BlockTrace

Every product has a story, we tell them all.

1. INTRODUCTION

The European Union has introduced new regulations requiring various classes of imported products to include a **Digital Product Passport (DPP)**, designed to enhance traceability, transparency, and lifecycle monitoring, from creation to recycling. The DPP functions as both an identity and description record for each product, raising the challenge of maintaining **trustworthy, immutable, and publicly accessible data**, while still allowing controlled updates across the supply chain.

To address this, we propose a solution based on **Hyperledger Fabric's permissioned blockchain**, where all data is cryptographically signed and immutably stored. The DPP is an on-chain file whose lifecycle is tracked through transactions that log both previous and updated values. Participants such as manufacturers, transporters, and recyclers have **role-based access**, ensuring they can only update the DPP within their designated scope and time window. This architecture ensures full auditability, data integrity, and accountability across all stages.

2. WHY BLOCKCHAIN

Blockchain functions as a distributed digital ledger that ensures **transparency, traceability, and data integrity** across all participants in a network. In the context of DPPs, it allows for **decentralized storage** of product information, reducing the risk of data tampering or loss.

This architecture enables **public visibility** of product records while maintaining **secure and verifiable update mechanisms**, reinforcing **trust throughout the product lifecycle**. By recording every change as a cryptographically signed transaction, blockchain guarantees **auditability and tamper-proof assurance**, which is critical for regulatory compliance, environmental reporting, and **fiscal accountability** in global trade.

3. WHY HYPERLEDGER FABRIC

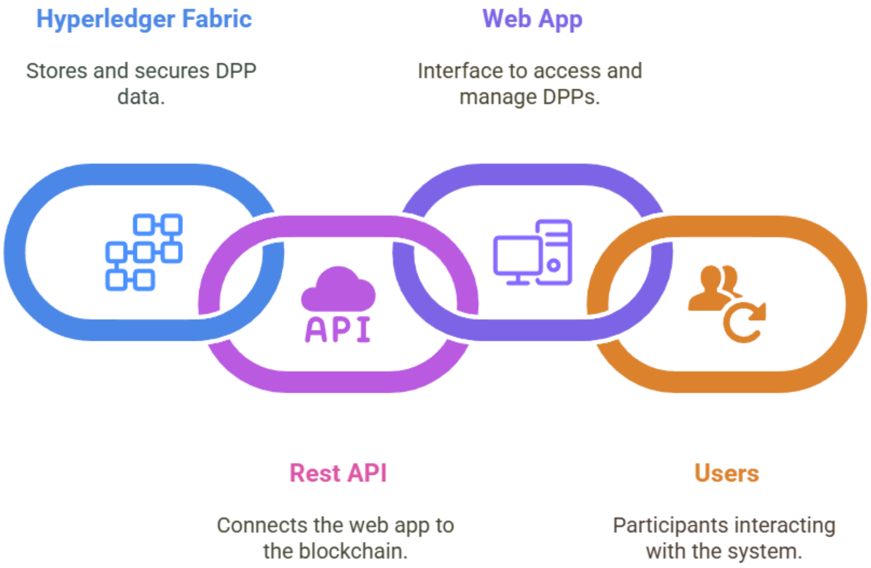
For our Digital Product Passport solution, **Hyperledger Fabric** was selected for its modular, enterprise-grade architecture and strong alignment with the needs of

compliance-driven environments like supply chains. As a **permissioned network**, it ensures that **only authorized participants** can write or update data, while maintaining **open read access** to promote transparency and public oversight.

Key features such as **channel-based communication**, **pluggable consensus**, and **identity management**, combined with a permission model where data is publicly visible but only modifiable by authorized entities, ensure that all transactions are **securely signed, immutably recorded**, and **verifiable by anyone**. These capabilities offer the right balance of **data protection**, **auditability**, and **trust**, making Hyperledger Fabric the ideal foundation for our application.

4. SYSTEM ARCHITECTURE OVERVIEW

The architecture of our solution is composed of four core components that work together to ensure data integrity, user interaction, and secure integration with the blockchain network. The diagram below provides a **high-level view** of how the system layers interact.


Hyperledger Fabric
Stores and secures DPP data.
Web App
Interface to access and manage DPPs.
API
Rest API
Connects the web app to the blockchain.
Users
Participants interacting with the system.

4.1. USERS

Final users of the system include both organizations and individuals who manage their DPPs through the *Web App*. Although their role within the blockchain network will be detailed in the **User Identity** section, they represent the entry point of interaction with the platform.

4.2. WEB APP

The Web App is the primary interface for user interaction. It provides a secure and intuitive environment to **view, register,** and **update** the state of DPPs. While it does not interact directly with the blockchain, it acts as the frontend layer connected to the *REST API*.

4.3. REST API

The REST API serves as the middleware between external applications and the Hyperledger Fabric network. It communicates with the ledger via Fabric's Gateway and is constrained by the chaincode logic deployed on the network. Essentially, it **translates blockchain logic into web-compatible operations**, enabling safe and controlled access to on-chain data.

4.4. HYPERLEDGER FABRIC

Hyperledger Fabric is the **core layer** of the system. As a permissioned blockchain platform, it is responsible for storing all DPP data securely, enforcing access control, and maintaining the integrity of every transaction within the network.

5. ARCHITECTURE COMPONENTS

Since the REST API and the user layer are relatively simple components, they do not require dedicated sections. Relevant details about their roles and interactions are addressed contextually throughout the whitepaper.

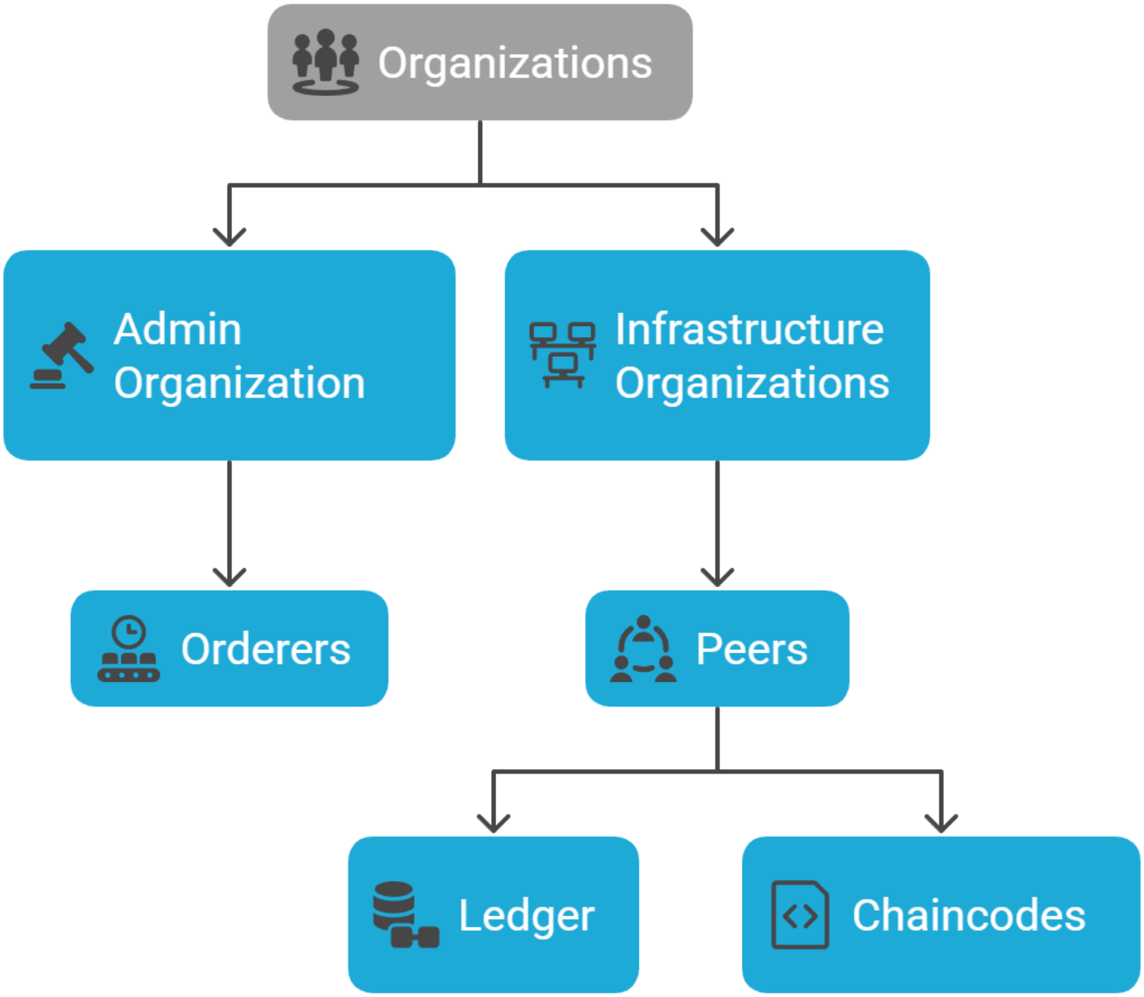
5.1. HYPERLEDGER FABRIC

Hyperledger Fabric serves as the **foundation** of the system, providing a secure and permissioned blockchain infrastructure. It is responsible for **storing** the entire **lifecycle** history of each DPP, including its creation, updates, and the entities responsible for each modification. Every **action is captured as a transaction**, cryptographically signed, and immutably **recorded on the ledger**.

The solution leverages Fabric's **world state** to represent the current state of each DPP, while **maintaining a full transaction log** for auditability. Access to data and write operations is controlled through **chaincode**, which defines the logic and rules governing how and when DPPs can be updated. This ensures that only **authorized participants** can perform specific actions, reinforcing both trust and compliance within the network.

The operational architecture of Hyperledger Fabric, as applied in our solution, is depicted in the diagram below.

Structure of the Hyperledger Fabric Network



5.1.2. Network Governance and Consensus

The governance model of this blockchain network balances decentralization among participants with operational efficiency and security. Each participating company joins the network as a distinct organization and is responsible for operating its own **peer node**. These peers maintain a copy of the ledger, execute smart contracts (chaincode), and endorse transactions related to their specific business roles. This architecture ensures that each organization maintains transparent access and **trust in how its data is represented and processed** on the network, even without direct control.

To manage the blockchain infrastructure and maintain consensus, the network employs a centralized ordering service operated by a dedicated **Administrative Organization**. This organization is exclusively responsible for running the **orderer nodes**, which order valid transactions, package them into blocks, and broadcast them to all peers. These nodes use the **Raft consensus protocol**, offering crash fault tolerance and deterministic block production.

It is recommended that the Administrative Organization be operated by a **governmental entity** or by the **lead company overseeing the network**, as these are typically the most trusted and neutral parties to manage such a critical component.

Beyond ordering, the Administrative Organization plays a key role in governance. It holds the necessary administrative credentials (MSP) and is defined in the network's policies as the sole authority to perform critical operations, such as:

- Creating or modifying channels
- Updating endorsement and access control policies
- Installing or upgrading chaincodes
- Onboarding or removing organizations from the consortium

Importantly, the **Administrative Organization does not operate any peer nodes**, preserving a clear separation of duties between governance and business-level transaction processing. This structure reinforces trust and neutrality in the network's infrastructure.

If needed, the architecture supports the addition of multiple Orderer Organizations to distribute ordering responsibilities. However, this adds governance complexity and is recommended only when it is strictly required.

This approach preserves transparency, compliance, and operational clarity, while reducing the risks.

5.1.3. Peers

Peers are the physical infrastructure components operated by organizations within the network. They are responsible for executing **chaincode logic** and maintaining local copies of the **ledgers** for the channels they are authorized to access.

It is important to distinguish between **organizations that operate peers** and the **entities** that interact with the solution. A real-world company may choose to become an **organization** in the network by hosting a peer node, thereby gaining infrastructure-level participation. However, this **organizational identity** is only used for maintaining the ledger and validating blocks at the network level. Notice that **you do not need to run a peer** or be part of an organization in order to participate in the chain, since some users may not have the structure required to do so.

In contrast, all **application-level transactions** must be signed by **individual entities** (such as users or companies) that hold decentralized identifiers (DIDs). These identities are the ones linked to asset ownership and authorization logic. The distinction ensures a clean separation between infrastructure governance and business-level interaction.

The concept of entities and their identities will be detailed further in the next sections.

5.1.4. Ledger and Chaincodes

The ledger will operate using **CouchDB** as the state database, enabling the storage and querying of data in **JSON format**, which simplifies integration and data manipulation.

Smart contracts are packed in a structure called **chaincode** that runs in the peers of the network. It defines the rules that govern all transactions within the network. These include operations such as:

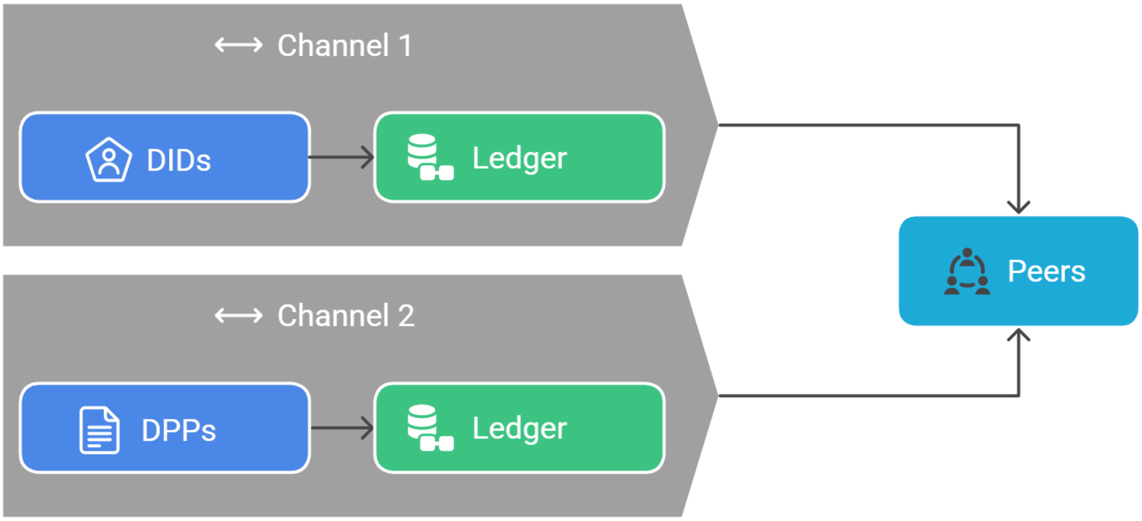
- **Creating and updating DPPs**
- **Transferring ownership**
- **Deleting records**
- **Creating DIDs** - further detailed in the **DID section**

Each DPP-related action is processed through the chaincode logic, which validates whether the invoking user has the appropriate permissions. This ensures strict access control over all stages of the product's lifecycle. A complete breakdown of DPP operations and their structure is presented in the section **"Digital Product Passport - Structure Overview"**.

5.1.5. Channels

The network defines two distinct channels: one dedicated to storing **Digital Product Passport** data, and another focused on **Decentralized Identifiers (DIDs)**. Each peer node is subscribed to both channels and holds a full copy of their respective ledgers. This architectural separation enables a clear distinction between **identity management** and **product traceability**, enhancing modularity and scalability.

By granting chaincodes access to both channels, the system supports more **granular and dynamic access control mechanisms**, enabling smart contracts to validate permissions based on DID-related credentials when processing DPP-related transactions.



5.1.6. Decentralized Identities (DIDs)

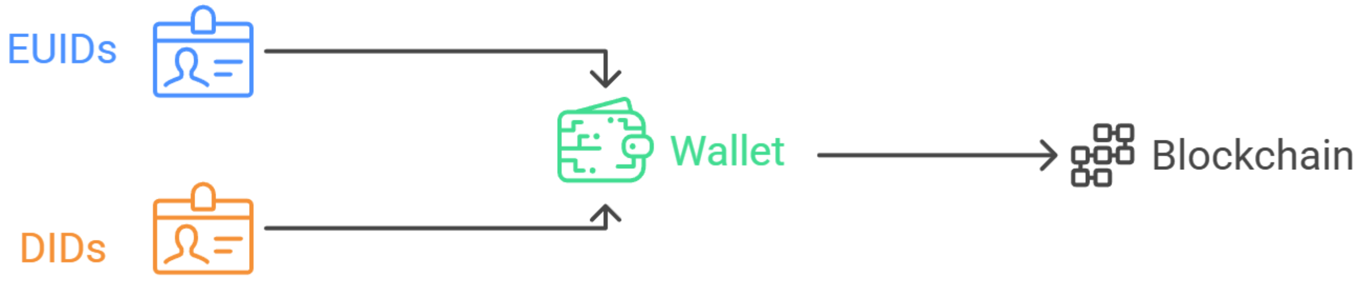
One of the key differentiators of this solution is the integration of **Decentralized Identifiers (DIDs)** with support for **European Unique Identifiers (EUIDs)**. Given that EUIDs are already being adopted under the new EU regulations, it is both strategic and necessary to support them within the identity model.

Wallets that follow the European framework will be able to **list transactions signed by an EUID** and **display files associated with that identity**, enabling a direct link between European citizens and their on-chain assets. However, relying solely on EUIDs would restrict participation from non-European users. To address this, we incorporate **DIDs** as a complementary, **universally accessible identity system**.

DIDs provide a **cryptographically verifiable identity** for each entity — individual or organization — interacting with the system. They serve as persistent identifiers used to **sign transactions and assert ownership**, linking on-chain actions to real-world identities. This is essential for enforcing access controls at the chaincode level, ensuring that only authorized parties can create, update, transfer, or delete Digital Product Passports (DPPs).

DIDs are created within the application environment using **Hyperledger Aries**, a framework designed to manage identities in a decentralized and privacy-preserving manner. Every participant in the network is required to have a DID to perform operations. Each DID is associated with a **Membership Service Provider (MSP)** and a **Certificate Authority (CA)**, which together define the trust model and permission structure for that identity within the blockchain.

In practice, users will **connect their wallets to the Web App** in order to sign transactions. While this solution currently focuses on the web interface, it's worth noting that the same identity mechanisms could support the future development of a **mobile wallet**, enabling secure identity management and transaction signing on portable devices.



5.1.7. Entities

Entities are the user, whether individuals or organization, that interact directly with the application. They play various roles throughout the product's lifecycle and are assigned specific permissions and responsibilities based on those roles. For example, a logistics company acting as a transporter should not be allowed to modify structural product data but must be able to update location-related information during shipment. To ensure that actions are secure, verifiable, and properly scoped, every entity is linked to a unique **DID**. These identifiers, governed by the network's **MSP** and **CA**, are cryptographically tied to every transaction, making it possible to enforce role-based access control and maintain full auditability. The precise set of entity roles supported by the system is outlined below.

The system defines several entity types, each with a distinct role in the product lifecycle. These roles determine what each entity is authorized to do via the chaincode logic:

- **OEM (Original Equipment Manufacturer):** Initiates the DPP by recording core manufacturing data, material composition, and production certifications.

- **Distributor / Logistics Provider:** Responsible for documenting product movement, ownership transfers, and transportation checkpoints.

- **Maintenance Provider:** Updates the DPP with service events, repairs, part replacements, and technical inspections throughout the product's use phase.

- **End User:** Accesses the DPP and adds information about the product's usage. Can also update information regarding product structure. *(A discussion addressing problems that may occur in this phase will be made further on.)*

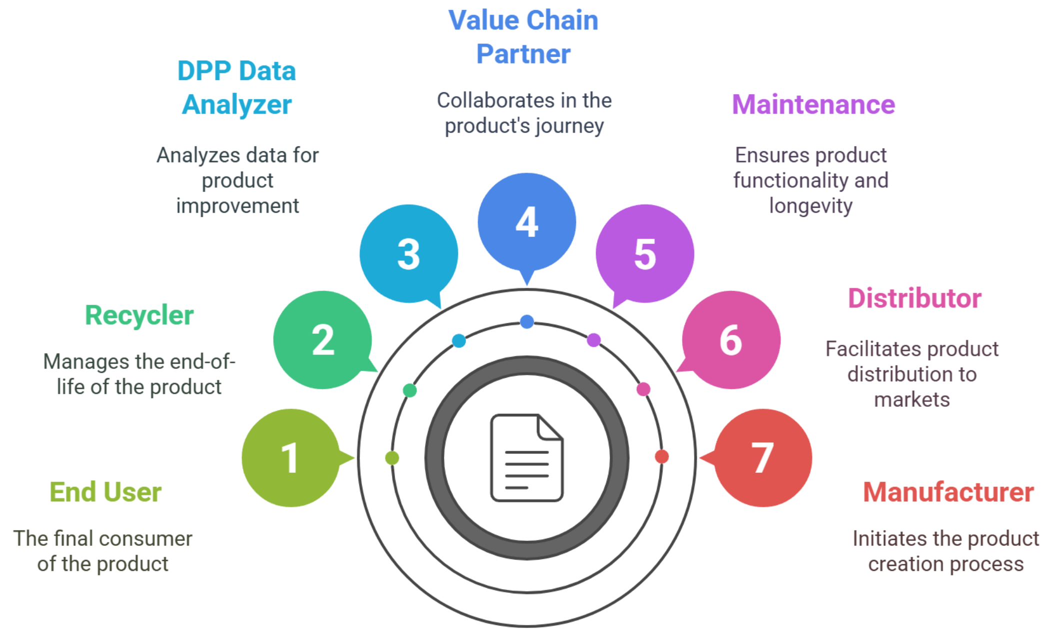
- **Recycler / Reuse Entity:** Registers end-of-life data, including recycling actions, reusability assessments, and final material recovery.

- **Value Chain Partner:** Contributes verified data within their domain (e.g., sustainability certifications or material sourcing validations).

- **Data Analyzer:** Reads DPP records to generate aggregated reports and insights related to circularity, environmental impact, and compliance performance.

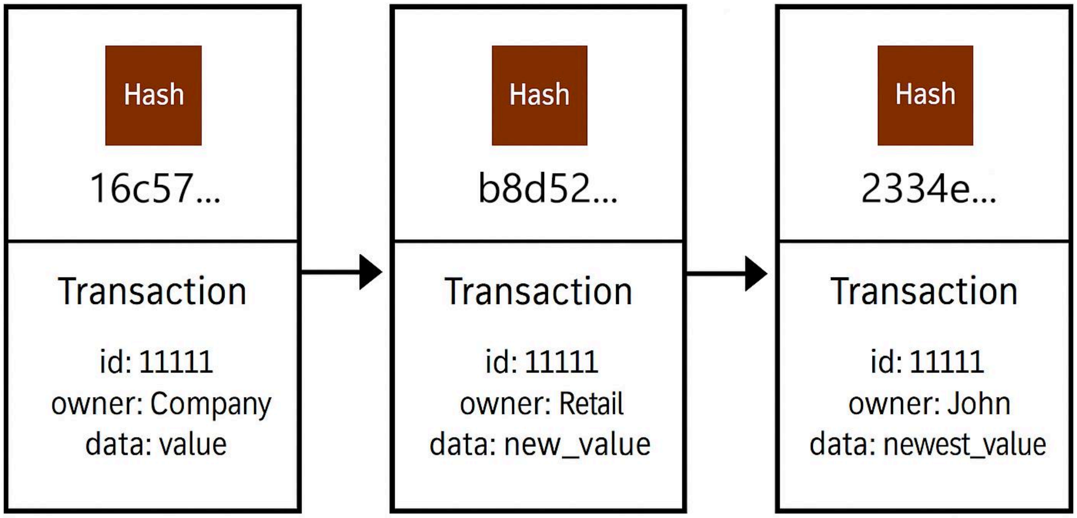
As a practical example, consider a **distributor entity**: it must possess a valid DID categorized as *Distributor*, and the target DPP must be in the *distribution* phase, with the `distributionEncharged` field explicitly referencing that DID. Only when these conditions are met can the entity submit a **valid transaction** to update the DPP. It is important to note that this does not imply a change in ownership, rather, it is the act of **appending new information** to the product's history, such as location, custody, or state transitions. In this model, entities do **not need to be the asset owner** to interact with a DPP; they only need to possess the correct **role-based credentials**, which are validated at runtime by the smart contract.

Product Lifecycle Ecosystem


Value Chain Partner
Collaborates in the product's journey
DPP Data Analyzer
Analyzes data for product improvement
Maintenance
Ensures product functionality and longevity
Recycler
Manages the end-of-life of the product
Distributor
Facilitates product distribution to markets
End User
The final consumer of the product
Manufacturer
Initiates the product creation process
1
2
3
4
5
6
7

5.1.8. Transactions

As previously discussed, transactions can occur for various purposes, such as creating a new DPP, updating product status, or recording lifecycle events. Each transaction must be **endorsed by peers** and **signed by the responsible entity** to be considered valid. Entities play a critical role in this process, as their **DIDs** and associated credentials determine the actions they are permitted to perform under the chaincode logic.

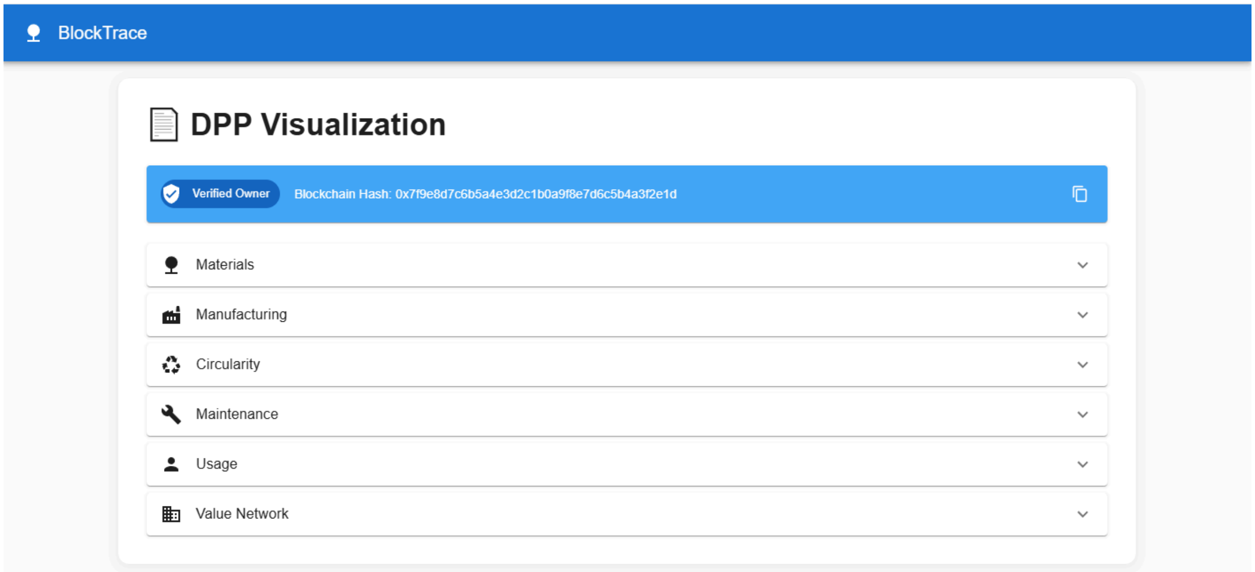


5.2. WEB APP

The **web application** provides a user-friendly interface that simplifies interaction with the blockchain-based system, aiming to facilitate stakeholder adoption of the technology. It abstracts the underlying complexity of decentralized infrastructure, offering intuitive tools for managing Digital Product Passports (DPPs).

Through the web app, users can **scan a QR code** attached to a product, which redirects them to a dedicated page containing that product's DPP. There, stakeholders can **view, register, or update** DPP records.

Additionally, the platform includes a **personal dashboard** where users can browse and manage all DPPs associated with their identity. To enhance user experience and engagement, the application can support the optional integration of **AI models trained on product data**, enabling clients to ask natural language questions and receive contextualized answers about specific DPPs, improving transparency and support.

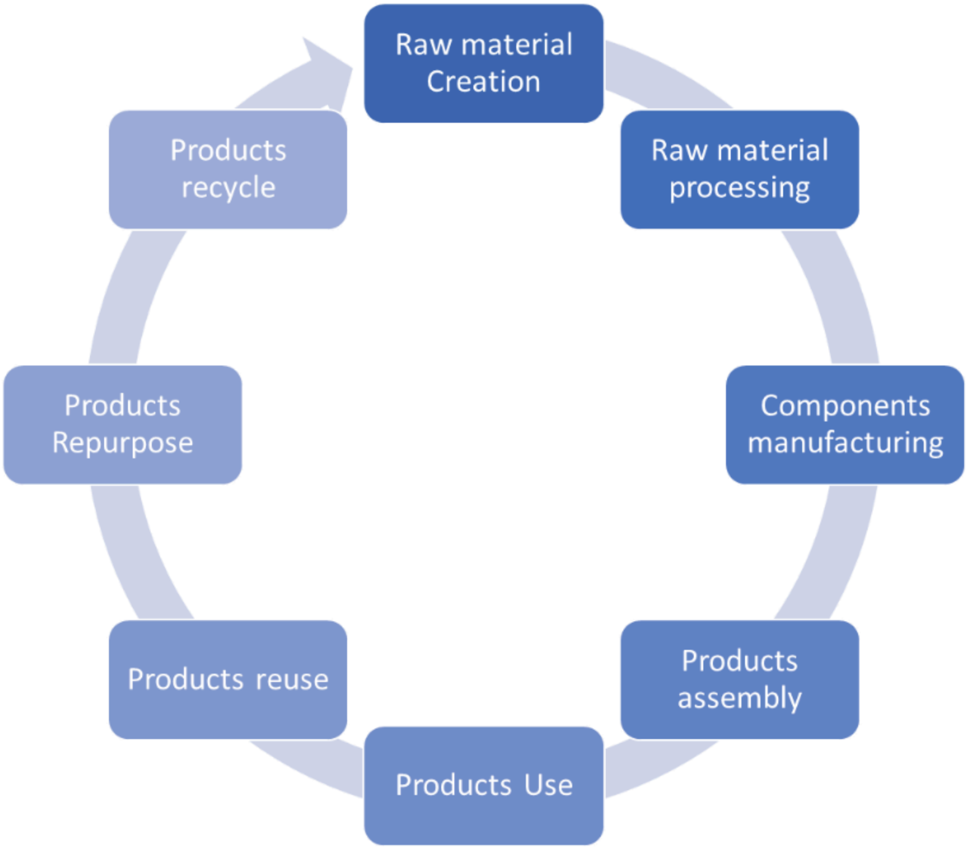


DPP Visualization

Verified Owner Blockchain Hash: 0x7f9e8d7c6b5a4e3d2c1b0a9f8e7d6c5b4a3f2e1d

6. Digital Product Passport - Structure Overview

The **Digital Product Passport** serves as a comprehensive record of a product’s entire lifecycle. To ensure that it fulfills its purpose in supporting the transition to a **circular economy**, and drawing upon established research in lifecycle analysis, we have defined a structured data model that reflects each key stage of the product journey.

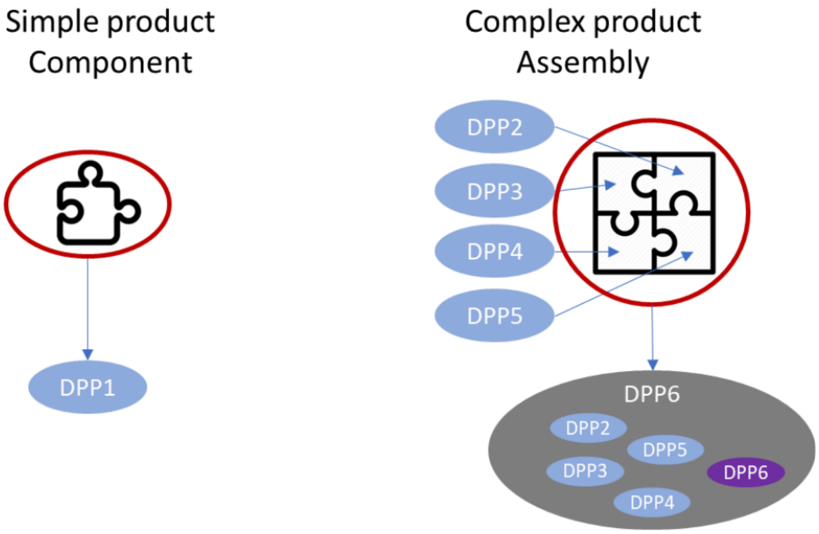


DPP framework across lifecycle stages. *Adapted from Psarommatis and May (2023).*

Digital Product Passports (DPPs) can be classified into two categories:

- **Single-component products**, represented by a standalone DPP
- **Multi-component products**, composed of multiple DPPs linked together

While single-component DPPs represent simple items, most real-world products fall under the second category. In these cases, the main DPP maintains **referential links** to other DPPs that represent its subcomponents, forming a **compositional hierarchy**. This mechanism enables accurate traceability across complex assemblies.



DPP's composition possibilities. *Adapted from Psarommatis and May (2023).*

It's important to note that **this composability refers only to structural linkage**, not to the internal data model of a DPP. Internally, each DPP follows a **standardized schema** aligned with the product's lifecycle and supported by a role-based access model.

Each DPP stores:

- **Identification data**: Unique ID, serial number, owner hash, etc.

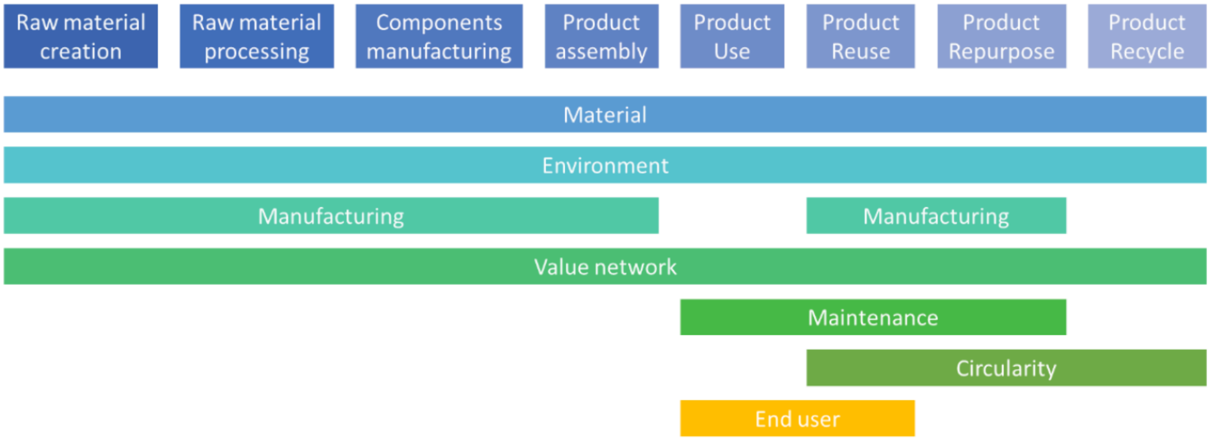
- **Lifecycle records**, aligned with specific **entity roles**, including:
    - **Materials Data** – Raw materials and components
    - **Environmental Data** – Emissions and sustainability metrics
    - **Manufacturing Data** – Information generated during production
    - **Value Network Data** – Logistics and distribution details
    - **Circularity Data** – Reuse, refurbishing, or recycling status
    - **End-User Data** – Information generated during product usage
    - **Maintenance Data** – Service, repairs, and upkeep logs

The **entity model** was designed specifically to match this data architecture. Each entity type (e.g., manufacturer, transporter, recycler) holds permissions to interact with lifecycle stages relevant to their role. These permissions are **defined and controlled by the DPP owner**, ensuring that data contributions are validated and tamper-proof throughout the product's entire life.

6.1. Product Lifecycle

The **lifecycle of a product** involves multiple transitions between actors, each with different permissions to view or update its Digital Product Passport (DPP). As illustrated in the lifecycle model adapted from *Psarommatis and May (2023)* [1], these permission stages evolve as the product moves through manufacturing, distribution, usage, and end-of-life. However, unlike the static permission layers proposed in their framework, we argue that **permission configurations are highly context-dependent**. They should be dynamically defined based on the **specific supply chain architecture** and **governance model** of the implementing party.

Therefore, **further research and development** are needed to support **adaptive permission frameworks**, ensuring flexibility and interoperability across diverse industry scenarios.



DPP’s lifecycle example. *Adapted from Psarommatis and May (2023).*

6.2. Data model example

To ensure consistency, auditability, and lifecycle tracking across all assets, each DPP follows a standardized JSON-based data model. This schema is optimized for CouchDB, enabling efficient querying and integration with Hyperledger Fabric’s world state.

Each DPP contains immutable identifiers and structured sections that represent the product's full lifecycle. It’s important to notice that some information can be encrypted if needed, so private and sensitive data are not leaked. Below is a simplified outline of the core data fields:

```
{
  "dpp_id": "uuid-1234",
  "serial_number": "SN-456789",
  "owner_did": "did:example:abcd1234",
  "status": "in_use",
  "components": ["dpp_id_sub1", "dpp_id_sub2"],
  "permissions": {
    "manufacturer": ["write:manufacturing_data"],
    "transporter": ["write:logistics_data"],
    "recycler": ["write:recycling_data"]
  },
  "lifecycle": {
    "materials_data": { ... },
    "manufacturing_data": { ... },
    "logistics_data": { ... },
    "usage_data": { ... },
    "maintenance_data": { ... },
    "recycling_data": { ... }
  },
  "audit_log": [
    {
      "timestamp": "2025-06-01T15:22:00Z",
      "action": "update:logistics_data",
      "signed_by": "did:example:transporter567"
    }
  ]
}
```

This is an **example** of how data can be stored, but **not necessarily a final version** of the DPP.

7. Security, Identification and Privacy

Individuals and companies can be registered on the network through one of two mechanisms: **EUIDs (European Unique Identifiers)** or **DIDs (Decentralized Identifiers)**. These identifiers serve as digital fingerprints, enabling the blockchain to validate and trace actions across the product lifecycle.

- **EUIDs** are created according to each country's eID regulations, often following official standards such as the EU Digital Identity Wallet framework.

- **DIDs** are generated directly through the Web Application and anchored on-chain using tools like **Hyperledger Aries**, ensuring compliance with W3C DID specifications. Once generated, a DID can be used to sign transactions, prove identity, and delegate roles securely.

To assign responsibility in the product lifecycle, the system can associate a DPP permission field with the public key of a valid DID or EUID. For example, a transport phase may require adding a transporter's public DID to the `distribution.permissioned_to` field of a DPP.

Additionally, national authentication systems (e.g., Brazil's **Gov.br**) could be integrated to issue enriched DIDs. These DIDs may contain encrypted metadata (such as CPF, full name, issuing authority) to support more descriptive and trusted identity assertions. While this feature is not part of the current implementation, the system's architecture allows future integration with national and international identity schemes.

The following identity record format **standardizes how DIDs, EUIDs, and other identifiers** can be registered and referenced within the blockchain system.

```
{
  "id": "did:web:blocktrace.io:users:0xA1B2C3",
  "type": "DID",            // or "EUID", "GovID", etc.
  "publicKey": "0x032f...acb9",
  "issuer": "Org1MSP",
  "roles": ["manufacturer", "transporter"],
  "metadata": {
    "country": "Belgium",
    "linkedEUID": "eu-wallet:BE:0999999999",
    "authMethod": "wallet",   // or "oauth", "gov.br", etc.
    "validFrom": "2025-04-01T12:00:00Z"
  }
}
```

8. Performance and Scalability

While **performance benchmarks are yet to be defined**, the **system was designed with scalability** as a core requirement. Hyperledger Fabric's modular and permissioned architecture enables the network to **scale horizontally by adding peers and orderers as needed**. Additionally, separating business logic into isolated channels and using CouchDB for flexible JSON-based data storage contributes to maintaining efficiency as the volume of transactions and participants grows.

9. Use cases

9.1. Regulatory Compliance

- Aligned with **Ecodesign for Sustainable Products Regulation (ESPR)** and EU Green Deal directives.
- Facilitates **regulatory audits** through transparent and verifiable records.
- Integrates with **EUID/eIDAS** and supports national digital identity systems.

9.2. Supply Chain Visibility

- Enables **immutable traceability** from production to end-of-life.
- Authorized participants can issue updates, cryptographically signed and timestamped.
- Ensures **data integrity** via digital signatures and cryptographic hashes.

9.3. Ownership Verification & Custody Transfer

- Leverages **DIDs and digital signatures** to authenticate and register custody transfers.
- Supports multi-role interactions (OEMs, logistics, recyclers) with **fine-grained access control**.

9.4. Aftermarket Services

- Empowers **post-sale services** like warranty, repair, and resale using verifiable product history.
- Enhances trust in second-hand markets and service records.

9.5. Circular Economy Enablement

- Tracks **reuse, reconditioning, and recycling** actions across the value chain.
- Enables reintegration of components with verified provenance and lifecycle transparency.

10. Benefits

10.1. Interoperability with Global ID Systems

- Compatible with **EUIDs** and adaptable to other digital identity frameworks.
- Supports non-EU users through **flexible DID architecture**.

10.2. Full Auditability & Immutable Logging

- Each transaction records the **changed field, previous value, and new value**.
- Enables full **historical reconstruction** and integrity validation.

10.3. Consumer Trust & Transparency

- **QR Code access** allows public verification of product data at any stage, without intermediaries.
- Builds credibility and transparency in B2B and B2C environments.

10.4. Modular & Scalable Design

- Supports both **single and multi-component products** through a structured, layered DPP format.

- Segments include: Material, Manufacturing, Environmental, Circularity, Maintenance, and more.

10.5. Regulatory Cost Reduction

- Automates compliance and audits via **smart contracts**, reducing manual work and third-party dependencies.
- With Hyperledger, transactions have **no cost per update**, unlike public blockchains that require gas fees. This enables **frequent, granular updates** without financial overhead.

11. Limitations & Open Challenges

Beyond the limitations already discussed in earlier sections, some components of this solution still require further research and refinement. While the architecture addresses key regulatory and operational needs, certain challenges remain open for future iterations or complementary systems:

11.1. Trust at Receipt and Responsibility Transfer

Once a **DPP ownership is transferred**, any **illegitimate edits or misrepresentations prior to that point become the responsibility of the new holder**. This introduces a trust dependency: entities receiving a product must either implement additional verification mechanisms upon receipt or place significant trust in their providers. While critical, this issue falls outside the scope of this system's responsibility.

11.2. Representing Raw Materials

Modeling raw material provenance and traceability poses substantial complexity due to the **volume, variability, and granularity of inputs**. Capturing such data might require a new class of DPPs dedicated to raw components and a third dedicated channel in the network. The feasibility and modeling of this structure must be further explored depending on the industry context.

12. Conclusion

The BlockTrace demonstrates how blockchain technology can be leveraged to enhance the **traceability and transparency** of complex supply chains. By implementing a decentralized system based on **Hyperledger Fabric**, the project provides a secure and **immutable environment for tracking products** and their components across multiple stakeholders.

Unlike traditional supply chain systems, which rely heavily on centralized databases and manual coordination, BlockTrace enables a **collaborative network** where each participant maintains their node and contributes to a shared, trustworthy ledger. This not only improves data accuracy and accessibility but also reduces the risk of fraud, delays, and inconsistencies.

The use of smart contracts further automates key processes, ensuring that transactions are executed according to predefined rules and that **all parties can verify compliance.** The platform's modular architecture, with defined roles for each organization, makes it scalable and adaptable to various industries.

By combining blockchain's core principles with modern supply chain requirements, BlockTrace offers an innovative alternative to conventional tracking systems. It lays the groundwork for a new generation of supply chain solutions that **prioritize trust, efficiency, and real-time visibility**.

13. References

- https://8112310.fs1.hubspotusercontent-na1.net/hubfs/8112310/Hyperledger/Offers/TYSWhitepaper2022-6.20.22.pdf
- https://www.mdpi.com/2076-3417
- https://ieeexplore.ieee.org/document/9387709
- https://keepelectronics.com/#/product/PF0H268N
- https://www.tandfonline.com/doi/full/10.1080/00207543.2018.1533261
- https://www.sciencedirect.com/science/article/abs/pii/S0007681318301472?via%3Dihub
- https://www.sciencedirect.com/science/article/pii/S2212827125000368?ref=pdf_download&fr=RR-2&rr=94921ace48603652
- https://hyperledger-fabric.readthedocs.io/pt/latest/whatis.html
- https://eu-digital-identity-wallet.github.io/eudi-doc-architecture-and-reference-framework/1.6.0/

14. Authors

Arthur de Lara Machado

Graduando em Ciência da Computação (UFSC), desenvolvedor de software e empreendedor. Atua na criação de soluções tecnológicas inovadoras, com foco em resolver problemas reais de forma prática e eficiente.

Email: arthurlaramachado@gmail.com

Felipe Fagundes Pacheco

Graduando em Ciência da Computação (UFSC). Pesquisador em redes no laboratório LRG, focado em aplicar conhecimentos técnicos a soluções práticas. Comprometido e curioso por inovação tecnológica.

Email: lipe.fagundespacheco@gmail.com

João Victor Cabral Machado

Graduando Ciência da Computação (UFSC), em transição para Sistemas de Informação. Trabalho em uma empresa que oferece serviços para o ecossistema Web3, e nas horas vagas me dedico ao estudo de game design, desenvolvimento de jogos e soluções envolvendo inteligência artificial.

Email: joao.cabraltt@hotmail.com

Leonardo Dias Guterres

Graduando em Sistemas de Informação (UFSC), atuando atualmente com desenvolvimento backend e em constante evolução para atuar como desenvolvedor fullstack.

Email: leogttrrs@gmail.com