Selective Data Sharing for Digital Product Passports using Heterogeneous Twins

Ranjit Kannappan,[1,2] Julien Hatin[1] Emmanuel Bertin[1,2] Noel Crespi[2]
[1]Orange Innovation, 14000 Caen, France
[2]Samovar, Telecom SudParis, Institut Polytechnique de Paris, France.
(ranjit.kannappan, julien.hatin, emmanuel.bertin)@orange.com, noel.crespi@telecom-sudparis.eu

***Abstract*—Continuous evolution of digital products makes them generate vast amounts of lifecycle data, creating new demands for secure, transparent, and efficient data sharing across stakeholders. While the EU's Digital Product Passport (DPP) initiative aims to address these challenges, current solutions lack robust architectures for secure, real-time, and selective data exchange, especially during the crucial middle-of-life (MOL) phase. This paper introduces a decentralized architecure where mini digital twins (mDT) project distinct product DT states for selective data disclosure, supporting multiple lifecycle stages and sustainable decision-making. Our approach demonstrates stakeholder-specific data sharing through heterogeneous mDT using OrbitDB, enabling low-latency lifecycle data access for Digital product passports.**

***Index Terms*—Digital Product Passport (DPP), Selective Data Disclosure, Lifecycle Management, Digital Twins**

I. Introduction

The rapid proliferation of connected products and the growing demand for sustainable resource management have made product lifecycle transparency a critical challenge for industry and regulators alike. The European Union's Battery Pass project exemplifies this shift, aiming to establish a Digital Product Passport (DPP) for batteries that ensures traceable, trustworthy lifecycle data is available to all authorized stakeholders [1], [2]. DPPs are envisioned as digital records that facilitate responsible end-of-life (EOL) decisions, such as recycling and repurposing, by providing recyclers and other EOL actors with essential information about a product's history, composition, and usage [3] .

However, the current focus of DPP development is predominantly on EOL processes, often overlooking the importance of dynamic information flow during the middle-of-life (MOL) phase, when products are actively used and maintained. This gap is significant: maintainers and users require timely, selective access to operational data to enable predictive maintenance, optimize resource use, and extend product lifespan, all of which are crucial for achieving true sustainability [4]. Without robust frameworks for secure, real-time, and stakeholder-specific data sharing, the potential of DPPs to drive circular economy outcomes remains limited.

Digital twin (DT) is a virtual representation of a physical asset, process or system. DTs have been emerged as a promising foundation for DPPs, enabling continuous monitoring and lifecycle management [5], [6]. DT ecosystems consists of multiple lifecycle parties, who require different pieces of information to perform their roles. Sharing an entire DT state is often problematic due to intellectual property concerns, sensitive private data regulations and bandwith/cost constraints for high frequency streams [7]. Yet, most existing DT architectures are either centralized, risking single points of failure and limited scalability, or rely heavily on blockchain, which introduces high latency, scalability bottlenecks, and privacy concerns [8].

To address these limitations, we propose a decentralized architecture utilizing mini digital twins (mDTs) that enables selective data disclosure allowing authorized stakeholders to access only relevant information based on their specific roles and operational needs.

This leads us to our central research question:

How can decentralized digital twins enable selective, low-latency sharing of lifecycle data for Digital Product Passports?

Contribution are

1) We propose a novel architecture where mDTs create heterogenous views of evolving product DT states for decentralised and controlled sharing of product lifecycle data, ensuring only the necessary information is disclosed to relevant stakeholders.
2) We present a discrete-time dynamic system model formalizing Product DT evolution and its projection into distinct mDT states (AMDT/SUMDT/SBMDT) for selective disclosure
3) The proposed system is evaluated against state of the art system for twin creation and sharing latency along with micro-benchmarking of encryption/decryption, storage, and data sharing overhead across varying data sizes.

II. Related works

Decentralization in digital twins addresses a fundamental challenge in multi stakeholder environment which is to enable secure verifiable data sharing without relying on single trusted authority. When considering a lifecycle environment of a product involving multiple parties such as manufacturers, maintainers, product users and recyclers, the DT data must be shared selectively while maintaining integrity, provenance and controlled access. As emphasied by [9] , the selective data sharing requirement becomes criticial particularly in cross-company networks where DT acts as a hub to share data while creating value through collaboration.

To achieve decentralized DT management, existing works heavily rely on blockchain based architecture. [7] propose a framework for secure DT data sharing using permissioned blockchain, implementing access control through smart contracts. [8] develop Ethertwin, an Ethereum based DApp for DT information management with fine grained access control, they explicitly evaluate the cost and performance implications of DT operations through blockchain operations. [10], [11] store DT data directly in blockchain to ensure authenticity and to prevent unauthorized modifications which increases the storage cost in blockchain. However, storing large arbitrary data such as lifecycle data directly on blockchain is expensive and reduces scalability [12]. Alternatively, [13] proposes a smart contract based DT creation that stores file in IPFS storage and commits the content identifiers to Ethereum smart contract. While this might improve immutability and traceability, it's reliance on frequent on-chain transactions for each DT lifecycle phase introduces scalability and latency limitations.

Based on extensive expert interviews, [9] predict that 80% of future digital twin applications will involve to cross-company usage within 5 years, emphasizing the need to ensure data sovereignty in such collaborative scenarios. Building on this, DPP leverages digital twins as dynamic, real time representation of physical products to enable continuous lifecycle data collection, processing and sharing across multiple stakeholders. As highlighted by [14], digital twins serve as foundational enabler for DPP. The dynamic data processing capability of digital twin allows DPP to extend beyond the static data documentation and facilitate predictive maintenance, usage optimization, and sustainability improvements during the middle of life phase, as emphsized in several review studies [15].

Despite these advantages, challenges remain in ensuring selective data disclosure of private information. [16] proposes multi-blockchain architecture to manage product data from DPP and likewise [17] demonstrate blockchain backed machine passports integrated with DT but both approaches acknowledge limitations in scalability and dynamic data processing across multi party infrastructure. [3], [18] further explore decentralized DTs for DPP, focusing on enhancing EOL stage data processing. Many of these existing approaches do not sufficiently address improving sustainability in MOL stage and acknowledge scalability issues by using blockchain.

Overall, blockchain-centric designs face two major drawbacks: high latency from frequent on-chain commits and potential privacy leakage due to publicly visible transactions. While private or permissioned blockchains can mitigate some issues, studies indicate they often struggle to meet the latency, scalability, and privacy demands of cross-company DT architectures [19]–[21].

III. Overview of Proposed System

The proposed system integrates data-driven digital twins that enable ubiquitous access to product information across its lifecycle. It has three main components that correspond to the formal notions of state, observation, and control: the Product Digital Twin (Product DT) as the stateful representation; and Mini Digital Twins (mDTs) as constrained observation/control views over selected attributes (ref Fig 1).

A. Product Digital Twin

The Product Digital Twin acts as a data-driven, continuous accumulation and extension of data representing the physical product. It is implemented using the Asset Administration Shell (AAS), the de-facto standard in Industry 4.0. As the physical product is assembled or modified, the digital representation is incrementally extended with modular AAS submodels. Each physical component maps to one or more submodels, which are created, linked, updated, or removed as assembly operations occur. The Product DT captures the evolving state and context of the product through sensors, usage logs, and service records. It provides rich contextual information across design, manufacturing, operation, and maintenance. In the formal view, the Product DT captures the evolving state as attribute sets, storing it on IPFS+OrbitDB which provides tamper free evolution of those attributes over time.

B. Mini Digital Twins

To support collaborative decision-making, the system introduces MiniDTs as lightweight, stakeholder-specific views of the Product DT. While the Product DT maintains complete information, stakeholders require selective access based on their needs.

MiniDTs are projections of the Product DT state over selected attributes. Three types are proposed: **Asynchronous Mini Digital Twin (AMDT)** offering read-only access to selected static attributes; **Synchronous Unidirectional Mini Digital Twin (SUMDT)** providing continuous observation of data streams without write capability; and **Synchronous Bidirectional Mini Digital Twin (SBMDT)** supporting real-time observation with authorized write capabilities.

IV. Architecture

The proposed system architecture (ref Fig 1) comprises of key components for creation of the product twin and sharing of data through different mDTs. As different mDTs are shared to appropriate stakeholders who holds different permission. This section explains in detail the components introduced in the previous section.

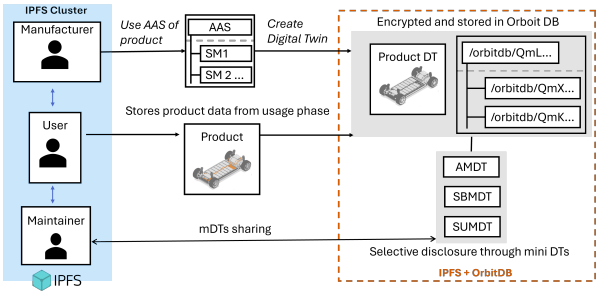


Fig. 1: Proposed Architecture with Product DT stored in IPFS OrbitDB and shared through mini DT to stakeholders

A. Creation of Product Twin

The creation of the Product Digital Twin (Product DT) forms the foundational layer of our architecture, representing the physical product throughout its lifecycle. Product DT creation begins during manufacturing, where an AAS file is generated to describe the product's initial state. As components are added or modified, corresponding AAS submodels are incrementally created and linked, each reflecting a logical or physical component with nested elements and properties representing sensor modules. Pre-approved submodels from the IDTA organization such as carbon footprint, predictive maintenance, nameplate define the base of the digital representation.

For decentralized and tamper-resistant storage, we use an OrbitDB database built on IPFS. OrbitDB employs CRDTs (Conflict-Free Replicated Data Types) to ensure data consistency across distributed peers, even during offline periods. Unlike standard IPFS, OrbitDB supports pubsub-based instant updates which is useful for sensor data sharing with low-latency and near real-time synchronization.

The storage process follows the hierarchical structure of the AAS document. Each submodel is processed through a StoreElement operation. If the element represents a property (e.g., temperature or charge level), an OrbitDB event log records its real-time updates. For composite modules, a key-value database stores encrypted addresses of child elements, secured with a locally retained symmetric key. Through recursive StoreElement operations, a hierarchical network of key-value and event log databases is constructed, mirroring the AAS structure and preserving modular separation.

Each module maintains its own encryption key and OrbitDB address, enabling fine-grained access control for stakeholder-specific data access. Using an IPFS Cluster ensures that Product DT data is automatically replicated across nodes through built-in replication policies, unlike a standalone IPFS node where data remains local until requested. The storage process produces a root OrbitDB address as the entry point for the Product DT, and in the following sections, we show how this hierarchical design supports stakeholder-specific views from the Product DT, enabling the creation of heterogeneous twins tailored to specific needs of the stakeholders.

B. Sharing of Digital Twin

1) Data replication: Data replication is handled by the IPFS cluster, which pins every ProductDT submodel and property log to a configurable replication factor. This ensures an up-to-date copy on all stakeholder nodes. Each stakeholder becomes a replication target and automatically pulls and stores new data when added. OrbitDB only allows authorized parties to update the database, so a canonical version is maintained. Only updates signed by authorized writer keys advance the log head; unauthorized edits produce a different CID and fail signature checks. Signed records and deterministic conflict resolution ensure that if nodes diverge, the latest valid signed version becomes canonical and tampering is exposed.

2) Mini DT based data sharing: Sharing ProductDT data requires a secure mechanism to preserve data sovereignty and enable selective disclosure. The modular storage architecture enables fine-grained selective disclosure. Our approach combines cryptographic key distribution with OrbitDB access control to create mDTs (AMDT, SUMDT, SBMDT) that provide stakeholder-specific views.

When a stakeholder requires access to specific submodels or properties, the data owner identifies the required symmetric keys by traversing the ProductDT structure. For example, creating a SUMDT for a battery maintenance provider involves sharing keys for selected submodels while excluding others. These keys and OrbitDB addresses are packaged into a payload, encrypted with the stakeholder's public key, stored in IPFS, and the content hash is shared through a pubsub channel.

The sharing process supports different mDT types. AMDT provides access to static metrics and historical data. SUMDT includes keys for real-time sensor data. SBMDTs require write permission managed through OrbitDB's access controller; for example, granting write access to a "Service History" log while maintaining read-only access to sensor data.

After receiving and decrypting the payload, the stakeholder's mDT maintains live references to the original OrbitDB addresses, ensuring updates to the ProductDT are reflected in all authorized mDTs.

C. Formalization of Product Digital Twin state and Mini Digital Twins

1) General Discrete-Time Dynamical System in Cyber-Physical Systems: A general discrete-time dynamical system in the context of cyber-physical systems (CPS) can be described as:

$$x(t+1) = f(x(t), u(t), \xi(t)), \quad y(t) = g(x(t)) \tag{1}$$

where:

- $x(t)$ represents the system state at time t,
- $u(t)$ represents control actions applied to the system,
- $\xi(t)$ represents external disturbances, and
- $y(t)$ is the observable output of the system.

In CPS, this formulation models how the physical system evolves over time under both controlled actions and external influences. For digital twins, $x(t)$ corresponds to the internal state of the physical asset, $y(t)$ is what the digital twin observes and visualizes, $u(t)$ represents decisions or automated interventions, and $\xi(t)$ captures uncontrollable environmental factors.

2) Ivanov's Data-Driven Digital Twin Formalization: Following Ivanov [22], a data-driven digital twin can be expressed as:

$$x(t+1) = f(x(t), u_{\text{pl}}(t), v(x(t), t), \xi(t)), \quad y(t) = g(x(t)) \tag{2}$$

where:

- $x(t)$: state of the system,
- $y(t)$: performance or observable output,
- $u_{\text{pl}}(t)$: planned control inputs,

- $v(x(t), t)$: adaptive control applied dynamically,
- $\xi(t)$: external disturbances.

Observability refers to the aspects of $x(t)$ that are visible through $y(t)$ over time. **Control** corresponds to the actions $u_{\text{pl}}(t)$ or $v(x(t), t)$ that influence the system state. **Adaptation** is represented by $v(x(t), t)$, which adjusts the system dynamically. **Disturbances** $\xi(t)$ are external events beyond the control of the system.

3) Product Digital Twin Formalization: Inspired by Ivanov's model, the proposed Product Digital Twin (Product DT) is formulated for a descriptive digital twin as follows:

$$x(t+1) = f_{\text{prod}}(x(t), \xi(t)), \quad y(t) = g_{\text{prod}}(x(t)) \tag{3}$$

where:

- $x(t)$: product attributes being observed,
- $y(t)$: observed outputs or visualizations in the twin,
- $\xi(t)$: disturbances affecting the product,
- no control ($u(t)$) or adaptation ($v(t)$) is applied at this stage.

Here, the Product DT functions solely as a descriptive twin, tracking the product's state and considering disturbances, without executing control or adaptive actions. Observability remains the key updating mechanism.

4) Mini Digital Twins (MiniDTs): To enable controlled information sharing with stakeholders, we define a MiniDT m as a restricted, shareable view derived from the main Product DT DT_{prod}:

$$m = (DT_{\text{prod}}, S, \text{Obs}, \text{Ctrl}, \text{Type}, t_0) \tag{4}$$

where:

- DT_{prod}: the main Product Digital Twin from which the MiniDT is derived,
- $S \subseteq x_{\text{prod}}$: selected product attributes from DT_{prod} to share with a stakeholder,
- Obs: observation policy, either static snapshot or continuous streaming of y_S from DT_{prod} (sensor data),
- Ctrl: optional updates that a stakeholder can provide to the shared attributes S,
- Type $\in$ {AMDT, SUMDT, SBMDT}: Each mDT type formalizes a distinct Product DT state projection,
- t_0: creation or snapshot time for the MiniDT.

*a) **Asynchronous MiniDT (AMDT)**:*

$$\text{Obs} = \{y_S(t_0)\}, \quad \text{Ctrl} = \emptyset \tag{5}$$

AMDT provides a static snapshot of selected attributes S from the Product DT at time t_0. It is read-only, allowing stakeholders to view the product state without making any modifications to DT_{prod}.

*b) **Synchronous Unidirectional MiniDT (SUMDT)**:*

$$\text{Obs} = \{y_S(t)\}_{t \geq t_0}, \quad \text{Ctrl} = \emptyset \tag{6}$$

SUMDT provides continuous, real-time observation of selected attributes S from DT_{prod}. Stakeholders can monitor changes as they occur but still cannot update the data in the Product DT. This mode is suitable for live monitoring scenarios where feedback to the product is not required.

*c) **Synchronous Bidirectional MiniDT (SBMDT)**:*

$$\text{Obs} = \{y_S(t)\}_{t \geq t_0}, \quad \text{Ctrl} = V_m \subseteq S \tag{7}$$

SBMDT allows authorized stakeholders to update a subset of attributes V_m in the MiniDT. These updates are propagated directly to the main Product DT, because the MiniDT references the Product DT attributes. Authorization is managed through the OrbitDB access controller, where the stakeholder is granted write permission for the specific attribute addresses.

D. Threat Model

a) Assets and Adversaries: Protected assets include lifecycle data stored in IPFS, symmetric keys held by data owners, OrbitDB access control rules, and audit trails for provenance. Adversaries include external attackers seeking unauthorized access or key interception, and dishonest stakeholders attempting data manipulation or false canonical versions.

b) Attack Vectors and Security Goals: Key threats include key compromise, unauthorized decryption, data tampering, and replay attacks. Security goals are confidentiality via selective disclosure, integrity through signatures and tamper-evident logs, availability via replication, and auditability through provenance records.

c) Assumptions:

- Stakeholders secure their private keys and storage.
- Channels are untrusted; encryption ensures confidentiality and authenticity.
- Only identified stakeholders join IPFS/OrbitDB; though some nodes may be compromised.
- OrbitDB access control is correctly configured and enforced.
- Majority of stakeholders behave honestly.

d) Mitigations: Data is encrypted with symmetric keys, which are encrypted with recipients' public keys and stored in IPFS for sharing. Encrypted payloads protect key locations over untrusted channels. OrbitDB's AccessController restricts writes to authorized nodes, while signatures and content-addressable storage in IPFS/OrbitDB provide tamper evidence and audit trails. Data owners manage keys securely.

V. Evaluation

A. Experimental Setup

We deploy an IPFS cluster with two consumer grade systems to simulate twin replication: both have Intel Core i5, 8GB RAM, 250GB storage; one runs Windows 11, the other Windows 10. The devices are connected over a local area network. Both systems use IPFS 0.35.1 and OrbitDB 0.28. This configuration ensures automatic replication: one node acts as the owner publishing and sharing data, the second as the consumer receiving and replicating data. We use a sample Asset Administration Shell (AAS) file from IDTA [23] as the workload. Since different mDTs can contain one or more submodels with different read/write permissions, the evaluation focuses on submodel-level data sharing rather than individual mini digital twins. Specifically, we measure the

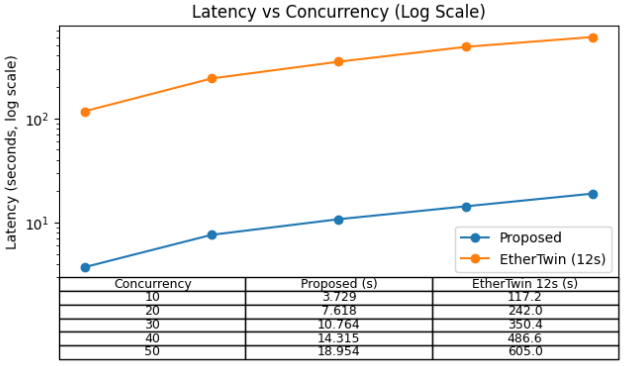
latency of sensor data propagation between peers, integral to SMBDT and SUMDT operation.

B. State of the art comparison

We evaluate our proposed architecture against the state-of-the-art implementation Ethertwin [8]. Ethertwin targets the same problem setting, allowing comparison of end-to-end latency for twin creation and data sharing. Both Ethertwin and our architecture were implemented in the same testbed. Ethertwin uses Ethereum blockchain for immutable record keeping while we rely on IPFS+OrbitDB.

As shown in Figure 2, we simulate concurrent twin creation and data sharing to evaluate end-to-end latency. Our median latency grows from 3.7s at 10 concurrent twin creation and sharing to 19s at 50 concurrent processes, whereas Ethertwin rises from 117s to 605s. This gap is due to Ethertwin using blockchain to record digital twin processes on-chain, involving blockchain transactions with Ethereum block time set at 12 seconds [24].

In our architecture, IPFS+OrbitDB achieves immutability of records, with modifications recorded using OrbitDB's *AccessController*. We use a log-scale y-axis because the latency difference spans multiple orders of magnitude, avoiding compression and allowing fair visual comparison.



Concurrency	Proposed (s)	EtherTwin 12s (s)
10	3.729	117.2
20	7.618	242.0
30	10.764	350.4
40	14.315	486.6
50	18.954	605.0

Fig. 2: End to end latency comparison with State of the art

C. Digital twin sharing

We evaluate concurrent data replication between two nodes in an IPFS cluster. When a digital twin or its submodels are created at one node, they must be replicated to another node to maintain trustworthy and consistent records. This is important in lifecycle scenarios where multiple stakeholders rely on synchronized product twin data, such as manufacturing with parallel twin creation or assembly with multiple submodels added in parallel.

We measured end-to-end replication latency under concurrency levels from 10 to 40 parallel replications. Each replication stores a submodel in IPFS+OrbitDB using Algorithm 1 and propagates it to the other node. For each level, 30 runs were conducted. Figure 3 shows that median latency increases from about 1.1 seconds at 10 concurrent replications to 6.1 seconds at 30, then slightly decreases to 5.5 seconds at 40, with small fluctuations likely due to Wi-Fi saturation and pubsub traffic.

We also evaluated payload sizes (1 MB to 100 MB) in Figure 4, showing a consistent upward latency trend with larger payloads, confirming that fluctuations in concurrency tests come from measurement variability. Both experiments show that decryption and receiver-side operations significantly contribute to overall latency as data volume and concurrency grow.

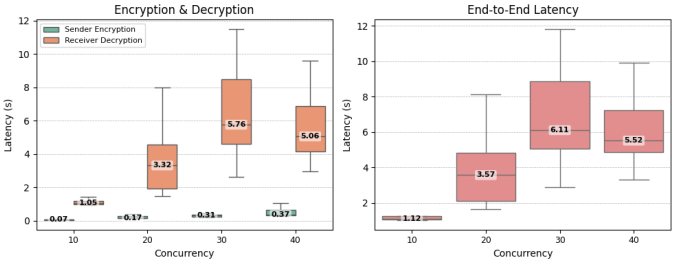


Fig. 3: Concurrent data sharing

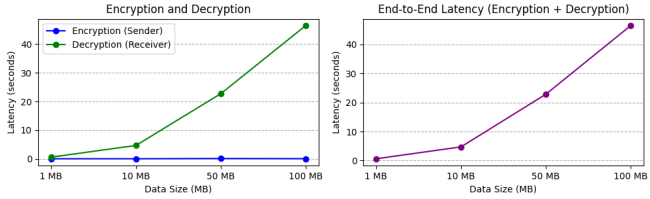


Fig. 4: digital twin sharing with different size

VI. Discussion

Our findings directly address the research question by demonstrating that formal mDT state projections (AMDT/SUMDT/SBMDT) implemented via modular OrbitDB encryption enable selective, low-latency data disclosure to stakeholders across MOL and EOL stages which are core focuses of DPP. The modularized encrypted submodels stored in IPFS+OrbitDB yields mDTs such as AMDT, SUMDT, SBMDT. These mDTs allow fine grained data sharing while preserving data sovereignty. The end to end results show large advantage over blockchain centric design [8] for twin creation and twin data sharing.

In line with Battery passport guidelines [1], sharing a SUMDT to a maintainer provides access to key parameters such as temperature and discharge. For end-of-life (EOL) processes, a recycler can utilize an AMDT to access only the relevant submodels, such as historical maintenance records, accident reports, and faulty component data. This selective disclosure enables recyclers to efficiently analyze the product's lifecycle and make informed decisions about recycling or repurposing, supporting more sustainable EOL outcomes.

This interpretation shows that selective disclosure via AMDT/SUMDT/SBMDT, the necessary lifecycle data from

beginning-of-life to end-of-life can be effectively shared to appropriate stakeholder and helps in the decision making process. The paper assumes that stakeholders are pre-defined and verified, and that request initiation, stakeholder authentication, and agreement management rely on fine-grained access control mechanisms that are beyond the core contribution of this work, integrating such access control is planned as future work.

While twin creation and twin sharing operations performed on blockchain [8] is measured with latency in higher order of magnitude as shown in Fig 2, our proposed architecture achieves the same functionality of immutability with low latency for DT operations. Blockchain may still be necessary for on-chain access control but relying on public blockchains can undermine confidentiality of granted accesses. Zero-knowledge proof mechanisms could attest compliance or prove policy satisfaction on-chain without revealing identities or policies. Future work includes exploring attribute-based encryption and zk proofs for access control validation and on-chain verifiability.

Blockchain-centric DTs incur fees and confirmation delays per DT operation, making frequent blockchain updates expensive and slow. In contrast, our architecture leverages an IPFS cluster, which shifts costs to cryptographic computation for encryption and decryption rather than on-chain transaction fees. By using an IPFS cluster, data is automatically replicated across multiple nodes, ensuring high availability and resilience without incurring additional costs for each update. This approach eliminates per-update blockchain transactions and associated fees, while also providing scalable, decentralized storage. As a result, our system can efficiently support large-scale, real-time digital twin operations at a fraction of the cost of blockchain-based solutions

VII. Conclusion

This paper introduced a decentralized mini digital twin (mDT) architecture for Digital Product Passports that targets multiple lifecycle stages and demonstrating through MOL phase, where continuous access to lifecycle data is critical for sustainability and operational efficiency. This work formally modeled Product DT state evolution as a discrete-time dynamical system, with novelty in heterogeneous mDT views (AMDT/SUMDT/SBMDT) projecting distinct states. On the systems side, the paper describes a hierarchical AAS-based Product DT stored on OrbitDB and IPFS, with per-module encryption and IPFS Cluster replication. Key-based selective disclosure is then used to construct heterogeneous, stakeholder-specific twins while preserving data sovereignty. The evaluation shows that the proposed approach achieves lower twin creation and sharing latency than a state-of-the-art baseline, while keeping encryption, storage, and sharing overhead manageable across varying data sizes. This demonstrates that selective, low-latency lifecycle data sharing is feasible without relying on blockchain-centric designs.

References

[1] "Battery pass project," https://battery-pass.org/.

[2] E. Commission, "Battery regulation (eu) 2023/1542," 2023.

[3] R. Kannappan, J. Hatin, E. Bertin, and N. Crespi, "Enhancing digital product passport through decentralized digital twins," in *12th IFIP International Conference on New Technologies, Mobility and Security (NTMS 2025)*, 2025.

[4] P. K. F. Wan and S. Jiang, "Enabling a dynamic information flow in digital product passports during product use phase: A literature review and proposed framework," *Sustainable Production and Consumption*, vol. 54, pp. 362–374, 2025.

[5] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary perspectives on complex systems: New findings and approaches*. Springer, 2016, pp. 85–113.

[6] M. Torzoni, M. Tezzele, S. Mariani, A. Manzoni, and K. E. Willcox, "A digital twin framework for civil engineering structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 418, p. 116584, 2024.

[7] M. Dietz, B. Putz, and G. Pernul, "A distributed ledger approach to digital twin secure data sharing," in *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer, 2019, pp. 281–300.

[8] B. Putz, M. Dietz, P. Empl, and G. Pernul, "Ethertwin: Blockchain-based secure digital twin information management," *Information Processing & Management*, vol. 58, no. 1, p. 102425, 2021.

[9] H. Haße, H. van der Valk, F. Möller, and B. Otto, "Design principles for shared digital twins in distributed systems," *Business & information systems engineering*, vol. 64, no. 6, pp. 751–772, 2022.

[10] S. Huang, G. Wang, Y. Yan, and X. Fang, "Blockchain-based data management for digital twin of product," *Journal of Manufacturing Systems*, vol. 54, pp. 361–371, 2020.

[11] S. Suhail, S. U. R. Malik, R. Jurdak, R. Hussain, R. Matulevičius, and D. Svetinovic, "Towards situational aware cyber-physical systems: A security-enhancing use case of blockchain-based digital twins," *Computers in Industry*, vol. 141, p. 103699, 2022.

[12] Y. Wang, H. Wang, and Y. Cao, "Comprehensive review of storage optimization techniques in blockchain systems," *Applied Sciences*, vol. 15, no. 1, p. 243, 2024.

[13] H. R. Hasan, K. Salah, R. Jayaraman, M. Omar, I. Yaqoob, S. Pesic, T. Taylor, and D. Boscovic, "A blockchain-based approach for the creation of digital twins," *IEEE Access*, vol. 8, pp. 34 113–34 126, 2020.

[14] J. Monteiro, J. Barata, and S. Gentilini, "A digital twin-based digital product passport," *Procedia Computer Science*, vol. 246, pp. 4123–4132, 2024.

[15] C. Lopes and J. Barata, "Digital product passport: a review and research agenda," *Procedia Computer Science*, vol. 246, pp. 981–990, 2024.

[16] M. Hulea, R. Miron, and V. Muresan, "Digital product passport implementation based on multi-blockchain approach with decentralized identifier provider," *Applied Sciences*, vol. 14, no. 11, p. 4874, 2024.

[17] J. Lutz and C. Reich, "Blockchain based digital passports for industrial digital twins," in *Joint International Conference on AI, Big Data and Blockchain*. Springer, 2025, pp. 39–48.

[18] R. Kannappan, J. Hatin, E. Bertin, and N. Crespi, "Decentralized product lifecycle management using blockchain and digital twins," in *The 45th IEEE International Conference on Distributed Computing Systems (IEEE ICDCS 2025)*, 2025.

[19] W. Li, A. Sforzin, S. Fedorov, and G. O. Karame, "Towards scalable and private industrial blockchains," in *Proceedings of the ACM workshop on blockchain, cryptocurrencies and contracts*, 2017, pp. 9–14.

[20] M. M. Khan, F. S. Khan, M. Nadeem, T. H. Khan, S. Haider, and D. Daas, "Scalability and efficiency analysis of hyperledger fabric and private ethereum in smart contract execution," *Computers*, vol. 14, no. 4, p. 132, 2025.

[21] F. Leal, A. E. Chis, and H. González-Vélez, "Performance evaluation of private ethereum networks," *SN Computer Science*, vol. 1, no. 5, p. 285, 2020.

[22] D. Ivanov, "Conceptual and formal models for design, adaptation, and control of digital twins in supply chain ecosystems," *Omega*, vol. 137, no. C, 2025.

[23] Admin Shell IO, "Samples for asset administration shell," https://admin-shell-io.com/samples/.

[24] Ethereum.org contributors, "Blocks," https://ethereum.org/en/developers/docs/blocks/, Ethereum Foundation, May 2024. [Online]. Available: https://ethereum.org/en/developers/docs/blocks/