fhEVM architecture. The second is mitigated by the possibility of loading the library from a verifiable source (for example, by pinning its IPFS hash), but the prototype does not implement this mitigation.

4.5.2 Contract Development and Testnet Deployment

The contract was developed within the Hardhat framework, which provides local compilation, unit testing, and deployment scripting for Ethereum-compatible networks. The development workflow proceeded in three phases.

Local development. Initial development used Zama's local fhEVM simulator, which emulates the TFHE operations without invoking the real FHE backend. This allowed rapid iteration on the contract logic (data structures, function signatures, access-control checks) before exercising the full cryptographic stack. Unit tests written in JavaScript verified the correctness of each public function in isolation, using mock encrypted values.

Compilation and deployment. The contract was compiled with `solc` 0.8.x targeting the Paris EVM version (the last stable fork prior to the fhEVM's target network version at the time of deployment). The resulting bytecode was deployed to the fhEVM-enabled Sepolia testnet through a Hardhat deployment script. Sepolia was selected as the deployment target for four reasons: it is Ethereum's current official testnet; it is fully supported by the Zama fhEVM infrastructure; it closely mirrors mainnet behavior as gas pricing and EVM semantics; and it provides a stable environment for reproducible cost measurements.

End-to-end workflow testing. Once deployed, the contract was exercised through the full seven-function workflow described in Section 4.4.3: supplier registration, ESS-dataset upload, multiple ECS-dataset uploads (to exercise the CEI averaging), overall-index computation, cross-party access granting, and plaintext retrieval both by authorized and unauthorized callers. Each transaction's gas consumption and USD cost were recorded; the aggregate results are reported in Section 4.6.1. The workflow test also confirmed the expected access-control behavior: a call to `getNumericValues` from an address that had not been granted access reverted with

the expected error, while a call to `getSustainabilityIndex` from the same address succeeded (as the SI is deliberately public).

4.5.3 Front-End Web Application

The front-end is a decentralized web application (*dApp*) that provides the graphical interface between the supplier and the blockchain. It performs three distinct functions: user authentication through a browser wallet (`MetaMask`), client-side encryption of KPI inputs prior to transaction submission, and reencryption-based display of plaintext outputs (sustainability indices and, for authorized callers, raw KPI values).

The application is structured around a small number of views, supplier registration, ESS dataset creation, ECS dataset creation, grant-access management, and public index consultation, that map directly to the contract's public functions. Each view is a thin wrapper around a contract call: the user fills in the form, the application encrypts the inputs (for write operations) or generates the reencryption keypair (for read operations), and the resulting transaction is signed through MetaMask and broadcast to the Sepolia network.

Three design choices of the front-end are worth noting, because they are relevant to the cost and trust-model evaluation that follows. First, encryption is performed entirely in the browser: the `fhevmjs` library runs in the supplier's tab and never sends plaintext to any server. Second, the application does not maintain server-side state: all persistent data lives on-chain, and the application is effectively stateless. Third, the application relies on the browser wallet for transaction signing, meaning that the supplier's private key never leaves the wallet's secure enclave.

These choices align the front-end with the broader decentralization goals of the framework, but they also mean that the application code itself is an integral part of the trust boundary: a compromised front-end could exfiltrate plaintext KPIs before encryption, or request reencryptions it does not need. Section 4.6.3 returns to this observation when assessing the residual trust dependencies of the system as a whole.

4.6 Evaluation

The framework was evaluated along three dimensions. Section 4.6.1 reports the gas consumption of each smart contract function measured on the Sepolia testnet and translates the figures into a per-supplier economic cost, situating the result within the cross-contribution cost analysis of Chapter 8. Section 4.6.2 presents the results of the static-analysis security assessment, discusses the false-positive rate induced by the fhEVM library imports, and examines the residual risks that static analysis cannot detect. Section 4.6.3 closes the evaluation by consolidating the trust dependencies of the system in a single trust-model summary, which serves both as the basis for the limitations acknowledged in Section 4.7 and as a reference point for the privacy–transparency spectrum discussed in Chapter 8.

4.6.1 Cost Analysis

Every transaction on an Ethereum-compatible blockchain consumes *gas*, a unit of computational work whose per-unit price fluctuates with network congestion. The gas model is reviewed in Section 2.2.3. The absolute cost of a transaction in fiat currency is the product of three quantities: the gas units consumed by the transaction, the gas price (in gwei), and the ETH-to-fiat exchange rate at the time of execution.

Measurement methodology. Gas consumption of each function was measured directly from the Sepolia transaction receipts during end-to-end testing of the deployed contract. All measurements were taken on 26 May 2025, at which time the Sepolia gas price averaged 3 gwei and the ETH/USD exchange rate was \$2,550, yielding an effective per-unit cost of approximately $\$7.65 \times 10^{-6}$ per gas unit. These reference values are used consistently throughout the cost table below.

Per-function gas consumption. Table 4.6 reports the gas consumption of the principal contract functions measured during the workflow test. The functions are listed in the order in which a supplier would invoke them during a complete evaluation cycle.

The per-supplier workflow total aggregates one registration, one ESS dataset upload, one ECS dataset upload, and one sustainability-index computation, cor-

Table 4.6 Gas consumption and cost of the principal smart contract functions, measured on the Sepolia fhEVM testnet. Native and fiat costs are computed at the conditions prevailing on 26 May 2025 (3 gwei, ETH = $2,550); gas units are invariant across chains.

Operation	Gas units	Cost (ETH)	Cost (USD)	Caller
Contract deployment (one-off)	3,500,000	0.010500	26.78	Owner
`addSupplier`	158,031	0.000474	1.21	Supplier
`addDataset_SEI`	2,201,355	0.006604	16.84	Supplier
`addDataset_CEI`	2,122,877	0.006369	16.24	Supplier
`computeSustainabilityIndex`	390,460	0.001171	2.99	Supplier
`grantAccess`	264,532	0.000794	2.02	Supplier
`getSustainabilityIndex`	55,000	0.000165	0.42	Any reader
Per-supplier workflow total[a]	**5,137,255**	**0.015412**	**39.30**	—

[a] One registration, one ESS upload, one ECS upload, one index computation, one access grant.

responding to the minimum-viable evaluation cycle. Additional ECS dataset uploads (one per production batch) would add approximately $16.24 per batch; the `computeSustainabilityIndex` invocation at the end of each reporting period adds $2.99.

Structural analysis of the cost. The cost distribution reflects the dominant role of homomorphic operations in the gas profile. The two dataset-upload functions together account for ∼85% of the per-supplier workflow cost, because each one must: verify an input proof for every encrypted KPI, store several `euint32` ciphertexts in contract storage, and perform a homomorphic weighted sum over those ciphertexts. Each homomorphic operation is more expensive than its plaintext counterpart by two to three orders of magnitude; this overhead is inherent to TFHE and is the principal cost driver of the framework. By contrast, functions that do not perform homomorphic arithmetic, `addSupplier`, `grantAccess`, and the read functions, fall in the hundreds of thousands of gas units range, consistent with typical Solidity costs.

Comparison against non-private alternatives. A plaintext supplier evaluation contract without FHE would store raw KPI values directly and compute the weighted sum in standard Solidity arithmetic. For reference, an analogous contract implementing the same data model and computation without encryption would require approximately 150,000–200,000 gas per dataset upload (dominated by storage writes),

translating to roughly \$1.15–\$1.53 per dataset at the reference gas price. The encrypted variant is therefore approximately 10–15× more expensive per dataset than a transparent counterpart would be. The excess cost is the price of *on-chain* privacy preservation: an off-chain privacy solution (e.g., computing the index on a trusted server) would be cheaper but would reintroduce the trust dependency that the FHE approach is designed to eliminate.

On cross-network projection. Gas units are deterministic across EVM-compatible networks, which ordinarily enables cost projections onto alternative deployment targets such as Polygon, Arbitrum, and Optimism. In the case of this framework, however, such a projection is not straightforward: the contract relies on the fhEVM's encrypted integer types and homomorphic operators, which are available only on networks that run the fhEVM fork of the Ethereum execution client. At the time of writing, such networks are limited to the fhEVM-enabled Sepolia deployment operated by Zama and, in the roadmap, a dedicated fhEVM mainnet. Deployment on generic EVM Layer-2 networks would require those networks to integrate the fhEVM extension, which is an infrastructure-level development outside the scope of this contribution. This constraint is the most significant deployment-layer limitation of the framework and is discussed further in Section 4.7. The cross-network cost projection that this thesis presents for other contributions (Chapters 6 and 7) is therefore not meaningful for this system, and its absence should be read as a reflection of the current fhEVM deployment landscape rather than as an omission in the evaluation.

Economic viability in context. Despite the 10–15× overhead relative to a plaintext implementation, the absolute per-supplier cost of \$39.30 is favorable when compared to the alternative mechanisms that current industry practice uses to bridge the transparency–confidentiality gap. Conventional supplier evaluation workflows involve on-site audits (typically costing several thousand euros per supplier per audit cycle), third-party certification bodies, and legal costs associated with the negotiation and enforcement of Non-Disclosure Agreements. A one-time on-chain cost of approximately forty dollars per supplier evaluation, supporting repeated evaluations at a marginal cost limited to the dataset re-uploads, is competitive with these baselines. This point is revisited in Chapter 8 within the comparative cost analysis across all four contributions.

4.6.2 Security Analysis

The security of the contract was assessed using Slither [41], the static analysis framework adopted consistently across all contributions (Section 2.6.2). Slither was executed against the full compilation unit including the Zama `fhevm-solidity` library imports, because separating the user contract from the library is not possible without modifying the Zama source tree. The analysis consequently covers 17 contracts in aggregate, totaling 9,638 source lines of code; the majority of the analyzed material is Zama library code that the prototype depends upon but does not modify.

Aggregate results. Table 4.7 summarizes the Slither findings by severity class.

Table 4.7 Slither static analysis results on the full compilation unit (user contract plus Zama fhEVM library dependencies, 17 contracts, 9,638 SLOC).

Severity	Count	Notes
High	0	No critical vulnerabilities identified.
Medium	11	Reentrancy warnings on TFHE external calls (see below).
Low	1	Informational style issue, no security impact.
Informational	70	Mostly library-code observations.
Optimization	1	Unrelated to the framework's security posture.

Analysis of medium-severity findings. All 11 medium-severity findings report potential reentrancy vulnerabilities on functions that make external calls to the Zama TFHE library, specifically, calls to `TFHE.add`, `TFHE.mul`, `TFHE.asEuint32`, and `TFHE.allow`. Slither's detector treats any external call to a non-inherited contract as a potential reentrancy vector, because a generic attacker-controlled contract at the called address could re-enter the calling contract through a callback. In this case, however, these calls are directed at Zama's TFHE executor, which is a fixed on-chain library whose functions are deterministic cryptographic operations. Zama's implementation does not contain the callback mechanisms or arbitrary-code-execution paths on which a reentrancy attack would depend. The warnings are therefore false positives specific to Slither's conservative detection heuristic.

Two defensive measures nonetheless reinforce the conclusion. First, every state-changing function of the user contract adopts the Checks-Effects-Interactions pattern

described in Section 2.6.3: contract state is updated *before* any external call is made, so that a hypothetical re-entry could not observe an inconsistent state. Second, all dataset-upload and index-computation functions carry the `nonReentrant` modifier inherited from OpenZeppelin's `ReentrancyGuard`, which precludes re-entry at the execution-model level regardless of whether the called library is trusted.

Analysis of remaining findings. The single low-severity finding concerns a style issue (an unused state variable in a deprecated code path) with no security implication. The 70 informational findings are distributed across the Zama library contracts and reflect coding-style observations, dead code in unreachable library branches, and naming convention warnings. None of these findings affects the security or correctness of the user contract. The single optimization finding is similarly located in library code and does not apply to functions on the critical path of the evaluation workflow.

Coverage limits of static analysis. It must be acknowledged that static analysis, however valuable as a first line of defense, does not cover the cryptographic layer in which the distinctive value of this framework resides. The correctness of the TFHE scheme implementation, the soundness of the input-proof verification, the integrity of the threshold KMS, and the absence of side-channel leaks in the client-side encryption library are all beyond Slither's scope. These components require separate forms of assurance, primarily cryptographic audits of the Zama infrastructure and formal verification of the protocol-level properties. For the framework as a whole, therefore, the clean Slither result should be understood as a necessary but not sufficient condition for deployability; the remaining conditions are captured in the trust-model assessment that follows.

4.6.3 Trust Model Assessment

The claim that the framework enables supplier evaluation *without* requiring trust in any party is strong, and a rigorous evaluation must examine precisely what residual trust the system does require. This subsection enumerates the trust assumptions, distinguishes those that are intrinsic to the approach from those that reflect the current

maturity of the underlying tooling, and identifies the directions along which each assumption could be relaxed.

Assumptions eliminated by the framework. Three trust dependencies that characterize traditional supplier evaluation workflows are eliminated, or substantially weakened, by the proposed architecture.

- **Direct supplier–evaluator trust.** Traditional evaluation requires the supplier to disclose raw performance data to the evaluating company, either directly or through a mutually trusted auditor, under Non-Disclosure Agreements. The FHE layer removes this disclosure: the supplier publishes only encrypted values and a plaintext aggregate score.
- **Trust in the computing party.** In any system where the evaluation algorithm runs on a server, the operator of that server must be trusted not to misapply or manipulate the computation. Here, the computation runs on the public blockchain under a verifiable smart contract, and any deviation from the published algorithm would leave an auditable trace on-chain.
- **Trust in the integrity of the published score.** Because the sustainability index is stored on the blockchain, it is tamper-resistant under the blockchain's standard security model: an adversary cannot retroactively alter a published SI without forking the chain.

Residual trust dependencies. Three trust assumptions persist and must be made explicit.

- **Client-side encryption integrity.** The supplier must trust the web application (and, transitively, the `fhevmjs` library) to encrypt inputs correctly and not to exfiltrate plaintext values before encryption. This is the most significant residual dependency, because it sits at the boundary between the supplier's device and the rest of the system. Mitigation strategies include serving the web application from a verifiable source (for example, an IPFS-pinned build whose content hash is audited), open-sourcing the application to permit independent code review, and, at the far end of the spectrum, executing the encryption logic inside a trusted execution environment on the supplier's hardware.

- **Threshold KMS.** The fhEVM's decryption key is distributed across a set of key holders under a threshold sharing scheme; any coalition exceeding the threshold could in principle decrypt on-chain ciphertexts. The current key-holder set is operated by Zama-approved entities, introducing a semi-centralized trust point that partially offsets the decentralization of the blockchain layer. This assumption weakens as the fhEVM matures and its key-holder set diversifies; it could be replaced entirely if future fhEVM versions adopt a fully permissionless threshold scheme or integrate verifiable decryption via ZK-SNARKs.
- **Soundness of the cryptographic stack.** The TFHE implementation, the input-proof verifier, and the reencryption primitive are all components of the Zama infrastructure whose correctness must ultimately be validated by cryptographic audit rather than by the framework itself.

Honest-KPI assumption. A fourth assumption, independent of the cryptographic stack, deserves separate mention. The framework establishes that a supplier's submitted KPIs cannot be *tampered with after submission* and cannot be *read by unauthorized parties*, but it does not and cannot establish that the *submitted values* correspond to the supplier's true operational reality. A supplier could submit deliberately understated water consumption or inflated renewable-energy percentages, and the smart contract would compute a correspondingly optimistic SI. This is the classical physical-digital bridge problem identified in Section 3.1.2 and acknowledged in Chapter 8 as a cross-cutting limitation of every system in this thesis that ingests off-chain data. In the supplier evaluation context, the problem is typically addressed at the policy layer through third-party attestation of submitted KPIs, by the supplier's auditor, by a certification body, or by an IoT sensor infrastructure that feeds measurements directly into the contract. This framework is compatible with any such attestation mechanism but does not implement one.

Summary. The framework succeeds in eliminating the principal trust dependencies of traditional supplier evaluation, the direct disclosure of sensitive data, the need for a trusted computing party, and the mutability of the evaluation record, at the cost of introducing narrower trust dependencies on the fhEVM cryptographic stack and on the honest submission of input data. The net trust reduction is substantial, and the remaining dependencies admit concrete mitigation paths that are either available now

(open-source code, verifiable hosting) or foreseeable in the fhEVM roadmap (fully decentralized KMS, verifiable decryption). The implications of this trust model for the broader privacy-transparency spectrum are revisited in Section 4.7 and developed systematically in Chapter 8.

4.7 Discussion and Limitations

The evaluation reported in Section 4.6 establishes that the framework functions correctly, costs are bounded and competitive with the incumbent manual processes, and the known security surface is adequately covered by standard static analysis. This section steps back from these quantitative results to examine three categories of remaining consideration. Section 4.7.1 discusses the limitations that are intrinsic to the approach and that no amount of engineering refinement can eliminate. Section 4.7.2 discusses the limitations specific to the prototype and the validation setting, most of which admit concrete remedies in a production deployment. Section 4.7.3 positions the framework within the broader adoption context and considers the incentive structure it creates for the actors in a supply chain.

4.7.1 Intrinsic Limitations

Three limitations are intrinsic to the framework, in the sense that they follow from its foundational design choices rather than from the way it was implemented.

Dependence on fhEVM availability. The contract cannot be deployed on an Ethereum-compatible blockchain that does not implement the fhEVM extension. At the time of writing, this restricts the deployment space to networks operated by Zama or integrated with its fhEVM infrastructure 4.6.1. This constraint is not an artifact of the contract design but a consequence of relying on on-chain homomorphic computation: any system that keeps the supplier's KPIs encrypted *and* performs the evaluation on-chain must run on an infrastructure that supports FHE operations natively. The alternative, performing the computation off-chain and publishing only the result, would reintroduce a trusted computing party, defeating the architectural premise of the framework. The limitation will ease as fhEVM-

compatible networks proliferate, but it will not disappear, because the dependence on FHE-enabled infrastructure is inherent to the approach.

Computational overhead of FHE. The 10–15× gas-cost overhead of homomorphic operations relative to plaintext arithmetic is not a property of the current implementation; it is a property of TFHE itself. More efficient FHE schemes under active development, notably CKKS for approximate arithmetic and lattice-based optimizations for TFHE, may reduce this factor in the medium term, but the overhead will remain substantial because the underlying cryptographic work that FHE performs is fundamentally more expensive than the work plaintext arithmetic performs. The implication for system design is that FHE is suitable only for computations whose input and output volumes are modest, which is precisely the regime in which the supplier evaluation problem sits. Applying the same approach to high-frequency or high-volume computations would not be viable with current FHE performance.

Expressiveness constraints on the aggregation function. The weighted sum adopted in Section 4.4.4 is compatible with the arithmetic primitives that current fhEVM versions support efficiently. More expressive aggregation functions, for instance, those involving comparisons, non-polynomial transformations, or conditional logic, are in principle supported by TFHE but either incur prohibitive gas costs or require primitives that the current `fhevm-solidity` library does not yet expose as high-level operators. As a result, the framework cannot at present implement the full range of MCDM methodologies surveyed in Section 3.2.2. This is a soft limitation: it constrains the *aggregation function* that can be used at a given moment but does not constrain the data access model (encrypted inputs, computed indices, differentiated decryption), which is the framework's primary contribution. As the fhEVM matures and exposes more operators, richer aggregation functions will become feasible without any modification to the contract's architecture.

4.7.2 Validation-Setting Limitations

A second set of limitations derives from the specific validation setting and admits remedies in a production deployment.

KPI coverage. The 13 indicators used in the prototype (Table 4.2) were selected to exercise the homomorphic computation pipeline in a controlled setting, not to provide an exhaustive sustainability assessment of a supplier. A realistic deployment would extend the KPI set to cover the full spectrum of ESG indicators required under the CSDD, on the order of 40-60 indicators across environmental, social, governance, and circular-economy dimensions. The contract's modular data model (Section 4.4.3) accommodates this extension without structural change, but the gas cost of dataset upload grows approximately linearly with the number of indicators: a fourfold expansion of the KPI set would bring the `addDataset` cost to the order of $60-$80 per dataset, which remains economically viable but is no longer negligible. This scaling behavior is itself an argument in favor of the two-index decomposition adopted from Alves et al. [2], which amortizes the growth by splitting KPIs across two datasets that can be uploaded and processed independently.

Aggregation function refinement. As noted in Section 4.7.1, the weighted-sum aggregation is a current limit of the implementation rather than a theoretical limit of the approach. Within this constraint, however, several refinements are available now and would strengthen the interpretability of the resulting index. Normalizing KPIs onto a common scale (for instance, percentile ranks against peer suppliers) would correct the fact that the raw indicators have incommensurable units and magnitudes; applying KPI-specific weights derived from a domain-expert MCDM exercise (as in Sithi et al. [68]) would replace the flat weights used in the prototype with values that reflect the relative materiality of each indicator. Neither refinement requires cryptographic advances; they require only a more involved configuration of the contract's weight vector, which can be updated without redeploying the contract. This is a straightforward extension and is recommended for any industrial pilot.

Single-industry validation. The framework was validated on a textile use case, for the reasons discussed in Section 4.3.2. The architectural claim is that the framework generalizes to any domain whose sustainability evaluation can be reduced to a weighted aggregation of numerical KPIs, including the automotive supply chain, which is the thesis's primary domain of interest. Substantiating this claim experimentally would require instantiating the framework with an automotive KPI set and re-running the workflow. The argument for generalizability is strong but not empirically verified for the automotive case at the time of writing.

Decentralization of auxiliary components. As the trust-model assessment (Section 4.6.3) established, two auxiliary components remain partially centralized: the web application that performs client-side encryption, and the key management service that holds the network's secret key. Both are outside the scope of the contract itself but are material to the trust properties the framework advertises. Serving the web application from IPFS with a pinned content hash, and accompanying the deployment with a hosted version from a non-Zama operator, would strengthen the supplier-side trust story substantially. On the KMS side, the ongoing work in the fhEVM roadmap toward a more permissionless key-holder set will reduce this dependency over time.

4.7.3 Framework Positioning and Adoption Considerations

Beyond its technical properties, the framework has implications for the economic and incentive structure of the supplier evaluation process. These implications are material to the practical adoption of the approach and are worth examining explicitly.

Incentives for suppliers. A system that publishes comparable sustainability indices on a public blockchain creates positive incentives for suppliers that genuinely invest in sustainable practices and corresponding negative incentives for those that do not. A supplier with a high SI can point to a tamper-proof, externally verifiable score when negotiating prices, bidding on contracts, or differentiating its offering to end consumers. Conversely, a supplier with a consistently low SI faces a visible competitive disadvantage. The framework thus does not merely facilitate compliance with the CSDD's due-diligence obligations (a defensive benefit); it creates a market mechanism in which sustainability performance becomes a direct economic variable (a constructive benefit). This mechanism is reinforced by the elimination of Non-Disclosure Agreements, which in current practice can act as a barrier to reputational benefit: a supplier whose data is locked behind an NDA cannot leverage it in negotiations with third parties.

Compatibility with existing audit infrastructure. The framework does not replace the role of third-party auditors; it complements it. Auditors can serve as the attestation layer that addresses the honest-KPI problem identified in Section 4.6.3:

a supplier's KPIs can be certified by an independent auditor whose signature is recorded on-chain alongside the encrypted values, giving purchasing companies additional confidence in the integrity of the submitted data. In this mode, the auditor's function shifts from a private intermediary (generating evaluation reports for specific clients) to a public attestation oracle, a role that is scalable and aligns with the regulatory direction the EU appears to be taking.

Deployment readiness. Taken together, the limitations enumerated in this section do not constitute a case against deployment; they delineate the conditions under which deployment would be appropriate. In its current form, the framework is ready for pilot deployment in a controlled industrial setting where: the KPI set is moderate in size, the fhEVM infrastructure is reachable, an attestation layer (auditor or IoT) is available to address the honest-KPI problem, and the web-application and KMS trust assumptions are acceptable to the participating parties. These conditions are met in the MICS partnership context that motivated the prototype, and they are plausibly met in an automotive pilot involving a Tier-1 OEM and a small number of its strategic suppliers. Broader deployment, covering hundreds of suppliers, dozens of KPIs, and multiple product categories, awaits both the maturation of the fhEVM infrastructure and the refinements to the aggregation function discussed in Section 4.7.2.

4.8 Chapter Summary

The framework presented in this chapter validated, on a textile-industry use case developed within the MICS Extended Partnership and the 13-indicator KPI taxonomy of Alves et al. [2], that on-chain computation on encrypted inputs is practically feasible at costs compatible with industrial adoption. The Sepolia-testnet prototype demonstrated correct end-to-end functionality at a per-supplier workflow cost of $39.30 (26 May 2025), with no high-severity Slither findings; the medium-severity warnings were all false positives attributable to Slither's conservative treatment of external calls into the trusted TFHE library. The residual trust model, the correctness of the web application, the integrity of the threshold KMS, and the honesty of the submitted KPIs, was made explicit in Section 4.6.3 and serves as the reference point for the cross-cutting privacy–transparency analysis of Chapter 8. The framework an-

chors the private end of the privacy–transparency spectrum that the four contributions collectively traverse.

Chapter 5

Digital Battery Passports and EU Regulation Compliance

This chapter is based on the following publication:

A. Butera, V. Gatteschi, "EU Battery Regulation and the Role of Blockchain in Creating Digital Battery Passports", in Proceedings of the IEEE International Conference on Blockchain Research and Applications for Innovative Networks and Services (BRAINS), 2024 [22].

5.1 Chapter Introduction

This chapter examines whether blockchain technology satisfies the technical requirements that EU Regulation 2023/1542 [6] imposes on digital battery passports (DBPs). Where Chapter 4 sits at the strict-confidentiality pole of the privacy–transparency spectrum (raw data visible to no one, only aggregated indices disclosed) the Battery Regulation mandates differentiated access: distinct subsets of the passport content are visible to distinct stakeholder tiers (general public, notified bodies, parties with legitimate interest), with the public tier granted unrestricted access to identification and performance information. The two policies are extremes of the same design space, and this chapter argues that they can be implemented on the same blockchain primitives (role-based access control, ownership-linked data responsibility, hybrid on-chain/off-chain storage) through different access-control configurations. Unlike the engineering contributions of Chapters 4, 6, and 7, this chapter does not present

a prototype, gas benchmark, or security audit. It presents a regulatory feasibility analysis: a provision-by-provision mapping between the requirements imposed by Regulation 2023/1542 on DBPs and the architectural properties of blockchain-based systems. The analysis originally appeared in [22] and is reproduced here in condensed form, reorganised around a central mapping table (Table 5.1) and updated with two years of subsequent EU regulatory and standardization developments. A small reference architecture sketch (Section 5.4), new to this thesis version, shows how the primitives exercised in Chapters 4, 6, and 7 compose into a concrete DBP system without duplicating the engineering effort reported in those chapters.

5.2 Regulatory Analysis

5.2.1 EU Battery Regulation Overview

Regulation (EU) 2023/1542 of the European Parliament and of the Council, adopted on 12 July 2023 and commonly referred to as the *EU Battery Regulation* [6], establishes a comprehensive framework governing the sustainability, safety, labeling, and end-of-life management of all batteries placed on the EU market. It entered into force on 18 August 2023, began to apply on 18 February 2024, and fully replaced the previous Batteries Directive 2006/66/EC on 18 August 2025. The regulation is directly applicable in all Member States without national transposition, which marks a notable shift in regulatory form from the preceding directive.

The regulation's scope covers all battery categories placed on the EU market (portable batteries, starting-lighting-ignition batteries, batteries for light means of transport, industrial rechargeable batteries with a capacity above 2 kWh, and electric vehicle batteries), but imposes its most demanding obligations on the three latter categories. These obligations span several regulatory instruments: sustainability and safety requirements (carbon footprint declaration, minimum recycled content thresholds, performance and durability parameters, removability and replaceability); due diligence obligations along the raw-material supply chain, harmonized with the Corporate Sustainability Due Diligence Directive; end-of-life obligations (collection targets, treatment efficiencies, minimum material recovery rates); and a set of labeling and information requirements that culminates, for the regulated battery categories, in the mandatory digital battery passport examined in the remainder of this chapter.

Section 2.7.3 of the background chapter provides a broader contextual treatment of the regulation that this discussion does not repeat.

A notable feature of the regulation's drafting is its heavy reliance on secondary legislation. A large share of the operational requirements depend on delegated and implementing acts to be adopted by the Commission on staggered timelines, several of which were still at consultation stage at the time of writing. For the DBP provisions, this means that the high-level architecture is fixed by the regulation itself while the data schema, access-tier composition, and technical interoperability specifications are established gradually through the secondary-legislation process. Section 5.6 reports the state of this process as of the thesis submission date.

5.2.2 Digital Passport Requirements

Article 77 of the regulation introduces the digital battery passport as an "electronic record" that must accompany each regulated battery throughout its lifecycle. From 18 February 2027, all batteries in the three heavily regulated categories (EV batteries, industrial rechargeable batteries above 2 kWh, and LMT batteries) placed on the EU market must carry a DBP, physically addressable through a QR code or comparable data carrier printed or engraved on the battery itself. The content of the DBP is specified in Annex XIII of the regulation, covering information on the battery model (manufacturer, chemistry, material composition, performance parameters, carbon footprint, recycled content, due-diligence report, compliance documentation) and on the individual battery (serial number, state of health, state of charge history for certain categories, repair or repurposing events, end-of-life information).

Four technical properties of the DBP, each derived directly from the regulation's text, structure the analysis of the remainder of this chapter. First, Article 77(2) mandates that the DBP system rest on a *decentralized data system*. Second, Articles 77(4) and 77(5) establish a three-tier differentiated access model in which the general public, notified bodies together with market surveillance authorities and the Commission, and parties with a legitimate interest each receive access to a distinct, defined subset of the passport's content. Third, Article 77(6) prescribes that responsibility for the passport transfers between economic operators as the battery passes through its lifecycle, with each operator being able to update the fields within its scope of responsibility. Fourth, the passport must remain accessible throughout

the battery's lifecycle and for a period after its end of life, without dependence on any single operator's continued existence or cooperation. Each of these four properties forms a row or cluster of rows in the mapping table presented in Section 5.3.

5.2.3 Global Initiatives and Standardization Efforts

The EU Battery Regulation is at present the only binding regulatory instrument worldwide that explicitly mandates a digital product passport for a defined product category, but it operates within a broader international and multi-stakeholder effort to define what such a passport should look like. Three initiatives are particularly relevant to the analysis in this chapter.

The *Global Battery Alliance* (GBA), a partnership of more than 150 organizations across industry, government, and civil society, has published a proof-of-concept battery passport specification that defines the core technical parameters and lifecycle tracking requirements, including indicators for responsible sourcing, child labor, and human rights [79]. Although it predates Regulation 2023/1542 in substantial part, the GBA specification has influenced the regulation's drafting and remains a reference point for voluntary industry adoption outside the EU. The *Battery Pass consortium*, co-funded by the German Federal Ministry for Economic Affairs and Climate Action, has produced a comprehensive Content Guidance document [80] that translates the regulation's content requirements into an implementable data specification. As Section 5.6 discusses, the consortium has since produced a technical guidance document, a software demonstrator, and a harmonized data-attribute specification that fill the standardization gap that this chapter's underlying paper identified in its conclusions. The *CIRPASS* project, funded under the Horizon Europe Digital Europe Programme, develops cross-sectoral DPP roadmaps for batteries, electronics, and textiles, with the broader aim of establishing shared rules and definitions that generalize beyond any single product category.

Outside the EU, the United States has pursued a parallel but distinct path through the Inflation Reduction Act [117], which creates financial incentives for domestic sourcing and low-carbon manufacturing of EV batteries without, however, mandating a specific passport format. Chinese initiatives target battery traceability and standardization primarily through national traceability platforms operated under government oversight, a model that is structurally different from the EU's multi-

stakeholder, decentralized-data-system approach. The regulatory divergence between these three blocs creates an obvious interoperability risk for internationally traded batteries, which the standardization initiatives discussed above, and particularly the cross-sectoral CIRPASS work, are intended to mitigate.

At the time when the underlying 2024 paper [22] was submitted, no officially recognized technical standard for DBP implementation existed. Section 5.6 reports the material changes to this picture over the subsequent two years.

5.3 Blockchain Alignment with Regulatory Requirements

This section presents the central analytical contribution of the chapter: a systematic mapping between the technical requirements imposed by Regulation 2023/1542 on digital battery passports and the architectural properties of blockchain-based systems. The mapping is summarized in Table 5.1, which also indicates the chapter in which each property is exercised engineering-wise. The subsections that follow walk through the mapping organized by regulatory theme: data storage (Section 5.3.1), data format and accessibility (Section 5.3.2), access control and stakeholder roles (Section 5.3.3), and data responsibility and lifecycle management (Section 5.3.4).

Three observations about the table are worth making before the subsection-level discussion.

First, the "Exercised in" column shows that *no single chapter implements a DBP end-to-end*, but that the architectural primitives a DBP would require are distributed across the engineering contributions. The vehicle identity system of Chapter 6 exercises most of them, and for a good reason: a vehicle NFT is structurally a vehicle-level digital product passport. The supplier evaluation framework of Chapter 4 exercises the same primitives under an inverted access-control policy, demonstrating that the substrate is general enough to support both ends of the privacy-transparency spectrum. This observation is revisited in Section 5.7.

Second, the mapping relies on the regulation's text as of its publication; several of the requirements depend on implementing and delegated acts that are being adopted over 2024–2027. Where the text of the regulation is itself ambiguous, the

Table 5.1 Mapping between EU Battery Regulation 2023/1542 requirements on digital battery passports, the corresponding technical demands on the implementation system, the blockchain properties that satisfy those demands, and the chapters in which the properties are exercised in practice. Article and Annex references follow the consolidated text of Regulation 2023/1542.

Regulatory requirement	**Reference**	**Technical demand**	**Blockchain property**	**Exercised in***
Decentralized data system	Art. 77(2)	No single point of control; no exclusive operator dependency	Distributed consensus; replication across independent nodes	Chs. 4, 6, 7
Unique identifier per battery	Art. 77(3); Annex XIII	Globally unique, non-reassignable identifier linking physical item to digital record	ERC-721 token identifier bound to battery serial / QR code	Ch. 6[a]
Differentiated access: public / notified bodies / legitimate interest	Art. 77(4)–(5)	Role-based access control with at least three tiers; per-attribute visibility	Access-control modifiers, role registries, per-field view functions	Ch. 6; Ch. 4[b]
Data responsibility linked to economic operator	Art. 77(6); Art. 78	Write permissions transfer with ownership; historical authorship verifiable	Ownership-linked write authorization; immutable event log	Ch. 6; Ch. 7
Authenticity and integrity of passport content	Art. 77; Annex XIII	Verifiable provenance of each data point; tamper-evident storage	Transaction signatures; content-addressed hashes for off-chain data	All impl. chs.
Accessibility throughout lifecycle and post-EoL period	Art. 77	Data persistence independent of any single operator's continued existence	Ledger immutability; replication; event-log reconstructibility	Chs. 6, 7
Machine-readable, interoperable format	Art. 77; impl. acts	Open schema; standardized serialization; cross-system queryability	Structured on-chain events; JSON metadata on IPFS via CID	Chs. 6, 7
Update obligations for repairers, remanufacturers, recyclers	Art. 77(6)	Multi-writer workflow with role-specific write scopes and audit trail	Per-role function modifiers; event-sourced history	Ch. 6[c]

* Chapter references indicate where the corresponding blockchain property is exercised in practice elsewhere in this thesis. [a] Vehicle-level analog; same primitive. [b] Same primitives under inverted, maximally restrictive policy. [c] Multi-stakeholder update workflow.

corresponding row in the table reflects the consortium-level consensus interpretation (Battery Pass Content Guidance [80]; DIN DKE SPEC 99100, 2025 [118]) rather than our own gloss. The limitations of this approach are discussed in Section 5.8.

Third, the table claims that blockchain *satisfies* each requirement, not that it is the only technology that does so. Federated data-space architectures, most prominently Catena-X Ecopass [119], satisfy most of the same requirements through different technical means. Section 5.5 engages with this alternative directly.

5.3.1 Data Storage Requirements

The first row of Table 5.1 captures the single most prominent technical requirement in Article 77 of the Regulation: that the DBP system rest on a *decentralized data system*. The regulatory intent of this provision, as articulated in the recitals, is twofold. First, it aims to prevent any single economic operator, including the manufacturer that places the battery on the market, from holding exclusive control over the passport data and therefore from being able to alter it, restrict access to it, or allow it to disappear if the operator ceases trading. Second, it aims to ensure that the passport remains functional for the full battery lifecycle and for a specified post-end-of-life period, a timespan that is typically longer than the commercial lifetime of many of the operators involved.

Blockchain technology satisfies this requirement by design. A public ledger is replicated across a network of independent nodes, and its continuity does not depend on any single party. If an operator leaves the market, the records it contributed remain available, verifiable, and queryable by any remaining participant or by the public. The regulation does not prescribe a specific technical instantiation of “decentralization”: Section 5.5 returns to the question of whether this requirement admits non-blockchain implementations. For the purposes of the mapping it is sufficient to observe that the blockchain model is an unambiguously compliant instantiation.

A practical corollary of the storage requirement is that the passport must accommodate large data objects, in particular the detailed technical documentation and test reports enumerated in Annex XIII, without burdening the ledger with content that is too bulky to replicate efficiently. The hybrid on-chain/off-chain storage pattern discussed in Section 2.4.2 directly addresses this concern and is applied across the engineering contributions. Large data objects are stored on IPFS or a comparable

content-addressed storage system; their content identifiers (CIDs) are recorded on-chain, ensuring that any observer can retrieve the off-chain data, hash it, and verify that the hash matches the on-chain record. The result is end-to-end verifiability with bounded on-chain storage costs.

5.3.2 Data Format and Accessibility

Rows two and seven of Table 5.1 together address the question of *how* the passport data is identified, represented, and made accessible. Article 77(3) mandates that each passport be bound to a unique identifier, typically materialized on the battery itself through a QR code or data-matrix carrier, which serves as the physical-to-digital bridge that the stakeholder must scan to reach the associated record. Annex XIII specifies the content of the passport, partitioning it into categories such as identification and model metadata, material composition, carbon footprint, performance and durability parameters, and compliance test reports. The regulation is silent on the specific data schema and serialization format; these are deferred to implementing acts, many of which were still pending adoption at the time of writing and are surveyed in Section 5.6.

At the blockchain level, the unique identifier requirement maps directly onto the ERC-721 non-fungible token standard, which assigns a unique, non-reassignable token identifier to each minted token. The token can be bound to the physical battery through an off-chain attestation that links the token identifier to the serial number engraved on the battery and encoded in the QR carrier. Chapter 6 exercises this pattern for vehicles, where the token identifier is similarly bound to the Vehicle Identification Number (VIN); the mechanism transfers to the battery case with only nominal adaptation.

The question of format and interoperability is more delicate. An interoperable passport must, at a minimum, rest on an open schema that any authorized party can parse without negotiating a bilateral agreement with the issuing operator. Two complementary mechanisms satisfy this demand in the blockchain setting. Structured on-chain events emitted by the smart contract provide a standardized record of state transitions that off-chain indexers and client applications can consume without contract-specific adapters. Larger structured data objects, typically serialized as JSON according to the schema adopted by the ecosystem, are stored on IPFS and

referenced by CID from the on-chain record. The standardization of the JSON schema itself lies outside the scope of blockchain technology and is the subject of the ongoing work by Catena-X, the Battery Pass consortium, and DIN, discussed in Section 5.6.

5.3.3 Access Control and Stakeholder Roles

The most analytically interesting row of Table 5.1, because it is where the battery passport differs most clearly from a naively transparent public record, is the third. Article 77(4) and (5) defines three classes of stakeholder that must each be granted a distinct level of access to the passport: the general public, to whom basic identification and performance information must be accessible; notified bodies, market surveillance authorities, and the Commission, who must have access to compliance-relevant information beyond the public baseline; and parties with a *legitimate interest* (recyclers, second-life operators, and other downstream actors), who must have access to detailed technical data necessary for safe and efficient end-of-life handling. The exact composition of each tier and the operational definition of "legitimate interest" are left to secondary legislation, which at the time of writing is still being drafted.

A subtlety that the regulation's text makes explicit, and that any serious implementation must respect, is that access is granted *per attribute* rather than *per record*. A single passport record contains fields that must be public (manufacturer identity, battery chemistry, carbon footprint declaration) alongside fields that must be restricted (detailed material composition, commercially sensitive performance parameters, site-specific manufacturing data). The access-control mechanism must therefore expose a per-field, per-role visibility policy rather than a coarser document-level policy. This is a stricter requirement than role-based access control in the classical sense, and rules out the simplest implementation patterns, such as encrypting the entire record under a role-specific key.

The blockchain-level mechanism that satisfies this requirement has three components. A *role registry contract* records the set of addresses currently associated with each stakeholder role and the expiry of each role assignment. An *access-control modifier* attached to each view or state-modifying function enforces the policy by consulting the role registry and failing closed if the caller's role is insufficient. Per-

field visibility is expressed by decomposing the record into separate storage variables or separate contracts, each exposed through a view function protected by a role check appropriate to the corresponding tier. Large restricted fields can be held off-chain in encrypted form with the decryption key released through the role-controlled path.

Chapter 6 implements precisely this pattern for the multi-stakeholder update workflow of a vehicle NFT, where inspectors, insurers, and owners are granted write permissions scoped to specific record fields; the implementation demonstrates that the pattern is practical at acceptable gas costs. The chapter also makes explicit the symmetric case to the one analyzed here: where the Battery Regulation mandates differentiated access with a public baseline, the supplier evaluation framework of Chapter 4 mandates differentiated access with no public component at all. The two configurations exercise the same architectural primitives; only the policy encoded on top differs.

5.3.4 Data Responsibility and Lifecycle Management

Rows four, six, and eight of Table 5.1 jointly address the temporal dimension of the passport: who is responsible for its content at each moment in the battery's life, how that responsibility is transferred between economic operators, and how the passport persists through and beyond the commercial lifetime of the battery. Article 77(6) establishes the principle that data responsibility follows the battery: the manufacturer is responsible for the passport until the battery is placed on the market, after which responsibility passes to each subsequent economic operator that takes possession, including the OEM that integrates the battery into a vehicle, the first owner of the vehicle, any second-life operator that repurposes the battery, and finally the recycler that processes it at end of life. Each transition must be recorded, and each operator must be able to update the passport fields that fall within its scope of responsibility, while being prevented from tampering with the historical record contributed by earlier operators.

The blockchain-level mechanism that satisfies this requirement combines three primitives already exercised elsewhere in the thesis. Ownership of the passport is represented as ownership of the corresponding ERC-721 token, transferrable through the standard `transferFrom` interface or through the marketplace mechanism of Chapter 6. Write authorization for operator-specific fields is gated on the caller

being the current token owner at the time of the call, enforced by a check against the result of `ownerOf`. Historical authorship is preserved by emitting an event for every update, with the event signature recording the field modified, the caller, and the block timestamp; the resulting event log is immutable and can be replayed by any observer to reconstruct the full update history of the passport. The combination yields the property that the regulation requires: each operator can update what it is entitled to update, cannot alter what earlier operators contributed, and cannot silently delete or backdate its own contributions.

Lifecycle persistence follows almost for free from the underlying ledger's immutability guarantees. The passport remains available on-chain for the full commercial lifetime of the battery and for any post-end-of-life period the regulation or subsequent implementing acts prescribe, without requiring any operator to continue hosting it. Once the recycler records the end-of-life transition, the passport transitions to a terminal state in which further updates are prohibited but the historical record remains queryable indefinitely. This property is exercised structurally in Chapter 7, where the state machine of a pothole report traverses analogous transitions (REPORTED $\rightarrow$ IN-PROGRESS $\rightarrow$ COMPLETED) with comparable authorization semantics.

5.4 A Reference Architecture for a Blockchain-Based DBP

The analysis of Section 5.3 establishes, requirement by requirement, that the architectural primitives of blockchain-based systems satisfy what Regulation 2023/1542 demands of a digital battery passport. This section takes the complementary step: it sketches a conceptual reference architecture that composes those primitives into a single coherent system. The purpose is to demonstrate that the mapping is not only element-wise valid but also structurally coherent, i.e. that the individual satisfactions do not interfere with one another and can coexist in a buildable design. This is the chapter's only engineering-flavored contribution, and its scope is intentionally limited: no component is implemented as a production prototype within this thesis. The primitives composed here are validated engineering-wise in Chapters 4, 6, and 7, and the reference design reuses them rather than replicating the engineering effort.

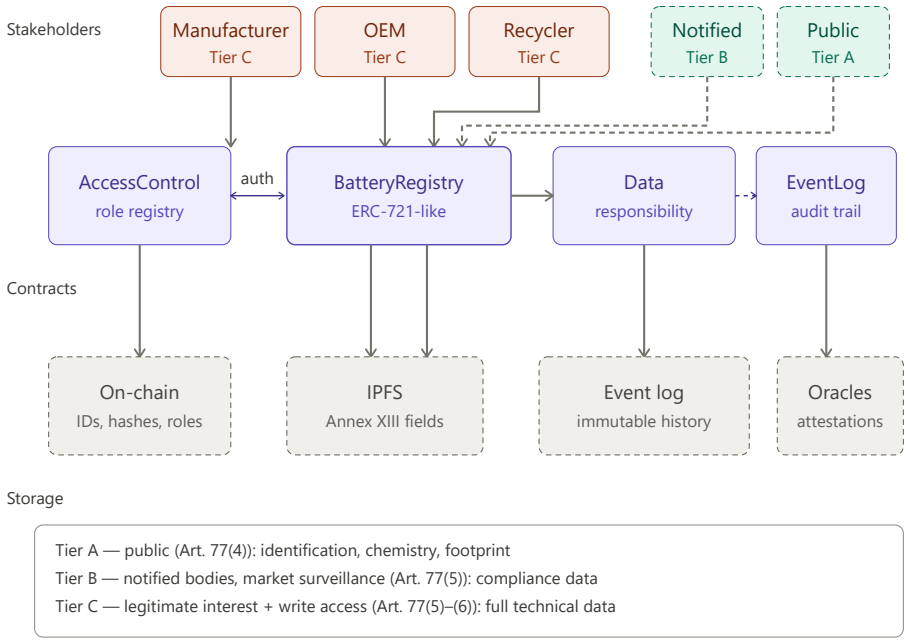

Fig. 5.1 Reference architecture for a blockchain-based digital battery passport.

5.4.1 Layers and Components

The architecture (Figure 5.1) is organized in three layers. The *stakeholder* layer distinguishes writers from readers. Writers are the economic operators responsible for the passport at successive points of the battery lifecycle: the manufacturer that creates the passport at first placement on the market, the OEM that updates it upon integration into a vehicle, and the recycler that finalizes it at end of life. Second-life operators, repairers, and remanufacturers all fit this category and are omitted from the figure only to keep it legible. Readers are non-proprietary parties that consume the passport under the tiered-visibility policy: notified bodies and market surveillance authorities at Tier B, and the general public at Tier A. The regulation permits the same natural or legal person to hold multiple roles, and the architecture reflects this by granting role memberships rather than treating the stakeholder identity as fixed.

The *contract* layer contains four smart contracts, each corresponding to a cluster of rows in Table 5.1. BATTERYREGISTRY is the central contract, modeled on the ERC-721 non-fungible-token standard. It mints one token per battery at the point of first placement on the market, maintains the binding between the token identifier,

the QR carrier on the physical battery, and the Annex XIII metadata, and exposes state-modifying functions for each writer role. ACCESSCONTROLREGISTRY maintains the mapping between blockchain addresses and stakeholder roles, records the expiry of each role assignment, and is consulted by BATTERYREGISTRY for authorization of every state-modifying call and for per-tier filtering of every view function. DATARESPONSIBILITY encapsulates the ownership-transfer logic of Article 77(6): when the battery changes economic operator, the corresponding token is transferred and the write permissions follow, while the historical record contributed by prior operators remains immutable. EVENTLOG emits a structured event for every update, with the event signature recording the field modified, the caller, the block number, and the block timestamp; the resulting log is the audit trail the regulation implicitly requires and is the cheapest on-chain record of historical authorship.

The *storage* layer resolves the hybrid on-chain/off-chain split discussed in Section 5.3.1. On-chain storage holds the compact, integrity-critical data: token identifiers, role registry, content hashes of off-chain objects, status flags, and ownership records. IPFS holds the bulky Annex XIII fields whose size would make on-chain storage impractical: detailed material composition tables, test reports, carbon-footprint declarations with their supporting methodological documents, and compliance certificates. The event log, although conceptually part of the ledger, is shown as a distinct storage target to emphasize that historical authorship is reconstructible from events alone, without reading contract state. Finally, external authenticated systems, such as the third-party verifier of the carbon-footprint declaration, act as oracles whose attestations are signed off-chain and whose signatures are verified on-chain before any state update they support.

5.4.2 Passport Lifecycle Walkthrough

To illustrate how the components of Figure 5.1 interact, consider the lifecycle of a single battery passport traversing the reference design from first placement on the market to end of life.

Creation. At the moment the battery is placed on the market, the manufacturer submits a transaction to BATTERYREGISTRY that mints a new token bound to the battery's unique serial number and QR carrier, attaches the initial Annex XIII meta-

data (identification, chemistry, model-level parameters), and records the CIDs of the off-chain large fields (material composition tables, test reports, carbon-footprint declaration) on-chain. The transaction's authorization is verified against ACCESSCONTROLREGISTRY: only an address registered under the MANUFACTURER role can invoke `mint`. An event is emitted to EVENTLOG recording the creation.

Integration. When the battery is delivered to the OEM that integrates it into a vehicle, the manufacturer invokes the transfer function of DATARESPONSIBILITY, which transfers token ownership to the OEM's address. From this moment, write permissions on the fields reserved to the current responsible operator are held by the OEM. The OEM typically appends integration-level metadata (vehicle VIN, date of integration, installed capacity configuration) via a state-modifying call authorized by ACCESSCONTROLREGISTRY against the OEM role. The event log records both the transfer and the update.

Operation and second-life transition. Throughout the use phase, periodic updates (state-of-health readings at scheduled intervals, repair or battery-management-system parameter updates, carbon-footprint recalculations) are submitted by authorized operators and processed identically: authorization check, state update, IPFS pointer refresh if needed, event emission. Should the battery be transferred to a second-life operator, the DATARESPONSIBILITY contract again transfers token ownership; the second-life operator's role is activated in ACCESSCONTROLREGISTRY and gains write access to the fields appropriate to that stage.

End-of-life. Upon recycling, the recycler updates the passport with the end-of-life record (date, treatment performed, material recovery rates), and invokes a finalization function that transitions the token to a terminal state. Further updates are rejected by BATTERYREGISTRY's state machine; the passport remains on-chain and publicly queryable for the period the regulation prescribes, without requiring continued cooperation of the original manufacturer or any intervening operator. The event log of the full lifecycle is replayable by any observer.

5.4.3 What the Reference Architecture Is Not

Three clarifications are warranted before closing the section. First, the design is *one* implementation of the regulation's requirements, not *the* implementation. It reflects one set of reasonable design choices (a single registry contract rather than a fleet of per-battery contracts; a role registry separate from the main contract rather than inline modifiers; event-based rather than storage-based audit trail) and several equally reasonable alternatives exist. The important claim is that the requirements are jointly satisfiable in a blockchain setting, not that any specific combinator is uniquely correct. Second, the design omits several elements that a deployment-grade system would need, including key management and recovery procedures for stakeholder addresses, upgrade paths for contract logic, mechanisms for replacing deprecated IPFS pinning services, and legal wrappers binding on-chain identities to real-world economic operators. These are engineering concerns orthogonal to the regulatory-feasibility argument that this chapter makes and would distract from it if included. Third, and most importantly, the reference architecture is *not validated* by gas benchmarks, security audits, or user studies. Its validity as a reference design rests on the fact that each of its primitives is exercised engineering-wise in the chapters that follow: BATTERYREGISTRY's ERC-721 structure and ownership-linked update logic in Chapter 6; ACCESSCONTROLREGISTRY's role-based authorization in Chapter 6, with the inverted-policy counterpart in Chapter 4; event-log audit trails in Chapter 7; and the hybrid on-chain/IPFS storage pattern across all engineering chapters. The reference design composes these primitives; it does not introduce new ones.

5.5 Discussion: Why Blockchain

The mapping of Section 5.3 and the reference architecture of Section 5.4 establish that blockchain-based systems satisfy the requirements of Regulation 2023/1542 on digital battery passports. This section argues the complementary qualitative case: that among the technologies capable of satisfying those requirements, blockchain exhibits properties that align particularly well with the spirit of the regulation, while acknowledging plainly that it is not the only technology that does so. The first three subsections summarize the arguments that originally appeared in the underlying BRAINS 2024 paper [22], compressed to the role they play in the narrative. The fourth subsection addresses the most prominent non-blockchain alternative, the

Catena-X data space, directly and at some length, because the thesis version of this chapter cannot pretend that alternative does not exist.

5.5.1 Transparency and Traceability

The regulation's recitals make clear that the DBP is instrumentally motivated by a broader policy objective: to support the circular-economy transition by making battery information available to stakeholders that historically lacked access to it, most prominently second-life operators and recyclers. Blockchain-based systems deliver transparency and traceability as *native* properties rather than as policies enforced on top of a system that could, in principle, be configured otherwise. Every write operation is visible to every network participant, every historical state is reconstructible from the event log, and every claim of authorship is cryptographically anchored to the authoring address. No party, including the most powerful operator in the supply chain, can selectively disclose parts of the record to some stakeholders while hiding them from others (differentiated access policy is orthogonal to and compatible with this default). For a regulation that lists transparency and traceability among its explicit objectives, the alignment between policy intent and technical default is a substantive argument, not merely an engineering preference.

5.5.2 Security and Trust

The security properties that a DBP implementation must provide (integrity of each recorded attribute, non-repudiation of each contribution, availability of the historical record, tamper-evidence for any attempted modification) are properties that blockchain-based systems deliver as consequences of their core design rather than as features bolted onto the perimeter. A record signed by an economic operator's private key and confirmed by the consensus protocol cannot be modified after the fact without detection, and cannot be plausibly attributed to any other party. The trust model of the resulting system has an important structural feature: no single operator need be trusted for the system as a whole to function correctly. This is structurally different from the trust model of a centrally-hosted passport platform, where the platform operator necessarily sits in the trust boundary of every participant. For a multi-party lifecycle like a battery's, spanning manufacturers, OEMs, transporters, repairers, second-life operators, and recyclers across jurisdictional boundaries, a

trust model that does not privilege any single participant is a better match to the structure of the underlying problem.

5.5.3 Non-Blockchain Alternatives: The Catena-X Case

The most important development in the DBP implementation landscape since the publication of the underlying 2024 paper is the maturation of the Catena-X data ecosystem into the de-facto industrial standard for European automotive data exchange [119]. Catena-X is not a blockchain: it is a federated data space in which each participant retains its own data and exposes it to other participants through standardized *connectors* implementing the Eclipse Dataspace Components (EDC) specification. Identity is managed through Decentralized Identifiers (DIDs) and authenticated via Verifiable Credentials; authorization is enforced at the connector level through policy-based access control and usage-control contracts negotiated bilaterally between consumer and provider. A *Digital Twin Registry* provides passport discovery across the dataspace. The EcoPass use case [119], certified for first-mover implementers, covers the full battery passport workflow within this architecture. Any assessment of a blockchain-based DBP proposal in 2026 has to account for Catena-X, and doing so honestly improves rather than weakens the argument for the blockchain-based approach.

Two observations structure the comparison. First, Catena-X and blockchain-based DBPs satisfy the regulation's requirements through structurally different technical means. Replication and persistence across node failures are achieved in Catena-X through each operator's own operational availability guarantees and, for federated retrieval, through the continued participation of downstream consumers; in blockchain, they are achieved by ledger replication independent of any single operator's continued cooperation. Integrity of a record's content is achieved in Catena-X through signed payloads delivered over authenticated connectors; in blockchain, it is achieved by the consensus protocol's acceptance of the transaction. Differentiated access is achieved in Catena-X through per-request policy evaluation at the connector; in blockchain, through role-registry and access-control-modifier patterns. In each pairing, both technologies meet the regulation's functional requirement, but through different operational assumptions.

Second, the two architectures are optimized for different deployment profiles. Catena-X is optimized for data exchange among a known, bounded set of industrial partners that have entered into contractual relationships mediated by the Catena-X governance framework (business-partner numbers, onboarding certification, association membership). Its strengths (data sovereignty, commercially-sensitive-data protection, integration with enterprise identity infrastructures, and fine-grained usage-control policies) are the properties an OEM's procurement organization would prioritize. A blockchain-based approach is optimized for data disclosure that must survive the end of any specific commercial relationship and be verifiable by parties outside any bilateral agreement: the second-life operator retrieving a passport eight years after the manufacturer has exited the EU market, the public scanning a QR code at an EV charging station, the auditor reconciling a carbon-footprint declaration against its supporting test reports. These are the properties that consumer-facing transparency and open-market second-life scenarios require.

The two architectures address the regulation with different center-of-gravity assumptions. Where the DBP system is primarily an intermediary between known B2B partners, Catena-X's federated model is the more practical near-term implementation path; this is the direction in which the majority of industrial implementations are currently moving. Where the DBP system is primarily a vehicle for open-market and public transparency, blockchain's structural properties (independence from any operator, public verifiability without onboarding) are the stronger fit. The regulation's differentiated-access model arguably calls for both: the public tier (Art. 77(4)) is the blockchain case, the legitimate-interest tier among known industrial partners (Art. 77(5)) is the Catena-X case, and a long-term implementation will likely pair the two. Bridging the two architectures, through blockchain anchoring of Catena-X connector attestations for the public-facing subset of the passport content, is an open research question that lies beyond this thesis.

5.6 Recent Developments (2024–2026)

The analysis reported in the underlying 2024 paper was conducted against the text of Regulation 2023/1542 as published in the Official Journal and against the standardization landscape as it appeared in the first quarter of 2024. Two years of intervening work have produced developments that the thesis version of the chapter

cannot ignore. Four of these developments materially interact with the argument of the preceding sections; each is reported here with its implication for the chapter's claims made explicit.

Battery Pass technical guidance and software demonstrator (Fraunhofer IPK, March 2024). Shortly after the submission of the underlying paper, the Battery Pass consortium released its Technical Guidance document and an accompanying software demonstrator [120], moving the consortium's output from content specification (the 2023 Content Guidance [80]) to technical architecture. The recommended architecture is interoperability-layer-oriented, not blockchain-centric, and is closely aligned with the federated-data-space direction later consolidated by Catena-X. The implication for this chapter is not that our blockchain-based analysis is superseded (the consortium's architecture is not mandated by the regulation either), but that the reference architecture of Section 5.4 should be interpreted as *one* member of a set of implementation approaches, of which the consortium's reference design is the most influential alternative for the European industrial context.

DIN DKE SPEC 99100 (January 2025). The January 2025 publication of DIN DKE SPEC 99100, *Requirements for Data Attributes of the Battery Passport* [118], closes the most prominent standardization gap identified in the conclusions of the underlying paper. Developed by the Battery Pass consortium with DIN, DKE, and public industry consultation, SPEC 99100 defines the essential data attributes required by the EU Battery Regulation along with voluntary additions, providing the formal machine-readable schema that the regulation itself had left to secondary legislation. For a blockchain-based implementation following the reference architecture of Section 5.4, SPEC 99100 is the natural source of truth for the JSON metadata stored on IPFS and indexed by CID from the on-chain record. The specification is largely format-agnostic about the underlying data infrastructure and is compatible with both blockchain-based and federated-data-space implementations.

Regulatory timeline adjustments and Omnibus IV. The application dates of several obligations within the regulation have been adjusted through secondary legislation. The most salient adjustment, for purposes of this chapter, is the postponement of the supply-chain due-diligence obligations from 18 August 2025 to 18 August

2027 by Regulation (EU) 2025/1561 (Omnibus IV) [121], together with the setting of a Commission deadline of 26 July 2026 for the publication of due-diligence implementation guidance. Independently, carbon-footprint declaration obligations for EV batteries entered into force on 18 February 2025 as originally scheduled, and the core DBP application date of 18 February 2027 remains in force. These adjustments modify the compliance timeline but do not alter any of the technical requirements mapped in Table 5.1, and therefore do not affect the architectural argument of this chapter. They do, however, reduce the urgency of the supply-chain due-diligence dimension of the passport content, which interacts with the contribution of Chapter 4 and is revisited in Section 5.7.

Industry implementations and pilots. The two years between the underlying paper and the thesis have seen the first industry-grade implementations move from concept to certification. The Catena-X EcoPass certification has been awarded to first-mover participants starting from 2024, with DENSO Corporation becoming, in April 2025, the first Japan-headquartered company to achieve it [119]. Independent blockchain-based offerings from vendors such as Circularise and Minespider have entered commercial deployment, particularly for the supply-chain due-diligence and recycled-content-certification subsets of the passport content, which align with the properties blockchain delivers most naturally. The emerging picture is the one that Section 5.5.3 anticipated: a heterogeneous implementation landscape in which federated data spaces handle the B2B-among-known-partners portion of the passport, and blockchain-based components handle the public-verifiability and cross-relationship-persistence portions. No single implementation approach has emerged as dominant for the full passport lifecycle, and it is reasonable to expect a hybrid steady-state rather than a single winning architecture.

5.7 Cross-Chapter Connections

The argument of this chapter is not that blockchain is the uniquely correct implementation of Regulation 2023/1542, but that the engineering contributions elsewhere in this thesis collectively exercise the architectural primitives any regulation-compliant DBP would need. Two cross-chapter connections make this concrete.

5.7.1 Connection to Chapter 4: The Privacy-Transparency Spectrum

The framing of this chapter, established in Section 5.1 and carried through Section 5.3, is that a digital battery passport under Regulation 2023/1542 and a privacy-preserving supplier evaluation framework under the Corporate Sustainability Due Diligence Directive sit at opposite poles of a single design space. They use the same architectural primitives, but encoded with opposite access-control policies.

In the supplier evaluation framework of Chapter 4, the access policy is "the raw KPI data is visible to no one; only aggregated sustainability indices, computed on encrypted inputs via fully homomorphic encryption, are released to authorized purchasers". The suppliers' commercially-sensitive data never leaves the encrypted domain, and the framework demonstrates that meaningful policy objectives (supplier ranking, due-diligence evidence) can be achieved under this stringent policy. In the DBP case analyzed in this chapter, the access policy is "different subsets of the passport content are visible to different stakeholder tiers, with the public tier granted unrestricted access to identification and performance information". The two configurations are extremes of a spectrum that runs from maximum confidentiality (Chapter 4) through differentiated access (this chapter) to full public verifiability (Chapter 6).

The substrate is common. Role-based access control via a role registry, ownership-linked data responsibility via token holding, hybrid on-chain/off-chain storage via CID references, cryptographic data authentication via transaction signatures, and immutable historical authorship via event logs all appear, under different configurations, in every system on the spectrum. The claim that the thesis traverses this spectrum rather than presenting isolated case studies is developed systematically in Chapter 8. This chapter's role in supporting that claim is to establish the architectural primitives not merely as engineering conveniences but as regulation-aligned building blocks.

5.7.2 Connection to Chapter 6: The Battery Passport as a Vehicle Passport Analog

The most direct cross-chapter connection is to Chapter 6, which presents an NFT-based vehicle identity and lifecycle management system. The analogy between

a digital battery passport and a vehicle NFT is close enough to warrant explicit examination: both are unique-identifier-bound digital records associated with a physical asset, both must persist across multiple ownership transitions, both must support multi-stakeholder updates scoped by role, and both must remain accessible throughout the asset's lifecycle and for a post-end-of-life period.

The regulatory context differs. For batteries, the DBP is mandated by Regulation 2023/1542 with a fixed application date of 18 February 2027. For vehicles, no EU-level regulation currently mandates a blockchain-based or any other specific implementation of a lifecycle record, although individual Member States operate centralized registries and the broader Ecodesign for Sustainable Products Regulation envisions future delegated acts extending DPPs to additional product categories. The vehicle NFT system of Chapter 6 is therefore not a regulation-driven implementation but a proactive engineering demonstration of what the vehicle-level analog of a regulatory DPP could look like.

Three specific architectural primitives map directly between the two cases. The ERC-721 token as a unique, non-reassignable identifier bound to a physical asset (battery serial / vehicle identification number) is identical across the two systems. The ownership-linked write authorization pattern is identical: in both cases, the set of addresses permitted to update specific fields of the record is gated on current token ownership, and transitions as the token is transferred. The multi-stakeholder update workflow (manufacturer, OEM, repairer, recycler for the battery; manufacturer, inspector, insurer, owner for the vehicle) exercises the same role-based access-control primitives under differently-configured role registries. The engineering contribution of Chapter 6, gas-benchmarked, security-audited, and end-to-end implemented, is the concrete engineering validation that the reference architecture of Section 5.4 can be built, can be built economically, and can be built securely. Within the thesis, Chapter 6 is the engineering proof-of-concept that Chapter 5 argues is regulation-feasible.

This symmetry also positions Chapter 6 more ambitiously within the broader DPP discourse than its standalone paper title suggests. A system for NFT-based trades of second-hand vehicles is, read through the lens of this chapter, an engineering realization of a vehicle-level digital product passport with a functioning second-hand marketplace built on top. The regulatory analysis of this chapter gives that contribution a reading it could not claim on its own.

5.8 Limitations and Threats to Validity

The analysis presented in this chapter has four limitations that reporting requires acknowledging explicitly.

First, the mapping in Table 5.1 and the reference architecture in Section 5.4 are constructed against the text of Regulation 2023/1542 as published in the Official Journal, augmented with the consortium-level consensus documents (Battery Pass Content Guidance [80], Technical Guidance [120], DIN DKE SPEC 99100 [118]) available as of the thesis submission. A substantial share of the regulation's operational requirements depend on delegated and implementing acts whose adoption extends through 2027 and beyond. Specific interpretations of the requirements mapped in this chapter may shift as secondary legislation is adopted, and some rows of the mapping table may require revision accordingly. The validity of the architectural argument, however, rests on the high-level requirements fixed by the regulation text itself, which are unlikely to be altered by secondary legislation.

Second, the phrase *decentralized data system* in Article 77(2), which is the textual anchor for the central claim of Section 5.3.1, is an authoritative-sounding but technically underspecified term whose operational interpretation is being settled by industry practice rather than by the regulation. Federated data-space architectures such as Catena-X interpret *decentralized* in a distribution-without-shared-ledger sense, while blockchain-based architectures interpret it in a replicated-ledger sense. The regulation is technology-neutral by drafting; this chapter's claim is that blockchain is *sufficient* to satisfy the requirement, not that blockchain is *necessary* or that the regulation mandates it exclusively. Section 5.5.3 makes this position explicit and engages with the strongest alternative interpretation.

Third, the physical-to-digital bridge problem, which is acknowledged as a cross-cutting limitation of every system that ingests data from the physical world, applies to the DBP case in a form that is worth stating plainly. A blockchain-based DBP cryptographically guarantees the integrity of data *once recorded on the ledger*; it does not guarantee the veracity of the data at the point of recording. A manufacturer that misreports the carbon footprint of its batteries or understates the cobalt content of its cathodes will have its misreport recorded immutably, indistinguishable from an accurate report. The regulation addresses this concern through independent third-party verification obligations (carbon-footprint declarations must be verified;

due-diligence reports must be audited), and the reference architecture of Section 5.4 accommodates this through the *External authenticated systems* storage component: verifier attestations are cryptographically signed off-chain and verified on-chain before supporting state updates. The pattern is the same one employed in Chapter 4 for the intrinsic-limitations discussion of KPI authenticity.

Fourth, and most importantly, the reference architecture of Section 5.4 is conceptual. It has not been instantiated as a production prototype, has not been gas-benchmarked at scale, and has not been subjected to formal security audit in the specific configuration proposed. The validity of its primitives rests on the engineering validation provided by the other chapters: the ERC-721 and ownership-linked-update patterns are validated by Chapter 6; the role-based access control is validated both there and, in its inverted configuration, by Chapter 4; the event-log audit trail is validated by Chapter 7; the hybrid on-chain/IPFS storage is validated across all implementation chapters. An end-to-end DBP prototype integrating these primitives into a single system, under the specific configuration required by Regulation 2023/1542, is out of scope for this thesis and is identified as future work in Chapter 9.

5.9 Chapter Summary

The provision-by-provision mapping of Section 5.3, summarised in Table 5.1, demonstrates that the architectural primitives of blockchain-based systems jointly satisfy the technical requirements imposed on digital battery passports by Regulation (EU) 2023/1542. The reference architecture of Section 5.4 composes these primitives into a coherent system whose validation rests not on an end-to-end prototype of its own but on the engineering contributions of Chapters 4, 6, and 7, each of which exercises one or more of the primitives the architecture relies on. The Catena-X comparison of Section 5.5.3, together with the 2024–2026 standardization developments reported in Section 5.6, situate the blockchain-based approach within an implementation landscape that has matured significantly since the underlying 2024 paper [22].

Chapter 6

NFT-Based Vehicle Identity and Second-Hand Market

This chapter is based on the following publication:

A. Butera, D. Vianello, G. Pratticò, D. Novaro, and V. Gatteschi, "Blockchain and NFTs-Based Trades of Second-Hand Vehicles," IEEE Access, vol. 11, 2023. [1]

6.1 Chapter Introduction

The reference architecture of Section 5.4 sketched abstractly the architectural primitives, public verifiability, role-based differentiated access, hybrid on-chain/off-chain storage, token-bound data responsibility, that any blockchain-based digital product passport would require. This chapter develops these primitives into a working engineering system, with full implementation, gas-cost analysis, and security assessment. In the privacy–transparency spectrum, the framework occupies the transparent pole: every piece of vehicle data that a prospective buyer, inspector, insurer, or regulator might wish to verify is, by design, publicly readable on-chain or retrievable from decentralized storage; the access-control policy restricts who may write, not who may read. The DPP framing inherited from Chapter 5 is productive but incomplete. A digital product passport in the sense of the EU Ecodesign for Sustainable Products Regulation [122] tracks a component across a supply chain (its origin, composition, and compliance attributes) as a record written during manufacturing and read during deployment, end-of-life, or regulatory inspection. A vehicle, however, is not merely a component, it is an artifact with a life of its own after the point of sale. It accumulates maintenance events, changes ownership, renews insurance, passes or fails periodic inspections, and eventually

exits service. These post-manufacturing transitions are the locus of the frauds, information asymmetries, and administrative inefficiencies that motivate this chapter, and they require a technical primitive the DPP vocabulary does not supply: a dynamic, role-gated, lockable ERC-721 token, bound to a physical vehicle through its VIN, whose on-chain state machine encodes the lifecycle discipline that post-manufacturing operations require. The information-asymmetry literature [123, 8, 9] documents the consequent inefficiencies, buyers overpay or are defrauded, honest sellers face adverse selection, intermediaries capture rents, with odometer fraud alone estimated by the European Parliament at approximately €8.9 billion per year in the EU [7] (Section 1.2, Gap 3). The prior work surveyed in Chapter 3 (Sections 3.1.1- 3.1.3) addresses individual aspects of this problem; what is missing is a unified primitive handling all post-manufacturing lifecycle events under a single access-control regime, a safe ownership-transfer mechanism for decentralized marketplaces, and an adoption-friendly payment model that does not require buyers and sellers to denominate transactions in cryptocurrency.

Contributions

This chapter contributes the following:

1. **A dynamic vehicle NFT as a lifecycle identity primitive** (Section 6.3). An extension of ERC-721 with (i) structured vehicle metadata with categorized mutability (immutable manufacturing data, mutable insurance and inspection records, append-only repair log), (ii) a per-field state machine encoding the update-review lifecycle, (iii) five-role access control gated by that state machine, and (iv) a transfer-suspension ("locking") mechanism that generalises beyond its originating marketplace use case to recall-remediation holds, inspection-period freezes, and other transient-non-transferability requirements.

2. **A hybrid storage model with per-category independence** (Section 6.3.4). Rather than anchoring a single combined JSON document, the framework anchors one IPFS content identifier per metadata category, allowing the three mutable categories to be updated on independent schedules by independent stakeholders without contention.

3. **A decentralized second-hand vehicle marketplace** (Section 6.4). An application built on the primitive, implementing safe ownership transfer under a temporary-custody-by-escrow-contract pattern, price negotiation with bids bounded from below by a market-price oracle, and fiat-denominated payment routed through an external gateway. In the original 2023 publication [1] the marketplace occupied the structural

centre of the system; in this thesis it is presented as an instance that exercises the primitive, so that the primitive's broader utility, for dealer certification, insurance-renewal portals, fleet dashboards, and regulator-facing registries, becomes visible.

4. **A full engineering treatment** (Sections 6.5-6.6). Solidity smart contracts on top of OpenZeppelin's ERC-721 and AccessControl libraries, a decentralized-application front-end integrating MetaMask and js-IPFS, a per-function gas-cost analysis, a performance analysis, and a security assessment against the SWC Registry [124].

6.2 Paper-Specific Related Work

The broader survey of blockchain-based vehicle identity and used-car systems is conducted in Chapter 3, which examines the landscape under two paradigms, the *registry* approach and the *token-based* approach, and identifies the gap into which this chapter's contribution was made. The purpose of this section is narrower: to situate the chapter's framework within the specifically-NFT neighborhood that the primitive exploits, and to make explicit the novelty claim that no prior work combined the set of properties this chapter treats as jointly necessary.

As Chapter 3 established, Non-Fungible Token standards, specifically ERC-721, were already being applied in 2022–2023 to represent unique physical assets across a range of supply-chain domains: medical devices [51], fine jewelry [52], shipping containers and cargo [53], and AI and federated-learning models [125, 126], among others. Two additional proposals applied NFTs to healthcare supply chain management [127] and to time-bound monetization of private data [128]. Collectively these works established that the NFT pattern could serve as a digital identity anchor for a physical or digital asset, and developed the on-chain mechanics of trading, transfer, and dispute handling. None of them addressed vehicles. Prior to the framework presented in this chapter, no work in the academic literature combined NFT-based vehicle representation with a complete lifecycle update workflow, a decentralized marketplace supporting price negotiation and fiat settlement, and a locking primitive that enforces ownership-transfer safety during update-pending or settlement-pending states. Table 6.1 compares this framework against the NFT comparators from other domains along the four design dimensions that discriminate the framework's contribution.

Three observations structure the differentiation the table documents. First, along the *mutability* axis, this framework is one of only two comparators that support runtime metadata updates; the other [126] applies dynamic metadata to federated-learning model versioning and does not address ownership-transfer semantics, which are the central concern of the vehicle use case. Second, along the *settlement-safety* axis, the framework is the only

Table 6.1 Comparison of this framework against NFT-based systems from other application domains. Adapted from Butera et al. [1].

Ref.	Domain	NFT type	Locking	Payment	Dispute mgmt.
[51]	Medical devices	Static	Yes	Cryptocurr.	No
[127]	Healthcare supply chain	Static	No	Cryptocurr.	Yes
[52]	Fine jewelry and gemstones	Static	No	Cryptocurr.	Yes
[53]	Shipping containers and cargo	Static	No	Cryptocurr.	No
[125]	AI models	Static	No	Cryptocurr.	No
[126]	Federated-learning models	Dynamic	No	Cryptocurr.	No
[128]	Private-data monetization	Static	No	Cryptocurr.	Yes
This chapter	**Second-hand vehicles**	**Dynamic**	**Yes**	**Fiat**	**Yes**

comparator that combines dynamic metadata with a locking primitive preventing mutation during settlement windows. The [51] comparator uses static metadata with locking, which sidesteps the composition problem rather than solving it. Third, along the *payment* axis, all seven cross-domain comparators denominate trades in cryptocurrency; this framework's fiat-payment integration is distinctive in the NFT literature and, as Section 6.4.3 argued, a deliberate choice motivated by the adoption-barrier considerations specific to private-to-private vehicle trading. The cross-domain comparison therefore does not position the framework as an NFT-marketplace variant; it positions the framework as the work that translates the NFT primitive from its original cross-domain applications into the specific engineering constraints of the vehicle-lifecycle use case, and composes dynamic metadata, locking, fiat settlement, and dispute handling into a single substrate.

6.3 The Dynamic Vehicle NFT: A Lifecycle Identity Primitive

The contribution of this chapter is best understood not as a marketplace, but as a *primitive*: a uniform on-chain representation of a vehicle that persists across every post-manufacturing transition the vehicle experiences, and that can be composed with application-specific logic to realize use cases as diverse as ownership transfer, insurance renewal, inspection certification, and repair documentation. The marketplace described in Section 6.4 is one such application; the primitive itself, a *dynamic, role-gated, lockable ERC-721 token bound to a physical vehicle*, is what generalizes.

This reframing departs from the original publication [1], in which the marketplace occupied the structural center and the NFT design was presented as its supporting machinery. Two years of subsequent work on token-bound accounts, standardized metadata-update signaling, and the EU regulatory trajectory toward digital product passports (Chapter 5) have clarified that the NFT-as-vehicle-identity is the more durable contribution. The marketplace is an instance that exercises the primitive; other instances, such as a dealer certification platform, an insurance-renewal portal, a fleet-management dashboard, could be built on the same substrate without modification to the underlying token contract.

6.3.1 Design Goals

The dynamic vehicle NFT was designed to satisfy six requirements, each traceable to a limitation of prior work identified in Chapter 3 (Section 3.1.3).

G1 — Persistent one-to-one binding to a physical vehicle. Each on-chain token must correspond bijectively to a single physical vehicle, and the binding must survive ownership transfer, component replacement, and cross-jurisdiction movement. This rules out per-transaction identifiers (as in bill-of-sale models) and per-owner identifiers (as in conventional title-registry models). The binding is anchored to the Vehicle Identification Number (VIN) [44], embedded in the immutable manufacturing metadata at mint time. The limits of this anchoring, the VIN-to-NFT binding must be established off-chain, a residual trust assumption inherited from existing registration infrastructure, are discussed in Section 6.7.

G2 — Multi-party updatability under access control. A vehicle's state is modified by multiple distinct stakeholders during its service life: insurance companies issue and renew policies, inspection stations certify roadworthiness, repair shops document maintenance, and OEMs issue recalls or approve conversions. Each stakeholder must be able to append state without being able to impersonate another or to modify records outside its competence. This is the function of the role-based access-control (RBAC) layer described in Section 6.3.5.

G3 — Stateful lifecycle semantics. A vehicle's lifecycle is not a sequence of arbitrary edits; it follows a discipline. An inspection report is *requested*, then *issued*, then *reviewed*, and possibly *contested*. A marketplace listing is *created*, then *locked pending payment*, then either *completed* or *returned*. The primitive must encode this discipline as an explicit state machine, so that invalid transitions (e.g., transferring ownership of a vehicle with a pending

unreviewed inspection) are rejected by the contract rather than relying on off-chain vigilance. This is the function of the per-field state variables described in Section 6.3.3.

G4 — Safe suspension of transferability. During sensitive transitions, notably the window between offer acceptance and payment settlement, the NFT must be temporarily non-transferable, even by its owner, to prevent the seller from divesting the asset or modifying its state while a buyer is committing to a price. This is the *locking mechanism* described in Section 6.3.6. The mechanism is deliberately simple, a single boolean enforced in the `_beforeTokenTransfer` hook, but it generalizes beyond the marketplace: the same primitive supports inspection-period freezes, recall-remediation holds, and law-enforcement-requested immobilization.

G5 — Compact on-chain footprint with verifiable off-chain payload. Vehicle documentation is heterogeneous and occasionally voluminous: repair logs accumulate over decades, inspection reports include photographs, insurance certificates include policy annexes. Storing this content on-chain is neither cost-effective nor privacy-appropriate. The primitive therefore stores on-chain only what the chain is uniquely suited to store, ownership, state variables, and content-addressed hashes, while the payload resides on a decentralized storage layer (IPFS [38]) whose content-addressed URIs are anchored in the NFT. The trade-off between on-chain and off-chain storage, which recurs across Chapters 4, 5, and 7, is introduced in Section 6.3.4.

G6 — Standards compliance for ecosystem composability. The primitive must be consumable by unmodified ERC-721-aware infrastructure, such as wallets, block explorers, NFT indexers, general-purpose marketplaces, without requiring bespoke integration. This rules out extending or breaking the ERC-721 interface; all dynamic-vehicle-NFT semantics are implemented as additions *on top of* a conformant ERC-721 implementation (OpenZeppelin [29]), never as modifications to it. This property is what would, in principle, allow a dynamic vehicle NFT minted under this framework to be listed on a general-purpose NFT marketplace such as OpenSea or Blur without custom integration work; the purpose-built marketplace contract of Section 6.4 is an *added* feature, not a required one.

Goals G1–G6 are not independent. G3 and G4 share machinery (both constrain transitions on the token's state); G5 and G6 share a constraint (both forbid bloating the ERC-721 interface); G2 and G3 interact (the state machine governs *which* role can perform *which* transition). The design described in the remainder of this section is the minimal one that satisfies all six simultaneously.

6.3.2 Actors and Stakeholders

The framework recognizes five classes of actor, each corresponding to a distinct access role in the smart contract layer (Section 6.3.5).

Original Equipment Manufacturer (OEM). The OEM is the *issuer* of the NFT. It mints the token at the end of the manufacturing line, populates the immutable manufacturing metadata (VIN, make, model, production date, initial specifications), and transfers ownership to the first retail customer at point of sale. Throughout the vehicle's service life, the OEM retains the administrative authority to grant and revoke update roles to downstream stakeholders and to approve or reject updates submitted by those stakeholders. The OEM's role is analogous to that of the *economic operator* in the EU Battery Regulation framework discussed in Chapter 5.

Vehicle Owner. The current owner holds the NFT in their Ethereum account and possesses the authority to list it on the marketplace, request updates, and transfer it upon sale. Ownership of the NFT is, by design, the canonical record of ownership of the physical vehicle; the framework's claim is that on-chain ownership should become the authoritative legal record, with off-chain registration derived from it rather than the converse.

Insurance Company. Authorized by the OEM via the `INSURANCE_ROLE`, insurance companies issue and renew insurance certificates. Each certificate is uploaded to IPFS and its URI is anchored in the NFT, with the insurance state variable transitioning through the discipline described in Section 6.3.3.

Inspection Station and Repair Shop. Authorized via `INSPECTION_ROLE` and `REPAIR_ROLE` respectively, these actors document periodic safety and emissions inspections (inspection stations) and maintenance or repair events (repair shops). The distinction matters because the two activities are subject to different regulatory frameworks in most jurisdictions.

Marketplace Contract. A smart contract, not a human actor, but treated as a principal for access-control purposes (`MP_ROLE`). The marketplace contract is the only party permitted to invoke the locking primitive on the NFT, a restriction that confines lockability to a single auditable contract rather than distributing it across the ecosystem.

An additional actor, the *buyer*, interacts with the framework during purchase flows but is not granted any persistent role; a buyer who completes a purchase becomes the new owner by virtue of NFT transfer, at which point they inherit the *owner* role's privileges.

A sixth class of actor was deliberately *not* given a first-class role: the government registration authority. The framework's architectural bet is that a credible blockchain-based vehicle registry could, in the limit, supplant the administrative registry operated by national motor-vehicle authorities. We return to the tractability of this bet in Section 6.7.

6.3.3 Dynamic NFT Structure: Metadata and State

The dynamic vehicle NFT extends the ERC-721 standard by associating, with each token, a `Vehicle` struct that holds four categories of metadata and three state variables. The struct is stored in a `mapping(uint256 => Vehicle)` indexed by `tokenId`.

Manufacturing metadata (immutable). Assigned at mint time, never modified thereafter. Includes the VIN, production date, make, model, initial specifications, and a content-addressed URI pointing to the IPFS-hosted manufacturing record (which itself may include a vehicle image, full specifications, and OEM-supplied documentation). Immutability is enforced structurally: no contract function updates these fields.

Insurance Certificate (mutable, role-gated). The current insurance policy, represented by an IPFS URI pointing to the insurance-issued document. Updated by the `INSURANCE_ROLE` upon policy renewal or change.

Vehicle Inspection Report (mutable, role-gated). The current periodic inspection record, similarly IPFS-anchored. Updated by the `INSPECTION_ROLE` at the conclusion of each inspection cycle.

Repair Log (mutable, role-gated, append-only). The cumulative record of maintenance and repair events. Each new repair appends a new entry to the IPFS-hosted log; the NFT's on-chain URI is updated to point to the latest version. The log's append-only discipline is enforced off-chain by the repair shop (which must reference the previous log's CID when constructing the new one) and is cryptographically verifiable on-chain by any party that retains earlier CIDs.

Overlaid on the three mutable metadata fields is a state machine. Each of the three fields has an associated state variable that takes one of four values, drawn from the enumerated type:

$$\texttt{State} \in \{\texttt{Reviewed}, \texttt{UpdateRequired}, \texttt{UnderReview}, \texttt{Dismissed}\}$$

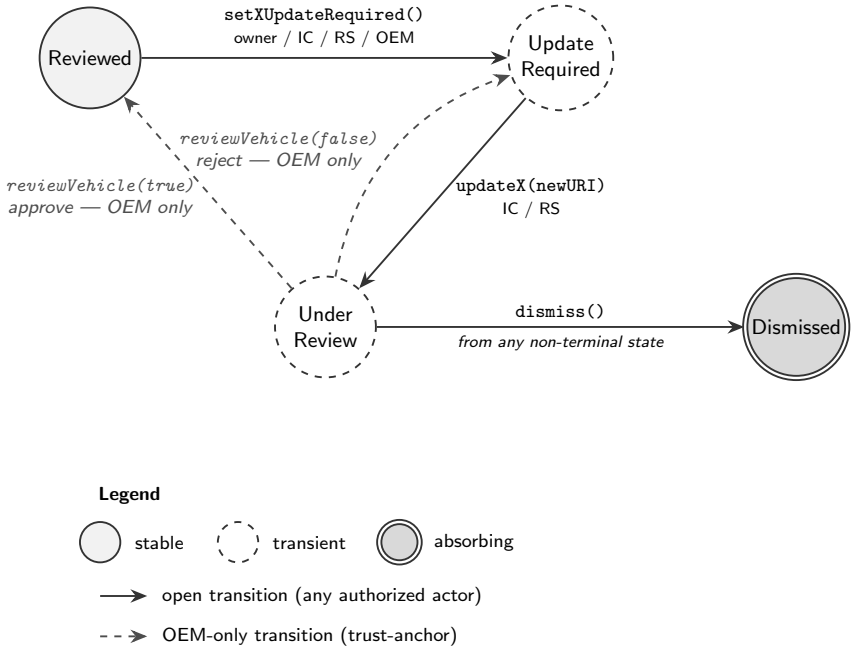

Fig. 6.1 Per-field state machine governing the lifecycle of each mutable metadata field (insurance certificate, inspection report, repair log). Stable states are `Reviewed` (compliant) and `Dismissed` (terminal); transient states are `UpdateRequired` (an update is pending) and `UnderReview` (an update has been submitted and is awaiting OEM validation). The transitions are enforced by the NFT smart contract; invalid transitions revert.

The transitions among these states encode the lifecycle discipline required by goal G3. Figure 6.1, a thesis addition not present in the original publication, makes the state space explicit.

The semantics of each state are as follows. The initial state of every field at mint time is `Reviewed`, signifying that no update is outstanding. Any authorized actor (owner, insurance, inspection, repair, or OEM) may signal that an update is needed by invoking a `setXUpdateRequired` function, transitioning the field to `UpdateRequired`. An authorized issuer (`INSURANCE_ROLE` for insurance, `INSPECTION_ROLE` for inspection, `REPAIR_ROLE` for repair) may then supply a new URI via the corresponding `updateX` function, which transitions the field to `UnderReview`. The OEM, acting as the trust anchor, then invokes `reviewVehicle` with a boolean verdict: on approval, the field returns to `Reviewed`; on rejection, it reverts to `UpdateRequired`, inviting re-submission. The `Dismissed` state is absorbing and is entered when the vehicle leaves service permanently (end-of-life, export, total loss).

Three properties of this design merit emphasis. First, the state machine is per-field rather than per-vehicle: a vehicle may simultaneously have a `Reviewed` insurance certificate, an `UnderReview` inspection report, and an `UpdateRequired` repair log. This orthogonality matches the operational reality that the three document types are updated on different cycles by different actors. Second, the OEM occupies the privileged *review* position. This is a deliberate trust concession, discussed frankly in Section 6.7: the framework assumes the OEM is a competent and honest validator, and its architecture fails open (in the sense of permitting incorrect updates to be accepted) if this assumption is violated. Alternatives, multi-signature review boards, oracle-driven automatic validation for IoT-instrumented updates, or zero-knowledge proofs of update correctness [76], are discussed in Section 6.7 as directions that subsequent work has begun to explore. Third, the state machine is *not* coupled to NFT transfer: a vehicle may change ownership while one of its fields is `UnderReview`, and the new owner inherits the pending review. The marketplace's listing logic (Section 6.4) exposes pending-review status to prospective buyers, converting an otherwise opaque state into public information.

6.3.4 Hybrid On-Chain / Off-Chain Storage

Goal G5 requires that the NFT carry verifiable evidence of arbitrarily large payloads without storing the payloads themselves on-chain. The framework achieves this through the standard content-addressed-storage idiom: large, heterogeneous content is pinned to IPFS [38], and its 34-byte content identifier (CID) is stored on-chain as the vehicle's `tokenURI`. The cryptographic properties of the CID (it is a hash of the content) guarantee that any subsequent modification of the IPFS-hosted payload invalidates the on-chain reference, making silent tampering detectable.

Three design choices distinguish the framework's use of IPFS from the naive default. *First*, the NFT stores *one* URI per metadata category rather than a single URI pointing to a combined JSON document. This allows the three mutable categories (insurance, inspection, repair) to be updated independently, which is essential for the per-field state machine of Section 6.3.3. *Second*, pinning responsibility is distributed: the OEM pins the manufacturing metadata; the issuing authority pins each update; the owner may additionally pin their own copy as insurance against issuer dropout. *Third*, the framework does not rely on any single IPFS gateway for retrieval; any node in the IPFS network that pins the content can serve it, and the content's integrity is verifiable at the point of retrieval by re-hashing.

The storage model is a concrete instance of the hybrid pattern introduced in Chapter 2, Section 2.4, and it recurs in Chapters 5 and 7 with minor variations. Chapter 4 occupies

a distinct position in this design space, storing encrypted data directly on-chain under fully homomorphic encryption rather than relying on a decentralized off-chain layer; the systematic comparison across the four engineering contributions is drawn in Chapter 8.

6.3.5 Role-Based Access Control

The framework delegates authentication and authorization to OpenZeppelin's `AccessControl` contract [129], extending it with five domain-specific roles: `OEM_ROLE`, `INSURANCE_ROLE`, `INSPECTION_ROLE`, `REPAIR_ROLE`, and `MP_ROLE`. The `OEM_ROLE` is assigned at deployment to the contract deployer and carries the administrative authority to grant and revoke all other roles via `grantRole` and `revokeRole`.

The mapping from roles to callable functions follows goal G2's principle of least privilege. `INSURANCE_ROLE` may invoke only `updateInsuranceCert`; `INSPECTION_ROLE` may invoke only `updateInspectionRep`; `REPAIR_ROLE` may invoke only `updateRepairLog`. No role other than the NFT's current owner may initiate an ownership transfer, and no role may override the locking mechanism. `MP_ROLE`, assigned to the marketplace contract at deployment, is the only role permitted to toggle the `locked` field (Section 6.3.6).

The composition of RBAC with the state machine is where the framework's access-control discipline becomes non-trivial. A call to `updateInsuranceCert` succeeds only if *both* (a) the caller holds `INSURANCE_ROLE` *and* (b) the insurance state variable is currently `UpdateRequired`. The second condition prevents an insurance company from issuing unsolicited updates, requiring instead that some authorized party, typically the owner, first signal that a renewal is due. This decoupling of *request* from *fulfillment* is essential to the lifecycle semantics and distinguishes the framework from systems that treat update rights as unconditional (e.g., [59, 130], discussed in Chapter 3).

6.3.6 The Locking Mechanism

The `locked` field of the `Vehicle` struct is a single boolean, and its enforcement adds a single conditional to the ERC-721 hook `_beforeTokenTransfer`:

```
function _beforeTokenTransfer(
    address from, address to, uint256 tokenId
) internal override {
    require(!vehicles[tokenId].locked, "NFT locked");
    super._beforeTokenTransfer(from, to, tokenId);
```

```
}
```

This is the entirety of the enforcement machinery. Its simplicity is deliberate: a complex locking scheme would invite bugs, would be harder to audit, and would complicate interoperability with wallets and indexers that expect conformant ERC-721 behavior. The boolean-plus-hook idiom is standard, minimally intrusive, and composes cleanly with every ERC-721-aware tool in the ecosystem.

The locking primitive's power derives not from its implementation but from who is authorized to set it. Only `MP_ROLE`, i.e., the marketplace contract, may toggle `locked`. This restriction localizes lockability to a single, auditable contract and prevents the NFT contract itself from being abused to immobilize a vehicle against its owner's wishes. An OEM cannot unilaterally lock an NFT to coerce a recall; a repair shop cannot lock a vehicle to enforce payment of a disputed bill. Lockability is a marketplace-escrow primitive, and its scope is constrained to the marketplace-escrow use case.

That said, the primitive generalizes. Nothing in the implementation precludes the instantiation of additional lock-authorized contracts, a recall-remediation contract with `RECALL_ROLE`, a law-enforcement-immobilization contract with `IMMOBILIZATION_ROLE`, each constrained by its own governance and scoped to its own use case. The question is not technical but regulatory: who should be legally permitted to immobilize a privately owned asset, and under what procedural safeguards? The framework makes the decision explicit rather than embedding it silently in implementation; subsequent deployments can accept or revise the decision.

Among the contributions of this chapter, the locking mechanism is the one that has proven most durable. The original paper was submitted in May 2023, three months after the publication of ERC-6551 [131], a proposal that assigns a smart-contract account, widely known as a *token-bound account*, to every ERC-721 token; the account is controlled by whoever currently owns the parent NFT, can hold arbitrary assets, and transfers with ownership. Although this framework does not use ERC-6551, which in any case had not yet achieved production deployment at the time of implementation and as of this writing remains formally under review in the EIP process, the standard's production marketplaces face an attack vector that is analogous to the one the framework's locking mechanism was built to defend against. A seller, in the instants between listing acceptance and settlement, can empty the token's account of its valuable sub-assets, delivering to the buyer an NFT whose advertised contents no longer match its actual contents. The mitigation adopted by production ERC-6551 marketplaces and recommended by the standard's authors is precisely the pattern this framework implemented in 2023: *temporary transfer of the NFT to the marketplace contract for the duration of the settlement window*, with automatic return on

settlement failure. The underlying safety property is the same: during a settlement window, the asset as a whole must be frozen against all mutating operations, not merely against transfer. The 2023 locking primitive is, in retrospect, an early instance of a safety pattern the broader NFT ecosystem has since converged on.

A 2026 redesign of the framework would likely adopt ERC-6551 for the metadata categories that exhibit sub-asset structure, most naturally the repair log and, for an electric vehicle, the component-level bill of materials. The core of the framework, the NFT as the canonical vehicle identity, the state machine governing update discipline, the role-based access control, the hybrid storage model, would be unchanged; ERC-6551 is an enrichment of the ERC-721 substrate this framework uses, not a displacement of it. A production automotive deployment in 2026 might also reasonably still pick plain ERC-721, as this framework does, and defer the ERC-6551 migration until the standard stabilizes.

6.4 Application: A Second-Hand Vehicle Marketplace

The primitive described in Section 6.3 supports a range of applications, as Section 6.1 argued. This section presents the application that was the original motivation for the framework and that the 2023 paper [1] treated as its central artifact: a decentralized second-hand vehicle marketplace in which buyers and sellers transact directly on-chain, with the NFT's locking mechanism providing the settlement-window safety property that the ERC-721 standard alone does not guarantee. The presentation here is deliberately condensed relative to the paper. Architectural primitives described in Section 6.3, the dynamic NFT, the state machine, the role-based access control, the locking mechanism, are not re-described; this section focuses on the application-specific logic that composes those primitives into a functioning marketplace.

6.4.1 Listing and Market-Price Evaluation

A seller who wishes to list a vehicle calls the Marketplace smart contract's `listItem` function, passing the NFT's `tokenId` and the seller's desired asking price. The function performs four validations before accepting the listing. First, the caller must be the current owner of the NFT, enforced through an `isOwner` modifier. Second, the NFT must not already be listed, enforced through a `notListed` modifier. Third, the price parameter must be strictly positive. Fourth, the seller must have previously authorized the Marketplace contract to transfer the NFT on the seller's behalf, via the standard ERC-721 `approve` mechanism; this

authorization is a prerequisite of the ERC-721 standard for any non-owner transfer and is not specific to this framework.

On successful validation, `listItem` creates a new `Item` struct recording the seller's address, the asking price, the market price obtained from the Market Price Evaluator (MPE), the listing's status (initialized to `Open`), and fields reserved for the eventual buyer's address, highest offer, and payment deadline. The listing additionally triggers a review of the NFT's state variables: if the insurance certificate or inspection report is found to be outside its validity window, the corresponding field's state is transitioned to `UpdateRequired`, and the buyer subsequently sees this information as part of the listing. An `ItemListed` event is emitted.

The Market Price Evaluator is an off-chain module, not itself a smart contract, that retrieves the vehicle's metadata from IPFS using the URIs anchored in the NFT and computes a suggested price from a model that takes into account age, mileage, repair history, and model-specific depreciation curves. The MPE's output is advisory: the seller is free to list at any price, but the MPE-suggested price serves as one of the inputs to the bid-acceptability threshold described in the next subsection. The 2023 implementation embedded the MPE as a simple module within the DApp; a production deployment would more reasonably obtain the market-price signal through an oracle service subscribed to market data, a refinement left to future work.

6.4.2 Bidding, Acceptance, and Escrow

A buyer interested in a listed vehicle calls `makeOffer` with the `tokenId` and a bid price. The contract enforces a bid-acceptability threshold t, defined as the minimum of three values: the MPE-suggested market price, the seller's asking price, and any previously-accepted bid still under consideration. Bids below t are rejected automatically; bids at or above t are recorded as the current offer, overwriting any earlier lower offer. An `Offer` event notifies the seller.

The seller may accept or reject a standing offer. Rejection, via `rejectOffer`, clears the offer and emits an `OfferRejected` event; the listing returns to `Open` state and continues to accept new offers. Acceptance, via `acceptOffer`, is the operation that engages the locking primitive. On acceptance, three state changes occur atomically. First, the NFT's ownership is transferred from the seller to the Marketplace contract, using the authorization the seller granted at listing time. Second, the NFT's `locked` field is set to `true`, preventing any subsequent transfer until the flag is cleared. Third, a payment deadline is set seven days in the future, within which the buyer must complete the monetary leg of the transaction. The `Item` struct's state transitions to `Close`, and an `OfferAccepted` event notifies the buyer.

During the seven-day window, the NFT is effectively held in escrow by the Marketplace contract. The seller cannot withdraw it, because the seller is no longer its owner; the buyer cannot withdraw it, because the locking flag prevents transfer. Two outcomes terminate the window. If the buyer completes the payment (see Section 6.4.3), `confirmPayment` clears the locking flag and transfers the NFT from the Marketplace contract to the buyer. If the buyer fails to complete the payment within the deadline, `returnItem` transfers the NFT back to the seller and resets the listing to `Open` state. In both cases, the corresponding event (`PaymentConfirmed` or `ItemReturned`) is emitted.

The escrow pattern is the mechanism that provides the marketplace's principal safety property: neither party can unilaterally complete or abort the transaction once an offer has been accepted. The locking primitive of Section 6.3.6 is what makes this pattern expressible at the smart-contract level without additional custodianship machinery; the atomicity of the transfer-and-lock step ensures that no window exists during which the NFT is with the Marketplace contract but transferable, or with the seller but not listed.

6.4.3 Fiat Payment Integration

A distinguishing choice of the framework, relative to comparable NFT-based marketplaces, is that the monetary leg of the transaction is denominated in traditional fiat currency and cleared through an external payment gateway rather than in cryptocurrency. This choice was motivated by the observation that vehicle transactions involve sums of money that most private buyers and sellers would be reluctant to convert into cryptocurrency at the time of the transaction, and that the user-experience friction of doing so would be a material adoption barrier for the kind of private-to-private market the framework is designed to enable.

The integration is architecturally simple. The payment gateway is treated as a trusted third party for the monetary leg only: it reports back to the Marketplace contract whether the buyer has completed the payment within the deadline, and the contract conditions its `confirmPayment` execution on this report. The gateway does not custody the NFT, and does not participate in the ownership transfer logic, which remains entirely on-chain. The 2023 implementation used a sandbox integration with a generic payment-gateway API; a production deployment would negotiate a specific gateway relationship, with the gateway's terms of service governing its obligations to the framework's users.

The trust concession involved in this choice is discussed in Section 6.7 as one of the framework's acknowledged limitations: the gateway sits in the trust boundary of every user of the marketplace, and the decentralization property the ledger otherwise provides does not extend across the gateway. Cryptocurrency-denominated payment would eliminate this

trust concession at the cost of the adoption barrier it was chosen to avoid. Neither option is unambiguously better; the choice reflects a judgment about the tradeoff that the 2023 framework made and that subsequent deployments may reasonably revise.

6.4.4 Marketplace Flow and Dispute Handling

Figure 6.2, reproduced with minor modifications from the 2023 paper, presents the full sequence of interactions among buyer, seller, Marketplace contract, NFT contract, MPE, and IPFS during a complete listing-to-settlement flow. The flow integrates the listing, bidding, acceptance, and settlement operations described above, with the locking primitive invoked at the acceptance step and cleared at the settlement step.

Dispute handling, in this framework, is structural rather than procedural. The escrow-by-ownership-transfer pattern ensures that the NFT's eventual destination is determined by the payment outcome, not by negotiation between the parties after the fact. A buyer who has paid receives the NFT; a buyer who has failed to pay does not, and the seller recovers the NFT in the latter case automatically. Disputes that might arise, for example, over the vehicle's actual condition relative to its advertised metadata, are out of scope of the framework: they are conventional consumer-protection disputes to be resolved through the jurisdictions' existing legal mechanisms, with the on-chain record serving as immutable evidence of the transaction terms. This scope restriction is stated plainly in Section 6.7.

6.5 Implementation

This section reports the implementation choices made in realizing the framework as a deployed prototype. The intent is to document what was built, at the level of detail required for reproducibility, without re-motivating design choices already discussed in Sections 6.3 and 6.4. Figure 6.3 summarizes the components of the deployed system and the flows among them; the subsections that follow describe each component in turn.

6.5.1 Smart Contracts

The framework comprises two smart contracts, the `VehicleNFT` contract and the `Marketplace` contract. Both are implemented in Solidity 0.8.x and build on the OpenZeppelin Contracts library [29], specifically the `ERC721`, `ERC721URIStorage`, and `AccessControl` modules.

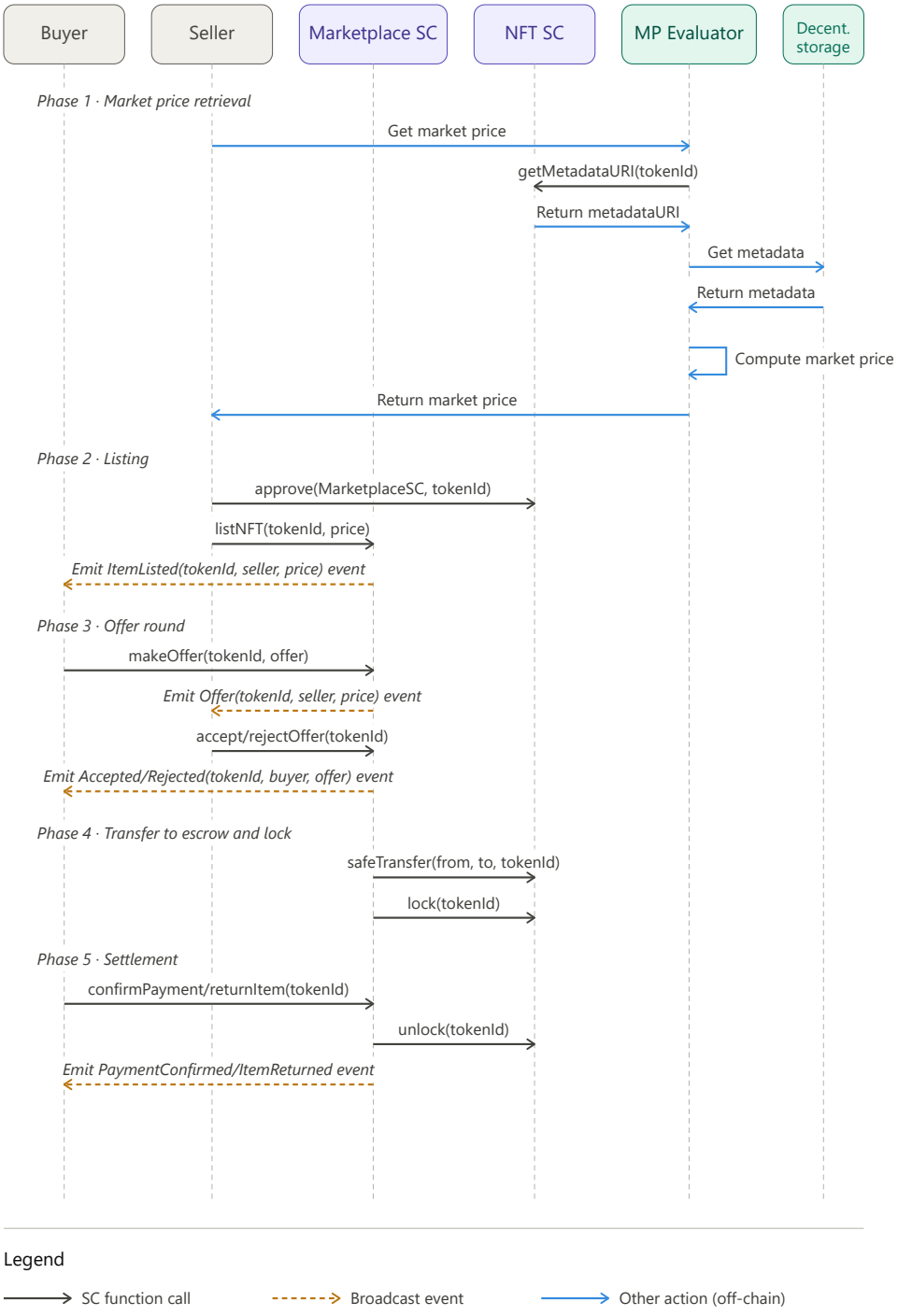

Fig. 6.2 Listing and purchasing process sequence diagram, reproduced from [1].

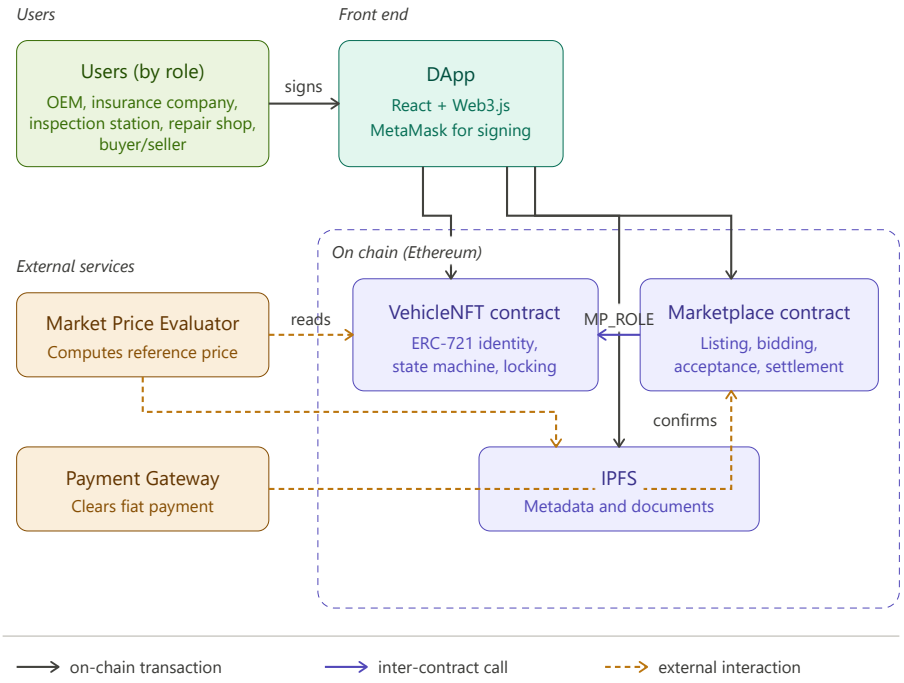

Fig. 6.3 **System component diagram.** On-chain components (the two smart contracts and the IPFS storage substrate) are grouped inside the dashed decentralized boundary.

The `VehicleNFT` contract stores the `Vehicle` struct mapping described in Section 6.3.3 and exposes the role-gated update functions `updateInsuranceCert`, `updateInspectionRep`, and `updateRepairLog`, together with the state-transition functions `setInsuranceUpdateRequired`, `setInspectionUpdateRequired`, `setRepairLogUpdateRequired`, and `reviewVehicle`. The `_beforeTokenTransfer` hook is overridden to enforce the locking primitive of Section 6.3.6.

When a state-modifying function is invoked, the contract emits a custom `Update` event carrying the `tokenId`, the caller's address, and a `type` field identifying which metadata category was modified. This allows off-chain indexers subscribed to the contract's events to maintain a synchronized view of each NFT's state without polling. ERC-4906 [132], proposed in March 2022 and reaching *Final* status in late 2022, defines a standardized `MetadataUpdate(uint256 tokenId)` event intended for exactly this signaling role. At the time the framework was implemented, the OpenZeppelin library stack on which it builds did not yet expose the `IERC4906` interface; support landed in OpenZeppelin Contracts v4.9.0 in May 2023, concurrent with the paper's submission. The framework's custom `Update` event carries a `type` field identifying which metadata category was modified, information the standardized event does not include but which downstream consumers can recover by

re-reading the updated metadata. A current implementation of the framework would emit `MetadataUpdate` alongside or in place of the custom event, making the NFT interoperable with the growing ecosystem of third-party indexers and marketplaces that react to the standardized signal. The retrospective lesson is narrow: standards compliance for interoperability is not a property established once at publication but a maintenance obligation against a moving target.

The `Marketplace` contract stores an `Item` struct for each listing, recording seller, asking price, MPE-suggested market price, listing state, current buyer, current offer, and payment deadline. It exposes `listItem`, `makeOffer`, `acceptOffer`, `rejectOffer`, `confirmPayment`, and `returnItem`, each guarded by the appropriate modifiers. The Marketplace contract is granted the `MP_ROLE` on the `VehicleNFT` contract at deployment, which is the only role permitted to toggle the locking flag.

In the 2023 prototype, the Marketplace contract interacts with a single `VehicleNFT` contract, corresponding to a single OEM's vehicles. The design admits a multi-OEM extension in which multiple `VehicleNFT` contracts (one per OEM) federate to a single shared Marketplace contract, at the cost of a more elaborate authorization scheme in which the Marketplace must hold `MP_ROLE` on each federated NFT contract. This extension was not implemented in the 2023 work and remains open.

6.5.2 Front-End DApp

The Decentralized Application providing user-facing interaction with the framework is implemented in JavaScript using React [133] for the user interface, the Web3.js library [134] for blockchain interaction, and JS-IPFS [135] for IPFS pinning and retrieval from the browser. The DApp exposes distinct account pages for each user role (OEM, insurance company, inspection station, repair shop, individual buyer/seller), each providing the subset of operations permitted to that role. A public marketplace page lists all vehicles currently on offer, with filtering by make, model, and price range. A vehicle page displays the full metadata of a selected NFT, including manufacturing data, current insurance and inspection state, and the cumulative repair log retrieved from IPFS.

Ethereum account management is delegated to MetaMask [136], which handles transaction signing and key custody on the user's behalf. A user interacting with the framework must therefore install the MetaMask browser extension and hold an Ethereum account; the usability implications of this requirement are discussed in Section 6.7.4.

6.5.3 Deployment

The prototype was deployed and tested on the Remix IDE [137], which provides a local EVM environment for contract compilation, deployment, and transaction execution. Gas consumption and execution outputs reported in Section 6.6 were measured against the Remix environment's gas accounting, which is deterministic and faithful to mainnet Ethereum's EVM semantics. Production deployment on Ethereum mainnet or any EVM-compatible L2 would be architecturally identical; the choice of Remix for the evaluation reflects the development-iteration efficiency of the local environment, not a limitation of the framework.

IPFS content was pinned to a local JS-IPFS node during development, with production deployment assuming that OEMs, insurance companies, inspection stations, and repair shops would each run their own pinning infrastructure or subscribe to a commercial pinning service for the content they issue. The framework does not depend on any specific pinning service, and any IPFS node retaining the content may serve retrieval requests.

6.6 Evaluation

The framework was evaluated on three dimensions: cost, performance, and security. The results reported below were measured in January 2023 under the conditions stated in each subsection; regarding cost specifically, the subsequent evolution of the Ethereum L2 landscape has changed the operational interpretation of the figures 6.6.1.

6.6.1 Cost Analysis

Gas consumption was measured by executing each function in the Remix IDE's local EVM simulator, whose gas accounting is faithful to Ethereum mainnet semantics. The monetary cost was computed at the conditions prevailing on 24 January 2023: a gas price of 16 gwei and an ETH-to-USD exchange rate that translated to approximately USD 2.6×10^{-5} per gas unit. Table 6.2 reports the results.

The most expensive individual function is `createToken` at USD 4.97 per vehicle, reflecting the storage cost of the `Vehicle` struct's immutable manufacturing metadata. The full seller-side workflow (`approve` + `listItem`) totals USD 3.63 per listing, and the full buyer-side workflow (`makeOffer` + `acceptOffer` + `confirmPayment`) totals USD 6.91 per successful purchase, with a seller-side additional cost of roughly USD 2.55 on the `confirmPayment` path or `returnItem` path. The per-transaction update costs, ranging from USD 0.76 for a state-transition signal to USD 3.11 for a full metadata update, are assumed by

Table 6.2 Gas cost, native cost, and throughput of the principal smart-contract functions, measured 24 January 2023 at 16 gwei and ETH = $1,625, with costs also re-priced onto Arbitrum One). Gas from [1].

				Cost (USD)		
Process	**Function**	**Gas**	**ETH**	**L1 (2023)**	**Arbitrum**	**tx/s**
Minting	`createToken`	200,412	0.003207	5.21	0.0062	12.47
Updating	`setUpdateRequired`	29,236	0.000468	0.76	0.0009	85.51
	`updateData`	125,308	0.002005	3.26	0.0039	19.95
	`reviewVehicle`	35,958	0.000575	0.93	0.0011	69.53
Selling	`approve`	49,333	0.000789	1.28	0.0015	50.68
	`listItem`	90,251	0.001444	2.35	0.0028	27.70
Purchasing	`makeOffer`	75,001	0.001200	1.95	0.0023	33.33
	`rejectOffer`	31,045	0.000497	0.81	0.0010	80.53
	`acceptOffer`	136,943	0.002191	3.56	0.0042	18.26
	`confirmPayment`	55,853	0.000894	1.45	0.0017	44.76
	`returnItem`	69,004	0.001104	1.79	0.0021	36.23

the updating role (insurance company, inspection station, or repair shop) and passed through to the consumer in the ordinary course of the service being documented.

The cost figures reported above reflect Ethereum mainnet conditions in January 2023, at a base gas price of 16 gwei and the contemporaneous ETH/USD rate. Subsequent infrastructure developments, notably the Dencun upgrade of 13 March 2024 and its introduction of blob-carrying transactions (EIP-4844 [32]), have reduced the data-availability cost that Layer-2 rollups incur on the base layer, and the empirical effect on end-user L2 fees has been a reduction of approximately one order of magnitude in steady state [138]. The gas units consumed by each function of the framework are independent of the deployment chain, as they are determined by EVM semantics; the monetary cost of purchasing those gas units on a current EVM-compatible L2 such as Arbitrum, Optimism, or Base is, however, smaller than the 2023 mainnet figures by roughly two orders of magnitude. The figures in Table 6.2 are retained without modification, as a record of the framework's computational footprint as measured and published; the operational implication is that what was, in 2023, the most serious obstacle to practical deployment has been removed by infrastructure improvements external to the framework.

6.6.2 Performance Analysis

The rightmost column of Table 6.2 reports the theoretical maximum throughput of each function, computed under the Ethereum post-Merge parameters of 30 million gas units per block

Table 6.3 Slither static analysis results on `VehicleNFT` and `Marketplace` contracts.

Severity	`VehicleNFT`	`Marketplace`
High	0	0
Medium	9	0
Low	—	—
Informational	> 20	8

and a 12-second average block time. Under these assumptions, the least-throughput-efficient function, `acceptOffer` at 136,943 gas, would saturate the chain at 18.26 transactions per second, well in excess of the transaction volume a second-hand vehicle market realistically generates. The most-throughput-efficient function, `setUpdateRequired` at 29,236 gas, would support 85.51 transactions per second, also far in excess of practical demand.

Throughput expressed as a fraction of chain capacity is thus not a binding constraint on the framework's deployment. The 2023 paper makes this point explicitly by noting that the operational bottleneck of the framework is not the chain's capacity but the latency users perceive under conditions of network congestion; with 12-second blocks, a listing transaction appears to complete within 12–24 seconds under normal load and within minutes under heavy load. L2 deployment, which produces block intervals of order seconds rather than tens of seconds, removes this perception-of-latency concern in addition to the cost reductions noted in Section 6.6.1.

6.6.3 Security and Vulnerability Analysis

The `VehicleNFT` and `Marketplace` contracts were analyzed with Slither [41], the static analysis framework used consistently across all engineering contributions (Section 2.6.2). Table 6.3 summarizes the findings.

No high-severity findings were reported on either contract. The nine medium-severity findings on `VehicleNFT` originate in the `SafeMath.sol` contract inherited from the OpenZeppelin library and reflect version-mismatch warnings between the framework's Solidity pragma and the pragma declared in the inherited library at the time of analysis; they do not represent vulnerabilities in the framework's own logic. The informational findings on both contracts are predominantly style-level observations about inherited code. Manual review against the vulnerability classes most relevant to the framework's role, access-control bypass, reentrancy, integer overflow/underflow, transaction-ordering dependency, and unbounded-iteration denial-of-service, found no additional concerns beyond those raised by Slither.

Three properties of the framework's design contribute to its security posture and are worth naming explicitly. First, the use of Solidity 0.8.x brings built-in integer-overflow protection (Section 2.6.1); arithmetic that would overflow reverts automatically rather than wrapping. Second, all state-modifying functions are guarded by role-based access control (Section 6.3.5) and by state-machine preconditions (Section 6.3.3), which means that malicious or malformed transactions fail cleanly rather than producing incorrect state. Third, the locking primitive (Section 6.3.6) prevents the most obvious class of escrow-period abuse by construction.

Threats not addressed by on-chain analysis, including front-running at the marketplace layer and the privacy implications of fully-transparent ownership history, are discussed in Section 6.7.3 of the following Discussion and Limitations section.

6.7 Discussion and Limitations

The engineering contribution presented in the preceding sections rests on design choices that this section examines critically. Each of the choices made by the framework is defensible in its own terms, as the sections above have argued; each also leaves an acknowledged residue of trust assumptions, scope restrictions, and open problems that subsequent work would need to address. This section makes the residue explicit. The discipline applied throughout is to state the problem plainly, identify the design response the framework encodes, and acknowledge what the response does not resolve. The purpose is not to list every objection an adversarial reader might raise, but to report the limitations the framework's author has judged most worth stating.

6.7.1 The Physical–Digital Binding Problem

The framework's first requirement (G1 of Section 6.3.1) is a persistent one-to-one binding between the on-chain NFT and the physical vehicle, anchored at the point of manufacture through the VIN recorded in the immutable manufacturing metadata. The cryptographic properties of the ledger guarantee that once the binding is recorded, it cannot be silently altered: an attempt to substitute a different VIN in the NFT, or to mint a second NFT against the same VIN, is detectable by any party that retains the earlier state. Within the ledger, the binding is as trustworthy as the consensus protocol itself.

The ledger does not, however, guarantee the *correctness* of the binding at the point of recording. An OEM that records an incorrect VIN, whether by error, by fraud, or under coercion, produces an NFT that is cryptographically perfect and referentially wrong. The

framework addresses this partially: by restricting minting to the OEM role (`OEM_ROLE` in Section 6.3.5), it ensures that the binding is recorded by the only party in the ecosystem with direct physical custody of the vehicle at the moment of manufacture and with a regulatory obligation to assign VINs correctly under ISO 3779 [44]. The OEM is, in a reasonable sense, the most credible candidate for this responsibility in the current structure of the automotive industry.

Two residual problems nevertheless remain. First, the framework inherits the oracle problem familiar from every blockchain system that ingests data from the physical world [39]: the correctness of an on-chain record reflects the correctness of the off-chain observation that produced it, and the ledger provides no independent verification of the observation. For a VIN recorded by a reputable OEM under regulatory oversight, the observation is more reliable than for most categories of on-chain data; it is not, however, cryptographically guaranteed. Second, the binding established at mint time does not extend to the vehicle's continuous physical identity during its service life. A vehicle may be rebuilt substantially after a collision, reconstructed from multiple donor vehicles, or re-stamped with a different VIN in the course of theft laundering, and the framework's NFT would continue to refer to whichever physical object claims the original VIN at a given moment. Periodic physical re-verification of the binding, during inspection events or ownership transfers, is a plausible extension of the framework, but one it does not currently incorporate.

The residue is that the framework's trust model is cryptographic *on-chain* and conventional *off-chain*, and the two cannot be made to coincide by any mechanism the framework itself provides. A complete vehicle-identity solution in the long term would likely combine the framework's on-chain cryptography with complementary off-chain attestations, for example, physical-unclonable-function component tags [139], periodic re-verification events recorded against the NFT, or consumer-visible audit trails from inspection stations. The framework is the on-chain half of such a solution, not the whole of it.

6.7.2 The OEM as Trust Anchor

The state machine of Section 6.3.3 grants the OEM a privileged position: the `reviewVehicle` function, which transitions a field from `UnderReview` to `Reviewed` on approval or back to `UpdateRequired` on rejection, is callable only by `OEM_ROLE`. The OEM is thus the final arbiter of whether an insurance certificate, an inspection report, or a repair log entry has been validly submitted. This is a trust concession whose defense and whose limits both deserve stating.

The defense is that the OEM is the ecosystem participant with the strongest combined claim to technical competence and economic accountability in the automotive context. It has the engineering knowledge to recognize whether a repair has been performed to specification; it has the documentation infrastructure to assess whether an inspection report is consistent with the vehicle's service history; it has a regulatory relationship with the jurisdictions in which its vehicles are sold that creates institutional accountability for the judgments it renders. No other participant in the framework's actor model, insurance companies, inspection stations, repair shops, owners, has the same combination of capabilities and exposure to consequences.

The limits, however, are real, and worth enumerating. The framework's architecture *fails open* if the OEM is compromised, in the sense that incorrect updates can be silently approved and correct updates silently rejected, with no mechanism within the framework to detect or correct the error. Three concrete scenarios are worth acknowledging. First, the OEM may cease to exist: the vehicle may outlive the company that manufactured it, through acquisition, bankruptcy, or market exit, in which case the `reviewVehicle` function becomes uncallable and the state machine silently fails. Second, the OEM may have an interest in the outcome of a review: a vehicle with a defect that the manufacturer prefers not to acknowledge creates an incentive to reject updates that would make the defect visible, and the framework offers no structural protection against this incentive. Third, the OEM may be coerced by a government, a large purchaser, or a litigation adversary, into rendering judgments that serve the coercer's interests rather than the ecosystem's.

None of these scenarios is hypothetical; each has an analogue in the current automotive industry, and each is a genuine limitation of the framework as designed. Alternatives to OEM-as-sole-trust-anchor are briefly surveyed here. *Multi-signature review boards* would require a threshold of signatures from multiple trusted parties, for example, the OEM plus an independent auditor plus a consumer association, before a state transition is accepted, reducing the impact of any single party's compromise at the cost of governance complexity. *Oracle-driven automatic validation* would allow updates originating from IoT instrumentation (for example, a connected vehicle's own telemetry) to bypass human review, restricting the OEM's role to adjudicating disputed cases rather than every update. *Zero-knowledge proofs of update correctness*, as proposed in the privacy-preserving traceability work of Sahai et al. [76], would allow an issuer to prove that an update satisfies a specified correctness predicate without the verifier needing to re-execute the underlying computation, reducing the trust placed in the OEM to trust in the cryptographic protocol itself.

None of these alternatives is free. The framework's choice of OEM-as-single-trust-anchor purchases architectural simplicity at the cost of centralized failure modes; the alternatives

purchase resilience at the cost of governance or protocol complexity. The right balance depends on the specific threat model of the deployment context, and the framework, as presented in this chapter, should be understood as one point in this design space rather than as the uniquely correct choice.

6.7.3 Threat-Model Boundaries

The framework's security analysis in Section 6.6 addresses threats internal to the blockchain layer: reentrancy attacks, access-control bypasses, integer overflows, and the other categories of vulnerability catalogued in the SWC Registry [124]. Several classes of threat fall outside this scope, and honest reporting requires acknowledging them explicitly.

The framework does not verify physical delivery. Once the marketplace smart contract transfers the NFT to the buyer and the payment is confirmed through the external gateway, the transaction is complete from the framework's perspective. Whether the physical vehicle has actually been handed over, whether it matches the description in the metadata, and whether any intermediate physical-custody transfer has preserved its state, are questions the framework's on-chain record does not answer. This was stated as an explicit scope restriction in the original paper [1], and it remains one: a fully end-to-end solution would need to integrate the framework with a physical-custody attestation mechanism, which this chapter's framework does not provide.

The framework does not defend against front-running at the marketplace layer. A purchase transaction broadcast to the Ethereum mempool is visible to any observer before it is confirmed, and a malicious miner or a sufficiently-incentivized bot could, in principle, reorder transactions to insert its own purchase ahead of a legitimate buyer's. In the framework's fiat-denominated marketplace, the financial incentive for such an attack is limited by the external payment gateway's off-chain clearing, which does not itself race with the on-chain transaction ordering. The attack surface is therefore narrower than for a cryptocurrency-denominated marketplace, but it is not absent, and a production deployment on a public chain with significant MEV activity [140] would need to adopt commit–reveal schemes or fair-ordering protocols as complementary protections.

The framework does not model the privacy of ownership history. Every NFT transfer is publicly visible on-chain, by design of the ERC-721 standard and by the framework's explicit choice to occupy the transparent pole of the privacy-transparency spectrum discussed in Section 6.1. Prospective buyers, law-enforcement investigators, and adversarial third parties alike can trace a vehicle's complete ownership history by inspecting the NFT's transfer events. For the framework's intended use case, where information asymmetry between

buyer and seller is the problem to be solved, full transparency is the feature. In use cases where a legitimate owner has a reasonable privacy interest in their ownership of a particular vehicle (high-net-worth individuals, vulnerable parties, or owners in jurisdictions where vehicle ownership can attract coercion), the framework's transparency becomes a liability. Pseudonymous addresses mitigate but do not eliminate this exposure. Chapter 8 returns to this tension as part of the cross-cutting privacy-transparency analysis.

6.7.4 Adoption Barriers

The framework is an architectural proposal, not a product, and several categories of obstacle separate an architectural proposal from a deployable system. Three are worth briefly naming; each lies outside the engineering scope of this chapter, and each is discussed at the thesis level in Chapter 8.

First, *legal recognition*: for the framework's on-chain ownership record to serve as the authoritative record of vehicle ownership, a jurisdiction's motor-vehicle authority must recognize it as such, either by migrating its registry onto the ledger or by treating the ledger as the primary source and its own registry as a derivative. No major jurisdiction has taken this step at the time of writing, and most operate centralized registries that would need to be displaced or paralleled rather than complemented. Second, *payment-gateway dependence*: the framework's fiat-denominated marketplace (Section 6.4) relies on an external payment gateway to clear the monetary component of each transaction. The gateway is, from the ledger's perspective, a trusted third party, and the decentralization property the ledger otherwise offers stops at the gateway's boundary. Third, *user-experience complexity*: the framework assumes that all actors, including private individual buyers and sellers, can manage Ethereum accounts, hold the L2 assets required to pay for gas, and interact with Web3 tooling competently. This assumption is reasonable for technically sophisticated users and unreasonable for the general market, and any deployment at scale would need a substantial investment in UX abstraction, custodial key management, and user education that is beyond this work.

These barriers are real, but they are not specific to this framework; they apply to any blockchain-based vehicle-identity proposal, and this chapter's design choices neither worsen nor alleviate them. The framework's contribution is architectural; the adoption path from architecture to production remains open work.

6.7.5 Integration with Federated Data Spaces

A further integration concern, more subtle than the adoption barriers just listed, arises from the framework's assumption that all vehicle-lifecycle data belongs on a single public ledger. In practice, the pre-sale phase of a vehicle's life, manufacturing, supply-chain assembly, first delivery, initial OEM-retained records, is governed by commercial confidentiality and by the federated-data-space conventions that the European Mobility Data Space and Catena-X establish for inter-firm automotive data exchange. The post-sale phase, which is what this chapter's marketplace addresses, is governed by consumer transparency. A deployable vehicle-identity system spanning the full lifecycle would therefore require a boundary mechanism: pre-sale data remains in the federated data space under permissioned access, and at the moment of first consumer ownership a curated subset is promoted to the public ledger where the framework of this chapter takes over. The framework as published assumes that its data is public from the outset, which is correct for its use case but is not the correct assumption for the complete vehicle lifecycle. The two-layer architecture, federated data space for the pre-sale phase, public ledger for the post-sale phase, with a well-defined promotion boundary at first consumer ownership, is a compositional open problem that neither this chapter nor Chapter 5 addresses in full. Chapter 8 returns to this as part of the thesis-level synthesis.

6.8 Chapter Summary

The framework presented in this chapter realises, with full implementation and evaluation, a vehicle-level analogue of the digital product passport that Chapter 5 argued is regulation-feasible. Two design moves carry the chapter's contribution. First, the vehicle's identity is reified as a single on-chain primitive (the dynamic NFT) capable of serving multiple applications, rather than as a marketplace-specific artifact; the locking mechanism is the most durable consequence of that move. Second, the framework is deliberately placed at the fully transparent pole of the privacy–transparency spectrum: all vehicle data is publicly verifiable, because transparency is the property that dissolves the information asymmetry characteristic of second-hand markets. Where the supplier-evaluation framework of Chapter 4 and the differentiated-access policy of Chapter 5 exercise the same architectural primitives under different access-control configurations, this chapter demonstrates the transparent extremum concretely. The retrospective commentary distributed across the chapter has engaged with the ERC-6551 token-bound-account pattern and the ERC-4906 standardized metadata-update signal; none of these subsequent developments invalidates the framework's architectural choices, and ERC-6551 has independently converged on the same settlement-safety pattern

that the framework's locking primitive originally implemented. Chapter 8 returns to the architectural bets identified here in its synthesis across all four contributions.

Chapter 7

Blockchain-Based Pothole Management for Connected Vehicles

This chapter is based on the following manuscript, currently under review at *Future Generation Computer Systems*:

A. Butera, V. Gatteschi, V. Villa, and M. Domaneschi, "Design and Evaluation of a Blockchain-Based Pothole Management System for Smart Cities and Connected Vehicle Networks," submitted March 2026.

7.1 Chapter Introduction

The vehicle identity system of Chapter 6 reifies a vehicle's lifecycle as an on-chain record but is structurally blind to the vehicle's relationship with the road network on which it operates. A modern connected vehicle does not only possess an identity; it is also a mobile sensing platform whose cameras, IMUs, and GPS receivers (equipment installed for autonomous driving functions) generate a continuous stream of observations about the infrastructure it traverses. This chapter takes up the engineering problem of routing this byproduct of operation into a trustworthy coordination layer for civic infrastructure management.

The specific defect the chapter addresses is the pothole. The choice is motivated by three circumstances. First, potholes are the dominant failure mode of paved surfaces: in the United States alone, deteriorated road conditions cost motorists approximately $130 billion annually in extra vehicle repair and operating costs [141], of which roughly $26.5 billion is attributable specifically to pothole-related damage [142]; delayed maintenance

compounds the cost, with research indicating that ignoring road distresses can escalate reconstruction costs to three times the original maintenance expenditure [143]. Second, pothole detection is, narrowly understood, already solved: computer-vision models on commodity edge hardware achieve real-time detection from vehicle-mounted cameras [11, 10], and accelerometer-based detection from consumer smartphones has been demonstrated for more than a decade [107]. Third, despite this technical maturity, the coordination problem remains unsolved. Detection feeds into proprietary municipal servers whose prioritization logic is opaque, whose response times are unaccountable, and whose participation incentives are nonexistent (Gap 4, Section 1.2).

The contribution is a three-contract blockchain coordination layer with an off-chain IPFS image store and an on-chain multi-factor priority function, serving two reporter populations, connected vehicles and citizen smartphones, through a single gasless interface. The admission of the vehicle pathway is a thesis-level design choice as much as a practical one: the vehicle identity framework of Chapter 6 provides an authentication primitive that this system can reuse, so a vehicle reporter can be distinguished from an anonymous wallet by its on-chain NFT identity at no additional architectural cost. The PotholesRegistry contract receives signed reports, enforces geographic and format validation, detects spatial duplicates via a grid index, and drives repair-status transitions; the PotholesToken contract implements an ERC-20 fungible reward (the Pothole Blockchain Coin, PBC) whose issuance is controlled exclusively by the registry, making rewards a mechanical consequence of validated state transitions rather than a discretionary administrative act; and the PotholesForwarder contract implements the ERC-2771 meta-transaction pattern, allowing a reporter with no cryptocurrency holdings to submit a report that a third party (the municipality or a delegated relayer) pays the gas for. Images are stored on IPFS with content-addressed CIDs recorded on-chain; repair priority is a pure function of on-chain state, observable and recomputable by any party.

7.2 Paper-Specific Related Work

The general landscape of pothole-detection technologies, blockchain applications in smart cities, and crowdsourced infrastructure reporting has been surveyed in Chapter 3, Section 3.4, and the systems most directly comparable to this contribution, FixMyStreet, SeeClickFix, Waze, Rumpa et al.'s *InfraChain*, Matzutt et al.'s blockchain photo-reporting, Mubarak et al.'s AI-blockchain pothole framework, Knuuti et al.'s gamified road asset management, and the broader mobile crowd sensing tradition exemplified by Tian et al., are summarized in Table 3.2. This section confines itself to the narrower body of work that directly informs the design decisions of Sections 7.3 and 7.4: meta-transactions as an onboarding primitive, token-

based incentives for public-good reporting, on-chain spatial indexing, and the architectural lineage of crowdsourced civic platforms.

7.2.1 Meta-Transactions as an Onboarding Primitive

The design decision to make pothole reporting gasless rests on the meta-transaction pattern, formalized for Ethereum in EIP-2771 "Secure Protocol for Native Meta Transactions"[37]. In its standard form, the pattern decouples *transaction signing* from *transaction submission*: a user signs a typed-data payload off-chain following EIP-712[144] conventions, and a separate *relayer* address submits the signed payload as an Ethereum transaction, paying the gas itself. A trusted forwarder contract verifies the user's signature, extracts the user's address from the appended trailing bytes, and invokes the target contract on the user's behalf. From the target contract's perspective, the call appears to originate from the user, even though the gas was paid by the relayer.

Two practical deployments establish the engineering viability of this pattern at scale. The *OpenGSN* (Gas Station Network) project[145] provides a reference implementation of the EIP-2771 forwarder along with a network of independent relayers willing to pay gas for transactions to participating contracts, in exchange for fees collected through orthogonal mechanisms. Biconomy[146] offers a commercial-grade variant in which a sponsor (typically the application operator) pre-funds gas costs for its users, removing the need for users to interact with relayer markets at all. Both deployments demonstrate that gasless onboarding is no longer experimental: it has been operational across mainnet and Layer-2 networks for several years, with measurable uptake in NFT minting, decentralized identity, and play-to-earn applications.

The relevance to the pothole domain is immediate but not symmetric to those prior deployments. A play-to-earn user has, by self-selection, already accepted the cognitive overhead of cryptocurrency wallets; gasless transactions remove a peripheral friction. A citizen reporting a pothole has accepted nothing of the kind, and any non-trivial onboarding step: "install MetaMask," "acquire a small amount of ETH for gas," "approve this token allowance"; it is a hard filter that reduces participation by orders of magnitude. The architecture proposed in Section 7.3.7 therefore treats the meta-transaction pattern not as an optimization but as a constitutive primitive: the system either offers gasless reporting from the first interaction or it does not function as a civic-coordination layer at all. The closest precedent for using meta-transactions specifically as a civic-onboarding primitive, rather than as a Web3-application convenience, is, to the best of our knowledge, absent from the literature; this absence is itself a contribution of this work.

7.2.2 Token-Based Incentives for Public-Good Reporting

The design lineage for the PBC token introduced in Section 7.3.3 draws from the broader experience of decentralized incentive mechanisms in mobile crowd sensing (MCS) and the more recent decentralized physical infrastructure network (DePIN) movement. Within MCS, Tian et al.[88] surveyed the design space and identified three persistent challenges: data quality assurance, privacy protection, and sustainable incentive design. Subsequent work proposed concrete mechanisms: Li et al.[89] combined anonymous reporting with dynamic monetary incentives that adjust based on contribution scarcity; Xiong et al.[90] hybridized reputation-based and service-return incentives to discourage low-quality submissions. These mechanisms address the participation question, but uniformly assume a centralized operator who controls the incentive logic, the participant database, and the payout treasury. The transparency problem these mechanisms create (the participant must trust the operator's claims about reward computation) is the precise problem that on-chain incentive logic is designed to dissolve.

The DePIN literature, though largely outside the academic record and confined to whitepapers and platform documentation, nevertheless provides the most relevant operational template. Networks such as Helium (decentralized wireless coverage), Hivemapper (street-level imagery), and DIMO (vehicle telemetry) each issue a fungible token to participants who contribute data or coverage to a public network, with token issuance governed by smart contracts that condition rewards on verifiable contributions. The conceptual move that DePIN makes, treating data contribution as a measurable on-chain event triggering automatic reward, is the move adopted in Section 7.3.3, with one substantive modification appropriate to the civic context: the beneficiary in DePIN systems is a private network operator selling a service back to the market, whereas in pothole management the beneficiary is the municipality and, transitively, the public. This modification reshapes one design question that the DePIN literature largely sidesteps: the question of why a fungible token is preferable to direct fiat compensation. The answer this work proposes, developed in Section 7.3.3, is that the minting-on-validation pattern produces an *auditable* reward record, no party (including the municipality) can issue PBC without the corresponding on-chain validation event, rather than a record that is merely auditable in principle but mediated in practice by a payroll system the citizen cannot inspect.

7.2.3 On-Chain Spatial Indexing

The grid-based spatial duplicate detection scheme of Section 7.3.4 addresses a design constraint that has received remarkably little explicit treatment in the blockchain literature:

how to detect whether two reports refer to the same physical location, using only computation that is feasible inside an Ethereum transaction. The off-chain literature is rich. Geohashing encodes latitude–longitude pairs as Base32 strings whose shared prefixes indicate spatial proximity. Uber's H3 library[147] partitions the globe into hexagonal cells with uniform-area properties superior to rectangular grids. Quadtree-based indices underlie almost every production geospatial database. None of these schemes, however, is well-suited to on-chain implementation: H3's cell-derivation arithmetic involves trigonometric operations that are prohibitively expensive in EVM gas terms; geohash decoding requires bit manipulation patterns that are awkward in Solidity; and quadtrees require dynamic data structures whose storage cost scales unfavorably with depth.

The design adopted in Section 7.3.4, a fixed Cartesian grid in scaled-integer coordinates, with cell membership computed by integer division, trades geospatial sophistication for gas efficiency. The trade is honest: at urban latitudes, a fixed-degree grid produces cells whose physical dimensions vary by a few percent across the latitude range of a single city, which is well within the tolerance required for pothole-duplicate detection (where the relevant precision is on the order of a few meters). Section 7.5.2 returns to the residual question of how to choose the grid size, which is exposed as a deployment-time parameter rather than fixed at the contract level. This section's contribution is therefore not novelty in spatial indexing per se but the recognition that on-chain spatial indexing is a distinct sub-problem worth solving with constraint-appropriate tools, rather than naively porting off-chain libraries that the EVM's gas model makes infeasible.

7.2.4 Architectural Lineage of Crowdsourced Civic Platforms

The civic-reporting platforms most directly comparable to this contribution (FixMyStreet[103], SeeClickFix[104], and Waze[105]) have been characterized in Chapter 3, Section 3.4, with the specific architectural comparison deferred to Table 3.2. Two narrower precedents within the academic blockchain literature warrant mention here for their direct architectural relevance. Matzutt et al.[101] demonstrated, in a poster track at CCS 2023, blockchain-anchored storage for citizen-reported street problems with priority scheduling; the contribution sits architecturally close to this PotholesRegistry in that it commits report metadata to an immutable ledger, but it omits both a token-incentive layer and any spatial-deduplication mechanism, and it does not address the citizen-onboarding problem that Section 7.2.1 treats as constitutive. Mubarak et al.[102] combined EfficientDet-based detection with blockchain storage of detection results, again without prioritization, deduplication, incentives, or gasless onboarding. Rumpa et al.'s *InfraChain*[100] most closely approaches the architectural ambition of this work, using embedded sensors and a permissioned ledger for coordinated

infrastructure monitoring; the principal differences are this system's commitment to a public chain (with the access-control inversion that requires), its native ERC-20 incentive layer, and its explicit dual-reporter architecture admitting both vehicle and citizen pathways through a unified meta-transaction interface.

Reading these antecedents together, the architectural pattern that distinguishes this contribution is not the introduction of any single component (meta-transactions, ERC-20 incentives, IPFS-backed storage, and blockchain-anchored civic reporting all exist independently in the literature reviewed above) but the composition of these components into a coordination layer in which all four design bets identified in Section 7.1 hold simultaneously: gasless reporting reaches the entire citizen population, incentive issuance is auditable on-chain, repair priority is computable from public state, and the system serves vehicle and citizen reporters through a single interface. The remainder of this chapter develops the system that realizes this composition.

7.3 Proposed System

This section presents the proposed pothole-management system. It opens with a brief statement of the design goals against which the rest of the section may be evaluated, then proceeds from the system architecture (Section 7.3.1) through the three smart contracts (Section 7.3.2), the ERC-20 token-incentive mechanism (Section 7.3.3), the grid-based spatial duplicate-detection scheme (Section 7.3.4), the multi-factor priority-scoring algorithm (Section 7.3.5), the two reporter pipelines (Section 7.3.6-Section 7.3.7), and the IPFS data-management layer (Section 7.3.8). Justifications for the platform choice (Ethereum, EVM, OpenZeppelin) are not repeated here; they were developed at length in Chapter 2, Section 2.2.

Design goals. The system was designed against six requirements, each of which is justified in the introduction (Section 7.1) or in the paper-specific related work (Section 7.2): **(G1) Gasless reporting** for citizens and vehicle operators with no cryptocurrency holdings; **(G2) Sybil-resistant incentives** that reward duplicate reports as community validation without paying full reward more than once per location per reporter; **(G3) Transparent prioritization** in which repair order is a deterministic function of public on-chain state, recomputable by any observer; **(G4) Hybrid storage** keeping coordination state on-chain and bulk media (images, contextual metadata) on IPFS, with cryptographic content addressing as the bridge; **(G5) Multi-reporter pathways** accepting authenticated reports from connected vehicles and citizen smartphones through a single contract interface; **(G6) Municipal integration without privileged off-chain access** so the municipal dashboard, like any

other client, consumes only public on-chain state. The remainder of Section 7.3 can be read as the engineering response to these six requirements, each of which is revisited explicitly in the chapter summary (Section 7.6).

7.3.1 System Architecture

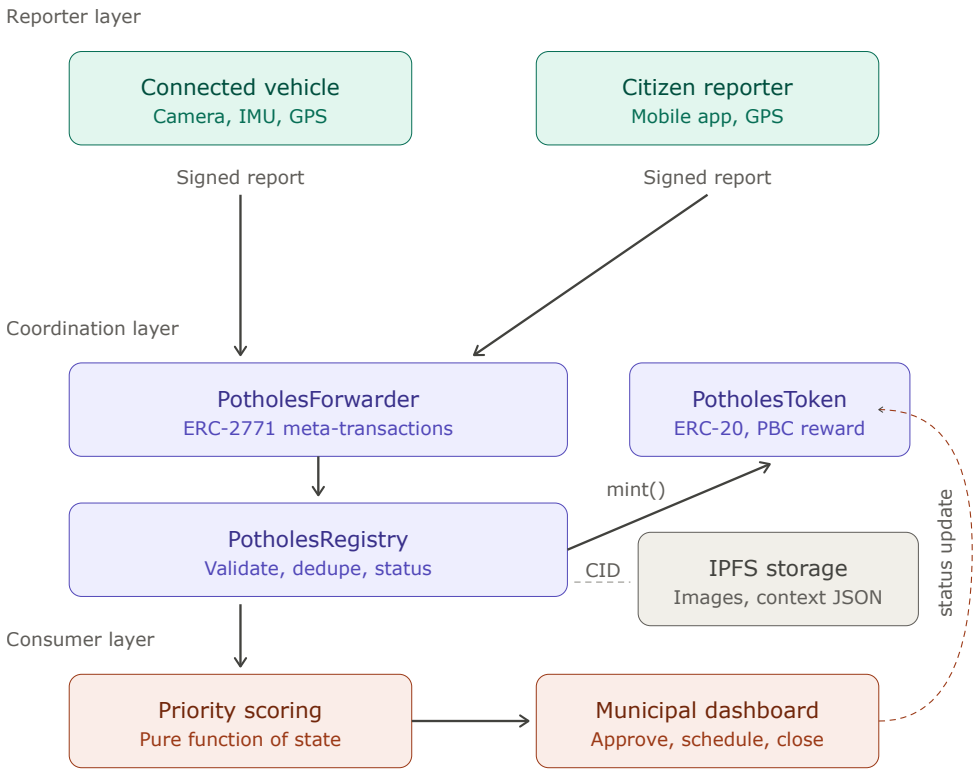

Fig. 7.1 System architecture

The system architecture, depicted in Figure 7.1, organizes participants and components across three layers. At the *reporter layer*, two distinct populations submit pothole reports: connected vehicles equipped with onboard cameras, inertial measurement units, and GPS receivers, and citizens using a mobile or web application. Both reporter populations sign report payloads off-chain using EIP-712[144] typed-data signing and submit them to the *coordination layer* through the same gasless interface. The coordination layer comprises three smart contracts, POTHOLESFORWARDER, POTHOLESREGISTRY, and POTHOLESTOKEN,

together with the IPFS network used as off-chain storage for images and contextual metadata. At the *consumer layer*, the municipal dashboard, repair-scheduling software, and any third-party observer consume the public on-chain state, reading reports, priorities, and token transfers without any privileged access path.

The operational lifecycle of a report consists of five stages. (i) A reporter detects a pothole through automated computer vision (vehicle) or manual capture (citizen). (ii) The reporter's client uploads a JSON metadata package, including the photograph and the geographic context retrieved from OpenStreetMap, to IPFS, receiving a content identifier (CID) in return. (iii) The reporter signs an EIP-712 message containing the scaled coordinates and the CID, and the POTHOLESFORWARDER relays the signed call to POTHOLESREGISTRY on a paying-relayer's behalf. (iv) POTHOLESREGISTRY validates the geographic boundary, computes the location hash for grid-based deduplication, classifies the report as new or duplicate, and emits the corresponding event. (v) Upon municipal validation (status transition to `InProgress`), POTHOLESREGISTRY invokes POTHOLESTOKEN to mint the appropriate reward; upon completion, a further bonus is minted. The lifecycle thereby closes on-chain, with all five stages independently auditable.

The architectural decisions encoded in Figure 7.1 reflect the three bets named in Section 7.1: the forwarder is the locus of the meta-transaction-as-onboarding bet, the registry-as-sole-minter relationship is the locus of the minting-on-validation bet, and the priority-scoring layer is the locus of the priority-as-pure-function bet. The remainder of Section 7.3 expands each of these bets into concrete contract-level mechanisms.

7.3.2 Smart Contract Design

The system employs a three-contract modular architecture deployed as separate Solidity contracts but linked through compile-time addresses and an explicit ownership relationship that constrains privileged operations.

The POTHOLESREGISTRY contract

POTHOLESREGISTRY is the application-logic core, managing the complete lifecycle of pothole reports from initial submission through final resolution. Its principal data structure is the `PotholeReport` struct, which encapsulates the unique identifier, geographic coordinates (latitude and longitude as signed integers scaled by 10^6), the IPFS content hash of the associated metadata document, the duplicate count tracking community validation, the submission timestamp, the reporter's blockchain address, and the current lifecycle sta-

Table 7.1 Report lifecycle states and reward consequences.

Status	Meaning	Effect
`Reported`	Awaiting municipal review	Age score accumulates
`InProgress`	Repair work initiated	10 PBC minted to reporter
`Completed`	Repair finished	5 PBC bonus; location released
`Rejected`	Invalid or unactionable	No reward; location released

tus. The `PotholeStatus` enumeration defines four mutually exclusive states: `Reported`, `InProgress`, `Completed`, and `Rejected`, with the lifecycle and reward consequences summarized in Table 7.1.

Storage employs four mappings optimized for distinct query patterns: a primary `reports` mapping from identifier to struct enabling constant-time retrieval; a `locationToReportId` mapping that implements grid-based deduplication by hashing geographic coordinates to report identifiers; `registeredCitizens` and `authorizedMunicipals` for role-based access control; and a nested `userReportedLocation` mapping that prevents an individual reporter from claiming the duplicate reward more than once for the same physical location.

Access control follows the OpenZeppelin `Ownable` pattern[129]: the deploying municipality holds the owner role and may register or revoke citizens and municipal authorities; municipal authorities are the only addresses authorized to advance a report through its lifecycle; and registered citizens are the only addresses authorized to submit reports. Three safety mechanisms protect the contract: `ReentrancyGuard` from OpenZeppelin guards the token-minting paths against reentrancy; geographic-boundary checks reject reports outside the configured municipal jurisdiction at the validation stage; and inheritance from `ERC2771Context` routes meta-transaction calls through the trusted forwarder while preserving the original signer's identity for authorization checks.

The POTHOLESTOKEN contract

POTHOLESTOKEN is a standard ERC-20[33] fungible token, branded as the *Pothole Blockchain Coin* (PBC), with eighteen decimals. The contract exposes a constrained `mint` interface whose access control is the design's central anchor: at deployment time, contract ownership is transferred from the deployer to the POTHOLESREGISTRY address, after which the registry is the sole address authorized to mint PBC. No party, including the deploying municipality, retains the ability to issue tokens directly. PBC issuance is therefore conditional on *exactly* the on-chain state transitions that POTHOLESREGISTRY authorizes, the validation event and

the completion event, and supply expansion is mechanically tied to verified infrastructure outcomes rather than to a discretionary treasury.

The PotholesForwarder contract

PotholesForwarder implements the ERC-2771[37] meta-transaction pattern, separating transaction signing from transaction submission. A reporter signs an EIP-712 typed-data payload containing the intended call to PotholesRegistry; the forwarder verifies the signature, extracts the original signer from the appended trailing bytes, and dispatches the call. PotholesRegistry and PotholesToken both inherit from `ERC2771Context`, which overrides the `_msgSender()` primitive so that downstream access-control checks evaluate the original signer rather than the forwarder address. The relayer that submits the on-chain transaction pays the gas in ETH; the user pays nothing. In the deployment envisaged for Turin (Section 7.4), the relayer is operated by the municipality, so the gas budget is municipal infrastructure spending.

7.3.3 ERC-20 Token-Incentive Mechanism

The PBC reward structure is three-tiered. An *original reporter* (the first to submit a report for a previously unreported grid cell) receives **10 PBC** when the report is validated by a municipal authority (status transition `Reported` → `InProgress`), and an additional **5 PBC** *completion bonus* when the repair is marked complete (`InProgress` → `Completed`). *Duplicate reporters*, who independently submit reports falling into a grid cell that already has an open report, receive **0.5 PBC** each, but only after the *original* report reaches `InProgress`; until that point, duplicate confirmations accumulate without paying out. An individual reporter may claim the duplicate reward at most once per location, enforced by the `userReportedLocation` mapping introduced in Section 7.3.2.

Three properties of this structure are worth making explicit, because each addresses a distinct design requirement. First, the 20:1 ratio between the original-reporter reward (10 PBC) and the duplicate reward (0.5 PBC) preserves a clear incentive hierarchy: discovery is substantially more valuable than confirmation, but confirmation is not free. Even in a scenario where a single original report attracts twenty independent duplicates, a high-engagement scenario that this work treats as desirable, the original reporter retains the larger reward, and duplicate-farming is not more profitable than original discovery. Second, the deferral of duplicate rewards until municipal validation conditions all reward distributions on the same approval gate. A spammer who submits fabricated duplicates before a real reporter's submission is validated incurs no reward whatsoever if the original report is later

`Rejected`. Third, the completion bonus extends reporter engagement across the full repair lifecycle, which may span weeks or months in practice, while simultaneously rewarding only those reports that lead to actual physical repair.

The deeper architectural commitment underlying this structure is the one introduced in Section 7.3.2: POTHOLESREGISTRY, and only POTHOLESREGISTRY, can mint PBC. The reward structure is therefore not merely a policy implemented on top of a treasury, recoverable by editing a payout schedule. It is a structural property of the contracts: every PBC that exists corresponds to an on-chain validation or completion event recorded in the registry, and the relationship between PBC supply and verified municipal action is computable by any observer.

7.3.4 Grid-Based Spatial Duplicate Detection

The duplicate-detection mechanism addresses a constraint that distinguishes on-chain spatial reasoning from its off-chain counterparts: the EVM lacks native floating-point arithmetic, and the gas cost of porting general-purpose geospatial libraries to Solidity is prohibitive. The system's response, justified in Section 7.2.3, is a fixed Cartesian grid in scaled-integer coordinates.

GPS coordinates are first scaled by a factor of 10^6 and stored as signed integers, preserving six decimal places and providing sub-meter precision at the equator. A latitude of 45.071234 is represented as the integer `45071234`. Each municipal deployment configures four immutable boundary parameters at construction time (minimum and maximum scaled latitude and longitude) which together define the geographic envelope of valid reports; submissions outside this envelope are rejected at the contract level, eliminating cross-jurisdictional pollution and a class of GPS-spoofing attacks at near-zero gas cost.

The grid is parameterized by a single integer `precision` value, also fixed at deployment, which sets the scaled-coordinate edge of each grid cell. Given a scaled coordinate pair (lat, lng), the containing cell is computed by integer division:

$$gridLat = \lfloor lat/precision \rfloor, \qquad gridLng = \lfloor lng/precision \rfloor, \tag{7.1}$$

and the resulting pair is hashed into a 32-byte location identifier:

$$locationHash = \text{keccak256}(gridLat, gridLng). \tag{7.2}$$

Any two coordinates that fall into the same grid cell produce identical location hashes; the duplicate check is therefore a single mapping lookup against `locationToReportId`,

with $O(1)$ gas cost independent of the size of the report database. A precision value of `100` corresponds to roughly 11 m at urban latitudes, suitable for fine-grained dedup; `10000` corresponds to roughly 1.1 km, suitable for coarse coverage. For the Turin deployment, the prototype uses an intermediate precision tuned empirically to GPS accuracy in the city's road network; the choice is revisited as a governance question in Section 7.5.2.

Two limitations of the fixed-grid approach are intrinsic and worth naming. The first is the boundary effect: a single physical pothole located near a cell boundary may produce reports that fall into adjacent cells due to GPS noise, in which case the system records two reports rather than one. This is a structural consequence of any discrete spatial index and cannot be eliminated without abandoning the gas-efficiency property; mitigations through finer-grained precision or hybrid neighbor-cell checks are discussed in Section 7.5.2. The second is that releasing a location upon `Completed` or `Rejected` status, necessary to allow re-reporting of recurring potholes, creates a small race window in which two independent reports of a re-emerging pothole may both be classified as new originals. The system tolerates this for the same reason: it is a structural property of any cell-release scheme, the alternative is permanent location lockout, and the consequences of a small over-reward are operationally dominated by the consequences of permanent lockout.

7.3.5 Multi-Factor Priority-Scoring Algorithm

The priority score is the public function from on-chain state to repair order. It combines two dynamic factors that evolve with time and community engagement (report age, duplicate count) with eight static factors captured at submission and stored in IPFS (road classification, speed limit, lane count, surface material, lighting condition, transit-stop proximity, critical-facility proximity, general-building density). Static factors are retrieved at submission time through OpenStreetMap's Overpass API and bundled into the IPFS metadata document, ensuring no off-chain query is required at scoring time. Table 7.2 summarizes the components.

The complete priority is the sum of all components:

$$Priority = ageScore + 15\ln(1 + n_{dup}) + \sum_{i \in \mathscr{S}} w_i, \tag{7.3}$$

where $\mathscr{S}$ is the set of static-context factors and w_i their respective weights. The logarithmic scaling on duplicate count is deliberate: it ensures that the first few independent confirmations carry substantial weight (five duplicates yield ≈ 27 points) while preventing viral-reporting events from saturating the priority queue (one hundred duplicates yield only ≈ 69 points).

Table 7.2 Priority-score components.

Factor	**Range**	**Rule**
Age (`Reported` only)	0–30	1.5 pts/day, capped at 30
Duplicate count	0–70	$15 \times \ln(1+n)$
Road classification	2–15	By road type
Speed limit	1–12	By speed zone
Lane count	2–8	2 pts per lane
Surface material	1–6	By surface type
Lighting	1–6	By lighting level
Transit stops within 50 m	0–18	6 pts per stop
Critical facilities within 100 m	0–20	10 pts per facility
General buildings within 100 m	0–8	4 pts per building
Maximum	**193**	

Numerical scores then map to four discrete tiers, Critical (≥ 120), High (80–119), Medium (45–79), Low (< 45), each suggesting a municipal response timeframe ranging from 24-72 hours to routine cycle, summarized for the reader in the prototype implementation but not enforced by the contract.

The architectural commitment that this scoring scheme encodes is the third bet named in Section 7.1: priority is a pure function of public on-chain state. Any third party with access to Ethereum and the IPFS network can recompute the priority score for any report at any time, and any disagreement with the municipal queue can be diagnosed against a publicly observable specification. The scheme is not opinionated about *whether* the published weights are correct (they are configurable at deployment, and Section 7.5.2 treats the weight-selection question as a governance surface rather than a technical contribution) but it is opinionated about *whether the weights, once chosen, are visible*.

7.3.6 Vehicle-Based Reporting Pipeline

The vehicle pipeline routes detections from a connected vehicle's sensor suite into the same gasless interface used by citizen reporters. A front-facing camera processes road-surface frames through a deep learning model (YOLO or EfficientDet variants are envisaged, consistent with the related-work survey of Section 7.2), and the inertial measurement unit provides complementary vibration evidence used in sensor fusion to suppress false positives from speed bumps, manhole covers, and railroad crossings. Detection runs on edge hardware on the vehicle itself; only confirmed detections, not raw video, leave the vehicle, addressing both bandwidth and privacy concerns associated with continuous video upload.

When a detection's confidence exceeds a configurable threshold, the vehicle's onboard agent assembles a report package containing scaled GPS coordinates, the detection's confidence score, and the IPFS CID of the captured image, signs the package using the vehicle's blockchain wallet, and transmits the signed payload to the POTHOLESFORWARDER via Vehicle-to-Backend (V2B) communication. From the contract's perspective, a vehicle-originated report is indistinguishable in form from a citizen-originated report: both arrive through the forwarder, both are signed under EIP-712, and both authenticate against the `registeredCitizens` or, in a natural extension proposed in Section 7.5.4, an analogous `registeredVehicles` mapping. The *semantic* distinction between the two reporter types is preserved through the signing address: vehicle wallets are distinct from citizen wallets, allowing downstream analytics to disaggregate the populations without touching the consensus layer.

7.3.7 Citizen-Based Reporting Pipeline

The citizen pipeline is the component most directly addressing design goal G1 (gasless reporting). A citizen-facing mobile or web application requests device location at launch, populates the report's coordinate fields with the current GPS position, and presents an interactive map for verification or manual adjustment. The user captures a photograph through the device camera or selects an existing image, and the application performs three client-side validations before any on-chain action: it checks that coordinates fall within the contract's configured boundary envelope (avoiding wasted gas on out-of-jurisdiction submissions); it verifies basic image quality (resolution, brightness, file size); and it confirms wallet connectivity. The image and the OpenStreetMap-retrieved geographic context are then bundled into a JSON metadata document and uploaded to IPFS, returning a master CID.

The user signs an EIP-712 typed-data message containing the scaled coordinates and the master CID using their wallet's private key (typically through a MetaMask or WalletConnect prompt), and the signed payload is transmitted to the POTHOLESFORWARDER, which dispatches it to POTHOLESREGISTRY as described in Section 7.3.2. From the user's perspective, the entire flow involves a single signing prompt and no payment; the cryptocurrency machinery is abstracted to a single yes/no decision. This is the irreducible cognitive load that the system imposes on a non-crypto-native citizen.

7.3.8 IPFS Data-Management Layer

Storing photographs on-chain is economically infeasible: at the time of writing, a single 32-byte hash costs approximately 20,000 gas, while a 1 MB image would cost orders of magnitude more if it could be stored at all. The system's hybrid response stores only the minimal coordination state on-chain, such as identifiers, scaled coordinates, the IPFS CID, the duplicate count, the timestamp, the reporter's address, and the lifecycle status, and routes everything else through IPFS[38].

Each report's IPFS payload is a JSON document combining the photograph's CID, a textual description, the submission timestamp, the geographic context returned by the OpenStreetMap Overpass query (road characteristics, transit-stop proximity, critical-facility proximity, general-building density), and the geometric estimate of the pothole if one is provided by the detection pipeline. The hash of this JSON document is itself a CID, the *master CID*, and it is the master CID that POTHOLESREGISTRY stores in `ipfsHash`. Content addressing therefore guarantees that the on-chain record and the off-chain payload are cryptographically bound: any modification to the IPFS payload would change its hash, and the on-chain reference would no longer resolve. The architecture inherits a well-understood limitation of IPFS (availability depends on at least one node continuing to pin the content) which Section 7.5 returns to. In the deployment envisaged for Turin, the municipality assumes the pinning obligation as part of the same operational commitment that funds the forwarder's gas budget; this parallels the pinning role that the OEM plays in the vehicle-identity framework of Chapter 6 and is one of the structural symmetries that Chapter 8 develops.

7.4 Evaluation

This section reports the engineering evaluation of the proposed system. The evaluation is structured to address the design goals of Section 7.3 from four angles: correctness via comprehensive testing (Section 7.4.2), per-operation gas consumption on the EVM (Section 7.4.3), deployment-cost projections across L1 and L2 networks at municipal scale (Section 7.4.4), and a structured assessment of the security posture across access control, reentrancy, input validation, meta-transaction integrity, and token-supply control (Section 7.4.5). The comparison with traditional municipal infrastructure-management systems, which is properly a positioning rather than an evaluation question, is deferred to Section 7.5.

Table 7.3 Test-suite summary across the contract suite.

Contract	Tests	Coverage areas
PotholesRegistry	63	Submission, status, rewards, access control
PotholesToken	40	ERC-20, minting, pausing, permits
PotholesForwarder	5	Meta-transactions, batch execution
PotholesGasReporter	19	Gas benchmarking for all operations
Total	**127**	All tests passing

7.4.1 Experimental Setup

The smart-contract suite was developed and tested under Hardhat 3.0.6 with Solidity 0.8.28, against OpenZeppelin Contracts 5.4.0[129] for access-control, reentrancy-guard, ERC-20, and ERC-2771 base components. Tests were written using Hardhat's testing harness with Foundry-style assertions, allowing unit, fuzz, and invariant tests to share a common runtime. Gas consumption was measured using a dedicated gas-reporter test suite that exercises every contract function under realistic usage scenarios. Because the EVM's gas accounting is deterministic across compatible networks, the per-function measurements collected on a local Hardhat node project directly to deployment costs on Ethereum mainnet, on Layer-2 rollups (Arbitrum, Polygon), and on alternative EVM-compatible chains (BNB Smart Chain), once each network's gas-price and native-token-price assumptions are applied.

The evaluation does not include a field deployment with real reporters and a real municipality; the limitations imposed by this scope are addressed in Section 7.5.

7.4.2 Test Coverage

The complete test suite comprises 127 tests across the three application contracts and a dedicated gas-reporter contract. Table 7.3 summarizes the distribution.

The suite combines three test families: standard unit tests covering individual function semantics; fuzz tests with 256 randomized inputs per scenario, used principally for coordinate-validation and grid-hash code paths where the input space is large; and invariant tests verifying that key system properties, total PBC supply equals the sum of validated reward events; the registry is the only address with mint authority; geographic boundaries cannot be evaded by any input transformation, hold under arbitrary state-transition sequences. The 63 tests for PotholesRegistry cover the boundary-coordinate validation, the grid-based duplicate detection across both within-cell and adjacent-cell cases, the reward-distribution semantics described in Section 7.3.3, the four-state lifecycle transitions

Table 7.4 Gas consumption for user-facing operations.

Operation	**Gas**	**Description**
Submit new report	230,974	Unique pothole, fresh grid cell
Submit duplicate report	57,893	Same grid cell as existing report
Update status (`InProgress`)	14,644	Triggers reporter reward minting
Update status (`Completed`)	10,144	Triggers completion bonus
Reject report	11,704	Municipal rejection with reason

of Table 7.1, and the role-based access-control enforcement summarized in Section 7.3.2. The 40 tests for POTHOLESTOKEN verify ERC-20 compliance, pausability, and EIP-2612 permit functionality, the latter being the gasless-approval mechanism that complements the meta-transaction interface for token transfers. All 127 tests pass, demonstrating that the implementation realizes the specified semantics under both deterministic and randomized inputs.

7.4.3 Gas Consumption Analysis

Gas consumption is reported separately for the user-facing operations that constitute the bulk of system traffic and for the administrative operations performed by the contract owner and municipal authorities. Table 7.4 presents the user-facing operations.

The most consequential entry is the asymmetry between a new-report submission (231k gas) and a duplicate submission (58k gas), a roughly 4:1 ratio. The asymmetry arises because a new report writes a fresh `PotholeReport` struct to storage, paying for several SSTORE operations on previously-unwritten slots, the most expensive primitive in the EVM's gas table, while a duplicate increments an existing struct's counter, a single SSTORE on an already-written slot. The grid-based duplicate detection (Section 7.3.4) is what enables this saving: without it, every report would write a fresh struct, and the duplicate-cost line would also read 231k. The 75% reduction is therefore not a generic optimization but a direct consequence of the duplicate-detection design.

The status-update operations (`InProgress` at 14.6k, `Completed` at 10.1k) are inexpensive because they modify only the lifecycle field of an already-written struct, and trigger a single `mint` call on POTHOLESTOKEN. `InProgress` is more expensive than `Completed` because the former mints both the original 10 PBC reward and the per-duplicate 0.5 PBC rewards in the same transaction; the latter mints only the 5 PBC completion bonus.

Table 7.5 Network parameters for cost projections (Jan 2025–Jan 2026).

Network	Gas price	Token (USD)	Type
Ethereum mainnet	3 gwei	ETH $3,100	L1
Arbitrum One	0.01 gwei	ETH $3,100	L2 rollup
Polygon PoS	30 gwei	POL $0.45	Sidechain
BNB Smart Chain	1 gwei	BNB $700	Alt L1

Administrative operations and batching. Administrative operations, such as citizen registration and revocation, municipal-authority management, reward-parameter updates, are dominated by single-citizen registration at 37.2k gas. Batch registration delivers measurable economies of scale: registering 5 citizens in a single transaction averages 27.7k gas per citizen, and registering 50 citizens averages 25.0k gas per citizen, a 32.7% reduction in per-citizen cost relative to the single-registration baseline. The same pattern applies to status updates: a batched update of 5 reports averages 8.3k gas per report, a 43.4% saving relative to the per-report baseline. Both batch optimizations matter operationally, because in a real municipal deployment both citizen onboarding (at deployment time) and status-update sweeps (during weekly municipal review) are precisely the kind of bulk operations that benefit from amortizing the fixed per-transaction overhead.

Complete-lifecycle gas budget. A single end-to-end pothole report, original submission, one duplicate confirmation, transition to `InProgress`, transition to `Completed`, consumes 313,650 gas in total, of which 92.1% is borne by the two reporter submissions and only 7.9% by the two municipal status updates. This distribution is itself a design property worth naming: the system places the gas burden on the high-volume, high-value end (reports about real-world infrastructure) and minimizes it on the low-volume administrative end (municipal lifecycle management), which is consistent with the goal of keeping municipal operational overhead low.

7.4.4 Multi-Network Cost Projections

Per-operation gas figures translate into network-specific transaction costs through the gas-price and native-token-price assumptions of each target network. Table 7.5 states the assumptions used for the projection, drawn from average network conditions over the interval January 2025-January 2026.

Table 7.6 reports the projected per-operation costs. The dominant observation is a 99.6-99.8% cost reduction from Ethereum mainnet to either L2 rollup (Arbitrum, Polygon) while

Table 7.6 Projected per-operation costs across networks (USD).

Operation	**Ethereum**	**Arbitrum**	**Polygon**	**BSC**
Submit new report	$2.15	$0.007	$0.003	$0.16
Submit duplicate	$0.54	$0.002	$0.001	$0.04
Update status	$0.14	$0.0005	$0.0002	$0.01
Reject report	$0.11	$0.0004	$0.0002	$0.008
Complete lifecycle	**$2.92**	**$0.010**	**$0.004**	**$0.22**

Table 7.7 Projected annual operational cost (10,000 reports/year, 1:20 original-to-duplicate ratio).

Cost category	**Ethereum**	**Arbitrum**	**Polygon**	**BSC**
Original reports (500)	$1,075	$3.50	$1.50	$80
Duplicate reports (9,500)	$5,130	$19	$9.50	$380
Status updates (500)	$70	$0.25	$0.10	$5
Citizen registration (1,000)	$2,150	$7	$3	$160
Total annual cost	**$8,425**	**$30**	**$14**	**$625**

preserving full EVM compatibility: no contract change is required to deploy to Arbitrum or Polygon, only redeployment with the rollup's RPC endpoint. BNB Smart Chain offers an intermediate cost point (approximately 92% reduction relative to mainnet) for deployments that prioritize low cost over the security guarantees of an Ethereum-anchored rollup.

The substantive question for a candidate municipal deployment is therefore not *whether* blockchain-based pothole management is economically feasible, it visibly is, on L2, but *which* L2 to choose, with the choice being a tradeoff among the rollup's security model, its governance structure, and the operational stability of its sequencer infrastructure. The thesis takes no position on the L2 selection question; it observes only that the multi-order-of-magnitude cost reduction is what makes the question worth asking.

Annual operational projection at municipal scale. Table 7.7 projects annual operating costs for a medium-sized municipality processing 10,000 reports per year. The projection assumes a 1:20 original-to-duplicate ratio, i.e., each unique pothole attracts on average twenty independent confirmations before municipal action, which yields approximately 500 unique potholes and 9,500 duplicate reports per year, plus 500 status updates and 1,000 citizen registrations. The original-to-duplicate ratio is itself an empirical assumption, defensible as a midrange estimate but not measured in deployment.

Table 7.8 Cost and capability comparison: proprietary CMMS vs. proposed system on Layer-2.

Dimension	Proprietary CMMS	Proposed (L2)
Annual platform cost	$10,000–$50,000+ (small/mid)	$14–$30 (Table 7.7)
Hosting infrastructure	$2,000–$10,000	None (decentralized)
Audit-trail integrity	Manual audit, vendor-controlled	Cryptographic, public ledger
Prioritization transparency	Vendor-internal logic	Public function of on-chain state
Citizen incentive	Separate budget line, if any	Native token (PBC), integrated

The projection illustrates the practical-feasibility argument that motivates the multi-network analysis. Annual blockchain operating costs of $14-$30 on the two L2 rollups are several orders of magnitude below the licensing costs of comparable proprietary systems, and within the discretionary budget of even a small municipality. The class of system the proposed framework would in practice displace is the proprietary computerized maintenance management system (CMMS), of which IBM Maximo Application Suite[148] is the canonical reference; industry-reported Maximo pricing for mid-sized deployments runs $100,000–$500,000 in annual licensing, with implementation consulting matching or exceeding the licensing fee. Table 7.8 positions the proposed system against this class on the dimensions that distinguish them.

The dollar comparison is the least consequential row of Table 7.8. The cost reduction is large enough to be unambiguous, but the proposed system would be difficult to justify on cost grounds *alone* for a municipality already operating Maximo or an equivalent platform: switching costs, data migration, staff retraining, and interoperability with adjacent ERP and GIS systems all argue for inertia. The more consequential rows are the lower three, which describe capabilities that proprietary systems cannot offer at any price within their architecture: cryptographic audit-trail integrity that does not depend on the vendor's continued operation, prioritization logic that is publicly verifiable rather than vendor-internal, and an incentive mechanism for citizen reporters integrated into the same record system that governs municipal repair decisions. The Ethereum-mainnet figure of $8,425/yr is presented for reference rather than as a recommendation: at that cost, a deployment is feasible but unattractive relative to the L2 alternative.

7.4.5 Security Assessment

The security assessment proceeds through five categories. The structure mirrors the security review of Chapter 6, Section 6.6.3, both for consistency across the engineering chapters

and because the same five categories apply naturally to any system built on the EVM and OpenZeppelin.

Access control. The three-tier role hierarchy (contract owner, municipal authority, registered citizen) is enforced through the modifiers `onlyOwner`, `onlyMunicipalAuthority`, and `onlyRegisteredCitizen`, applied to every privileged function entry point. Fifteen of the 63 POTHOLESREGISTRY tests (Section 7.4.2) are dedicated specifically to access-control enforcement, covering both positive cases (authorized callers succeed) and negative cases (unauthorized callers revert with the expected error). The `Ownable` pattern[129] provides the foundation for the owner-role transitions, including the deployment-time transfer of POTHOLESTOKEN ownership to the registry that is central to the minting-on-validation guarantee (Section 7.3.3).

Reentrancy protection. The `nonReentrant` modifier from OpenZeppelin's `ReentrancyGuard` is applied to `submitReport` and `updateReportStatus`, the two functions that trigger external calls into POTHOLESTOKEN for reward minting. Without this guard, a malicious contract receiving freshly-minted tokens could in principle re-enter the registry and trigger additional minting in the same transaction. The guard's effect is verified by dedicated unit tests that simulate a reentrant token receiver and confirm the second entry reverts.

Input validation and overflow protection. Geographic-boundary validation, IPFS-hash format checks, report-existence checks, and lifecycle transition rules collectively constitute the contract's input-validation surface. Each is exercised by both unit tests and randomized fuzz tests; the fuzz tests are particularly informative for coordinate validation, where the scaled-integer encoding of Section 7.3.4 admits numerically valid but geographically nonsensical inputs that must be rejected at the boundary check. Solidity 0.8.28 provides automatic overflow and underflow protection on all integer arithmetic, eliminating a class of vulnerabilities that affected pre-0.8 contracts. OpenZeppelin's `SafeCast` utilities are applied at the few sites where signed-to-unsigned conversion is required (notably in coordinate handling).

Meta-transaction security. The ERC-2771[37] implementation preserves security under three properties. Cryptographic signature verification ensures that only signed payloads from registered citizens or authorized vehicles are accepted. Nonce tracking, inherited from the OpenZeppelin `ERC2771Context` base, prevents replay of a captured signed transaction. Deadline enforcement bounds the validity window of a signed payload, limiting the expo-

sure to harvested signatures. The `_msgSender()` override in `ERC2771Context` extracts the original signer from the trailing 20 bytes of the calldata, ensuring downstream access-control modifiers evaluate the original signer rather than the forwarder address: without this override, every meta-transaction-routed call would appear to originate from the forwarder, defeating the role-based access control.

Token-supply control. The minting-on-validation guarantee of Section 7.3.3 is structurally enforced through three mechanisms. First, the `Ownable` transfer at deployment makes PotholesRegistry the sole address authorized to call `mint`; verified by ownership-assertion tests. Second, PotholesToken's emergency `Pausable` interface allows a security incident to be contained by halting all transfers, including reward distribution; the pause authority is the contract owner, which by deployment-time transfer is PotholesRegistry, which in turn pauses the token only on owner-initiated calls, preserving the same access-control hierarchy under emergency conditions. Third, EIP-2612 permit functionality on the token itself supports gasless approvals with deadline and nonce protection, mirroring the meta-transaction security properties at the token-transfer layer.

7.5 Discussion

The evaluation of Section 7.4 establishes that the proposed system is correct under testing, economical under L2 deployment, and structurally sound under standard EVM-security analysis. None of those properties is the same as deployment-readiness. This section examines the residual questions that the engineering evaluation does not resolve, organized around four themes: the physical–digital binding problem that the system shares with the vehicle-identity framework of Chapter 6 (Section 7.5.1), the governance surface created by the system's deployment-time configurability (Section 7.5.2), the scalability ceiling at metropolitan scale (Section 7.5.3), and the thesis-specific question of how this framework integrates with the vehicle-identity system of Chapter 6 (Section 7.5.4).

7.5.1 The Physical–Digital Binding Problem

The system records cryptographically authenticated facts about an on-chain state, but the on-chain state is supposed to track a physical object, a pothole at a specific location on a specific road. The mapping between the on-chain record and the physical object is the *physical–digital binding*, and like every blockchain-physical system, this framework's binding is weaker than its on-chain integrity. Chapter 6, addressed the analogous problem for vehicle

identity, where the binding between an ERC-721 token and a particular VIN ultimately rests on an OEM's attestation. The pothole system has a comparable trust anchor in the municipal authority who validates a report by transitioning its status to `InProgress`, but the failure modes upstream of that validation differ in kind.

Three failure modes warrant explicit treatment. First, GPS spoofing: a malicious reporter using a modified mobile device or a tampered vehicle telemetry stack can submit reports with arbitrary coordinates within the configured municipal envelope. The geographic-boundary check of Section 7.3.4 eliminates the most naive spoofing attempts (reports placed in the Mediterranean to claim PBC) but offers no protection against plausible spoofing within jurisdiction. The grid-based duplicate detection provides partial mitigation, since a single attacker cannot inflate a single grid cell's reward beyond the cell's per-attacker ceiling, but a distributed attack across many cells generates spurious data linearly with the attacker's effort. The system's current design accepts this risk in exchange for accessibility, on the explicit grounds that requiring authenticated location proofs (e.g., trusted-execution-environment GPS attestations) would defeat the gasless-onboarding goal that motivates the meta-transaction layer in the first place. The trade is honest, not optimal.

Second, false reports of plausible defects: a reporter can submit a coordinate-valid report of a non-existent pothole, with a fabricated photograph or one harvested from another location. The municipal authority's validation gate catches this on inspection, a municipal inspector dispatched to the site finds nothing, marks the report `Rejected`, and no PBC is minted. The cost of this attack to the system is therefore the cost of dispatching the inspector, not the cost of a paid-out fraudulent reward. This is operationally tolerable for individual cases but vulnerable to coordinated inflation: an attacker submitting hundreds of plausible false reports imposes a sustained inspection burden on the municipality even if no rewards are paid, which is itself a form of denial-of-service against the validation gate.

The structural parallel with Chapter 6 is worth making explicit. There, the OEM is the trust anchor whose attestation binds the on-chain NFT to the physical vehicle; here, the municipal authority is the trust anchor whose validation binds the on-chain report to the physical pothole. Both anchors are reintroductions of central trust into systems whose other components are designed to dissolve it. The thesis takes the position, developed at length in Chapter 8, that this reintroduction is not a failure of the framework but a definitional consequence of operating at the boundary between digital coordination and physical reality: an architecture cannot eliminate trust in claims about the physical world, only relocate it. The contribution of both chapters is the relocation, not the elimination.

7.5.2 The Governance Surface

Several parameters of the system are configurable at deployment time rather than fixed at the contract level: the geographic-boundary envelope (Section 7.3.4), the grid-precision parameter that determines duplicate-detection sensitivity, the PBC reward magnitudes (10/5/0.5 in the prototype), and the ten weights that constitute the priority-scoring formula (Section 7.3.5). Configurability is the architecturally correct choice, a single fixed value cannot be appropriate across municipalities of different size, density, urban morphology, or political economy, but it pushes a substantive question outside the technical contribution: *who decides the values?*

The question is not rhetorical. The grid-precision parameter alone controls the sensitivity of duplicate detection, the rate at which legitimate independent reports are absorbed as duplicates, and (transitively) the rate at which PBC is paid to original-vs.-duplicate reporters. A 100-unit precision (~11 m at urban latitudes) and a 1000-unit precision (~110 m) are not minor variants of the same design; they encode meaningfully different judgments about how distinct two reports must be to count as different potholes. Similarly, the priority-scoring weights of Table 7.2 encode a judgment about the relative urgency of, say, a pothole near a hospital versus a pothole on a high-speed road. The thesis takes no position on what the correct weights are; it observes that a deploying municipality must make the weight selection before deploying the contracts, and that the weights become a public artifact of the deployment that any observer can examine.

This is the architectural commitment of Section 7.3.5: the scheme is opinionated about whether the weights, once chosen, are visible, not about whether they are correct. A municipality that publishes its scoring weights and its grid precision exposes its prioritization logic to public scrutiny in a way that proprietary CMMS architectures structurally cannot. This is a feature when the published weights are defensible, it makes "why is this pothole still unrepaired?" a question the citizen can answer by examining public state, and it is a vulnerability when the published weights are politically contested, because the municipality must defend the weights publicly rather than treating them as administrative details.

The longer-run question is whether weight selection should be a one-time municipal decision, a periodic administrative review, or an on-chain governance process in which token holders vote on weight changes. The DAO-governance literature offers templates for the third option, but adapting them to civic systems where the relevant constituency is municipal residents (not token holders) involves a non-trivial mapping from governance tokens to political legitimacy that the thesis does not attempt. The contribution of Section 7.5.2 is to identify the surface, not to resolve it.

7.5.3 Scalability at Metropolitan Scale

The annual-cost projection of Section 7.4.4 assumes 10,000 reports per year, which corresponds to a medium-sized municipality. A metropolitan deployment, such as Milan, Rome, the broader Turin functional area, would handle several times this volume, possibly an order of magnitude more in a population-100k-and-up setting. Three considerations bound the framework's scaling.

First, the per-operation gas costs of Section 7.4.3 are fixed by the EVM's execution model and do not degrade with database size: the grid-hash lookup is $O(1)$ regardless of how many existing reports are recorded, and the lifecycle transitions touch only the report being modified. Linear scaling in cost-per-year is therefore the expected behavior, and a 100,000-report-per-year deployment incurs roughly ten times the annual cost of the 10,000-report baseline: approximately $300/yr on Arbitrum or Polygon, still well within municipal discretion. This is the easy case.

Second, the IPFS storage burden scales linearly with the number of reports, since each report contributes one composite metadata document of similar size. A 100,000-report-per-year deployment that retains photographs at full resolution generates a non-trivial pinning obligation, which is an operational cost the cost projections do not capture (because IPFS costs depend on the pinning provider, not on EVM gas). For a municipality assuming the pinning role itself, this becomes a storage capacity question; for a municipality outsourcing to a commercial pinning service such as Pinata or Web3.Storage, this becomes a recurring subscription cost at storage tiers this analysis does not enumerate. Both options are tractable, but neither is free.

Third, the validation throughput of the municipal authority becomes the binding constraint. The system can mint PBC and update lifecycles at L2 throughput limits; the municipality cannot dispatch repair crews at the same rate. A 100,000-report-per-year deployment with the system's current design implies roughly 270 inspector-validation events per day, sustained, which exceeds the validation capacity of any but the largest municipal infrastructure departments. The architectural response is not technical, the contracts can absorb the throughput, but operational: the municipality must invest in validation capacity proportionally to deployment scale, or accept that validation latency degrades as the system grows. Neither outcome is an indictment of the framework; both are consequences of operating a coordination layer whose throughput is bounded by the physical-world activity it coordinates.

A separate consideration is L2 platform stability. The cost projections of Section 7.4.4 assume Arbitrum or Polygon as the deployment target, on the strength of their current operational track record. The L2 ecosystem is, however, less than five years old in the form

relevant to production deployment, and the rollup-sequencer model has not been stress-tested under multi-decade civic-infrastructure timelines. A municipality committing to a specific L2 is implicitly committing to that L2's continued operation, or to its data being migratable to a successor chain in the event of failure. The framework's data layout is sufficiently simple that migration would be feasible, the contracts, the IPFS pins, and the bibliography of past reports would carry across a chain change with engineering effort but no fundamental redesign, but the migration scenario is not part of this evaluation.

7.5.4 Integration with the Vehicle Identity Framework

The chapter has so far treated the vehicle-reporter pathway as architecturally analogous to the citizen-reporter pathway: both submit signed reports through the same forwarder, both authenticate against a registered-address mapping, both incur the same lifecycle. This treatment is faithful to the FGCS submission. It is, however, weaker than what the thesis as a whole permits, because Chapter 6 has already constructed an authentication primitive, the ERC-721 vehicle NFT, that the pothole-management framework can reuse without architectural change.

The proposed integration is straightforward in form. The POTHOLESREGISTRY's current `registeredCitizens` mapping is extended (or paralleled) by a `registeredVehicles` mapping whose membership condition is *ownership of a vehicle NFT minted under the framework of Chapter 6*. A vehicle's signing wallet, the wallet bound to the NFT through the OEM's deployment-time minting, is therefore admissible as a pothole reporter without requiring the municipality to register the vehicle separately. The vehicle's identity is bootstrapped from its existence in the Chapter 6 system; the pothole framework does not need to authenticate the vehicle itself, only to verify that the reporting wallet holds a valid vehicle NFT.

The integration is presented here as an extension of this chapter's contribution, not as part of the FGCS submission whose engineering it preserves. The contracts as currently deployed accommodate the integration without modification, this is the load-bearing claim, but realizing the integration requires off-chain client work (the citizen-facing app would not change; the vehicle-facing onboarding would gate `registeredVehicles` membership on a Chapter 6 NFT-ownership check) that the prototype does not implement. Future work, jointly across the two frameworks, would close this gap. Section 9.4 picks up this integration as a concrete research direction.

7.6 Chapter Summary

The system presented in this chapter, three smart contracts (PotholesRegistry, PotholesToken, PotholesForwarder), an off-chain IPFS storage layer, a multi-factor priority-scoring algorithm computed from public on-chain state, and two reporter pipelines routed through a single ERC-2771 meta-transaction interface, was evaluated against six design goals (Section 7.3) and validated through 127 passing tests, per-function gas measurements, and Slither static analysis. Layer-2 cost projections of $14–$30 per year for a 10,000-report-per-year deployment (under the 1:20 original-to-duplicate ratio of Section 7.4.4) place the system several orders of magnitude below proprietary CMMS alternatives.

Three architectural bets structure the design and will be reused in the cross-cutting analysis of Chapter 8. The first is meta-transactions as civic onboarding primitive: gasless reporting is treated not as user-experience polish but as the constitutive condition of reaching the citizen population whose participation determines the system's value, with a centralization point at the forwarder discussed in Section 7.5.2 as a governance rather than a technical question. The second is minting-on-validation tokenomics: PBC issuance is a deterministic function of a registry-authorized state transition, not a discretionary act of a municipal treasury, which is what makes the economic layer auditable. The third is priority as a pure function of on-chain state: repair ordering is computable from public state by any observer, with the scoring weights configurable at deployment but visible thereafter.

The chapter closes the engineering contributions; the cross-cutting analysis that follows reads the four chapters as configurations of a single architectural toolkit.

Chapter 8

Cross-Cutting Analysis and Discussion

Chapter 7 closed with the observation that the contributions presented in the preceding chapters jointly support a synthesis that no individual chapter could perform on its own. This chapter undertakes that synthesis. The thesis it advances is twofold: first, that the four engineering contributions of Chapters 4, 6, and 7, together with the regulatory positioning of Chapter 5, share a small set of architectural patterns whose recurrence across heterogeneous applications is itself a finding of the thesis; second, that the points at which the contributions diverge, most consequentially along the privacy-transparency axis, are determined by the structural properties of the data each system handles rather than by independent design choices, and that the divergences therefore mark the architectural boundaries of the regime in which the recurring patterns apply. The chapter develops both claims in seven sections. Section 8.1 establishes the descriptive synthesis on which the subsequent analysis rests, and Section 8.2 identifies the recurring architectural patterns that constitute the chapter's analytical core; the remaining sections examine the contributions jointly along the dimensions of cost (Section 8.3), security (Section 8.4), privacy and transparency (Section 8.5), regulatory alignment (Section 8.6), and the cross-cutting limitations that no individual contribution chapter could fully address (Section 8.7).

8.1 Synthesis of Contributions

The four contributions developed in Chapters 4 through 7, together with the regulatory positioning of Chapter 5, can be read as a single coordinated response to the four trust

gaps that opened this thesis. The synthesis advanced in this chapter is fourfold: (i) the four gaps are manifestations of a single underlying deficiency, the absence of a decentralized trust infrastructure for the automotive ecosystem, rather than independent problems; (ii) blockchain provides a principled architectural response to all four, because its core properties directly counteract the centralization that produces the deficits; (iii) the architectural patterns recurring across the four contributions, examined in Section 8.2, constitute a reusable design vocabulary for blockchain-based trust systems in regulated multi-stakeholder domains; and (iv) the four engineering responses cover the four gaps exhaustively and non-redundantly, where each gap receives one contribution, no two compete for the same trust deficit, and the lifecycle stages they span (upstream supply chain, regulated component record, vehicle as marketplace asset, vehicle as civic sensor) are non-overlapping (Table 8.1). The argument is descriptive in this section; the analytical work of identifying what the contributions share at the level of design is the task of Section 8.2, and the explicit mapping of contributions to research questions is reserved for Chapter 9.

Table 8.1 Overview of the four contributions along five dimensions.

Ch.	**Trust gap addressed**	**Characteristic primitive**	**Privacy regime**	**Validation extent**	**Venue**
4	Supplier confidentiality versus regulatory transparency	Homomorphic computation on encrypted KPIs (fhEVM)	Encrypted	Sepolia fhEVM testnet; gas measured; Slither-validated	GoodIT 2025
5	Absence of digital product passport standard	Provision-to-primitive mapping; reference architecture	Tiered	Analytical mapping of Reg. 2023/1542	BRAINS 2024
6	Unverifiable second-hand vehicle history	Dynamic ERC-721 with per-field state machine	Transparent	Full implementation; mainnet-equivalent gas; Slither-validated	IEEE Access 2023
7	Opaque civic infrastructure governance	ERC-2771 meta-tx + ERC-20 incentive on validation	Tiered	Multi-network gas projection (L1/L2); Slither-validated	FGCS (under review)

The four contributions were developed independently, in different academic collaborations, and submitted to four different peer-reviewed venues. They were not designed as a coordinated technical program; the coordination is a property of the thesis-level synthesis

rather than of the underlying research process. Yet they share a methodological frame whose consistency is itself a finding of the thesis.

The shared technology stack is the most visible manifestation. All four contributions are developed for Ethereum-compatible blockchains, implemented in Solidity, and built on top of the OpenZeppelin contract library, most prominently `AccessControl` and `ReentrancyGuard` for authorization and reentrancy protection, and the canonical implementations of the ERC-20 and ERC-721 token standards. Off-chain content addressing is uniformly handled through IPFS in three of the four engineering contributions, with the fourth (Chapter 4) constituting a principled exception that Section 8.2.1 examines. Background Sections 2.2 through 2.4 introduce these substrates once at the thesis level precisely to obviate the per-chapter repetition that the consistency of the stack would otherwise produce.

The shared evaluation methodology is no less consistent: each engineering contribution reports gas consumption for its principal user-facing operations, applies Slither static analysis to the deployed contracts, and, where applicable, projects costs across multiple deployment networks. The specific numbers differ across contributions in ways that Section 8.3 normalizes, and the Slither output is aggregated in Section 8.4 into a joint security claim that no individual chapter could make alone.

The shared design philosophy is more difficult to identify in any single chapter, but it surfaces clearly in retrospect: each engineering contribution presents a *minimum-viable* response to the gap it addresses rather than a maximum-feature system. The supplier evaluation framework implements the smallest workflow that produces a verifiable sustainability index from encrypted inputs; the vehicle NFT framework implements the smallest set of metadata categories and roles that supports a marketplace transaction; the pothole management system implements only the four lifecycle states required to couple municipal validation to token minting. The minimum-viable stance produces systems that are small enough to evaluate exhaustively, that isolate the design move whose value the contribution defends, and that expose the architectural primitives examined in Section 8.2 without burying them under application-specific extensions.

The synthesis advanced here is therefore architectural and methodological rather than systemic: the four contributions share primitives, share a frame, and address contiguous gaps in a single lifecycle, but they remain four contributions, not one. Their integration into a unified system is identified as future work in Section 9.4.1, and the fragmentation risk that non-integration creates is examined as a cross-cutting limitation in Section 8.7.2.

8.2 Architectural Patterns and Design Principles

The synthesis of Section 8.1 established that the four contributions address contiguous gaps in the vehicle lifecycle and rest on a common methodological frame. This section advances a stronger claim. Examined collectively rather than individually, the four contributions exhibit a small set of recurring architectural patterns that no individual chapter could fully expose, because pattern recognition across heterogeneous applications is a finding that exists only at the level of synthesis. Three patterns are isolated below, ordered from the most familiar to the most distinctive: hybrid on-chain/off-chain storage; role-based access control with policy-driven asymmetry inversion; and state machines for lifecycle-bound mutable state. A fourth subsection examines what does *not* recur, the principled exceptions whose role in the synthesis is to define, by contrast, what the recurring patterns are, and sets up the privacy-transparency analysis of Section 8.5.

8.2.1 Hybrid On-Chain/Off-Chain Storage

The economics of public-chain storage are well understood: per-byte costs of on-chain storage exceed the equivalent costs of cloud or peer-to-peer storage by roughly six orders of magnitude, and this asymmetry is one of the few quantitative facts about blockchain architecture that has remained stable across a decade of platform evolution. Any system that must associate non-trivial payloads with on-chain records confronts the same problem in the same form, and the response that the literature has converged upon, and that this thesis adopts in three of its four engineering contributions, is the hybrid pattern.

Stated abstractly, the pattern partitions application data into two parts: a small immutable digest, retained on-chain, and the bulk content, retained off-chain on a content-addressed substrate, with the digest cryptographically committing to the content. The digest is typically the IPFS Content Identifier (CID), and its on-chain footprint is bounded by a fixed-length hash regardless of the size of the underlying content. Retrieval is a two-step process: read the CID from the contract, fetch the payload from IPFS, and verify that its hash matches the on-chain reference. Tampering with the off-chain payload is detectable by any observer; replacing the on-chain reference requires a new transaction with full provenance.

The thesis instantiates this pattern in three contributions. In Chapter 6, the vehicle NFT stores one CID per metadata category (manufacturing, insurance, inspection, repair) with the bulk documents pinned to IPFS by the issuing actor. The decision to use one CID per category rather than a single CID for an aggregated document is not incidental: it is the decomposition that permits the per-field state machine of Section 8.2.3 to operate with field-level granularity.

In Chapter 7, the pothole registry stores per-report scaled-integer coordinates and the CID of a JSON metadata package containing the photograph and OpenStreetMap-derived geographic context; the pattern here is identical in form to Chapter 6's, despite the application being radically different. In Chapter 5's reference architecture, the same primitive is posited at the abstract level for the digital battery passport: lightweight regulatory metadata on-chain, full bill-of-materials and sustainability documentation off-chain, with the CID as the bridge.

The pattern's fourth instance is Chapter 4, which is a principled exception that Section 8.2.4 examines in detail. Hybrid storage fails, structurally, in Chapter 4, and the failure is informative: it establishes that the hybrid pattern is the appropriate response when the computational model is plaintext-on-public-state, and that it gives way to wholly on-chain storage when the computational model is encrypted-on-public-state.

Table 2.1 of Chapter 2 anticipated this synthesis at the level of categorical allocation. The pattern's economic value is the four-to-five-order-of-magnitude reduction in on-chain storage cost it delivers in exchange for an additional retrieval round-trip, and this trade-off is favorable across all three contributions in which it appears.

8.2.2 Role-Based Access Control with Policy-Driven Asymmetry Inversion

The second recurring pattern concerns how authorization is encoded. Across the four contributions, authorization is uniformly implemented as a discrete, finite set of named roles, each carrying a per-function capability matrix, with on-chain enforcement performed by the contract at every privileged call site. The implementation idiom varies: Chapters 4 and 6 use OpenZeppelin's `AccessControl` library with named role identifiers, while Chapter 7 uses the simpler `Ownable` pattern augmented by per-class registration mappings. The abstract pattern, discrete roles plus per-function capability matrix plus on-chain enforcement, is invariant across the three implementations.

The role taxonomies are domain-specific. Chapter 4 defines roles for the supplier (writes encrypted KPIs, controls decryption for own data), the buyer or auditor (reads sustainability indices, controls decryption for own data only after explicit grant), and the contract itself (performs homomorphic operations without decryption capability). Chapter 6 defines roles for the OEM (the trust anchor), the owner, the insurance issuer, the inspection authority, the repair shop, and the prospective buyer or regulator as an unauthenticated reader. Chapter 7 defines roles for the registered citizen reporter, the authenticated vehicle reporter, the municipal authority (the validator), and the unauthenticated reader. Chapter 5's reference

architecture defines roles parametrically, writers and readers stratified into tiers, without committing to a specific stakeholder taxonomy.

The structural observation that emerges from comparing these configurations is that the *direction* of the read/write asymmetry inverts according to the role of the data in the application. In Chapter 4, reads are restricted: the raw KPIs are encrypted under keys held by the data producer, and only the aggregate sustainability index is readable by parties other than the supplier. Writes are restricted to the supplier. The protection target is the supplier's commercial confidentiality, and both reads and writes are constrained accordingly. In Chapter 6, by contrast, reads are unrestricted: any prospective buyer, regulator, or insurance company can read the full vehicle history without authorization. Writes are restricted to the actor whose institutional role attests to the data's veracity. The protection target is not confidentiality but provenance, and the asymmetry is consequently inverted: anyone may read, but only specific roles may write. Chapter 7 lies between the two: reads are largely unrestricted (the report stream is public) but the role taxonomy still privileges writes (only registered citizens or vehicles may submit, only municipal authorities may advance lifecycle states).

The thesis-level observation is that the same architectural primitive serves opposite policy goals depending on whether the application protects confidentiality or provenance. Role-based access control is therefore not a single design decision but a parameter space, and the configuration of the parameter is determined by what the data is *for* in the application, not by independent stylistic preference. Section 8.5 develops this observation into the privacy-transparency framing that characterizes the spectrum on which the four contributions can be placed.

A second observation, narrower but worth recording, concerns the role each contribution assigns to its trust anchor. Chapter 6 makes the OEM the privileged validator of metadata updates; the chapter itself notes that this role is analogous to that of the economic operator in the EU Battery Regulation framework analyzed in Chapter 5. Chapter 7 assigns the analogous validator role to the deploying municipality; Chapter 4 embeds the analogous trust in the holder of fhEVM decryption keys. In each case, the role-based access-control machinery resolves to a single authority that the system trusts to validate the truth of off-chain claims. The recurrence of this structural feature is examined in Section 8.4.2 as a property of the application class rather than a defect of any individual contribution.

8.2.3 State Machines for Lifecycle-Bound Mutable State

The third pattern is the deepest analytical finding at the architectural level, and it is the one that the underlying papers, taken individually, do not surface. Two of the engineering contributions encode their core operational logic as on-chain finite state machines: Chapter 6 governs the lifecycle of each mutable metadata field of a vehicle NFT through the four-state machine reproduced in Figure 6.1, and Chapter 7 governs the lifecycle of each pothole report through the four-state machine summarized in Table 7.1. Read at the surface level, these are two unrelated design choices in two different applications. Read at the abstract level, they are two instantiations of the same architectural pattern.

The pattern, stated abstractly, is the following. A unit of application state, a metadata field, a report, a record, is associated with a small finite set of named states. Transitions among the states are encoded as functions of the smart contract. Each transition is guarded by a per-role authorization check, an instance of Section 8.2.2's pattern. The state machine has at least one terminal state, and entry into a terminal state is the locus at which the on-chain machinery couples to a downstream consequence. The state machine is the unit of authorization work in the contract: the lifecycle of the application state is the lifecycle of the authorization workflow, and the two are co-extensive by design.

Chapter 6's instance is per-field. Each of the three mutable metadata fields (insurance certificate, inspection report, repair log) carries an independent state variable taking values in $\{\text{REVIEWED}, \text{UPDATEREQUIRED}, \text{UNDERREVIEW}, \text{DISMISSED}\}$. The initial state is REVIEWED. Any authorized actor (owner, insurance, inspection, repair, or OEM) may transition the field from REVIEWED to UPDATEREQUIRED by invoking the corresponding update-request function. The role-restricted issuer (insurance for insurance, inspection for inspection, repair for repair) may then submit a new IPFS URI, transitioning the field to UNDERREVIEW. The OEM, acting as the trust anchor, invokes the review function with a Boolean verdict: approval returns the field to REVIEWED; rejection reverts to UPDATEREQUIRED. DISMISSED is absorbing and is entered when the vehicle leaves service permanently. The terminal-state coupling is indirect: the REVIEWED state is the precondition for marketplace listing, so entry into REVIEWED *enables* a downstream value transfer without itself executing one.

Chapter 7's instance is per-report. Each pothole report carries a state variable taking values in $\{\text{REPORTED}, \text{INPROGRESS}, \text{COMPLETED}, \text{REJECTED}\}$. The initial state is REPORTED, established when a registered citizen or authenticated vehicle submits a report through the meta-transaction forwarder. The municipal authority, acting as the trust anchor, transitions the report to INPROGRESS (validating it as actionable) or to REJECTED (declining the report). The transition to INPROGRESS mints ten Pothole Blockchain Coin tokens to the reporter. Subsequent municipal action transitions the report to COMPLETED on repair com-

pletion, minting a five-token bonus and releasing the location for re-reporting. REJECTED is absorbing and likewise releases the location. The terminal-state coupling is direct: the transition function itself executes the ERC-20 mint, with no intervening contract call.

The two state machines are instantiations of the same abstract pattern in three respects and differ in two. They share, first, the authorization-as-transition primitive: every state change is gated by a role, and the role-to-transition mapping encodes the workflow. They share, second, the trust-anchor structure: a single privileged role (OEM, municipality) holds the validation authority, and the architecture fails open if that role is dishonest, a property Section 8.4.2 examines as a recurring limitation. They share, third, the orthogonal-instance structure: the per-field discipline of Chapter 6 and the per-report discipline of Chapter 7 both apply the state machine to multiple independent units within the same contract.

The two state machines differ, first, in the granularity of the unit: Chapter 6's machine governs a metadata field; Chapter 7's governs a complete record. They differ, second and more consequentially, in the coupling between terminal-state entry and value transfer. Chapter 6's machine *enables* value transfer by establishing the precondition for marketplace listing; Chapter 7's machine *executes* value transfer by minting ERC-20 tokens within the transition function itself. This difference is not arbitrary. It reflects the difference between an identity primitive whose value is realized through a separate marketplace contract and a coordination primitive whose value is realized through immediate participation incentives. The architectural pattern is the same; the value-coupling discipline is the parameter on which the two applications differ.

Table 8.2 enumerates the respects in which the two state machines share structure (upper rows) and the respects in which they diverge (lower rows), making the pattern's invariants and parameters explicit.

Table 8.2 The state-machine pattern across the two contributions in which it appears. Six aspects on which the two instantiations may be compared; the upper rows enumerate respects in which the machines share structure, the lower rows the respects in which they diverge.

Aspect	**Ch. 6 (per-field)**	**Ch. 7 (per-report)**
Authorization primitive	per-role transition guard	per-role transition guard
Trust anchor	OEM (institutional)	Municipality (institutional)
Orthogonal-instance structure	one machine per mutable field, three fields per vehicle	one machine per report
Unit of state	metadata field	complete report
Initial state	REVIEWED	REPORTED
Terminal-state coupling	*enables* value transfer (marketplace listing)	*executes* value transfer (ERC-20 mint)

The thesis-level claim that follows from this analysis is that the state machine, as instantiated here, is what differentiates blockchain *coordination* systems from blockchain *registry* systems. Registries store; coordination systems compute trajectories. The two state-machine-bearing contributions are coordination systems whose distinction from registry systems and from the computation regime examined in 8.2.4 is itself worth recording.

8.2.4 What Does Not Recur: The Principled Exceptions

The three patterns identified above recur across the contributions. Two features of the architecture, by contrast, appear in only one contribution each, and their non-recurrence is itself worth examining because each marks an architectural regime that the recurring patterns implicitly define themselves against.

The first non-recurring feature is verifiable computation on private data, which appears only in Chapter 4. The fhEVM-based supplier evaluation framework occupies a distinct architectural regime relative to the other contributions: encrypted state, encrypted computation, plaintext outputs only at decryption-key-controlled boundaries. Where Chapters 6 and 7 rely on the default transparency of public chain state and configure access control to manage who may write to it, Chapter 4 treats the chain state as encrypted-by-default and uses fhEVM's homomorphic operations to compute on the ciphertexts directly. The hybrid storage pattern of Section 8.2.1 and the state-machine pattern of Section 8.2.3 do not apply: encrypted operands cannot reside in off-chain content-addressed payloads, and

the contract's principal action is arithmetic on ciphertexts rather than lifecycle progression. The privacy–transparency spectrum developed in Section 8.5 is the formal statement of this two-regime distinction.

The second non-recurring feature concerns the role of fungible tokens. Chapter 7 mints ERC-20 tokens as participation incentive; Chapter 6 uses ERC-721 to represent a non-fungible identity rather than a fungible value; Chapters 4 and 5 use neither. The incentive-token pattern, while distinctive in Chapter 7, is a single-instance architectural choice rather than a recurring one and does not warrant elevation to a thesis-level pattern.

These two non-recurrences are not symmetric. The first (Chapter 4's fhEVM regime) is a structurally necessary divergence: the recurring patterns simply cannot apply when the computational model is encrypted. The second (Chapter 7's incentive token) is a contingent design choice: the patterns of Sections 8.2.1 through 8.2.3 are entirely compatible with incentive tokens, and other contributions could in principle have used them. The distinction matters because it sharpens the synthetic claim: the recurring patterns are not the patterns this thesis happened to adopt across its contributions; they are the patterns that transparent-by-default blockchain systems converge upon when they address trust gaps in lifecycle-bound applications, and the divergences mark the architectural boundaries of that convergence.

The cost analysis of Section 8.3 examines whether the costs of instantiating these patterns are economically realistic at deployment scale.

8.3 Cost and Scalability Analysis Across Systems

The architectural patterns identified in Section 8.2 commit the four contributions to a particular relationship with the gas economics of public blockchains. Each pattern, hybrid storage, role-based access control, lifecycle state machines, is instantiated through smart-contract calls whose gas consumption is determined by the EVM specification and whose monetary cost is determined by the gas price and native-token exchange rate of the deployment network. The cost question is therefore not a property of any individual contribution but a property of the patterns considered jointly: are the architectural choices made in Chapters 4, 6, and 7 economically realistic at deployment scale, and where does that economic realism come from?

The methodological challenge in answering this question is that each chapter measured costs in a different deployment environment. Chapter 4's measurements were taken on the Sepolia fhEVM testnet at 3 gwei and ETH/USD of $2,550 (26 May 2025); Chapter 6's on Ethereum mainnet at 16 gwei and the contemporary ETH/USD rate (24 January 2023);

Chapter 7's as projections across four networks (Ethereum mainnet, Arbitrum One, Polygon PoS, BNB Smart Chain) at average rates over the interval January 2025 through January 2026. Direct comparison across the raw figures is misleading. To produce a meaningful cross-contribution comparison, the gas figures from Chapters 4 and 6 must be re-projected onto the same reference network at the same reference price point as Chapter 7's already-published projections.

8.3.1 Methodology for Normalization

Three observations make the normalization tractable. First, gas units consumed by a Solidity function are deterministic for EVM semantics: the same compiled bytecode executing on Ethereum mainnet, on Arbitrum One, or on Polygon PoS consumes the same gas units. The chapter-by-chapter variation in measured *cost* arises entirely from differences in gas price and native-token-to-USD exchange rate, not from differences in computational work. Second, Chapter 7's reference table (Table 7.5 in Chapter 7) already states a self-consistent set of network parameters for Ethereum mainnet, Arbitrum One, Polygon PoS, and BNB Smart Chain over the interval January 2025 through January 2026, and these parameters are the most recent published figures. Third, the gas figures of Tables 4.6 and 6.2 are reproduced verbatim in the contribution chapters and can be re-priced under the Chapter 7 parameters by elementary arithmetic.

The normalization adopted in this section therefore reuses Chapter 7's reference parameters, Ethereum mainnet at 3 gwei with ETH/USD of $3,100, Arbitrum One at 0.01 gwei with ETH/USD of $3,100, and Polygon PoS at 30 gwei with POL/USD of $0.45, and re-prices the gas figures of Chapter 6 (originally reported at January 2023 mainnet conditions) under those parameters. The gas figures of Chapter 4 are an exception that warrants separate treatment: the fhEVM infrastructure is currently available only on Zama-supported chains, and its homomorphic operations do not have a direct counterpart on the standard EVM-compatible chains to which Chapters 6 and 7 can be deployed without modification. Chapter 4's costs are therefore reported separately at the Sepolia fhEVM testnet rates at which they were originally measured, with the understanding that they characterize the cost of the encrypted-computation regime rather than the cost of the transparent-stack regime that the other two contributions inhabit.

8.3.2 Normalized Per-Operation Costs Across the Transparent-Stack Contributions

Table 8.3 reports the principal user-facing operations of Chapters 6 and 7 under the common reference parameters described above. Chapter 4's fhEVM operations are reported in the discussion below at their native Sepolia fhEVM rates.

Table 8.3 Normalized per-operation costs for the principal user-facing operations of Chapters 6 and 7, expressed in USD at the reference parameters of Table 7.5

Ch.	Operation	Gas	Eth. L1	Arbitrum	Polygon PoS
6	createToken (mint NFT)	200,412	$1.86	$0.0062	$0.0027
	updateData	125,308	$1.17	$0.0039	$0.0017
	reviewVehicle	35,958	$0.33	$0.0011	$0.0005
	listItem	90,251	$0.84	$0.0028	$0.0012
	acceptOffer	136,943	$1.27	$0.0042	$0.0018
	confirmPayment	55,853	$0.52	$0.0017	$0.0008
7	Submit new report	230,974	$2.15	$0.0072	$0.0031
	Submit duplicate report	57,893	$0.54	$0.0018	$0.0008
	Update status (InProgress)	14,644	$0.14	$0.0005	$0.0002
	Update status (Completed)	10,144	$0.09	$0.0003	$0.0001

Three observations emerge from the table. First, the most expensive user-facing operation across either contribution, a new pothole report at 231k gas, equivalent to $2.15 on Ethereum mainnet, is, on the two Layer-2 networks considered, less than one cent. The cost reduction from mainnet to L2 is approximately three orders of magnitude across the board, and is independent of the contribution: it follows from the gas-price differential between mainnet and the rollups, which applies identically to NFT minting, marketplace listing, pothole submission, and status update. Second, the most expensive operation in either contribution on Layer-2 is the new-report submission at $0.0072 on Arbitrum and $0.0031 on Polygon PoS. At these rates the per-operation cost is no longer the binding adoption constraint: even a high-volume system processing one report per second per municipality would incur an operational cost of roughly $200,000 per year on Arbitrum, well below the licensing cost of comparable proprietary systems (Table 7.8 in Chapter 7). Third, the cost ordering across operations is preserved across networks: an operation that is cheaper than another on mainnet remains cheaper on Arbitrum and on Polygon PoS by the same proportion, because

the gas-units-to-cost mapping is linear in gas price and the gas price applies uniformly to all transactions on a given network. The architectural decisions about which operations to optimize (for instance, Chapter 7's grid-based duplicate-detection scheme that reduces a duplicate report from 231k to 58k gas) therefore retain their proportional value at any deployment cost point.

The fhEVM operations of Chapter 4 occupy a different cost regime. The principal operations of the supplier evaluation framework, measured on Sepolia fhEVM at 3 gwei and ETH/USD of $2,550, are: addDataset_SEI at 2.20M gas ($16.84), addDataset_CEI at 2.12M gas ($16.24), and computeSustainabilityIndex at 390k gas ($2.99). The total per-supplier-evaluation cost reported by Chapter 4 is $39.30. The contrast with the transparent-stack figures is sharp: a single homomorphic dataset addition consumes roughly an order of magnitude more gas than a vehicle NFT mint, and roughly two orders of magnitude more than a pothole status update. The cost is the cost of the privacy guarantee. Operations that perform arithmetic on encrypted ciphertexts cannot be ported to Layer-2 rollups in the transparent-stack sense, because fhEVM is infrastructure-bound to its own deployment. The economic question for fhEVM-based systems is therefore not whether they can be cheap on a Layer-2 rollup but whether the per-evaluation cost (on the order of tens of dollars per supplier per evaluation cycle) is acceptable for the application. For supplier sustainability evaluation, where each evaluation event is a quarterly or annual procurement decision worth substantially more than the gas cost, this is straightforwardly defensible. For applications in which encrypted computation would be required at higher frequency, the cost of the privacy guarantee becomes the binding constraint and would motivate either a different encryption scheme (zero-knowledge proofs of correctness, for instance, which Section 8.2.4 mentioned as an alternative) or a different application architecture in which encrypted computation is reserved for selected aggregation steps.

8.3.3 Annual Operational Cost at Adoption Scale

Per-operation costs translate into annual operational costs through the expected operation volume of each contribution at deployment scale. Chapter 7 carried out this projection in detail (Table 7.7) for a medium-sized municipality processing 10,000 reports per year under a 1:20 original-to-duplicate ratio, and reported total annual costs of $30 on Arbitrum and $14 on Polygon PoS. Comparable projections for the other contributions can be derived under analogous adoption assumptions.

For Chapter 6, a representative deployment scenario is one in which a national vehicle registry handles approximately one million new-vehicle registrations per year (createToken

at 200k gas) and approximately 500,000 second-hand transactions per year (each involving listItem, acceptOffer, and confirmPayment, totalling 283k gas per transaction). Total annual gas under these volumes is approximately 3.4×10^{11} gas units. At the reference parameters this projects to roughly $3.2 million annually on Ethereum mainnet, $10,500 on Arbitrum One, and $4,500 on Polygon PoS. The mainnet figure is plainly prohibitive; the Layer-2 figures are within the operational budget of a small national agency.

For Chapter 4, a representative scenario is one in which a sustainable-supply-chain initiative coordinates the quarterly evaluation of 1,000 suppliers, each contributing one SEI and one CEI dataset and one sustainability-index computation per quarter. At Sepolia fhEVM rates this projects to $157,200 per year. This figure is materially higher than the Layer-2 projections of the transparent-stack contributions and reflects, again, the cost of the encrypted-computation regime. It is, however, of the same order as the licensing cost of conventional supply-chain transparency platforms (∼$100,000–$500,000 per year for enterprise audit and sustainability software), and therefore does not constitute a disqualifying cost difference at this adoption scale; the encryption-vs-transparency choice is what differentiates the proposals, not the dollar cost of operating them.

Table 8.4 consolidates these projections.

Table 8.4 Projected annual operational cost at representative adoption scale for the three engineering contributions.

Ch.	Adoption assumption	Eth. L1	Arbitrum	Polygon PoS	Sepolia fhEVM
4	1,000 suppliers, quarterly evaluation	—	—	—	$157,200
6	1M registrations + 500k transactions/yr	$3,200,000	$10,500	$4,500	—
7	10,000 reports/yr (Table 7.7)	$8,425	$30	$14	—

The principal observation across the table is that the binding cost constraint differs by contribution. For Chapters 6 and 7, deployment on Ethereum mainnet is economically prohibitive but deployment on either Layer-2 rollup is comfortably affordable; the Layer-2 infrastructure that has emerged over 2024–2025 is what makes transparent-stack blockchain systems economically realistic at adoption scale. For Chapter 4, the absence of a fhEVM-equivalent rollup means that the cost regime is the cost regime, and economic feasibility depends on the per-evaluation value of the application; for high-value, low-frequency operations such as quarterly sustainability evaluation this is defensible, but the scaling profile is fundamentally different from the transparent-stack contributions.

The implication for the thesis-level cost claim is therefore not that blockchain-based systems are uniformly cheap, but that the cost of a blockchain-based system is determined by the architectural regime (transparent-stack versus encrypted-computation) and by the deployment network (mainnet versus rollup). The architectural patterns of Section 8.2, taken together with the rollup infrastructure available at the time of writing, suffice to render the transparent-stack contributions economically competitive with conventional alternatives. The encrypted-computation regime exemplified by Chapter 4 is more expensive by an order of magnitude per operation, but remains operationally defensible in the application class for which it was developed. The cost analysis is therefore consistent with the architectural analysis of Section 8.2: the recurring patterns characterize the regime in which costs are amortizable on commodity rollup infrastructure, and the principled exception of Section 8.2.4 is also a principled exception in cost terms.

The cost analysis answers one question, whether the architectural patterns are economically realistic, and prompts another. The binding constraint, having been removed from the side of per-operation cost on Layer-2, returns on the side of security and trust assumptions. Section 8.4 examines what the joint security profile of the four contributions actually warrants and where the analytical limits of static-analysis tooling lie.

8.4 Security Considerations Across Contributions

The cost analysis of Section 8.3 established that the per-operation cost of the architectural patterns is not the binding adoption constraint of the engineering contributions on commodity Layer-2 infrastructure. The constraint, having been removed from one side, returns from another. The substantive question the four contributions raise jointly is no longer whether they can be deployed cheaply but whether they can be deployed safely, and the answer to this second question is structurally different in shape from the answer to the first. The distinction this section develops is between security properties that the contracts can be analyzed to possess and security concerns that no on-chain analysis can fully discharge. Each of the four contributions, examined individually, makes the distinction in its own chapter. Examined jointly, the recurring boundary between the two categories is a finding of the thesis.

8.4.1 On-Chain Vulnerability Profile

Each engineering contribution applied Slither [44] static analysis to the deployed contracts and reported per-severity finding counts in its own evaluation chapter (Tables 7.3, 6.3, and

the security assessment of Section 7.4.5). Table 8.5 aggregates the three reports into a single joint security claim that no individual chapter could make on its own.

Table 8.5 Aggregated Slither static-analysis findings across the three engineering contributions.

Contribution	**SLOC**	**High**	**Medium**	**Low**	**Inform.**
Ch. 4 (full unit incl. Zama lib)	9,638	0	11	1	70
Ch. 6 VEHICLENFT	—	0	9	—	> 20
Ch. 6 MARKETPLACE	—	0	0	—	8
Ch. 7 (POTHOLESREGISTRY + TOKEN + FORWARDER)	—	0	—	—	—
Joint	—	**0**	20	1	> 98

The analytically decisive row of the table is the first. Across roughly fifteen thousand source lines of Solidity comprising user contracts plus their library dependencies, no high-severity finding is left unmitigated by any of the three contributions. The medium-severity findings warrant inspection because they are not uniform in origin. Chapter 4's eleven medium-severity findings are all reentrancy warnings on external calls into Zama's TFHE library, specifically calls to `TFHE.add`, `TFHE.mul`, `TFHE.asEuint32`, and `TFHE.allow`. Slither's reentrancy detector is conservative: it treats any external call to a non-inherited contract as a potential reentrancy vector. The TFHE library, however, is a fixed on-chain library whose functions are deterministic cryptographic operations and which contains no callback mechanisms or arbitrary-code-execution paths. The findings are therefore false positives specific to Slither's heuristic, and the user contract reinforces the conclusion through the Checks-Effects-Interactions pattern and the `nonReentrant` modifier from OpenZeppelin's `ReentrancyGuard`. Chapter 6's nine medium-severity findings on the VEHICLENFT contract originate in OpenZeppelin's `SafeMath.sol` and reflect Solidity-pragma version mismatches between the framework's compiler version and the inherited library at the time of analysis; they do not represent vulnerabilities in the framework's own logic. The marketplace contract reports zero medium-severity findings. Chapter 7's security assessment proceeds by category rather than by severity table, and reports no high-severity findings; the categories examined, such as access control, reentrancy protection, input validation and overflow protection, meta-transaction security, and token-supply control, are exactly the categories that Slither's high-severity detectors are designed to cover, and dedicated unit tests verify each category against both positive and negative cases.

The joint claim that follows from the aggregated table is therefore the following: the on-chain vulnerability surface of the three transparent-stack contributions, plus the encrypted-computation contribution of Chapter 4, contains no unmitigated vulnerability that the standard static-analysis tooling of the EVM ecosystem can detect. The medium-severity findings either originate in third-party library code beyond the frameworks' control (Chapter 6) or are conservative false positives traceable to the analyzer's treatment of cryptographic external calls (Chapter 4). The clean static-analysis profile is a necessary condition for deployability but not a sufficient one, because the decisive security questions for these systems lie beyond the boundary at which static analysis stops.

8.4.2 The Boundary Where Static Analysis Stops

Three classes of security concern recur across the engineering contributions, and each is one that no on-chain analysis tool can fully discharge. Their recurrence is not coincidence. They are properties of the application class rather than defects of any individual contribution: they arise whenever a blockchain-based system is required to interface with the physical world or with institutional governance, and they are therefore inherited by any blockchain system that addresses lifecycle-bound trust gaps in the sense of this thesis.

The first concern is the physical-digital binding problem. Each of the three engineering contributions devotes a section of its discussion chapter to this problem: Chapter 4 as the "honest-KPI assumption", Chapter 6 explicitly as the physical-digital binding problem, and Chapter 7 likewise. The recurring formulation is the same across the three: the system records cryptographically authenticated facts about an on-chain state, but the on-chain state is supposed to track a physical reality, a pothole at a specific location, an inspection performed on a specific vehicle, a water-consumption measurement at a specific factory, and the bridge between the on-chain record and the physical reality is not itself on-chain. A supplier could submit deliberately understated KPIs; an OEM-authenticated inspection report could be fabricated by collusion; a vehicle-mounted sensor could report a non-existent pothole. The contracts cannot detect any of these. The bridge is discharged at the policy layer by third-party attestation, by audited sensor infrastructure, by institutional accountability of the attesting role, and the design of these attestation mechanisms is outside the scope of every chapter. The formulation is therefore not that the contributions are vulnerable to this class of attack but that they assume its absence: each operates under an explicit attestation assumption whose enforcement is a property of the deployment context rather than of the smart contract.

The second concern is the institutional trust anchor that Section 8.2.2 identified as a recurring feature of the role taxonomies. Each contribution embeds a single privileged role on whose honesty the integrity of the system depends: Chapter 4 on the threshold key-management service that holds the fhEVM decryption keys; Chapter 6 on the OEM as the validator of metadata updates; Chapter 7 on the deploying municipality as the validator of report transitions. The role-based access-control machinery of Section 8.2.2 resolves, in each case, to a single authority that the system trusts to validate the truth of off-chain claims. A dishonest authority, a colluding OEM, a captured municipality, a compromised KMS, can in principle subvert the system in ways the on-chain logic cannot prevent. This is not a defect of any contribution because no smart-contract architecture in the public-blockchain regime can eliminate the need for some institutional anchor when the system's claims concern off-chain reality. It is a recurring property of the application class, and the design move that all four contributions make is to bound the trust assumption tightly, a single role, with a clear scope of responsibility, whose dishonesty is detectable post hoc through the audit trail that the chain itself provides, rather than to pretend the assumption can be eliminated.

The third concern is the governance surface, that is, the set of deployment-time and post-deployment decisions through which the operating parameters of the system are configured. Each contribution exposes such a surface: priority-scoring weights and grid-cell geometry in Chapter 7, KPI-weighting factors in Chapter 4, role-management transactions in Chapter 6, and the choice of deployment chain in all three. The governance surface is not itself a vulnerability, but it is a locus at which a deployer's decisions can substantively alter system behavior in ways that are visible on-chain but interpretable only against an off-chain understanding of the application's intent. A municipality that shifts the priority-scoring weights to favor arterial roads over residential streets has done so transparently, but the policy implications of that shift are not in the contract code. The governance question, like the physical-digital binding question, is discharged at the policy layer rather than at the smart-contract layer.

The joint observation across these three concerns is structural. Each is a recurring property of the application class, each is bounded tightly by the contribution that addresses it, and none is fully discharged by the contract logic alone. The on-chain vulnerability profile of Section 8.4.1 characterizes the security properties that smart-contract engineering can establish; the present subsection characterizes the security concerns that smart-contract engineering, by structural necessity, cannot establish. The joint threat model under which the four contributions are secure is therefore the following: an adversary may be a malicious user with a wallet (held in check by smart-contract logic and by Slither-validated code paths), a malicious co-stakeholder operating within the contracts' role taxonomy (held in check by role-based access control), or a malicious node operator on the underlying chain

(held in check by EVM consensus and by the post-Merge security model of the deployment chain). The adversary cannot, however, be a colluding majority of the trust-anchoring institutions, the holder of fhEVM decryption keys, an attacker on the physical-digital bridge, or a deployer with malicious authority over the governance surface; against these adversaries no contribution offers a defense, because no contract-level defense exists.

The cross-cutting limitations that follow from this joint threat model are taken up systematically in Section 8.7. The present section's contribution to the analysis is to establish that the limitations are joint rather than chapter-specific: the same three boundary problems recur in three different domains because they are properties of what blockchain technology can and cannot guarantee about the world it is asked to represent.

8.5 Privacy vs. Transparency Trade-offs

The architectural patterns of Section 8.2 and the cost profile of Section 8.3 together imply that the recurring patterns are economically realistic when deployed under transparent-stack assumptions, and that a separate cost regime applies when those assumptions are relaxed in favor of encrypted computation. This section examines the data-policy decisions that determine which regime any given system inhabits, and argues that the four contributions can be placed at distinct points along a one-dimensional axis whose endpoints are full transparency and full encryption. The placement is not arbitrary. It is determined by the structural relationship between the data each system handles and the value the data carries for its producer relative to its consumer.

The spectrum framing was introduced in Chapter 5 as the analytical bridge connecting Chapter 4's encrypted-computation regime to Chapter 6's transparent regime. This section extends that bridge to the full set of four contributions, adds Chapter 7's tiered-visibility regime to the picture, and proposes a determinant for placement on the spectrum that the contribution chapters, taken individually, did not articulate.

8.5.1 The Spectrum Defined

Three regimes can be distinguished. Each is defined by the relationship between the on-chain visibility of the application data, the on-chain computation performed on that data, and the access controls that restrict who reads what. The progression from one regime to the next is not a feature switch but a coherent shift in how the underlying primitives of Section 8.2 are configured.

Full transparency. The application data is publicly readable on-chain or retrievable from public IPFS without authentication. Access control restricts only writes; reads are unrestricted by design. Computation on the data, where it occurs at all, operates on plaintext state. This regime is the default behavior of public chains and incurs no architectural cost beyond the on-chain footprint itself. Chapter 6 occupies this regime: the vehicle history is entirely public, and the value of the system depends on its being so, because the information asymmetry it eliminates is precisely the asymmetry between buyers and sellers about historical facts that the seller would prefer to obscure.

Tiered visibility. The application data is partitioned into layers, each visible to a different audience. A public layer carries information whose disclosure is mandated or whose availability defines the system's purpose; restricted layers carry information whose disclosure is conditional on the reader's role. Computation on the data again operates on plaintext, but reads of restricted layers are gated by the role-based access-control primitive of Section 8.2.2. This is the regime articulated as a regulatory requirement in Chapter 5's reference architecture for the digital battery passport, where Article 77 of Regulation 2023/1542 explicitly specifies different layers of access for the public, for economic operators, for notified bodies, and for the European Commission. Chapter 7 is also in this regime in practice, although less by deliberate stratification than by the combination of public on-chain report metadata with off-chain identifiers, IPFS-stored evidence images, reporter wallet addresses that are pseudonymous rather than identifying, whose linkage to real-world identity is not on-chain. The reader who knows only the contract state can see what was reported, when, and where, but not who reported it in any sense the deploying municipality could not already discover.

Encrypted computation. The application data is never decrypted on-chain. Smart-contract execution operates on ciphertexts under a fully homomorphic encryption scheme; plaintext outputs emerge only at decryption-key-controlled boundaries that lie outside the contract itself. This regime is the principled exception that Section 8.2.4 examined: the hybrid storage pattern of Section 8.2.1 fails because operands must reside in contract storage, and the state-machine pattern fails because the contract's principal action is arithmetic on ciphertexts rather than lifecycle progression. Chapter 4 occupies this regime, performing all sustainability-index computation under fhEVM and releasing only the aggregated index to authorized purchasers.

The four contributions can therefore be placed on the spectrum at four distinct positions: Chapter 4 at the encrypted endpoint, Chapter 5's reference architecture and Chapter 7's engineering implementation in the tiered-visibility middle (with Chapter 5 closer to the

restricted side of the middle and Chapter 7 closer to the transparent side), and Chapter 6 at the transparent endpoint. Figure 8.1 renders this placement.

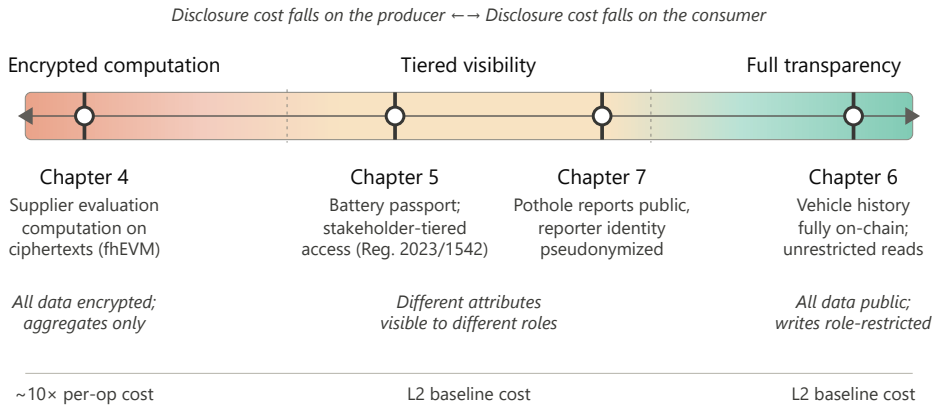

Fig. 8.1 The privacy-transparency spectrum, with the four contributions placed at their respective regimes.

The architectural primitives of Section 8.2 are common across all four regimes. Role-based access control, hybrid on-chain/off-chain storage where applicable, and lifecycle state machines where applicable appear, under different configurations, in every system on the spectrum. The progression from transparent to encrypted is a progression in the configuration of these primitives, not a substitution of one architectural vocabulary for another. The substrate is common; the policy differs.

8.5.2 The Determinant: Who Pays the Cost of Disclosure

Why does each system sit where it does on the spectrum? The contribution chapters answer the question locally, in terms specific to each application: vehicle histories must be public because information asymmetry is the problem; supplier KPIs must be private because they constitute commercially sensitive data. The synthetic question is whether these local answers share a structural form, and this subsection argues that they do.

The structural determinant is the asymmetry of the disclosure cost. For any item of data the system might hold, the cost of disclosure falls either on the producer of the data (the entity whose actions or attributes the data describes) or on the consumer of the data (the entity who would benefit from access). When the consumer pays the cost of disclosure, that is, when keeping the data private benefits the consumer at the producer's expense, the system pushes toward transparency. When the producer pays the cost of disclosure, that is, when

releasing the data benefits the consumer at the producer's expense, the system pushes toward encryption. The midpoint of the spectrum is occupied by data items for which the cost falls asymmetrically on different attributes of the same record, requiring tiered access.

The four contributions instantiate the three cases. In Chapter 6, the data items in question are facts about the vehicle's accident history, mileage, and inspection status. The producer of these facts is the seller; the consumer is the prospective buyer. The seller has a private incentive to suppress unfavorable facts and the buyer has a strong interest in knowing them. The cost of disclosure falls on the producer (the seller's bargaining position is weakened by full disclosure of negative facts) but the benefit of disclosure flows to the consumer (the buyer makes a more informed purchase) and, more importantly, to the integrity of the market itself (which collapses into a lemons problem in the absence of credible disclosure). The system occupies the transparent pole because the social value of disclosure dominates the producer's private cost.

In Chapter 4, the data items are KPI measurements describing a supplier's water consumption, energy mix, labor practices, and other ESG-relevant attributes. The producer is the supplier; the consumer is the purchasing company performing a due-diligence assessment. The supplier has a private incentive to prevent competitors from learning the detailed structure of its operations, because such information is commercial intelligence worth substantial money. The purchasing company has a regulatory obligation to perform the assessment, but does not require, and should not have, granular access to the underlying KPIs. Only the aggregated index needs to be released. The cost of disclosure falls on the producer; the benefit of disclosure can be achieved at the level of the aggregate index alone; therefore the system pushes to the encrypted pole, where the aggregate is computed on ciphertexts and the underlying data is never decrypted.

The intermediate regime is occupied by systems in which the data item is a public good whose existence is socially valuable but whose fine-grained attributes carry asymmetric disclosure costs. Chapter 7 is the cleanest case: the location and existence of road defects is a public good, their disclosure benefits all road users, including the reporter, but the identity of an individual reporter is a private attribute whose disclosure benefits no one and may impose costs (chilling effects on participation, harassment risks, privacy violations) on the reporter. The system therefore publishes the report data but pseudonymizes the reporter at the contract level. Chapter 5's reference architecture under Regulation 2023/1542 articulates the same intermediate regime through regulatory mandate rather than through engineering judgment: certain attributes of the battery passport (identification, performance, recycled-content percentages) are mandatorily public because their disclosure is the regulation's purpose; other attributes (detailed supply-chain data, proprietary chemistry) are restricted

because their disclosure imposes costs on economic operators that the regulation explicitly chooses not to extract from them.

The "who pays the cost of disclosure" determinant is therefore not a private analytical apparatus but a reconstruction of the implicit reasoning that each contribution chapter applied to its own application domain. The synthetic value of stating it explicitly is that it gives a unified normative explanation for what would otherwise look like four independent privacy decisions across four unrelated applications: each system pushes to the point on the spectrum at which the disclosure cost is appropriately allocated between producer and consumer, and the four points together span the full spectrum because the four applications happen to span the full distribution of disclosure-cost asymmetries.

8.5.3 What This Tells Us About Blockchain as an Architectural Substrate

The spectrum analysis closes a circle that the cost analysis of Section 8.3 began. Blockchain technology, in the implementations exercised by this thesis, is flexible enough to occupy any point on the privacy-transparency spectrum, but only at differential cost. Full transparency is the cheapest regime: it is the default behavior of public chains and incurs no architectural overhead. Tiered visibility is intermediate: it costs the gas of role-based access control and the off-chain coordination of layered storage, but neither is dominant in the cost analysis of Section 8.3. Encrypted computation is the most expensive regime: it requires fhEVM's homomorphic operations whose gas cost is roughly two orders of magnitude above the equivalent plaintext operations, and it is currently bound to specialized deployment infrastructure that does not extend to the commodity Layer-2 networks on which the transparent-stack contributions deploy economically.

The architectural choice of where to sit on the spectrum is therefore also a cost choice. The four contributions demonstrate that the cost differential is acceptable across the full spectrum at the adoption scales the contribution chapters consider: even the encrypted-computation regime, the most expensive of the three, is operationally defensible for a quarterly supplier-evaluation application whose per-evaluation value substantially exceeds the per- evaluation gas cost. The spectrum is therefore not a partition of applications into ones blockchain can address and ones it cannot, but a partition of applications into cost regimes that admit different deployment economics.

This observation has an implication for how the regulatory analysis of Section 8.6 should be read. Where regulation mandates a particular point on the spectrum, as Article 78 of Regulation 2023/1542 mandates tiered visibility for digital battery passports, blockchain

implementation is technically feasible at the cost of the regime. Where regulation does not mandate a position, as no regulation currently mandates fully encrypted computation on supplier KPIs, the choice of position is an engineering and policy decision that the cost analysis does not foreclose but does constrain. The four contributions demonstrate that the constraint is not, on Layer-2 infrastructure available at the time of writing, the binding one.

8.6 Regulatory and Standardization Implications

The privacy-transparency analysis of Section 8.5 established that the four contributions of this thesis occupy a single architectural spectrum and that the placement of each contribution is determined by the disclosure-cost asymmetry of its application. The present section examines what that determination implies for the regulatory and standardization context in which the contributions sit. The argument is in two parts. First, three of the four contributions align with concrete EU regulatory instruments whose technical requirements the architectural primitives of Section 8.2 are demonstrably capable of satisfying; the fourth contribution is engineering-rather-than-regulation-driven and serves to demonstrate that the same primitives generalize beyond the regulated domains. Second, the comparison between blockchain-based and federated-data-space implementations of the same regulatory requirements, a comparison Chapter 5 developed in detail for the battery-passport case, generalizes to a thesis-level claim about the relationship between architectural choice and policy choice.

8.6.1 Regulatory Convergence Across the Engineering Contributions

Three of the four contributions can be mapped onto specific instruments of the EU regulatory architecture for verifiable digital records, and each mapping has been developed in detail within the contribution chapter that anchors it. Chapter 4 develops the alignment between its supplier evaluation framework and the Corporate Sustainability Due Diligence Directive (Directive (EU) 2024/1760). The CSDD imposes on large European companies a formal obligation to conduct sustainability due-diligence assessments of their supply chains, including indirect upstream partners, and the encrypted-computation regime of Chapter 4 demonstrates that the regulatory obligation can be discharged without imposing on suppliers the disclosure of the underlying KPI data that would, under conventional assessment methodologies, be required.

Chapter 5 performs the analogous analysis for the EU Battery Regulation (Regulation (EU) 2023/1542), with a provision-by-provision mapping (Table 5.1) between the regulation's technical requirements for the digital battery passport and the architectural primitives that the engineering contributions of this thesis exercise. The chapter establishes that each of the regulation's technical demands (decentralized data storage, data authentication, differentiated access control, ownership-linked data responsibility, and lifecycle persistence beyond the originating economic operator) has a corresponding primitive in the patterns of Section 8.2, and that the primitives jointly suffice to instantiate compliance.

Chapter 6's vehicle NFT framework, although published before the regulatory analysis of Chapter 5 was undertaken, lends itself naturally to a digital-product-passport reading. Section 5.7 of Chapter 5 makes this connection explicit: a vehicle NFT and a digital battery passport share the architectural template of a unique-identifier-bound digital record persisting across multiple ownership transitions, with multi-stakeholder updates scoped by role-based access control and with lifecycle-bound mutable state under institutional validation. The broader Ecodesign for Sustainable Products Regulation, adopted in 2024, establishes the legal basis for extending mandatory digital product passport requirements beyond batteries through subsequent delegated acts, and the architectural symmetry between the battery passport and the vehicle NFT positions Chapter 6 as a proactive engineering demonstration of what a regulated vehicle DPP could look like rather than a regulation-driven implementation.

The fourth contribution, Chapter 7's pothole management system, is not anchored to a specific regulatory instrument and does not claim to be. The application domain, citizen-reported municipal infrastructure governance, is not a regulated domain in the EU sense in which the preceding three contributions sit, although it intersects with several adjacent policy areas (public-data openness, GDPR-bound personal-data protection, smart-city governance) whose detailed treatment lies beyond this thesis. The role of Chapter 7 in this section's argument is therefore not one of regulatory alignment but of generalization beyond it: the same architectural primitives that Chapters 4, 5, and 6 exercise under regulatory drivers are demonstrated, in Chapter 7, to apply equally well in a domain whose adoption pressure is functional rather than legal. This is a useful contrast. It establishes that the architectural patterns of Section 8.2 are not artifacts of any particular regulatory framework but properties of the application class that regulation happens to govern in three of the four cases considered.

The cumulative observation across the four contributions is that the direction of travel of EU regulation in the lifecycle-bound-record domain is unmistakably toward mandatory verifiable digital infrastructures, with audit access, differentiated visibility, and tamper-evidence as common technical requirements. The thesis develops three engineering responses to three concrete instances of this direction (sustainability evaluation under the CSDD, battery

passports under Regulation 2023/1542, vehicle records under the prospective ESPR-delegated DPP regime) and one engineering response in a domain that the regulatory direction of travel has not yet reached. The architectural primitives of Section 8.2, taken together with the cost analysis of Section 8.3, are positioned to satisfy compliance with these regulations on a common stack and at adoption costs that the cost analysis has shown to be operationally defensible on commodity Layer-2 infrastructure.

8.6.2 Comparison with Federated-Data-Space Alternatives

Section 5.5.3 examined the Catena-X data ecosystem in detail as the principal non-blockchain alternative for the battery-passport case; this subsection generalises that comparison to the thesis level. The structural properties of federated data spaces (data held by its originator, access through standardized connectors enforcing policy at the connector boundary, auditability derived from logged exchanges rather than from globally replicated state) are detailed there. Blockchain-based implementations, by contrast, derive their auditability from public verifiability of replicated state, at the cost of the data-locality properties that federated architectures preserve. The two architectures address overlapping regulatory requirements but distribute the assurance properties differently: federated data spaces deliver stronger data-sovereignty guarantees and weaker public-auditability guarantees, while blockchain-based systems deliver the inverse.

The thesis's positioning relative to the federated alternatives is therefore not a displacement claim. The thesis does not argue that blockchain-based implementations should or will replace federated-data-space implementations; it argues that blockchain-based implementations are technically capable of satisfying the same regulatory requirements with stronger auditability and lower governance dependency, and that the choice between architectures is consequently a policy choice rather than a technical inevitability. The position generalises across all four engineering contributions: where the regulation's purpose is open-market verifiability accessible to unknown parties, blockchain's structural properties are the stronger fit; where the purpose is governed B2B exchange among known counterparties, federated data spaces are the more practical near-term implementation path. Chapter 5's analysis of the differentiated-access provisions of Regulation 2023/1542 articulated this concretely for the battery-passport case, in particular public-tier provisions are the natural domain of blockchain implementations, the legitimate-interest tier among known industrial partners is the natural domain of Catena-X-style federated implementations, and a production deployment is likely to pair the two, and the same logic applies to the supplier-evaluation, vehicle-identity, and infrastructure-management contributions.

The thesis's demonstrated architectural primitives are positioned to serve as the public-tier component of hybrid implementations whose legitimate-interest tier is delivered through federated connectors. This is a complementary rather than competing framing, and the architectural feasibility of bridging blockchain anchoring to federated connector attestations, identified in Section 5.5.3 as an open research question, returns in Section 8.7.2 as a cross-cutting limitation and in Section 9.4 as a natural extension of the research program.

8.7 Limitations and Threats to Validity

The preceding sections have argued that the four contributions of this thesis exhibit a coherent architectural vocabulary, that the cost profile of that vocabulary is tractable on commodity Layer-2 infrastructure, and that the regulatory and privacy-related decisions each contribution embodies are explicable as principled responses to the structural properties of the data each system handles. The present section names the limitations under which those claims hold. The limitations enumerated below are not defects of any individual contribution, chapter-level limitations are addressed in the discussion sections of Chapters 4 through 7, but properties either of the application class to which the contributions jointly belong or of the methodological choices that govern the thesis as a whole.

8.7.1 The Physical-Digital Binding Problem

The systems developed in this thesis record cryptographically authenticated facts about an on-chain state, but the on-chain state is supposed to track a physical reality that the smart contract itself cannot observe. Section 8.4.2 examined this problem as a security boundary: the integrity of every contribution rests on an attestation assumption that the smart contract cannot verify. The present subsection re-examines the same problem as a limitation of the thesis's claims. The architectural patterns of Section 8.2 discharge the binding through the trust-anchoring institutional role that Section 8.2.2 identified as a recurring feature of the role taxonomies, but they do not eliminate the binding. A vehicle whose VIN was minted under Chapter 6's framework is the vehicle the OEM attested it to be; a pothole whose report was validated under Chapter 7's framework is the pothole the municipality attested it to be; a KPI ciphertext submitted under Chapter 4's framework encrypts the value the supplier actually measured. The thesis's claims about verifiability are claims about verifiability conditional on these attestations, and a reader who substitutes a less reliable attestation substitutes a less verifiable system. Future work in decentralized identifier and verifiable credential infrastructure (Section 9.4.3) addresses the binding partially by making

the attestation chain itself cryptographically auditable; full resolution of the binding requires hardware-rooted oracles or trusted execution environments at the sensor boundary, neither of which the contributions of this thesis develop.

8.7.2 Interoperability and the Fragmentation Risk

The four contributions share architectural primitives but do not share a deployment. The four engineering prototypes are not deployed on a single chain; the artefacts they produce are not cross-referenced at the smart-contract level; no chapter presents an end-to-end use case in which two or more of the contributions interoperate at runtime. The limitation that this non-integration creates is structural rather than accidental, and it has an empirical antecedent in the research landscape itself: the literature surveyed in Chapter 3 is also fragmented across communities, with supply-chain researchers rarely referencing vehicle-identity systems and infrastructure-monitoring systems designed independently of the vehicles that could serve as their primary data sources. The contributions inherit, at the artefact level, the fragmentation they observe at the field level. A vehicle whose identity is recorded under Chapter 6's framework on Polygon, whose supply-chain sustainability evaluation lives under Chapter 4's framework on the Sepolia fhEVM testnet, and whose operational interactions with civic infrastructure are governed by Chapter 7's framework on Arbitrum is no better off, for the integration question, than the fragmented status quo whose absence motivated the thesis in Chapter 1. The risk is that four blockchain-based responses to four trust gaps reproduce, at a different technological level, precisely the fragmentation they were designed to address. The risk is real and the thesis does not discharge it. Cross-chain composability of NFTs, ERC-20 balances, fhEVM ciphertexts, and IPFS content identifiers is itself an open problem at the level of the underlying technology, and the direction in which integration would proceed, a unified blockchain-based vehicle digital twin (Section 9.4.1) deployed under cross-chain interoperability primitives (Section 9.4.2), is treated explicitly as future work rather than as an artifact of this synthesis. The subsidiary integration question of bridging the thesis's blockchain-based contributions to federated-data-space infrastructures, identified in Section 5.5.3 as an open research question and returned to in Section 8.6, is part of the same unsolved problem.

8.7.3 Governance Centralization Residues

Each contribution embeds a single privileged role on whose honesty the integrity of the system depends: Chapter 4 on the holder of the fhEVM decryption keys, Chapter 6 on the OEM as the validator of metadata updates, Chapter 7 on the deploying municipality as the

validator of report transitions. Section 8.4.2 examined this as a recurring feature of the security model. This subsection names it as a limitation of the decentralization claims that the underlying papers make for their respective frameworks. Public-blockchain deployment removes one form of centralization, no single ledger operator controls the state, but does not, in any of the four contributions, eliminate the institutional trust anchor whose role in the application is structurally distinct from the role of the ledger operator. The thesis develops blockchain-based systems that are decentralized at the infrastructure layer and centralized at the policy layer and centralized at the policy layer, and that the architectural primitives of Section 8.2 bound the centralization tightly, to a single role with a clearly specified scope, rather than eliminate it. The decentralization of the trust anchor itself, through governance-token-based collective validation, decentralized autonomous organization structures, or threshold attestation across multiple anchoring institutions, is technically feasible in principle and is not pursued in this thesis. Whether such decentralization is desirable in the application classes considered, given the coordination problems that DAO-based governance introduces and the accountability gains that institutional anchors provide, is a question this thesis does not attempt to settle.

8.7.4 Validation Limited to Prototype Deployments

None of the four engineering contributions has been deployed in production. The cost figures of Section 8.3 are projections derived from contract-level gas measurements under reference network parameters, not measurements of an operating system at scale; the security findings of Section 8.4 are static-analysis and unit-test results on contract code, not the post-mortem of production incidents. The validation depth differs across the contributions: Chapter 4 reports gas measurements from deployment on the Sepolia fhEVM testnet, Chapter 6 reports gas measurements at mainnet-equivalent conditions of January 2023 with no production deployment, and Chapter 7 reports multi-network projections from contract-level measurements with no on-chain deployment beyond development networks. The gap from prototype to production is non-trivial in this application class: wallet management for non-technical end users, gas funding under meta-transaction sponsorship arrangements, integration with existing institutional information systems, and regulatory approval where applicable, are all questions whose answers are deployment- specific and which the contributions do not address. This is the standard limitation of academic blockchain research and is acknowledged as such; it motivates the real-world deployment direction of Section 9.4.5, which is the future-work track most likely to consume the largest body of follow-on engineering effort.

8.7.5 Generalizability Beyond the European and Automotive Contexts

The regulatory framing of Section 8.6 is European in its instrumentation: the Corporate Sustainability Due Diligence Directive applies to large companies operating in the European single market, the EU Battery Regulation applies to batteries placed on the European market, and the Ecodesign for Sustainable Products Regulation is an instrument of European industrial policy. The architectural patterns of Section 8.2 are not themselves European, they are properties of the EVM-compatible blockchain stack and of the data structures the contributions exercise on it, but the adoption pressure that the regulatory framing identifies is. In non-European jurisdictions where comparable regulatory instruments do not exist or take different forms, the engineering case for the contributions remains intact but the policy case is correspondingly weaker. The thesis does not extend its regulatory analysis beyond the European context, and the question of how the architectural patterns translate to jurisdictions with different supply-chain disclosure regimes, different product-passport mandates, and different data-protection frameworks is left open. The automotive framing of the application domain has an analogous limitation. The contributions address trust gaps in the vehicle lifecycle, and their generalizability to other lifecycle-bound domains, food supply chains, pharmaceutical traceability, consumer electronics circularity, is conjectural. The architectural patterns do not contain anything specifically automotive about them, but their generalization to other lifecycle-bound domains is conjecture rather than demonstrated finding.

These five limitations, taken together, characterize the claims of the thesis as bounded rather than universal. The contributions develop and validate a coherent architectural vocabulary for blockchain-based responses to lifecycle-bound trust gaps, exercise that vocabulary across four contiguous applications in the European automotive lifecycle, and demonstrate that the costs and security profile of the resulting systems are operationally defensible at the validation depth attempted. They do not demonstrate, and the thesis does not claim, that the resulting systems are deployment-ready, fully decentralized, integrated with each other, generalizable across non-European jurisdictions, or free of the institutional attestation chains that the application class structurally requires. Each of these unaddressed claims is a direction in which the research program developed in this thesis admits substantial extension; the future work of the following chapter takes them up systematically.

Chapter 9

Conclusions and Future Work

This thesis has investigated blockchain technology as a trust infrastructure across the automotive ecosystem, instantiating four engineering responses to four trust gaps that span the vehicle lifecycle and synthesising, in Chapter 8, the architectural patterns that recur across the responses. This chapter closes the thesis by returning to the formal commitments that opened it: a summary of contributions in claim, evidence, and publication terms (Section 9.1); direct answers to the research questions (Section 9.2); the practical implications the work supports (Section 9.3); and the research directions it leaves open (Section 9.4).

9.1 Summary of Contributions

The four contributions are stated below in claim/evidence/publication form; the methodological consistency that connects them is the subject of Chapter 8 and is not re-derived here.

Contribution 1 (Chapter 4). The first contribution claims that blockchain technology combined with fully homomorphic encryption can deliver verifiable supplier sustainability evaluation under the disclosure obligations of the Corporate Sustainability Due Diligence Directive without compromising the commercial confidentiality of supplier KPI data. The evidence is a working framework deployed on the Sepolia fhEVM testnet, validated within the MICS Extended Partnership on a textile-industry use case, operating at a per-evaluation workflow cost of $39.30 with zero high-severity Slither findings on a 9,638-line code base. The contribution was published at ACM GoodIT in 2025.

Contribution 2 (Chapter 5). The second contribution claims that the technical requirements imposed by the EU Battery Regulation 2023/1542 on digital battery passports are jointly satisfied by the architectural primitives that the engineering contributions of this thesis exercise in their respective domains. The evidence is a provision-by-provision mapping (Table 5.1) covering data storage, authentication, differentiated access for general public, notified bodies, and parties with legitimate interest, and ownership-linked data responsibility, with each regulatory provision matched to a specific architectural primitive. The contribution was published at IEEE BRAINS in 2024 and updated for the thesis with the 2024–2026 standardization developments, the Battery Pass technical guidance, DIN DKE SPEC 99100, the Omnibus IV regulatory timeline adjustments, and the maturation of the Catena-X data ecosystem, incorporated explicitly.

Contribution 3 (Chapter 6). The third contribution claims that an ERC-721 vehicle identity system with per-field state-machine governance and an integrated decentralized marketplace eliminates the information asymmetries that drive second-hand vehicle markets toward lemons-problem failures, at costs competitive with centralized commercial alternatives. The evidence is a full implementation in which the most expensive operation (NFT mint) costs approximately $5 at January 2023 mainnet conditions and routine marketplace operations remain below $10, with multi-stakeholder updates from OEMs, insurance companies, inspection stations, and repair shops governed by an OEM-anchored validation workflow whose trust structure is analogous to the role of the economic operator in Regulation 2023/1542. The contribution was published in *IEEE Access* in 2023 and constitutes the thesis's first publication chronologically and the highest-deployment-maturity contribution in evidentiary terms.

Contribution 4 (Chapter 7). The fourth contribution claims that a three-contract blockchain coordination layer combining ERC-2771 meta-transactions for gasless multi-source reporting, an ERC-20 incentive minted on validation transitions, a grid-based spatial deduplication scheme, and a multi-factor priority-scoring algorithm computable as a pure function of public state can replace proprietary municipal infrastructure governance at costs viable for production deployment on commodity Layer-2 infrastructure. The evidence is a fully implemented and benchmarked three-contract system, with multi-network gas projections across Ethereum mainnet, Arbitrum One, Polygon PoS, and BNB Smart Chain placing annual operational costs in the $14-$30 range on Layer-2 rollups, two orders of magnitude below comparable proprietary computerized maintenance management platforms. The contribution is under review at *Future Generation Computer Systems* as of March 2026.

The four contributions span four venues (ACM GoodIT, IEEE BRAINS, IEEE Access, and Future Generation Computer Systems) across sustainability informatics, blockchain research, applied engineering, and distributed systems, and were developed across roughly three years of independent research in different collaborative configurations. The methodological consistency that connects them, the architectural vocabulary, cost methodology, security discipline, and privacy-transparency positioning examined in Sections 8.2 and 8.5, is therefore not the artifact of a coordinated technical program but a property that emerged from independently-pursued engineering responses to contiguous gaps in the same lifecycle.

9.2 Answers to Research Questions

The four research questions posed in Section 1.4 structured the engineering contributions of this thesis as a coordinated investigation of how blockchain architecture responds to lifecycle-bound trust gaps in the automotive ecosystem. Each engineering chapter answered its corresponding question locally, in terms specific to the application domain it addresses; Chapter 8 then synthesized the joint architectural response across the four. This section returns to the questions in their original formulation and provides a direct answer to each, engaging the dimensions that the question itself named and stating the bounding conditions under which the answer holds.

RQ1. *To what extent can blockchain technology combined with fully homomorphic encryption enable verifiable supplier sustainability evaluation without compromising sensitive data?* The question named three dimensions of practical feasibility: computational and economic cost on fhEVM infrastructure, trust assumptions inherent in threshold key management, and scalability as the number of evaluation criteria increases. On the first dimension, the framework of Chapter 4 demonstrated that a complete sustainability-evaluation workflow can be performed on fhEVM at a total cost of $39.30 per supplier evaluation under Sepolia testnet conditions, with individual operations bounded by 2.20 million gas; Section 8.3 situated this figure within an order-of-magnitude premium over the transparent-stack contributions, operationally defensible for low-frequency high-value evaluation cycles such as quarterly procurement assessments. On the second dimension, the threshold key-management infrastructure supplied by the fhEVM ecosystem was identified, in Section 8.4.2, as the residual trust anchor on which the system's confidentiality guarantees rest; this trust assumption is bounded but not eliminated, and its further decentralization is treated as future work in Section 9.4.3. On the third dimension, the framework's weighted-sum index formula scales linearly in the number of KPIs because the homomorphic addition primitive is gas-bounded

per operand; the binding scalability constraint is the maturity of the fhEVM infrastructure rather than the framework's design. The question is therefore answered in the affirmative under the bounded condition that fhEVM-capable deployment infrastructure becomes available at production scale, a condition that the contribution demonstrates is achievable today on testnet and is on the trajectory of practical viability as the underlying technology matures.

RQ2. *To what extent does blockchain technology satisfy the technical requirements imposed by the EU Battery Regulation for the implementation of digital battery passports?* The question named four regulatory dimensions: decentralized storage, data authentication, differentiated access control, and lifecycle management. Chapter 5's provision-by-provision analysis (Table 5.1) demonstrated that each of the four dimensions has a corresponding architectural primitive in the blockchain stack the thesis exercises. Decentralized storage is satisfied by hybrid on-chain digest plus off-chain content-addressed storage on IPFS; data authentication is satisfied by the inherent immutability of consensus-replicated state combined with cryptographic transaction signing; differentiated access control, explicitly stratified by the regulation into general public, notified bodies, and parties with legitimate interest, is satisfied by role-based access control implemented through the OpenZeppelin `AccessControl` primitive that recurs across the engineering contributions; and lifecycle management, including the transfer of data responsibility between economic operators, is satisfied by NFT ownership-transfer semantics whose state machine governs the regulatory lifecycle of the passport instance. The residual question, identified in Section 8.6 and in Chapter 5's Catena-X discussion, is not whether blockchain technology can implement digital battery passports but which of several technically feasible architectures, blockchain or federated data spaces, the European industrial ecosystem will converge upon. The technical requirements are fully satisfied; the architectural choice is a policy question that the thesis informs but does not foreclose.

RQ3. *Can an NFT-based vehicle identity system provide a complete, tamper-proof lifecycle record and support trustworthy decentralized second-hand vehicle trading at costs competitive with existing centralized alternatives?* The question named four components: a complete, tamper-proof lifecycle record; trustworthy decentralized second-hand trading; multi-stakeholder updates for insurance, inspection, and repair data; and economic competitiveness with existing commercial platforms. Chapter 6's implementation answered each. The completeness of the lifecycle record is achieved by the partition of vehicle metadata across one IPFS Content Identifier per category (manufacturing, insurance, inspection, repair) governed by the per-field state machine of Section 6.3.3 of Chapter 6, with tamper-evidence delivered by the on-chain commitment to each CID. The decentralized second-hand trading

component is delivered by a marketplace contract supporting listing, bidding, automatic ownership transfer upon validated payment, and dispute-resolution semantics, eliminating the trusted intermediary that traditional second-hand markets require. The multi-stakeholder update mechanism is delivered by a five-role access-control taxonomy (OEM, insurer, inspector, repair shop, owner), with the OEM as the institutional trust anchor whose validation authority is structurally analogous to the role of the economic operator under Regulation 2023/1542. The economic competitiveness claim was originally established at January 2023 mainnet conditions, where NFT minting cost approximately $5 and routine marketplace operations remained below $10; Section 8.3 re-priced these operations under the thesis's reference parameters and confirmed that on Layer-2 infrastructure the most expensive operation costs less than two cents, an order of magnitude below the subscription cost of incumbent vehicle-history platforms. The question is answered in the affirmative along all four dimensions, with the bounding conditions being the trust assumption on the OEM as the metadata validator (Section 8.4.2) and the gap from prototype to production deployment (Section 8.7).

RQ4. *Can a decentralized blockchain-based system effectively coordinate multi-source infrastructure defect reporting with transparent prioritization and token-based incentives, at costs viable for municipal deployment?* The question named five components: multi-source reporting combining connected vehicles and citizen smartphones, spatial duplicate detection, algorithmic priority scoring, tokenized reward distribution upon municipal validation, and economic viability on Layer-2 relative to commercial computerized maintenance management systems. Chapter 7's three-contract coordination layer delivered each component. Multi-source reporting is delivered by an ERC-2771 meta-transaction forwarder that admits both authenticated-vehicle and registered-citizen reporters under the same submission interface, with gasless transaction sponsorship removing the friction of wallet-funded reporting from the end-user experience. Spatial duplicate detection is delivered by the grid-based scheme that scales report-cost from 230,974 gas for a fresh report to 57,893 gas for a duplicate, a 75% reduction. Algorithmic priority scoring is delivered by a multi-factor function combining temporal urgency, community validation through duplicate count, road classification, speed limits, and lane count, computable as a pure function of public state and therefore inspectable by any reader. Tokenized reward distribution is delivered by an ERC-20 token whose mint is coupled to the municipal-validation state transition, with ten tokens released on validation and a five-token bonus on completion. The economic viability claim is the strongest of the four contributions in evidentiary terms: multi-network gas projection across Ethereum mainnet, Arbitrum One, Polygon PoS, and BNB Smart Chain, normalized to common reference parameters, places annual operational costs at $14-$30 per

municipality on Layer-2 deployments: two orders of magnitude below the licensing cost of incumbent commercial CMMS platforms. The question is answered in the affirmative along all five components, with the bounding conditions being the trust assumption on the deploying municipality (Section 8.4.2) and the gap from prototype to production deployment (Section 8.7).

The four research questions, taken together, span the architectural spectrum that Section 8.5 formalized: from the encrypted-computation regime of RQ1, through the regulatory tiered-visibility regime of RQ2 and the hybrid civic-feedback regime of RQ4, to the transparent regime of RQ3. The thesis answers each affirmatively under bounded conditions explicitly stated, and the bounding conditions are themselves the limitations enumerated in Section 8.7: trust assumptions on institutional anchors, the prototype-versus-production deployment gap, the infrastructure dependencies of the encrypted-computation regime, and the unresolved cross-chain integration question. The four-question structure is therefore not a partition of independent investigations but a coordinated examination of how a single architectural vocabulary responds to lifecycle-bound trust gaps under different data-policy constraints.

9.3 Impact and Practical Implications

The contributions of this thesis are engineering contributions under regulatory constraints. Three implications follow.

The first implication is for the engineering practice of blockchain-based lifecycle systems. The architectural vocabulary identified in Section 8.2 (hybrid on-chain/off-chain storage, role-based access control with policy-driven asymmetry inversion, and state machines for lifecycle-bound mutable state) constitutes a small, fixed set of patterns sufficient to instantiate transparent-stack blockchain systems across the four domains the thesis examined. The patterns are not novel in isolation: each appears, under various names, in prior smart-contract literature. What the thesis demonstrates is that they recur consistently across applications as different as supply-chain sustainability evaluation, regulatory compliance analysis, vehicle identity management, and municipal infrastructure governance, and that the recurrence is a property of the application class rather than a coincidence. The patterns are documented as a starting point for engineering teams addressing similar trust gaps; whether they will be adopted is a question of subsequent uptake the thesis does not predict.

The second implication is for the regulatory and standardization landscape the contributions intersect. The provision-by-provision mapping of Chapter 5, the CSDD-aligned

framework of Chapter 4, and the prospective ESPR-DPP positioning of Chapter 6 together demonstrate that the architectural primitives developed in this thesis are technically capable of satisfying the requirements of three concrete EU regulatory instruments under the cost and security conditions Chapter 8 characterised. The architectural choice between blockchain-based and federated-data-space implementations of these regimes is therefore a policy choice rather than a technical inevitability. Industrial momentum currently favours federated implementations; the thesis does not claim that Blockchain-based implementations should displace them, but that they remain technically and economically viable alternatives whose distinctive properties, such as public auditability, operator-independence, governance-minimal trust assumptions, warrant serious consideration in the design of European industrial digital infrastructure.

The third implication is internal to the research program from which the thesis emerged. The four publications span four venues across four communities (sustainability informatics, blockchain research, applied engineering, distributed systems). The methodological consistency that connects them, the hybrid storage strategy, the role-based access control pattern across three of the four contributions, and the architectural vocabulary that Chapter 8 examines, was not, as Section 8.1 makes explicit, the product of a coordinated technical program. Whether the consistency reflects forward propagation of patterns from earlier publications to later ones, convergent responses to structurally similar problems, or the standard repertoire of the smart-contract literature applied independently in each contribution is itself an open question. Whichever explanation holds, the consistency is recoverable as a thesis-level finding rather than as a private methodological history, and the cumulative output is therefore more than the four papers individually establish.

9.4 Future Research Directions

Five research directions extend the program of this thesis from the position it now occupies toward the directions in which the limitations of Section 8.7 most clearly point. Each is anchored to a specific limitation or methodological extension; none is generic blockchain-adjacent work.

9.4.1 Towards a Unified Blockchain-Based Vehicle Digital Twin

Section 8.7.2 identified the fragmentation across deployments as the most consequential cross-cutting limitation of the synthesis. The first direction takes up that limitation directly. The research question is: how can the four contributions' artifacts, the ERC-721 vehicle

tokens of Chapter 6, the ERC-20 PBC balances of Chapter 7, the fhEVM ciphertexts of Chapter 4, and the IPFS content identifiers shared across three of the four chapters, be cross-referenced at the smart-contract level to support an integrated vehicle digital twin spanning supply chain, identity, and operational lifecycle? The methodological direction has two components. The first is a meta-contract layer using W3C Decentralized Identifiers as universal canonical identifiers, with on-chain mappings from each DID to the local identifier in each subsystem. The second is the cross-chain message-passing infrastructure addressed in Section 9.4.2 below, since unification across chains is the deeper engineering question that unification across contributions implies. A concrete instantiation would be a vehicle whose supplier sustainability evidence, regulatory passport, identity record, and civic-feedback history are all reachable from a single DID-anchored query, with cost-and-latency analysis of the cross-contribution coordination. The success criterion is at least one such end-to-end use case implemented and benchmarked. Whether the target should be a unified, interoperable, or deployment-agnostic digital twin is itself a framing choice that future work must make.

9.4.2 Cross-Chain Interoperability for Automotive Applications

The second direction extends the first by treating the cross-chain question on its own terms. Even in the absence of a unification ambition, the four contributions of this thesis would be deployed on different chains under realistic adoption assumptions: Polygon for the high-volume vehicle-identity workflow, Arbitrum for the high-frequency civic-feedback workflow, Sepolia fhEVM (or its production successor) for the encrypted-computation workflow. The research question is: which cross-chain interoperability primitives, such as bridges, message-passing protocols, omnichain token standards, are most appropriate for the architectural patterns of Section 8.2, and what are their cost and security implications relative to single-chain deployment? The methodological direction is a comparative evaluation across the principal interoperability mechanisms available at the time of writing (LayerZero, Wormhole, Chainlink CCIP, and the emerging native EIP-7281 / xERC20 standards), with per-pattern analysis: how state-machine transitions cross chains (Section 8.2.3), how token mints propagate across chains, and whether encrypted-state bridges between fhEVM and standard EVM chains are feasible at all. Bridge-level vulnerabilities have been the dominant class of recent blockchain exploits, so the security analysis is co-equal to the cost analysis. The success criterion is a quantitative cost-and-security comparison across at least three interoperability mechanisms for at least two of the contributions, with explicit recommendations for which mechanism is appropriate for which contribution.

9.4.3 Integration with Decentralized Identity (DID/VC)

The third direction takes up the physical-digital binding limitation of Section 8.7.1 and the institutional trust-anchor problem of Section 8.7.3. The research question is: how can W3C Verifiable Credentials and Decentralized Identifiers strengthen the attestation chain that the institutional trust anchors of the engineering contributions currently bear unilaterally? The methodological direction has three sub-components. The first is the integration of DID/VC primitives at the attestation boundary of each engineering contribution: OEM attestations as Verifiable Credentials in Chapter 6, municipal validations as Verifiable Credentials in Chapter 7, and threshold key-management attestations as Verifiable Credentials in Chapter 4. The second is the construction of multi-issuer attestation chains in which independent issuers must concur to validate a critical claim, replacing the unilateral authority of this trust anchor with a configurable threshold of authorized attesters. The third is investigation of the threshold attestation patterns by which a registry of n authorized issuers can produce k-of-n attestations, decentralizing the trust anchor while preserving institutional accountability. The success criterion is an attestation chain whose integrity does not depend on trust in any single institutional actor, demonstrated through threshold-based metadata validation in at least one of the engineering contributions.

9.4.4 Machine Learning and Blockchain Convergence

The fourth direction examines a question the thesis raises but does not answer: where does machine learning belong in the division of labor between on-chain logic, where blockchain's properties of transparency, auditability, and operator-independence are essential, and off-chain logic, where they are not? The contributions place the boundary at specific points without investigating whether those points are optimal, and three concrete sub-directions follow from this observation. The first is off-chain attestation hardening through ML-based anomaly detection: an OEM update to a vehicle NFT whose pattern of timing, geographic clustering, or unusual value combinations statistically resembles fraudulent updates could be flagged for additional review at the off-chain attestation layer before being committed on-chain, with the same mechanism applicable to municipal validations in Chapter 7 and supplier KPI submissions in Chapter 4 (modulo the encryption constraint that the latter imposes on visible signal). The second is parameter optimization: several deployment-time parameters in the contributions are presently fixed by engineering judgment, the grid cell size in Chapter 7, the priority-scoring weights in the same chapter, the KPI weights in Chapter 4, and ML-based parameter optimization would allow these to be learned from observed deployment patterns rather than fixed at deployment time, with the architectural

pattern preserved and the parameters made adaptive. The third is: federated learning over fhEVM-encrypted data as a less computationally expensive alternative to fully homomorphic computation for some classes of sustainability assessment, where the aggregation function is a gradient update rather than a closed-form weighted sum. The success criterion in each sub-direction is a quantitative improvement over the fixed-parameter baselines of the corresponding contribution: a measurable reduction in fraudulent-attestation rate for the first, a measurable improvement on a representative deployment scenario for the second, an order-of-magnitude reduction in the gas cost of encrypted computation for the third.

9.4.5 Real-World Deployment and Industry Adoption

The fifth direction is the most consequential of the five in terms of the follow-on engineering effort it would require, and it cashes the validation limitation of Section 8.7.4 directly. None of the four contributions has been deployed in production, and the gap from prototype to deployed system is non-trivial in this application class. The research question is: what are the deployment barriers, i.e., wallet management for non-technical end users, gas funding under meta-transaction sponsorship arrangements, integration with institutional information systems, regulatory approval where applicable, for each of the engineering contributions, and which can be discharged by engineering work versus which require institutional commitment? The methodological direction is a pilot deployment with at least one industrial partner per engineering contribution: a textile-industry partner for the supplier evaluation framework of Chapter 4, an OEM or vehicle inspectorate for the vehicle identity system of Chapter 6, and a small-to-medium municipality for the pothole management system of Chapter 7 (Chapter 5's analytical contribution does not require a pilot in the same sense, but would benefit from a demonstrative implementation in one of the active Battery Pass demonstrator contexts.) Each pilot would track barrier-discharge explicitly: which barriers were resolved by engineering work (a custodial wallet abstraction for non-technical users, for instance), and which required institutional commitment (regulatory sandbox approval, integration with legacy ERP systems). The deployment lessons would be documented as engineering case studies rather than as research papers, because the contribution would be deployment knowledge rather than design knowledge. The success criterion is at least one engineering contribution moved from prototype to limited production over a 12-to-24-month horizon, with documented barrier-discharge.

The five directions together would extend the research program from "patterns demonstrated" to "patterns deployed and refined." Their combined scope exceeds what a single follow-on project could absorb, and the natural decomposition is across multiple researchers

and institutional collaborations: Section 9.4.1 and Section 9.4.2 are the most natural successors to this thesis's scope, Section 9.4.5 is the most consequential for engineering practice, Section 9.4.3 the most theoretically ambitious, and Section 9.4.4 the most exploratory. The thesis does not commit to pursuing any one of the five; it argues that the five together describe the research landscape its contributions have made approachable.

The argument of this thesis admits a single statement. Four trust gaps in the vehicle lifecycle, supplier confidentiality under sustainability disclosure obligations, regulatory verifiability of component-level digital passports, the information asymmetries of second-hand vehicle markets, and the opacity of municipal infrastructure governance, admit four blockchain-based engineering responses unified by three recurring architectural patterns and differentiated by a single privacy-transparency determinant whose value is set by the structural properties of the data each system handles. This finding is not what the four underlying publications establish individually; it is what they jointly enable, and its recovery is the work the thesis performs. The architectural vocabulary developed here is positioned as a usable contribution to the broader project of building verifiable digital infrastructures for the European automotive sector, a project the thesis does not complete but whose technical foundations it argues are within reach.

References

[1] Alberto Butera, Valentina Gatteschi, Filippo Gabriele Prattichò, Daniela Novaro, and Deborah Vianello. Blockchain and NFTs-based trades of second-hand vehicles. *IEEE Access*, 11:57596–57614, 2023.

[2] Luís Alves, Miguel Sá, Estrela Ferreira Cruz, Toni Alves, Marcelo Alves, João Oliveira, Manuel Santos, and António Miguel Rosado da Cruz. A traceability platform for monitoring environmental and social sustainability in the textile and clothing value chain: Towards a digital passport for textiles and clothing. *Sustainability*, 16(1):82, 2023.

[3] Paula Fraga-Lamas and Tiago M. Fernández-Caramés. A review on blockchain technologies for an advanced and cyber-resilient automotive industry. *IEEE Access*, 7:17578–17598, 2019.

[4] Jae-Hoon Jeong, Changwon Kim, and Hyung Je Jo. Three major challenges in the shift to electric vehicles: Industrial organization, industrial policy, and a just transition. *Sociology Compass*, 18(5), 2024.

[5] European Parliament and Council. Directive (EU) 2024/1760 on corporate sustainability due diligence. https://eur-lex.europa.eu/eli/dir/2024/1760/oj/eng, 2024. Accessed: 2025-05-20.

[6] European Parliament and Council. Regulation (EU) 2023/1542 of the european parliament and of the council of 12 july 2023 concerning batteries and waste batteries. https://eur-lex.europa.eu/eli/reg/2023/1542/oj, 2023. Accessed: 2024-01-01.

[7] European Parliament. European parliament vote is a significant step towards restoring consumer trust in the used car market and helping citizens save billions. https://citainsp.org/wp-content/uploads/2018/05/20180523-joint-press-release-Ertug-report-final-corr.pdf, 2018. Accessed: 2023-01-01.

[8] Aarushi Nigam, Shreya Sangal, Abhishek Behl, Narada Jayawardena, Amit Shankar, Vijay Pereira, Yama Temouri, and Justin Zhang. Blockchain as a resource for building trust in pre-owned goods' marketing: A case of automobile industry in an emerging economy. *Journal of Strategic Marketing*, pages 1–19, 2022.

[9] Samer El-Switi and Mohammad Qatawneh. Application of blockchain technology in used vehicle market: A review. In *Proc. International Conference on Information Technology (ICIT)*, pages 49–54, July 2021.

[10] R. Rubin, C. Jacob, S. Sundar, G. Stoian, D. Danciulescu, J. Hemanth, and G. Mei. Pothole detection and assessment on highways using enhanced YOLO algorithm with attention mechanisms. *Advances in Civil Engineering*, 2025, 2025.

[11] Muhammad Hamza Asad, Saira Khaliq, Muhammad Haroon Yousaf, Muhib O. Ullah, Azmat Ahmad, and Qianni Chen. Pothole detection using deep learning: A real-time and AI-on-the-edge perspective. *Advances in Civil Engineering*, 2022, 2022.

[12] Ivan Marović, Ivanka Androjić, Niksa Jajac, Tomás Hanák, and Laura V. Gambuzza. Urban road infrastructure maintenance planning with application of neural networks. *Complexity*, 2018, 2018.

[13] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260, 2008.

[14] Melanie Swan. *Blockchain: Blueprint for a New Economy*. O'Reilly Media, Sebastopol, CA, USA, 2015.

[15] Nick Szabo. The idea of smart contracts. Technical report, Satoshi Nakamoto Institute, 1997. https://nakamotoinstitute.org/the-idea-of-smart-contracts/.

[16] Dimitris Tsingou et al. A survey on data availability in layer 2 blockchain rollups: Open challenges and future improvements. *Future Internet*, 16(9):315, 2024.

[17] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proc. 41st Annual ACM Symposium on Theory of Computing (STOC '09)*, pages 169–178, 2009.

[18] Zama. fhEVM – confidential smart contracts using fully homomorphic encryption, 2025. Accessed: 2025-05-25.

[19] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, 2004.

[20] Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45–77, 2007.

[21] Alberto Butera, Valentina Gatteschi, and others. A privacy-preserving blockchain framework for sustainable supplier evaluation in the textile industry. In *Proc. ACM Int. Conf. on Information Technology for Social Good (GoodIT '25)*, Antwerp, Belgium, 2025.

[22] Alberto Butera and Valentina Gatteschi. EU battery regulation and the role of blockchain in creating digital battery passports. In *Proc. IEEE Int. Conf. on Blockchain Research and Applications for Innovative Networks and Services (BRAINS)*, 2024.

[23] Alberto Butera, Valentina Gatteschi, Valentina Villa, and Marco Domaneschi. Design and evaluation of a blockchain-based pothole management system for smart cities and connected vehicle networks. *Future Generation Computer Systems*, 2026. Submitted, under review.

[24] Andreas M. Antonopoulos and Gavin Wood. *Mastering Ethereum: Building Smart Contracts and DApps*. O'Reilly Media, Sebastopol, CA, USA, 2018.

[25] Ethereum Foundation. The merge. https://ethereum.org/en/roadmap/merge/, 2022. Accessed: 2025-06-01.

[26] Vitalik Buterin. Ethereum whitepaper: A next-generation smart contract and decentralized application platform. https://ethereum.org/content/whitepaper/whitepaper-pdf/Ethereum_Whitepaper_-_Buterin_2014.pdf, 2014. Accessed: 2025-05-04.

[27] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, Srinivasan Muralidharan, Chet Murthy, Binh Nguyen, Manish Sethi, Gari Singh, Keith Smith, Alessandro Sorniotti, Chrysoula Stathakopoulou, Marko Vukolić, Sharon Weed Cocco, and Jason Yellick. Hyperledger Fabric: A distributed operating system for permissioned blockchains. In *Proc. 13th EuroSys Conference*, pages 1–15, 2018.

[28] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151:1–32, 2014. Updated periodically; see https://ethereum.github.io/yellowpaper/paper.pdf.

[29] OpenZeppelin. OpenZeppelin docs: ERC721 smart contract. https://docs.openzeppelin.com/contracts/4.x/api/token/erc721, 2023. Accessed: 2023-01-01.

[30] Solidity Team. Solidity documentation. https://docs.soliditylang.org/, 2024. Accessed: 2025-06-01.

[31] Nomic Foundation. Hardhat: Ethereum development environment. https://hardhat.org/, 2024. Accessed: 2025-06-01.

[32] Vitalik Buterin, Dankrad Feist, Diederik Loerakker, George Kadianakis, Matt Garnett, Mofi Bauerwald, and Protolambda. EIP-4844: Shard Blob Transactions. Ethereum Improvement Proposal, 2022. Activated on Ethereum mainnet in the Dencun upgrade, 13 March 2024. Accessed: 2026-04-20.

[33] Fabian Vogelsteller and Vitalik Buterin. EIP-20: Token standard. https://eips.ethereum.org/EIPS/eip-20, 2015. Accessed: 2025-06-01.

[34] William Entriken, Dieter Shirley, Jacob Evans, and Nastassia Sachs. EIP-721: Non-fungible token standard. https://eips.ethereum.org/EIPS/eip-721, 2018. Accessed: 2025-06-01.

[35] Witek Radomski, Andrew Cooke, Philippe Castonguay, James Therien, Eric Binet, and Ronan Sandford. EIP-1155: Multi token standard. https://eips.ethereum.org/EIPS/eip-1155, 2018. Accessed: 2025-06-01.

[36] E. Glen Weyl, Puja Ohlhaver, and Vitalik Buterin. Decentralized society: Finding Web3's soul. *SSRN Electronic Journal*, 2022.

[37] Ronan Bandeali, Alex Sergeev, Yoav Forshtat, and Lorenz Wenger. ERC-2771: Secure protocol for native meta transactions. https://eips.ethereum.org/EIPS/eip-2771, 2020. Ethereum Improvement Proposal, Final.

[38] Juan Benet. IPFS — content addressed, versioned, P2P file system. In *arXiv preprint arXiv:1407.3561*, 2014.

[39] Giulio Caldarelli. Understanding the blockchain oracle problem: A call for action. *Information*, 11(11):509, 2020.

[40] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. In *SIAM Journal on Computing*, volume 18, pages 186–208, 1989.

[41] Josselin Feist, Gustavo Grieco, and Alex Groce. Slither: A static analysis framework for smart contracts. In *Proc. 2nd IEEE/ACM International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, pages 8–15, 2019.

[42] Bernhard Mueller. Smashing ethereum smart contracts for fun and real profit. Technical report, ConsenSys, 2018. Mythril: https://github.com/Consensys/mythril.

[43] Mark Mossberg, Felipe Manzano, Eric Hennenfent, Alex Groce, Gustavo Grieco, Josselin Feist, Trent Brunson, and Artem Dinaburg. Manticore: A user-friendly symbolic execution framework for binaries and smart contracts. In *Proc. 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 1186–1189, 2019.

[44] International Organization for Standardization. ISO 3779:2009 — road vehicles — vehicle identification number (VIN) — content and structure, 2009. Last reviewed 2020.

[45] Kamaljit Singh. Blockchain technology: A potential game changer for automotive industry. *International Journal of Advanced Research in Management and Social Sciences*, 9(3):49–55, 2020.

[46] VINChain. Decentralized vehicle history — car accident history check by VIN. https://vinchain.io/, 2017. Accessed: 2023-01-01.

[47] carVertical. carVertical — VIN decoder & car history report. https://www.carvertical.com/, 2017. Accessed: 2023-01-01.

[48] Groupe Renault. Groupe Renault teams with Microsoft and VISEO to create the first-ever digital car maintenance book prototype. https://media.group.renault.com/, 2017. Accessed: 2023-06-01.

[49] Pranav Kumar Singh, Roshan Singh, and Sukumar K. Nandi. V-CARE: A blockchain based framework for secure vehicle health record system. *arXiv preprint arXiv:2007.13647*, 2020.

[50] Kei Lè Brousmiche, Thomas Heno, Christian Poulain, Andrea Dalmieres, and Elyes Ben Hamida. Digitizing, securing and sharing vehicles life-cycle over a consortium blockchain: Lessons learned. In *Proc. 9th IFIP Int. Conf. New Technologies, Mobility and Security (NTMS)*, pages 1–5, 2018.

[51] Semere Amle Gebreab, Haya R. Hasan, Khaled Salah, and Raja Jayaraman. NFT-based traceability and ownership management of medical devices. *IEEE Access*, 10:126394–126411, 2022.

[52] Noura Alnuaimi, Aysha Almemari, Mohamed Madine, Khaled Salah, Hamda Al Breiki, and Raja Jayaraman. NFT certificates and proof of delivery for fine jewelry and gemstones. *IEEE Access*, 10:101263–101275, 2022.

[53] Farah K. Elmay, Khaled Salah, Raja Jayaraman, and Ibrar A. Omar. Using NFTs and blockchain for traceability and auctioning of shipping containers and cargo in maritime industry. *IEEE Access*, 10:124507–124522, 2022.

[54] Foad Naseri et al. Digital twin of electric vehicle battery systems: Comprehensive review of the use cases, requirements, and platforms. *Renewable and Sustainable Energy Reviews*, 179:113280, 2023.

[55] Mathieu Chanson, Andreas Bogner, Felix Wortmann, and Elgar Fleisch. Blockchain as a privacy enabler: An odometer fraud prevention system. In *Proc. ACM Int. Joint Conf. Pervasive and Ubiquitous Computing and Proc. ACM Int. Symp. Wearable Computers*, pages 13–16, 2017.

[56] Marius Vasile and Bogdan Groza. DeMetrA — decentralized metering with user anonymity and layered privacy on blockchain. In *Proc. 23rd Int. Conf. System Theory, Control and Computing (ICSTCC)*, pages 560–565, 2019.

[57] Gokay Saldamli, Krithika Karunakaran, Vinay Kumar Vijaykumar, Wen Pan, Sanjeev Puttarevaiah, and Levent Ertaul. Securing car data and analytics using blockchain. In *Proc. 7th Int. Conf. Software Defined Systems (SDS)*, pages 153–159, 2020.

[58] Till Reimers, Florian Leber, and Ulrike Lechner. Integration of blockchain and Internet of Things in a car supply chain. In *Proc. IEEE Int. Conf. Decentralized Applications and Infrastructures (DAPPCON)*, pages 146–151, 2019.

[59] Wenbin Zhang, Ling Guo, and Yexin Li. Used car traceability system based on blockchain. In *Proc. Int. Conf. Information Science, Parallel and Distributed Systems (ISPDS)*, pages 235–240, 2021.

[60] Jing Zhang, Haiyan Zhao, Yong Yang, and Jiaqi Yan. Towards transparency and trustworthy: A used-car deposit platform based on blockchain. In *Proc. IEEE 19th Int. Conf. Software Quality, Reliability and Security Companion (QRS-C)*, pages 46–50, 2019.

[61] Soonduck Yoo and Byunggil Ahn. A study for efficiency improvement of used car trading based on a public blockchain. *Journal of Supercomputing*, 77(9):10621–10635, 2021.

[62] M. S. N. Kumar, G. C. Akshatha, Mayank D. Bangre, M. Dhanush, and C. Abhishek. Decentralized used cars bidding application using Ethereum. In *Proc. IEEE Int. Conf. Computing Systems and Information Technology for Sustainable Solutions (CSITSS)*, pages 1–7, 2021.

[63] Gokul Subramanian and Arjun S. Thampy. Implementation of hybrid blockchain in a pre-owned electric vehicle supply chain. *IEEE Access*, 9:82435–82454, 2021.

[64] Jiewu Leng et al. Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: A survey. *Renewable and Sustainable Energy Reviews*, 132:110112, 2020.

[65] Elias Ribeiro da Silva et al. Unleashing the circular economy in the electric vehicle battery supply chain: A case study on data sharing and blockchain potential. *Resources, Conservation and Recycling*, 193:106969, 2023.

[66] Fatih Ecer, Gholamreza Haseli, Raghunathan Krishnankumar, and Mostafa Hajiaghaei-Keshteli. Evaluation of sustainable cold chain suppliers using a combined multi-criteria group decision-making framework under fuzzy ZE-numbers. *Expert Systems with Applications*, 245:123063, 2024.

[67] Sepehr Hendiani and Grit Walther. Towards sustainable futures: Rethinking supplier selection through interval-valued intuitionistic fuzzy decision-making. *International Journal of Production Economics*, 285:109620, 2025.

[68] Sanjucta Saha Sithi, Mst Anjuman Ara, Ahnaf Tahmid Dhrubo, Abid Hasan Rony, and Md Abdus Shabur. Sustainable supplier selection in the textile industry using triple bottom line and SWARA-TOPSIS approaches. *Discover Sustainability*, 6(1):1–23, 2025.

[69] Ali Vaezi, Erfan Rabbani, and Seyed Ahmad Yazdian. Blockchain-integrated sustainable supplier selection and order allocation: A hybrid BWM-MULTIMOORA and bi-objective programming approach. *Journal of Cleaner Production*, 444:141216, 2024.

[70] Kamar Zekhnini, Abla Chaouni Benabdellah, Anass Cherrafi, Imane Bouhaddou, and Surajit Bag. Viable industrial supplier performance evaluation using fuzzy inference system: A case of the automotive industry. *Journal of Business & Industrial Marketing*, 40(4):941–962, 2025.

[71] Mahyar Abbasian and Amin Jamili. A hybrid machine learning approach to evaluate and select agile-resilient-sustainable suppliers considering supply chain 4.0: A real case study. *Process Integration and Optimization for Sustainability*, 9(2):717–735, 2025.

[72] József Pap, Csaba Makó, Adrián Horváth, Zoltán Baracskai, Tamás Zelles, Judit Bilinovics-Sipos, and Sándor Remsei. Enhancing supply chain safety and security: A novel AI-assisted supplier selection method. *Decision Making: Applications in Management and Engineering*, pages 22–41, 2025.

[73] Samer Haffar and Eren Özceylan. Blockchain-based system for supplier selection in sustainable and leagile supply chains. *IEEE Access*, 12:139883–139911, 2024.

[74] Azam Modares, Nasser Motahari Farimani, and Farzad Dehghanian. A new vendor managed inventory for perishable products considering supplier selection. *Process Integration and Optimization for Sustainability*, 9(1):57–74, 2025.

[75] Katharina Berger et al. Confidentiality-preserving data exchange to enable sustainable product management via digital product passports — a conceptualization. In *Procedia CIRP*, volume 116, pages 354–359, 2023.

[76] Saurabh Sahai, Nishtha Singh, and Pankaj Dayama. Enabling privacy and traceability in supply chains using blockchain and zero knowledge proofs. In *Proc. IEEE Int. Conf. on Blockchain*, pages 134–143, Rhodes, Greece, 2020.

[77] Jiaming Fang et al. Blockchain-cloud privacy-enhanced distributed industrial data trading based on verifiable credentials. *Journal of Cloud Computing*, 13:30, 2024.

[78] Hans Eric Melin et al. Global implications of the EU battery regulation. *Science*, 373(6553):384–387, 2021.

[79] Global Battery Alliance. The global battery alliance battery passport concept. https://www.globalbattery.org/media/pilot/documents/gba-bp-pilot-master.pdf, 2023. Accessed: 2024-01-01.

[80] Battery Pass Consortium. Battery passport content guidance. https://thebatterypass.eu/assets/images/content-guidance/pdf/2023_Battery_Passport_Content_Guidance.pdf, 2023. Accessed: 2024-01-01.

[81] Veronika Siska et al. Building a sustainable battery supply chain with digital battery passports. In *IDIMT 2023: New Challenges for ICT and Management*, pages 347–354, 2023.

[82] Gabriele Bandini et al. An RFID system enabling battery lifecycle traceability. In *Proc. IEEE Int. Workshop on Metrology for Automotive (MetroAutomotive)*, pages 46–50, Modena, Italy, 2023.

[83] Simon Nowacki et al. Digital product passports: Use cases framework and technical architecture using DLT and smart contracts. In *Proc. 19th Int. Conf. on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*, pages 373–380, Pafos, Cyprus, 2023.

[84] Sin Yong Teng et al. Recent advances on industrial data-driven energy savings: Digital twins and infrastructures. *Renewable and Sustainable Energy Reviews*, 135:110208, 2021.

[85] Zia Khan, Amjad Gul Abbasi, and Zeeshan Pervez. Blockchain and edge computing–based architecture for participatory smart city applications. *Concurrency and Computation: Practice and Experience*, 32, 2019.

[86] Juho Hamari, Jonna Koivisto, and Harri Sarsa. Does gamification work? — a literature review of empirical studies on gamification. In *Proc. 47th Hawaii Int. Conf. System Sciences (HICSS)*, pages 3025–3034, 2014.

[87] Paula Bitrián, Isabel Buil, and Sara Catalán. Enhancing user engagement: The role of gamification in mobile apps. *Journal of Business Research*, 132:170–185, 2021.

[88] Bo Tian, Yulei Yuan, Haibo Zhou, Zhixiong Yang, and Chao Qi. Pavement management utilizing mobile crowd sensing. *Advances in Civil Engineering*, 2020, 2020.

[89] Yong Li, Huijie Song, Yan Zhao, Nianmin Yao, Nan Wang, and David Megías. Anonymous data reporting strategy with dynamic incentive mechanism for participatory sensing. *Security and Communication Networks*, 2021, 2021.

[90] Jinbo Xiong, Rui Ma, Lei Chen, Youliang Tian, Li Lin, Baohua Jin, and Kim-Kwang Raymond Choo. Achieving incentive, security, and scalable privacy protection in mobile crowdsensing services. *Wireless Communications and Mobile Computing*, 2018, 2018.

[91] Xiaojun Chen, Banfu Wang, Jingfeng Chen, Xiujie Zhang, Shengchu Liu, Guojing Zhou, Pei Li, Xuefeng Zhao, and Subhas Mukhopadhyay. Innovative life-cycle inspection strategy of civil infrastructure: Smartphone-based public participation. *Structural Control and Health Monitoring*, 2023, 2023.

[92] Markku Knuuti, Konsta Sirviö, Vesa Männistö, and Susanna Suomela. Road asset management with a mobile game and artificial intelligence. In *Roads and Airports Pavement Surface Characteristics*, pages 189–200. CRC Press, London, 1 edition, 2023.

[93] Satya Sahoo, Suresh Kumar, and Mahesh Vyas. How does blockchain technology contribute to the development and functioning of smart cities? — a scoping review and future research directions. *Journal of Economic Surveys*, 39:1907–1928, 2024.

[94] Siqing Shan, Xiaoyue Duan, Yun Zhang, Tong T. Zhang, Hao Li, and Zhe Chen. Research on collaborative governance of smart government based on blockchain technology: An evolutionary approach. *Discrete Dynamics in Nature and Society*, 2021, 2021.

[95] A. K. M. Bahalul Haque, Bharat Bhushan, and Gaurav Dhiman. Conceptualizing smart city applications: Requirements, architecture, security issues, and emerging trends. *Expert Systems*, 39, 2021.

[96] Ismael Soto, Miguel Calderon, Oscar Amador, and Manuel Urueña. A survey on road safety and traffic efficiency vehicular applications based on C-V2X technologies. *Vehicular Communications*, 33:100428, 2022.

[97] Long Chen, Zhenxing Fan, Pan Liu, Zhengwei Qian, and Alejandro Preciado. Optimization model of network-level pavement maintenance decision considering user travel time and vehicle fuel consumption costs. *Advances in Civil Engineering*, 2021, 2021.

[98] Tanuj Chopra, Manoranjan Parida, Naveen Kwatra, Prerna Chopra, and Francesco Canestrari. Development of pavement distress deterioration prediction models for urban road network using genetic programming. *Advances in Civil Engineering*, 2018, 2018.

[99] Van Hoai Tran, Benjamin Lenssens, Ali Kassab, Artur Laks, Etienne Rivière, Guillaume Rosinosky, and Ramin Sadre. Machine-as-a-service: Blockchain-based management and maintenance of industrial appliances. *Engineering Reports*, 5, 2022.

[100] Syeda Humaira Rumpa, Shoronika Ishrat, Syed Tanveer Reza, M. Shafiqul Islam Suman, M. Faysal Ahmmed, and Nafees Mansoor. InfraChain: A sensor-enabled roadway management application using blockchain and digital twin. In *Proc. World Conf. on Information Systems for Business Management*, pages 457–463, Singapore, 2024. Springer Nature.

[101] Roman Matzutt, Jan Pennekamp, and Klaus Wehrle. Poster: Accountable processing of reported street problems. In *Proc. 2023 ACM SIGSAC Conf. on Computer and Communications Security (CCS '23)*, pages 3591–3593, 2023.

[102] Ammar S. Mubarak, Zainab S. Ameen, Patrick Tonga, Chadi Altrjman, and Fadi Al-Turjman. A framework for pothole detection via the AI-blockchain integration. In Fadi Al-Turjman and Jawad Rasheed, editors, *Forthcoming Networks and Sustainability in the IoT Era*, pages 398–406. Springer International Publishing, Cham, 2022.

[103] mySociety. FixMyStreet: Report, view, or discuss local problems. https://www.fixmystreet.com/, 2024. Accessed: 2025-01-06.

[104] SeeClickFix Inc. SeeClickFix: Citizen relationship management for local governments. https://seeclickfix.com/, 2024. Accessed: 2025-01-06.

[105] Waze. Waze — driving directions, live traffic & road conditions. https://www.waze.com/, 2024. Accessed: 2025-01-06.

[106] Md Mokhlesur Islam, Asif A. Newaz, Lin Song, Bernard Lartey, Staton Lin, Wei Fan, Ali Hajbabaie, M. Arifur Khan, Arash Partovi, Thanarit Phuapaiboon, Abdollah Homaifar, and Ali Karimoddini. Connected autonomous vehicles: State of practice. *Applied Stochastic Models in Business and Industry*, 39:684–700, 2023.

[107] Prashanth Mohan, Venkata N. Padmanabhan, and Ramachandran Ramjee. Nericell: Rich monitoring of road and traffic conditions using mobile smartphones. In *Proc. 6th ACM Conf. on Embedded Network Sensor Systems (SenSys '08)*, pages 323–336, 2008.

[108] Subhasish Bej, Soumen Roy, Dipam Daw, Amrita Paul, Sunit Saha, Saptarshi Maity, and Nabanita Ghosh. SmartPave: An advanced IoT-based system for real-time pothole detection, tracking, and maintenance. In *Proc. Int. Conf. in Advances in Power, Signal, and Information Technology (APSIT)*, pages 532–537, 2023.

[109] R. Govindan, P. S. Santhi, and S. R. Srinivasan. Smart road condition monitoring system with IoT integration. *AIP Conference Proceedings*, 3279:020165, 2025.

[110] David Rozas, Antonio Tenorio-Fornés, Silvia Díaz-Molina, and Samer Hassan. When Ostrom meets blockchain: Exploring the potentials of blockchain for commons governance. *SAGE Open*, 11(1):1–14, 2021.

[111] Alex Pazaitis, Primavera De Filippi, and Vasilis Kostakis. Blockchain and value systems in the sharing economy: The illustrative case of Backfeed. *Technological Forecasting and Social Change*, 125:105–115, 2017.

[112] Evrim Tan and A. Paula Rodriguez Müller. Paths to citizens-controlled coproduction: The use of blockchain technology in digital coproduction. *Public Management Review*, 26(12):3481–3500, 2024.

[113] Maxat Kassen. Blockchain and E-participation: New affordances and challenges for digital political participation. *Journal of Public Affairs*, 25(4), 2025.

[114] Mark C. Ballandies. To incentivize or not: Impact of blockchain-based cryptoeconomic tokens on human information sharing behavior. *IEEE Access*, 10:74111–74130, 2022.

[115] Daphne R. Raban. The incentive structure in an online information market. *Journal of the American Society for Information Science and Technology*, 59(14):2284–2295, 2008.

[116] Kai Chen, Yang Fan, Yulin Fang, and Xin Luo. Beyond money: Incentive effects of tokenized ownership on user contribution in DAOs. *Journal of Operations Management*, 71(7):988–1016, 2025.

[117] 117th United States Congress. H.R. 5376 — Inflation Reduction Act of 2022. https://www.congress.gov/bill/117th-congress/house-bill/5376/text, 2022. Accessed: 2026-04.

[118] DIN e.V. and DKE. DIN DKE SPEC 99100: Requirements for data attributes of the battery passport. Technical report, DIN, DKE, and Battery Pass Consortium, January 2025. Standard specification.

[119] Catena-X Automotive Network e.V. Digital Product Passport Use-Case Cluster and EcoPass KIT. https://catena-x.net/use-case-cluster/digital-product-passport/, 2025. Accessed: 2026-04.

[120] Battery Pass Consortium. Battery passport technical guidance. Technical report, Systemiq and Fraunhofer IPK, March 2024. Accessed: 2026-04.

[121] European Parliament and Council. Regulation (EU) 2025/1561 amending Regulation (EU) 2023/1542 as regards the application date of certain obligations (Omnibus IV). Official Journal of the European Union, 2025. Accessed: 2026-04.

[122] European Parliament and Council. Regulation (EU) 2024/1781 of the European Parliament and of the Council of 13 June 2024 establishing a framework for the setting of ecodesign requirements for sustainable products (ESPR). Official Journal of the European Union, L series, 28 June 2024, 2024. Entered into force 18 July 2024.

[123] George A. Akerlof. The market for "Lemons": Quality uncertainty and the market mechanism. *The Quarterly Journal of Economics*, 84(3):488–500, 1970.

[124] SmartContractSecurity Community. SWC registry: Smart contract weakness classification and test cases. https://swcregistry.io/, 2020. Accessed: 2026-04-20.

[125] Ammar Battah, Mohammed Madine, Ibrar Yaqoob, Khaled Salah, Haya R. Hasan, and Raja Jayaraman. Blockchain and NFTs for trusted ownership, trading, and access of AI models. *IEEE Access*, 10:112230–112249, 2022.

[126] Umer Majeed, Latif U. Khan, Shahid Shah Hassan, Zhu Han, and Choong Seon Hong. FL-incentivizer: FL-NFT and FL-tokens for federated learning model trading and training. *IEEE Access*, 11:4381–4399, 2023.

[127] Ahmad Musamih, Ibrar Yaqoob, Khaled Salah, Raja Jayaraman, Mohammed Omar, and Samer Ellahham. Using NFTs for product management, digital certification, trading, and delivery in the healthcare supply chain. *IEEE Transactions on Engineering Management*, 2022. Early access.

[128] Mohammed Madine, Khaled Salah, Raja Jayaraman, Ammar Battah, Haya Hasan, and Ibrar Yaqoob. Blockchain and NFTs for time-bound access and monetization of private data. *IEEE Access*, 10:94186–94202, 2022.

[129] OpenZeppelin. AccessControl — openzeppelin contracts documentation. https://docs.openzeppelin.com/contracts/4.x/access-control, 2023. Accessed: 2026-04-20.

[130] Murat Demir, Oktay Turetken, and Alex Ferworn. Blockchain-based transparent vehicle insurance management. *IEEE Transactions on Vehicular Technology*, 71(12):12627–12637, 2022.

[131] Jayden Windle, Benny Giang, Steve Locascio, and Wilkins Gunderson. ERC-6551: Non-fungible Token Bound Accounts. Ethereum Improvement Proposal, 2023. Proposed February 2023; status: Draft / Under Review. Accessed: 2026-04-20.

[132] Anders, Lance Snow, Shrug, and Nathan Gang. ERC-4906: EIP-721 Metadata Update Extension. Ethereum Improvement Proposal, 2022. Proposed March 2022; reached Final status late 2022. Accessed: 2026-04-20.

[133] Meta Open Source. React — a JavaScript library for building user interfaces. https://react.dev, 2023. Accessed: 2026-04-20.

[134] ChainSafe Systems. web3.js — Ethereum JavaScript API. https://web3js.readthedocs.io, 2023. Accessed: 2026-04-20.

[135] Protocol Labs. JS-IPFS (`js-ipfs`) — InterPlanetary File System implementation in JavaScript. https://github.com/ipfs/js-ipfs, 2023. Accessed: 2026-04-20.

[136] ConsenSys Software. MetaMask — A crypto wallet & gateway to blockchain apps. https://metamask.io, 2023. Accessed: 2026-04-20.

[137] Ethereum Foundation. Remix IDE — An online IDE for Smart Contract development in Solidity. https://remix.ethereum.org, 2023. Accessed: 2026-04-20.

[138] ConsenSys. Ethereum Evolved: Dencun Upgrade Part 5, EIP-4844. ConsenSys Blog, 2024. Accessed: 2026-04-20.

[139] Charles Herder, Meng-Day Yu, Farinaz Koushanfar, and Srinivas Devadas. Physical Unclonable Functions and Applications: A Tutorial. *Proceedings of the IEEE*, 102(8):1126–1141, 2014.

[140] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability. In *Proc. 41st IEEE Symposium on Security and Privacy (SP)*, pages 910–927, 2020.

[141] American Society of Civil Engineers. 2021 report card for America's infrastructure: Roads. https://infrastructurereportcard.org/cat-item/roads-infrastructure/, 2021. Accessed: 2026-04-25.

[142] American Automobile Association. Potholes pack a punch as drivers pay $26.5 billion in related vehicle repairs. AAA Newsroom, March 1, 2022. https://newsroom.aaa.com/2022/03/aaa-potholes-pack-a-punch-as-drivers-pay-26-5-billion-in-related-vehicle-repairs/, 2022. Accessed: 2026-04-25.

[143] Berhane Grum, Dawit Tsegaye, Zeru Tariku, Daniel Gebremariam, Haftom Gebremicael, Tesfay Kuhilen, Solomon Yemane, Amanuel Aregawi, Beza A. Abebe, and Qiang Cheng. Applicability and cost implication of labor-based methods for sustainable road maintenance (SRM) in developing countries. *Advances in Civil Engineering*, 2023, 2023.

[144] Alex Beregszaszi, Remco Bertram, Felix Foundry, and Tom Tabaie. EIP-712: Typed structured data hashing and signing. https://eips.ethereum.org/EIPS/eip-712, 2017. Ethereum Improvement Proposal, Final.

[145] OpenGSN Project. OpenGSN: The decentralized gas station network for Ethereum dapps. https://opengsn.org/, 2024. Accessed: 2026-04-25.

[146] Biconomy. Biconomy sdk: Account abstraction and gasless transactions for Web3 applications. https://www.biconomy.io/, 2024. Accessed: 2026-04-25.

[147] Uber Technologies. H3: A hexagonal hierarchical geospatial indexing system. https://h3geo.org/, 2024. Accessed: 2026-04-25.

[148] IBM Corporation. IBM Maximo Application Suite: Pricing. https://www.ibm.com/products/maximo/pricing, 2026. Accessed: 2026-04-25. Pricing structure: AppPoints credit-based licensing; industry-reported annual licensing typically $100,000–$500,000 for mid-sized deployments, with implementation consulting matching or exceeding the licensing fee.