Chain2Sustain

A solution for sustainable supply chains via blockchain

Distributed Ledger Technology for Private Sector Innovation
Cooperation of Technical University of Munich and fortiss

Contents

Chain2Sustain
A solution for sustainable supply chains via Blockchain

Mehmet Eren Turanboy
mehmet.turanboy@gmail.com

David Gasser
ge52noz@tum.de

Felix Myhsok
felix.myhsok@tum.de

July 22, 2025

Abstract

This paper presents a sustainable supply chain system that leverages blockchain technology for enhanced traceability and transparency. This project was developed in response to the increasing global demand for sustainable supply chain networks, such as in the elecetrical car battary manufacturing industry. The proposed system allows supply chain stakeholders to manage and monitor environmental emissions associated with product manufacturing. Utilizing Fablo, a blockchain configuration tool for Hyperledger Fabric, the system manages private data collections, ensuring that sensitive information is only accessible to authorized parties. The system's core chaincode is categorized based on its application area – emission audits and asset transfers. Auditors play a critical role by invoking chaincodes to assign organizational roles and create manufacturing process recipes. The system supports asset creation and transfer, beginning with mines that serve as initial asset creators. Users utilize these assets to manufacture new products bundled together as needed. The final product, viewable by consumers, provides detailed emission figures related to its production. The paper acknowledges the challenge of maintaining decentralization, user confidentiality, and data traceability simultaneously and offers solutions to these issues in the context of the blockchain system. While the study builds upon previous research on blockchain applications in various industries, it further addresses the challenge of balancing decentralization, user confidentiality, and data traceability in sustainable supply chains.

1 Introduction

In an era of unprecedented environmental challenges and growing concerns over their long-term effect on human life on earth, sustainability has become a principal priority worldwide. Recognizing the urgent need for transformative action, a growing number of governments, organizations, and individuals strive towards transforming today's world into a more sustainable version of itself as fast as possible [1].

As a result, there has been a discernible shift in the market. With individual consumer awareness and consciousness on the rise, the demand for products that are manufactured and distributed in a sustainable manner is growing rapidly [2]. Furthermore, governments worldwide are enacting laws and regulations to impose specific requirements on companies as well as individuals, compelling them to operate in a more environmentally responsible and socially conscious manner [3]. Aside from this extrinsic motivation, businesses also hold intrinsic motivation to integrate sustainability into their operations as they can effectively manage possible risks to their undertaking, drive cost savings and efficiency gains, access new markets, and strengthen their public opinion [4][5].

The growth trend of sustainable products exemplifies a transition towards more responsible and conscious consumption patterns, wherein consumers are increasingly seeking products that align with their values and are not hesitant to expend more for a sustainably than for a comparable unsustainably manufactured product [2]. Thus, companies, more specifically the Original Equipment Manufacturers (OEMs), receive a solid incentive to pivot towards sustainability. However, a course adjustment according to their more sustainable ambitions requires the OEMs to often not only modify the workings of their own company but also demand similar adjustments from companies they source from. Generally, for a product to be sustainable, it must have a reduced environmental footprint throughout its life cycle, from sourcing raw materials to disposal or recycling. Additional deciding factors are renewable resources, energy efficiency, waste reduction, and ethical production practices [6]. In consequence, both the OEM and its supply chain must be examined to determine if a product is truly sustainable.

The recent shift in consumer behavior gives rise to the need for systems that determine how sustainable a product is. This can be done in the form of verification or quantification. Multiple institutions address this need and provide certifications according to their self-defined or legislative standards. However, the application area of these certifications remains specialized on certain product aspects or parts and, as of now, is furthest developed

and integrated with the food industry sector [7]. Furthermore, certifications sometimes lack credibility as the self-defined standards lack transparency or are generally unknown to the customer [8].
Consequently, there is a need for a system capable of encompassing and monitoring the entire product supply chain while operating as transparently and reliably as possible. Our approach utilizes a blockchain-based solution that guarantees sustainability by structural and functional soundness. To achieve this, the system must model the actions performed along the supply chain, monitor the network and check incoming data for irregularities, as well as automate vulnerable manual operations. Moreover, the solution must ensure traceability for audit. The correct balance of traceability and data privacy proved to be an essential factor in the design of the system. Lastly, our approach focuses on an exemplary application in the automotive industry.

The structure of the paper is as follows. To design the system, we first analyze sustainability standards and study the research conducted in this field. Furthermore, we discuss our Methodology for the development of the system. Thereafter, we assess the requirements for such a system utilizing a Stakeholder Diagram and a DLT decision framework [9]. Based on this analysis, a system design is developed and explained in Section 5. We further explain the implementation of the system's prototype in depth. Lastly, the system, the results, possible limitations, and further research are discussed.

2 Standards & Related Research

Understanding reporting frameworks and defining scopes is crucial when designing systems for tracking and tracing a company's sustainability and its products. This section introduces the most widely recognized and significant emissions standards applicable to supply chain tracing use cases. The following section presents relevant research conducted in supply chain tracking utilizing blockchain technology. Additionally, real-world deployments of blockchain technology for supply chain management are highlighted as illustrative examples.

2.1 Emission Standards

Sustainability is a somewhat ambiguous term. To define it in the corporate world and make it measurable Environmen- tal, Social and Governance (ESG) reporting is commonly used [10]. Thereby, ESG reporting goes beyond traditional financial reporting and recognizes that companies have broader responsibilities. The main factors are climate change, social inequality, human rights, and ethical business conduct. While the environmental dimension of ESG reporting focuses on an organization's impact on the natural environment, including areas such as energy consumption, Greenhouse Gas (GHG) emissions, waste management, and resource conservation, the social dimension encompasses factors such as labor practices, employee welfare, community engagement, and diversity and inclusion. The governance dimension addresses corporate governance structures, board composition, risk management, and adherence to ethical standards.

Stakeholders, including investors, customers, employees, and regulators are increasingly demanding transparency and accountability from organizations regarding their sustainability practices. Thus, a common standard for ESG reporting needs to be defined. As of today, there exists a range of well-defined approaches. However, there is no general agreement on one specific solution.

In this project, we are focusing on the reporting of environmental sustainability. More precisely on the accounting of GHG emissions produced along the supply chain of a car manufacturing process.

GRI 305 One of the standardized emissions reporting frameworks is called GRI 305 and was defined by the Global Reporting Initiative in 2016 [11]. When complying with GRI 305, companies are required to report their management style about their emissions as well as any potential offsets and reduction techniques like for example emissions certificate trading. Emissions need to be separated into listings of **direct GHG Emissions (Scope 1)** which are emitted during the manufacturing process and used materials, **indirect GHG Emissions (Scope 2)** resulting from the used energy and **other indirect GHG Emissions (Scope 3)** which summarized all other emissions produced by the company and their employees which are not defined in previously mentioned scopes. Examples for scope 3 emissions are business travels or the emissions produced by a company's office space. Even though GRI 305 requires individual statements about the amount of emitted gases like $\mathrm{CO_2, CH_4, N_2O, FC, PFC, SF_6, NF_3}$, it foresees that all emissions are calculated into $\mathrm{CO_2}$ equivalent to make them comparable.

Greenhouse Gas Protocol The Greenhouse Gas Protocol [12] is another standard for carbon accounting that was developed by the World Resource Institute and the World Business Council for Sustainable Development. Besides calculation guidance for corporations, it also provides standardized approaches for cities and communities, mitigation projects, and the transfer of emissions along the value chain. Especially the latter one provides important aspects for the sustainable supply chain project. Similar to the GRI 305 standardization, the GHG Protocol defines 3 different scopes of emissions. However, for supply chain accounting, only scope 1 and scope 2 [13] emissions are mandatory. This is due to the difficulty in accounting for scope 3 emissions, especially across multiple companies in a supply chain [14].

The GHG protocol defines two methods for calculating emissions. The **primary data** approach uses individual data from the company, e.g. energy bills, smart meters installed in the manufacturing plant, or other directly measurable characteristics. Another approach is to use **secondary data** which consists of average emissions values across the industry.

According to their information, the GHG Protocol is one of the most popular and adopted frameworks for emissions accounting, with an adoption rate of 9 out of 10 Fortune 500 companies [15].

2.2 Related Research on sustainable supply chains based on blockchain

The distributed ledger technology, characterized by its primary features of immutability, decentralization, distribution, and security, provides a robust foundation for modeling supply chains that frequently extend globally without a central trusted or managing entity. This technology offers significant advantages in terms of transparency, traceability, and trust, addressing key challenges modern supply chains face.

Extensive research has been conducted in this field, exploring diverse industry sectors and examining various aspects within the realm of supply chain management. A few examples are shown in the following.

Liu et al. [16] adopted a general approach to ESG reporting using blockchain technology. The researchers developed a system comprising three main components: a Fact Telling Gateway, enabling access to the blockchain network and verification of data from IoT sensors; Versioning Smart Contracts, facilitating standardized calculations and reporting; and a Blockchain-Based Issuing Mechanism, automating the generation of ESG reports.

In their study, Agrawal et al. [17] examined the textile industry and developed a blockchain-based framework for tracing cotton along the supply chain. Their design involved assigning a separate ledger for each product line, which was shared among all suppliers involved in the production of the final product. While the researchers acknowledged the presence of varying levels of confidential data, they did not provide a comprehensive explanation of the mechanisms employed to achieve data confidentiality.

While Kshetri et al. [18] investigated strategies for addressing ethical and responsible sourcing in the mineral and metal industry, Ma et al. [19] concentrated on the financial transactions within a supply chain by employing a Hyperledger Fabric Network and leveraging its private data collections to share transaction details exclusively among authorized parties involved in the transactions.

Similar to the use case scenario in this project, Lu et al. [20] and Wessel et al. [21] are focusing work on the automotive industry. The former proposes a system with a single shared ledger between all endorsing peers that uses cryptographic encryption to ensure the confidentiality of specific attributes. In their detailed description of their Hyperledger Fabric Network, they further describe that each organization in the supply chain requires its customized chaincode to meet the different functional requirements. Wessel et al. [21] took a different approach to modeling dependencies and responsibilities along the supply chain by introducing ontologies into a system for battery cell manufacturing. In their design, they are assuming the existence of various intelligent appliances in the factory for data acquisition. However, they also allow for manual data input. The data is then combined with an ontology model to create traceable objects stored on the public ledger. With that design, they can uniquely trace the batteries' components and materials through the entire supply chain. Confidentiality of data was not part of their work. In a following study, Wessel et al. further showed how real-world materials could be identified and traced in a manufacturing process [22].

Kim et al. [23] also developed an ontology-based approach for achieving traceability in the supply chain, although their use case example was more general in nature.

Researchers have also investigated methods aimed explicitly at tracking GHG emissions within the supply chain. For instance, Seidenfad et al. [24] conducted a study on carbon accounting for coffee, comparing various consortium architectures. They implemented a Hyperledger Fabric network, employing a single ledger as a persistent record layer.

Similarly, Chen et al. [25] focused on emissions accounting and proposed a design that enables the transfer of emissions both upstream and downstream along the supply chain. Their approach included a fine-grained differentiation of emissions by categorizing them into different scopes (refer to Section 2.1).

In conclusion, the general research in supply chain modeling utilizing blockchain-based technology often revolves around the framework Hyperledger Fabric. The designs frequently exhibit similarities in terms of channel usage, with a single channel serving an entire product supply chain. While product traceability is common in all reviewed architectures, privacy and confidentiality are less emphasized, with only a few authors incorporating functions that ensure confidentiality. Additionally, implementation guidelines are consistently absent for realizing the presented high-level architectures.

2.3 Real World Examples

In addition to the aforementioned research, several real-world deployments of distributed ledger technology in supply chain contexts have been observed. The following examples highlight some of these deployments.

IBM Food Trust, one of the prominent supply chain and blockchain applications, has garnered significant attention [26]. Leveraging Hyperledger Fabric as its underlying technology, IBM Food Trust offers services for efficient digital supply chain management. This includes the capability to track sustainability by recording and referencing certificates issued by trusted third-party authorities such as RCS Global [27]. However, specific implementation details and architectural design of the IBM Food Trust system remain undisclosed as of the time of this report.

Another example of sustainability, more precise responsible sourcing of rare earth like Cobalt is the car manufacturer Volvo [28]. They claim to use a "responsible sourcing blockchain network" to trace minerals used for battery production from the mine all the way into the final product.

An important observation is that all the systems currently deployed in the real world rely solely on 3rd party auditors like RCS Global [27] to verify companies, their materials, and compliance with ESG requirements. Subsequently, these auditors provide the companies with certificates that are often valid for extended periods. These certificates are also used in the supply chain network to certify that the required standards are met.

3 Methodology

Figure 1: Illustration of the methodology

For such a time-intensive group project, our team agreed that a well-structured methodology is critical for the project's success. Therefore, we decided to follow the design methodology illustrated in Figure 1. Generally, this approach is split into two: understanding the task environment and the issues of the current system and then leveraging this knowledge in order to explore possible solutions.

In the 'Empathize' step, we familiarized ourselves with supply chains, their stakeholder, and the regulations in this space. We saw it as essential to fully understand the problems and where they stem from before beginning to search for possible solutions. Consequently, we first focused on the regulatory standards and related research presented in Section 2. Through the identification of the main stakeholder of the supply chain, we were able to assess the environment in a more structured manner. In the following, the stakeholder's roles and motivations were analyzed, and challenges as well as priorities were determined. Additionally, we used this step to establish a common baseline by discussing general supply chain principles and task-significant related concepts. In order to gather our findings and obtain a better overview of the domain, we continued with the second step.

In the 'Define' step, we aimed to define what general specifications the supply chain environment required. For this analysis, a Stakeholder Diagram containing the main stakeholder's actions, goals, and connections was constructed. Furthermore, the elements were evaluated according to their significance for the system design, as certain aspects of the diagram are an integral part of the supply chain and need to be modeled by the system. In contrast, others might hold general importance but are mostly irrelevant to the system's design. As a prerequisite to exploring the solution space for the relevant requirements, we first had to perform an analysis to verify that DLT-based methods are a viable approach for the determined problem. To do so, we leveraged the "DLT decision framework" tool provided by fortiss [9]. The decision framework confirmed that DLT is applicable to this issue and additionally provided us with helpful suggestions for the system design.

In the 'Ideate' step, we focused on designing the system. For this, we first developed multiple systems under the guidance of the decision framework's results. The framework helped us decide what form of DLT (public/hybrid/consortium blockchain) we should use. In addition, it gave us meaningful insight into crucial factors concerning the Application, Middleware, and Infrastructure Layer of the system. The Stakeholder Diagram was then used to design multiple candidates for the system's layout, each candidate focusing on different aspects of the problem. During this process, it became clear that the focal point for this system's design would be the trade-off between traceability and confidentiality. At the end of this step, all candidates were compared, and the most promising, a compromise between traceability and confidentiality, was selected for our implementation of the solution.

In the 'Prototype' step, we implemented our system design as a proof of concept on how the use of DLT can accomplish a sustainable supply chain. For this, the prototype only concentrates on the most vital elements of the supply chain that are required for the modeling via blockchain. The implementation utilizes

Docker Containers to run the organizations of a Hyperledger Fabric network, which in turn was configured and created by Fablo. Lastly, we developed a preliminary user interface in order to showcase the operations on the blockchain.

4 System Requirements

As mentioned in Methodology (section 3), this section compiles our findings in order to formally define the requirements for the system. We decided to utilize a Stakeholder Diagram and the DLT Decision Framework as the basis for the system design. Moreover, we closely inspected the trade-off between traceability and confidentiality, as they are the two key factors determining the system design.

4.1 Stakeholder Diagram

The Stakeholder Diagram in Figure 2 depicts the main stakeholders in the supply chain environment as well as their respective Actions(A), Goals(G), and their relationships amongst each other. The complete diagram considers elements that are not crucial to the supply chain. Therefore, these elements are grayed out and are not considered during the design process. For example, the system did not handle the exchange of money as we considered it to be unimportant for the fundamental functionality.

Furthermore, we decided against explicitly modeling the Shipping Companies as part of the system but instead implicitly integrated them as a part of the organization selling a product. Consequently, the Mines represent the starting link of the supply chain, as they source the raw materials, the Original Equipment Manufacturer (OEM) represents the end link that not only manufactures the final product but also sells it to the consumers, and the suppliers represent the connecting middle links that produce smaller sub-products from the Mines' resources and sells them to the OEM for the assembly of the final product. To limit the project's scope, the tracking of a product by the system ends with the sale to the consumer. After-life processes are not considered.

4.2 Traceability vs. Confidentiality

Our project on the automotive manufacturing supply chain emphasizes the confidentiality of trading information. This is especially crucial given the sensitive nature of data exchanged among raw material suppliers in our use case.

Before implementing the project, we identified essential requirements for our use case. Among these, three significant objectives were crucial: eliminating reliance on centralized authorities, ensuring user confidentiality, and maintaining data traceability. However, striking a balance between these objectives using a blockchain system posed a challenge.

Suppliers of raw materials for manufacturing cars, who sell their goods to car manufacturers, are keenly interested in keeping their transaction data confidential. Any disclosure could enable rival companies to form strategies to gain a competitive edge. For instance, if a competitor gains access to this data, they could offer higher bids to the raw material supplier, leading to potential legal issues in several countries. Also, exposure to profit margins is an undesirable outcome for these suppliers.

Conversely, our system necessitates traceability, as we must guarantee customers the integrity of data regarding goods in the supply chain network. For instance, the system should not blur crucial information or product history, enabling customers to discern whether the product attaches to environmentally friendly practices.

Therefore, we strive to achieve equilibrium in our system, maintaining acceptable traceability to allow customers to assess a product's environmental friendliness while ensuring sufficient confidentiality to protect sensitive trading information between parties.

4.3 DLT Decision Framework

For general guidance on systems requirements, we consulted the DLT Decision Framework provided by fortiss [9]. The framework was capable of determining three major questions:

1. Is DLT applicable for building a sustainable supply chain system?
2. What type of DLT do we need for such a system?
3. What additional requirements are likely?

There was some uncertainty about how particular framework questions were supposed to be answered, as not all questions seemed relevant to our case. However, overall we felt like the framework's output could be trusted as it logically matched well with the questions that were answered confidently.

The results of the DLT decision framework stated that the case is DLT-suitable following the general evaluation and that there is a likely use case according to more subjective fitting criteria. Furthermore, the framework recommended a hybrid/private DLT as the most appropriate and a hybrid DLT as also viable. Lastly, the following additional requirements were recommended for the system's design:

- Hash on-chain and raw data in external storage.
- On-/off-chain connection including delegated computation pattern, content addressable storage pattern, and off-chain signature pattern.

The full report sheet of the DLT decision framework can be found on the project's GitHub page.

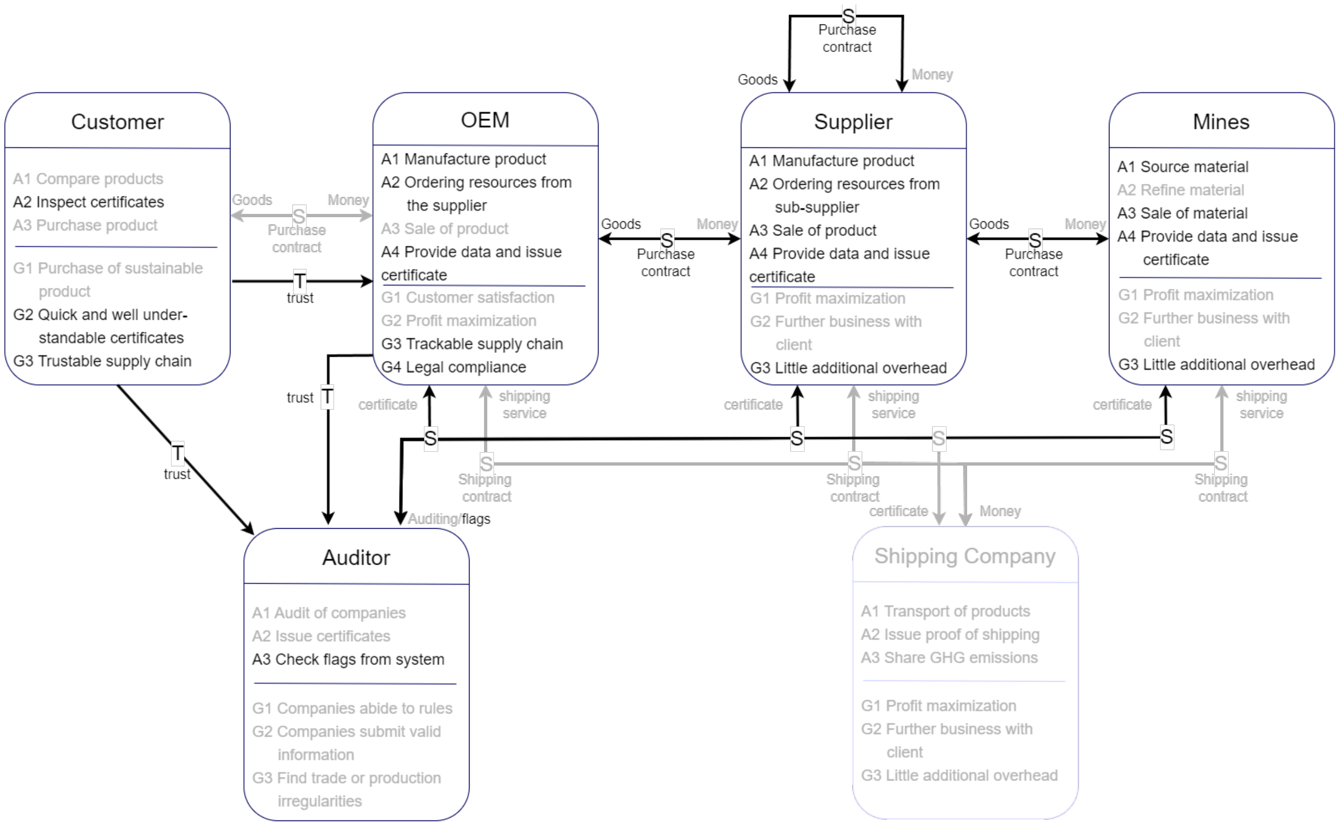


Figure 2: Stakeholder Diagram with system irrelevant information grayed out. A equals Actions and G equals Goals

5 System Design

In this chapter, we present a system design that enables a traceable supply chain infrastructure for tracking GHG emissions while organizations involved maintain confidentiality. We establish the design's scope and introduce the key components (Organizations, Channels, and Chaincode). Furthermore, we provide a comprehensive explanation of how traceability is achieved while upholding confidentiality.

5.1 Scope of our System Design

As stated in Section 2.1, this project emphasizes the tracing and transfer of GHG emissions throughout the supply chain. Our approach aligns with the GRI 305 emissions standard and the GHG Protocol, which assumes that all company emissions can be converted to a CO_2 equivalent in kilograms. This assumption simplifies the design and minimizes the tracking complexity of various emission types. We primarily concentrate on Scope 1 and Scope 2 emissions, as defined in Section 2.1, since they can be mapped to individual product emissions compared to Scope 3 emissions. This condition aligns with the GHG protocol, which mandates only Scope 1 and 2 emissions.

5.2 Organizations & Peers

The blockchain network consists of organizations that establish the organizational structure. Each organization utilizes its own access control and membership service. In this project, we assign each company in the supply chain to its individual organization within the blockchain infrastructure. In this design, each organization has a single member, referred to as a peer. Peers maintain a replicated copy of the distributed ledger and possess the ability to invoke blockchain functionality. To interact with and modify the ledger's state, chaincode is deployed on the peers, which subsequently serve as an interface for organization clients. Since a strict one-to-one mapping exists between organizations and peers, they are modeled as one unit as depicted in Figure 3.

Furthermore, the network incorporates a dedicated and autonomous ordering service responsible for ordering the transactions from the peers into blocks based on a first-come-first-serve principle. This approach guarantees a uniform and consistent state of the ledger. Subsequently, the ordered transactions are transmitted to the peers for validation. In the component diagram (Figure 3), other components, like Certification Authorities, are excluded if they are unrelated to the application logic. Nevertheless, these entities remain crucial in maintaining the overall network's functionality.

Auditors A distinct role known as "Auditors" exists within the proposed architecture. Auditors are similar to companies; they are assigned to their own organization and peer to interact with the network. Moreover, their organization is equipped with additional access rights

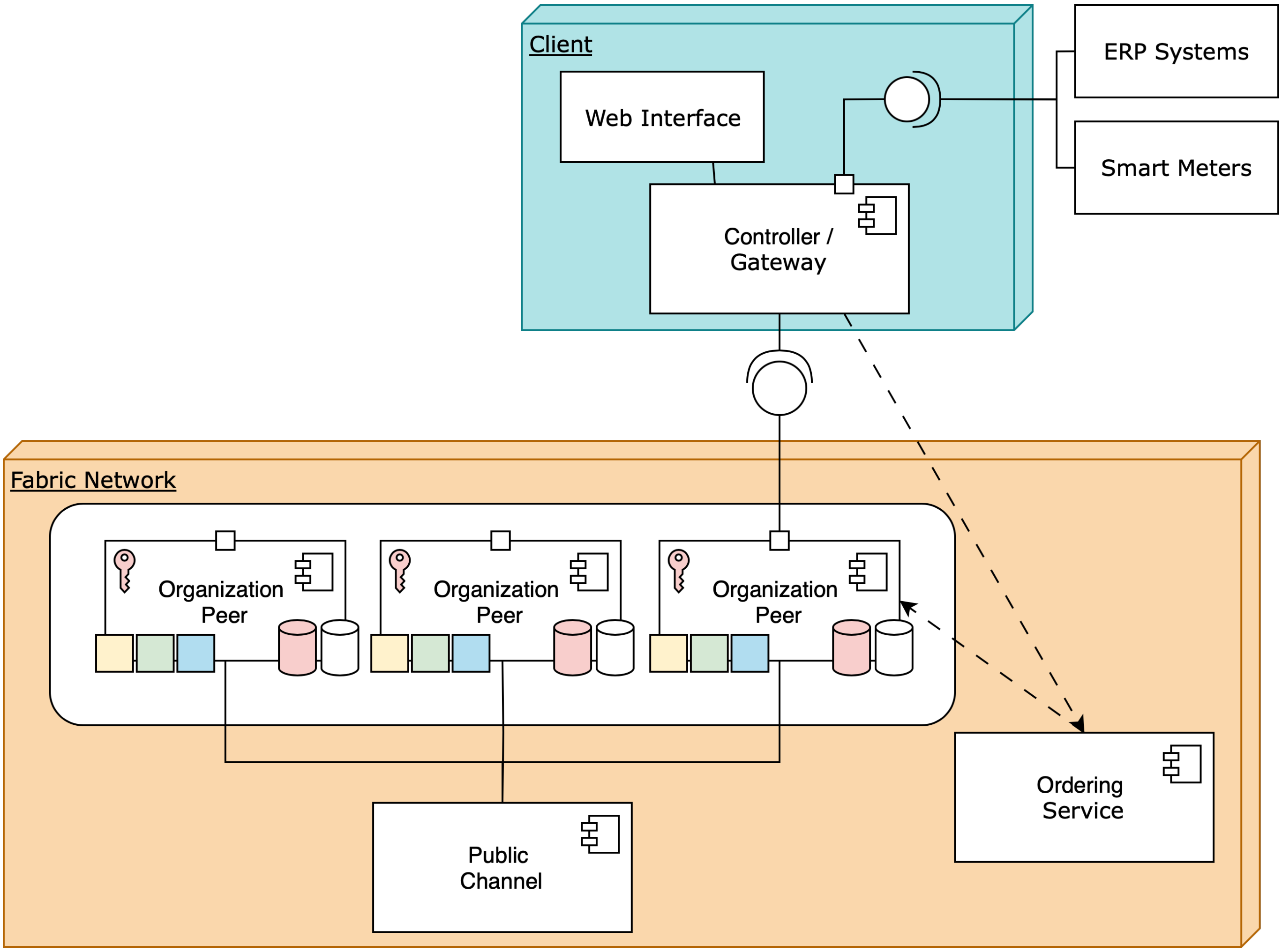


Figure 3: Components Diagram of the proposed architecture

and permissions, which are further explained in later sections.

5.3 Channels

Channels in the context of the blockchain network represent a layer to communicate application logic and build an overlay of the infrastructure. Each channel contains a distributed ledger shared among all the peers within that specific channel. To participate in a channel, organizations require to be granted access. The visibility of the ledger's state is restricted to organizations and their peers with the required permissions, ensuring data isolation from external entities and confidentiality. To facilitate traceability throughout a supply chain, all transactions are systematically recorded on a single channel, inclusive of all organizations involved in the supply chain.

5.4 Main Functionality

The supply chain application comprises three primary components. Firstly, companies participating in the supply chain must be able to record their emissions and establish a linkage between emissions and their respective products. Secondly, a company needs to be able to merge and transform existing products into novel ones, similar to the process in a manufacturing plant. Subsequently, in adherence to the fundamental principles of a supply chain, the newly created products must be transferred to the subsequent company within the supply chain. The following paragraphs elaborate on the mechanisms to achieve these three core functionalities.

During the design of these functionalities, it is imperative to establish a chain of custody for the materials and subproducts, ensuring persistent traceability. While doing so, confidentiality among the involved business partners needs to be guaranteed.

Emissions Recording Companies gather the necessary data from their plants to accurately record their emissions, preferably utilizing Enterprise Resource Planning (ERP) Systems or smart meters. This data is then transmitted to a client responsible for interacting with the network. However, if on-site deployment of such systems and sensors is not feasible or integration is not possible, the client provides an interface to enable manual data input. Once enough emissions data is collected, the client connects to their organization's peer and invokes the *emissions auditing* function. The execution

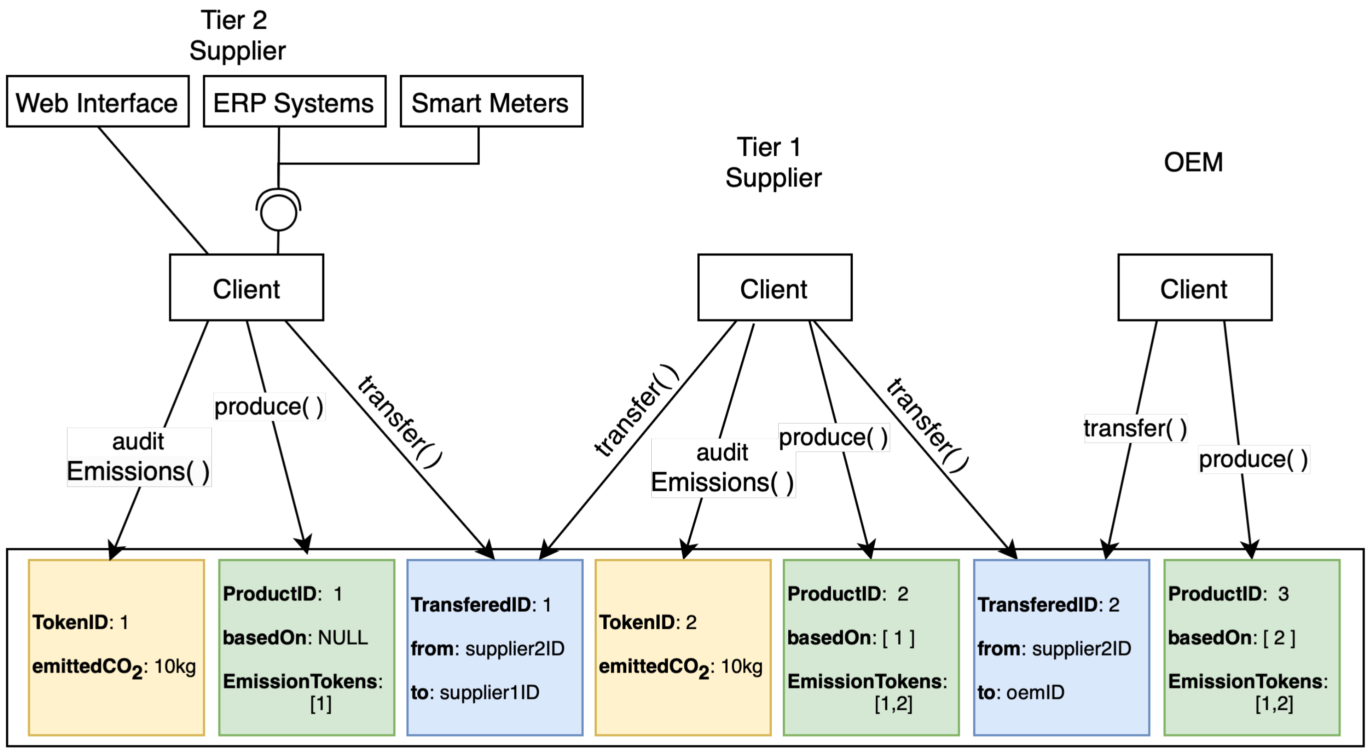


Figure 4: System Design of the proposed architecture

of this emissions audit functionality occurs within the blockchain infrastructure and requires consensus from the majority of organizations on the channel. This function aims to assess the plausibility of emissions reported by the company's client and verify their alignment with actual emissions. Various methods can be employed for these checks, including comparing the data to industry averages, detecting outliers based on historical data, or engaging independent third-party auditors to certify the emissions amount. The integrity and accuracy of emissions accounting rely fundamentally on this auditing mechanism. Once the plausibility checks are completed, an *emissions record* is generated and stored on the shared ledger (displayed as the Yellow Square in Figure 4). This record consists of the quantity of emitted CO_2 in kilograms, along with a unique identifier to reference it in relation to products. An essential concept for *emissions records* is that they are immutable and cannot be modified or deleted after creation, just like the permanence of emissions in our atmosphere. To compute the emissions of a product, it is necessary to query the *emissions records* associated with this product and then calculate the sum of their emissions.

Product Creation To accurately capture the production of a product within the system, the company inputs relevant product information into the client interface. At a minimum, the system requires information regarding the materials and sub-products used in the manufacturing process and a comprehensive list of corresponding emissions records. Other details, such as product names or product-specific certificates, can also be listed.

Equipped with the necessary data, the client invokes the *produce* functionality on the blockchain infrastructure through the organization's peer. This function generates a *product record* (depicted as the Green Square in Figure 4) on the ledger, with a unique identifier assigned to it.

To ensure companies include all relevant sub-products during production correctly, the *produce* functionality incorporates a comprehensive list of all components required in the manufacturing process, similar to a recipe. Each product's recipe is specifically tailored and validated by independent third-party auditors, safeguarding the accuracy and completeness of the information.

Product Transfer The network incorporates a tamper-proof functionality to facilitate the secure transfer of products among companies in the supply chain. Like the previously discussed functions, the client gathers relevant input data to initiate the transfer process. Essential information for product transfers includes 1. the identifier of the product record being transferred, 2. the organizational identifier of the current product owner, and 3. the identifier of the organization receiving the product. Using this information, the client indirectly invokes the *transfer* function on the network. The function creates a record of the product transfer on the ledger. This *transfer record* is represented as the Blue Square in Figure 4.

To finalize the transfer, the receiving company must confirm the shipment by invoking the same *transfer* function with identical input parameters. The transfer is considered successful only if the execution of the *transfer* function by both parties is accurately recorded on the ledger. This mechanism ensures that a single company cannot falsely transfer products and associated emissions to another company without proper consent.

5.5 Traceability

Traceability in the context of sustainable supply chains means the ability of authorized participants to comprehend and retrace the composition of products with their components, materials, and emission footprint throughout the production process.

The proposed architecture in this research ensures comprehensive traceability by referencing unique product IDs. As described in the preceding section, the *produce* functionality assigns a distinct ID to each product and persistently stores it on the ledger. Furthermore, the *product record* also captures information about the components used in its creation.

This framework enables a recursive approach to traceability, where one can initiate the process with a given product ID, retrieve the corresponding *product record* from the ledger, and subsequently access the *product records* of each sub-component. This iterative procedure identifies all materials and components involved in the product's manufacturing and their associated emission records.

5.6 Privacy & Confidentiality

In order to protect trade secrets within the supply chain, the system must maintain confidentiality. Companies rely on the guarantee that their partnerships and supplier relationships remain undisclosed to prevent exclusion from the supply chain. Additionally, it is imperative to prevent unauthorized individuals from inferring the specific components and materials used in product manufacturing, guarding the company's business interests. These objectives of absolute confidentiality are contradictory to achieving full traceability within the system. The presented architecture finds a suitable trade-off by restricting traceability only to authorized parties, prioritizing privacy preservation.

Private Data Multiple techniques are utilized to keep data confidential while storing proof of its existence on the ledger for future verification and tamper resistance. Instead of storing plain data on the public ledger, only the hash of the data is publicly visible on a record. Thus anyone with access to the original data can quickly verify the legitimacy of the transaction and its attributes, ensuring data immutability. The actual data is securely transmitted between authorized parties through encrypted communication. It is then stored in separate, encrypted databases (indicated by the red colored database in the component diagram, Figure 3). This data storage approach is called Private Data Collection within Hyperledger Fabric Networks [29].

To fulfill the confidentiality requirements, the components of a product from the *product record* (labeled "basedOn" in Figure 3) and the entire *transfer record* are stored within Private Data Collections (PDCs). The confidential information in the *product records* is accessible only to the organization that owns the product, resulting in a PDC which is not shared with any other organization. On the other hand, the *transfer records* need to be accessible to both involved parties. Therefore, when two companies transfer products, each company has access to a shared state of a PDC that exclusively contains the respective *transfer records*.

Through this need-to-know principle, companies and their organizations have only access to relevant information. If a company for example leaves the supply chain network, their automatically removed from the respecitve PDC and thus don't have access to any future private information.

In cases where a third-party auditor conducts a full supply chain audit, each company needs to grant the auditor access to their PDCs. Such access is necessary for the auditor to fully comprehend the transactions occurring within the network. This design empowers individual companies to maintain control over their data.

Identity Mixing Although the actual data remains private, another challenge persists within blockchain technology. Due to the inherent nature of blockchain, every transaction is recorded, including meta information such as the caller's identity. However, this information can disclose sensitive details about a company and its business practices. For instance, by examining the created *emission records* of an organization, anyone could conclude the emissions produced during a specific time frame.

To address this issue, it becomes crucial to obfuscate the peers' identities invoking the network functions. In Hyperledger Fabric, this can be accomplished using a technique known as Identity Mixing [30]. Through this approach, access rights and peer authorization are assessed using zero-knowledge proofs without revealing the actual identity of the peer. Consequently, the peer is assigned a unique random identifier for every transaction.

5.7 Scalability

The system can scale in two directions to accommodate the needs of a growing supply chain. When the supply chain extends by adding more companies (vertical scaling), these companies can be incorporated into the existing channel. To add another organization onto a channel, a one-time reconfiguration of the channel and the authorizations is required alongside with the instantiating of new PDCs.

Hyperledger Fabric, in terms of transaction throughput, has been proven to handle up to 350 transactions per second with limited resources [31]. These rates should be sufficient even for extensive supplier networks with thousands of companies, as only a limited amount of transactions from each company are expected to occur daily.

6 System Implementation

In this project, we used Hyperledger Fabric because Hyperledger Fabric is one of the most suitable blockchain platforms for supply chain projects. Although most well-known blockchain platforms (like Ethereum and Bitcoin) are permissionless blockchains, Hyperledger is a permissioned blockchain. Permissioned blockchains are platforms where only authorized people can join the network, whereas, in permissionless platforms, everybody can join the network. In our project, only allowed parties will have read and write access to the blockchain. This approach has some valuable advantages for us. Firstly, it enhances the system's security by limiting network access to approved parties only. Secondly, it promotes scalability and robustness since only authorized participants have read and write access to the blockchain. By leveraging Hyperledger Fabric's permissioned architecture, we create a more controlled and secure environment for our supply chain solution. This strategic choice enhances the overall reliability and efficiency of our system.

Hyperledger also enables us to write and deploy smart contracts called chaincode and send them to the network. These chaincodes are classes used for automating business processes and agreements by modifying the network's state. Additionally, these chaincodes are visible to all peers in the network, allowing different parties to trust the business logic as all participants can see transaction content before committing a transaction [32].

Moreover, we implemented our smart contracts using the programming language Golang due to its strong community support among Hyperledger users. This choice allowed us to find solutions to our problems more quickly. Golang is also known for its high performance and fast execution times, making it an ideal language for our smart contract development.

Building a network with Hyperledger Fabric can be challenging at times, especially during the development phase, due to the complexity of network configuration and the requirement of intricate scripts and Docker images. For example, adding a new peer to the Hyperledger Fabric network involves configuring TLS communication between peers and editing numerous network configuration files. This process impedes the quick deployment and teardown of the network, significantly slowing down our development phase. Fablo enabled us to specify the network via a configuration file. These files contain information about the details of the network.[33]

After specifying these features in a configuration file, we were able to start our Hyperledger network in a containerized environment. Using Hyperledger CLI commands, we interacted with the blockchain, with each organization having its own container. Moreover, when we needed to update the chaincodes, we no longer had to bring down the network and prune every image. Fablo allows us to upgrade chaincodes without any problem.

We implemented two chaincodes for our business logic which are EmissionsAuditChaincode and TransferAssetsChaincode. The EmissionsAuditChaincode provides functionalities for creating emissions records, adding private details to the private data collection, retrieving emissions records, auditing emissions, and more. On the other hand, the TransferAssetsChaincode focuses on handling transactions between peers, such as creating an asset or creating a shipping, etc. This chaincode heavily utilizes private data collections, as it manages a significant amount of sensitive data.

Hyperledger Fablo has other interesting and useful features that are not used in the current state of the implementation. However, these features of Fablo might be useful for potential project deployment. For instance, by using Fablo, we can start a blockchain explorer to observe what happens in the blockchain. This explorer can be broadcast in localhost, and we can monitor the blockchain with a simple user interface. Additionally, Fablo has a REST-API for managing identities, discovering the network of a channel, and querying and invoking chaincode[34].

For development purposes, we interacted with docker and communicated with the blockchain network through the terminal. However, we also developed a backend service to interact with the Hyperledger CLI. As middleware, we use Express.js, and for communicating with the Hyperledger services, we utilize gateways as they abstract the communication details and handle contract retrieval and configuration changes.

In our use case, we have different roles for organizations and their peers. These roles can be categorized into OEM, supplier, Mine, auditor, and customer. In the chaincode, we specified some characteristics for these roles. For instance, we allow organizations with the role of "Mine" to create assets and final products. Final products can also be created by an OEM. Auditors in our implementation are the organization with a powerful role in the network as they can read and write every private data collection.

6.1 Network

To expedite the development process of our Hyperledger Fabric network, we utilized Fablo, a versatile toolkit known for its capabilities in creating Docker containers and customizing network behavior. With Fablo, we efficiently generate Docker containers for peers and customize network aspects like consensus mechanisms, channel creation, and arrangements of organizations and peers. The seamless integration of channel code further simplified smart contract deployment within the network. By leveraging Fablo, we achieved a faster and more streamlined development cycle for our Hyperledger Fabric network, enhancing efficiency and customization possibilities[33]. By leveraging Fablo, we gained the ability to construct a Hyperledger Fabric Network with ease by simply specifying a single JSON or YAML file. This streamlined approach allowed us to define the network's configuration, including the desired organizations, peers, channels, and chaincodes, all within a single file.

Our architecture features a single channel with deployed chaincodes and facilitates peer communication. Within this configuration, we have successfully deployed two chaincodes encompassing the essential business logic for asset transformation and emission recording. Additionally, our architecture comprises three organizations, each consisting of multiple peers. This setup enables seamless collaboration and data sharing among the organizations while ensuring efficient execution of transactions and operations within the network.

Within our network configuration, we successfully implemented private data collections for each peer by utilizing Fablo's capabilities. Through Fablo, setting up private data collections proved to be a seamless process. We specified the desired names and participants of the private data collections, allowing us to access and invoke them within our chaincode. This feature enhanced privacy and data isolation within the network, ensuring that sensitive information is shared only with authorized participants. By leveraging Fablo's functionality, we achieved efficient management of private data collections, further enriching the academic exploration of blockchain privacy mechanisms.

Fablo offers the flexibility to select the consensus mechanism and Fabric version for our network setup. We opted for Hyperledger Fabric 2.4.2, which is one of the latest versions available. In terms of consensus mechanisms, we had the choice between RAFT and SOLO. RAFT is a Byzantine Fault Tolerant consensus mechanism commonly used in production environments due to its robustness and safety features. On the other hand, SOLO, which relies on a single party for transaction validation, is typically employed during the development phase. Although SOLO provides faster transaction processing and simplified network construction, it lacks the decentralized nature of RAFT. We chose to utilize SOLO for our development phase, and when transitioning to production, we can easily switch to RAFT to ensure a more robust and secure system.

6.2 Chaincode

The chaincodes are divided based on their application area, as it allows for better organization and management within the supply chain system. The two categories are chaincodes related to handling emissions and those responsible for asset transfers. However, both categories have some inherent similarities. Transient data is always used for the input parameters when invoking chaincode with private data, as it obfuscates its content and keeps it confidential. Furthermore, certain helper functions are also shared.

6.2.1 Emission Audits

Section 5.4, explains how the system conducts an audit and verification process for the emissions of a company. This verification process is implemented by chaincode functionality, which is described below.

- **Audit Emissions** The chaincode function receives its input parameters through function arguments. In particular, it requires the emitted $\mathrm{CO_2}$ in kilograms, which should be recorded, and a list of the previous emissions of the organization, represented by their corresponding *emission records* IDs. The chaincode performs a query to retrieve the previous emission of the organization, based on the provided *emission record* IDs. Subsequently, it applies an outlier detection algorithm [35], which behaves as follows:
  - If no base value exists, the tested value is accepted.
  - For less than five base values, the algorithm calculates the median and classifies the tested value as an outlier if it deviates by more than 50% from that median.
  - For more than four base values, the algorithm calculates quantiles and classifies the tested value as an outlier if it lies outside the range between the first and third quantile.

  If the validation step confirms that the tested value passes and aligns with the emissions profile of the company, the system creates an *emissions record* on the ledger which contains the amount of emitted $\mathrm{CO_2}$. Furthermore, the ID of the *emissions record* is stored in a PDC exclusively accessible by the invoking organization. This PDC ensures that the company can maintain an overview of the *emissions records* it has generated and allows for the traceability of these records back to the company. In case the emissions fail the outlier detection, the function terminates with a failure and provides an error message to the caller.

  In the current implementation, obtaining the input of previous emissions of an organization from the organization itself is problematic due to potential tampering. However, this is necessary because the network restricts querying a PDC and altering the state of the public ledger within the same transaction.

The Audit Emissions Chaincode represents the only emissions chaincode functionality that requires invocation in a production scenario. However, additional functions for state representation and debugging purposes are implemented, which this work does not explain further.

6.2.2 Asset Transfers

The auditor is tied to a specific organization within the network's structural framework. As a result, the chaincode effectively restricts access to functions exclusively intended for the auditor by verifying whether the invoking peer corresponds to the auditor's organization. When a mismatch is detected, the system raises a flag, which is then stored in the respective peer's PDC. Both the auditor and the peer can access the flag, which contains detailed information about the problematic attempt and the corresponding date. This flagging mechanism is primarily applied to segments of code deemed highly susceptible to misuse and aims to identify potentially malicious activities. Auditors can assess the severity of such flagged events based on their frequency and the information provided and conduct an audit if deemed necessary. The deployment of flags is implemented system-wide and will be explicitly referenced whenever utilized.

The *auditor* invokes two critical chaincodes as part of the system's configuration:

- **Give Rights**: In the system, every organization is mandated to have a specific role that governs its access to various functions. These roles include Mines, Suppliers, and OEMs, and they determine the actions each organization is permitted to perform. For instance, Suppliers positioned in the middle of the supply chain are restricted from creating assets, as this privilege is reserved exclusively for Mines, which initiate the chain. The assignment of roles for each organization is stored in their respective PDCs. A unique ID identifies the rights element across all PDCs. Consequently, any attempt by a company to tamper with or overwrite this ID while invoking different chaincodes triggers a flag. The system meticulously assesses the severity of such malicious attempts and raises a flag accordingly, providing a detailed description of the incident. By enforcing this role-based access control and vigilant ID monitoring, the system effectively safeguards the integrity and security of the supply chain network.
- **Create Recipe**: The auditor creates a recipe after thoroughly auditing the manufacturing process. This recipe serves as a comprehensive blueprint, delineating the specific assets and their respective quantities essential for the production of a new asset. Analogous to a cooking recipe that meticulously outlines the ingredients and their precise measurements required to create a particular dish, this manufacturing recipe plays a comparable role in ensuring the accurate and authorized production of new products. The recipe itself is stored within the PDC of the organization for which it is issued. By deploying this recipe-based approach, the auditor effectively controls and monitors the production processes, mitigating the risk of unauthorized or inappropriate use of assets.

As previously stated, any attempt by an organization other than the auditor to invoke these chaincodes will result in access being denied, and a flag to indicate the unauthorized access attempt.

With the configuration and access controls in place, the blockchain is now fully capable of facilitating the creation and transfer of assets. The supply chain begin with the *Mines*, which serve as the initial link in the chain:

- **Create Asset**: The Mine initiates the asset creation process on the blockchain by providing essential details such as the asset's unique ID, the product name, and the Emission IDs related to its creation. Upon receiving this information, the chaincode generates a product record that outlines what assets a product is based on, in this case "null", and its Emission IDs. This element is then pushed to the chain, acting as a starting block for the publicly visible traceability elements. Thereafter, the asset is stored in the PDC of the invoking organization. However, if any organization other than a Mine attempts to create an asset, a flag is instead stored in their PDC. In our current implementation, the assets created by the Mines must be used to manufacture new assets and cannot be shipped out directly. Nevertheless, this can be customized according to the need of the supply chain members.

Once the first assets are created, the manufacturing process of new assets can commence, adhering to the specified recipes. Afterward, manufactured assets can be bundled together into shipments in order for them to be exchanged with other organizations. These function can be executed by *Mines, and the Suppliers*. The *OEM* can *only* access *shipment-related chaincode*:

- **Manufacture Asset**: The chaincode receives the recipe, emission IDs, and the assets necessary for the manufacturing process. The recipe verification entails checking whether all the required assets exist within the organization's PDC and confirming that their quantities match the specifications outlined in the recipe. Upon successful verification, the chaincode proceeds with the manufacturing process. First, the old assets involved in the manufacturing are deleted from the PDC of the organization. Subsequently, the newly created asset is added to the manufacturing organization's PDC. Furthermore, the product record outlining the used assets, e.g. "[A1,A2]", and its Emission IDs, is pushed to the chain.

- **Create Shipping**: Manufactured assets of the same type can then be packaged together as a shipment. Information about the shipment also includes the date of shipping. If the date is incorrect, a flag is thrown. However, if there are no issues, the packaged assets are deleted, a detailed representation of the shipment is saved to the shipping organization's PDC, and a reduced representation of the shipment containing the ID, the Seller, and the type of the assets is saved to a PDC that the Buyer and the Seller share.
- **Claim Shipping**: The Buyer organization can see if a shipment exists by querying the PDC that it shares with the Seller. If there is a shipment, it can be claimed by the buying organization. If the Seller tries to claim it, a flag will be thrown. The Buyer must enter the same detailed shipment information as is stored in the Seller's PDC. The hashes of the Seller's and the Buyer's info are compared. If they match, the shipment is deleted from the shared PDC, and the assets are unpacked in the Buyer's PDC. Otherwise, a flag is thrown. The shipment from the Seller's PDC gets deleted automatically the next time the Seller invokes a chaincode.

To conclude the process and allow the final product to exit the supply chain, only the *OEM* is granted the privilege to invoke the following final chaincode:

- **Final Product**: is a modified version of the normal Manufacture Asset chaincode. Instead of saving the manufactured product on the OEM's PDC, the asset is pushed to the chain unencrypted. Consequently, the final product is publicly visible and consumers can inspect the product's emission figures.

In addition, there are several query functions to read all of the elements on the blockchain and from the PDCs of organizations.

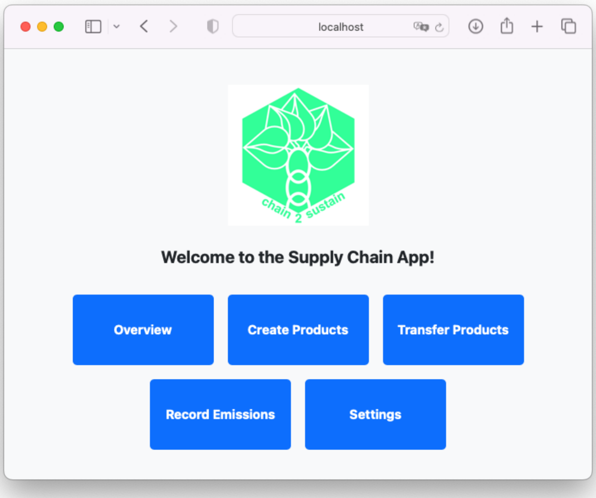
6.3 Client

The client acts as the interface for companies using the supply chain network, offering diverse interaction methods to ensure a comprehensible and user-friendly experience. However, the client is not required to interact with the supply chain network, it merely functions as a convenience product to abstract interactions with the network. The client provides two interfaces to the user: a web interface and an API for smart appliances in the manufacturing plant to connect. It is developed as a NodeJS application and, upon execution, starts a local web server. The graphical user interface is created using the Express.JS web framework [36], along with Embedded JavaScript templates, and can be accessed through the local port 3000. Figure 5 displays examples of the user interface. The web pages primarily consist of input fields to gather the necessary data for invoking chaincodes on the network. Moreover, they are designed to display responses, query results, and error messages as required.

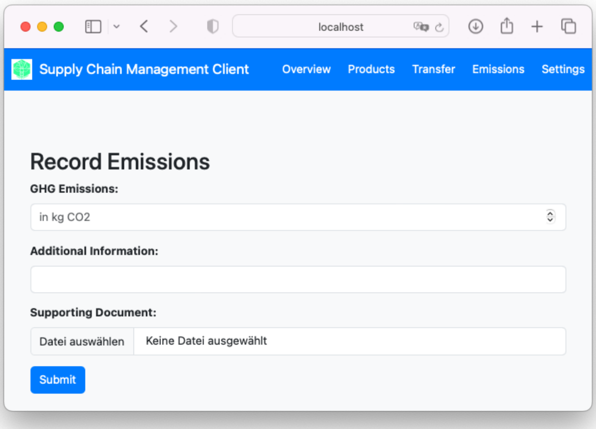
Smart appliances can establish a connection with the client through a REST-API. The comprehensive documentation of the endpoints can be found in a Markup File alongside the source code. The client utilizes the Fabric Gateway [37] to connect to the organization's peer. The gateway abstracts complex calling patterns and simplifies the invocation of chaincode through the peer.

7 Discussion & Limitations

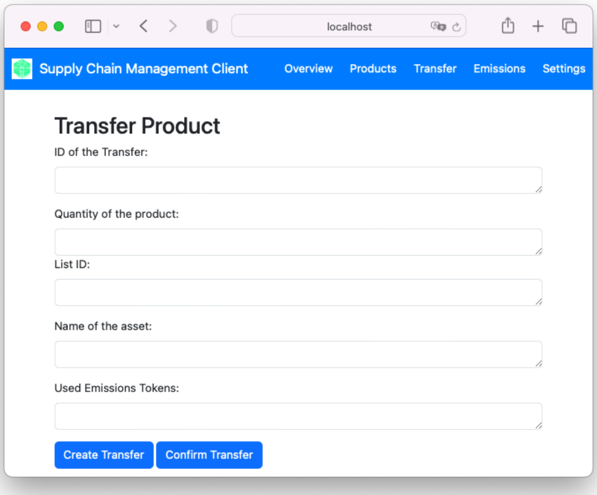
Throughout the development phase of our project, we examined numerous frameworks aiming to secure privacy without compromising the system's traceability features. Generally, we sensed that maintaining traceability often risks exposure to excessive data. Hence, we explored various designs, such as employing multi-channel networks. Furthermore, we considered tools compatible with Hyperledger Fabric, including private data collections and identity mixer, to blur the sensitive data of network participants. In the subsequent sections, we delve into these strategies' challenges, advantages, and drawbacks. Our project has to maintain sensitive trading data to give auditors a clear picture of the transactions between car manufacturers and suppliers.
Initially, we aimed to create multiple dedicated channels for specific stakeholders in the supply chain, including a separate "Channel All" for facilitating communication between auditors and customers. We hoped to ensure confidentiality for car manufacturers and raw material suppliers through this design. However, we realized this would compromise actual traceability since there isn't a direct way to transfer assets between channels without involving an off-chain entity. The involvement of auditors to facilitate this transfer increased reliance on central authorities, creating a bottleneck for customers seeking access to the transaction history.



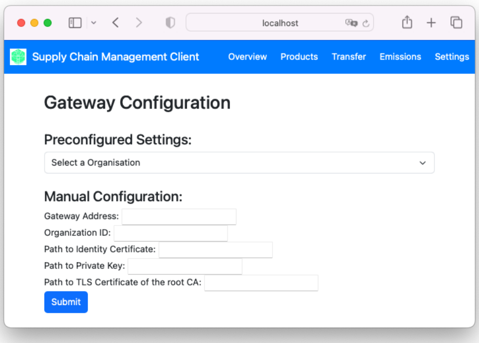
(a) Start Page



(b) Record Emissions



(c) Transfer Products



(d) Settings

Figure 5: Sample UI of the Client

We then explored the use of Identity Mixer, intending to tokenize companies' carbon footprints and energy usage. The tokenized data, however, required sensitive information like energy payments made by companies or suppliers. This raised concerns about the potential exploitation of such data by rival companies. Moreover, while Identity Mixer can hide token creators, it renders the system non-traceable. After careful consideration, we decided not to use Identity Mixer, ensuring customers can confidently identify token owners and understand the supply chain processes.

Finally, we opted to use Hyperledger Fabric private data collections, which allow us to conceal specific information on the blockchain while retaining its on-chain storage. However, this decision partially compromised transparency, as detailed confidential trading data became inaccessible to users. Despite the trade-off in traceability, we prioritized protecting sensitive information within the system.

7.1 Confidentiality & Privacy Multi-Chain Approach

There are some methods to ensure confidentiality in Hyperledger fabric, and one of these approaches we used was creating separate channels for organizations that engaged trading and auditors. To show the supply chain data to users, we planned to create a separate big channel that will be maintained with auditors. However, every channel acts like a different blockchain, bringing some design problems. Companies need to trust the auditors since the only bridge between the trading and customer channels will be the auditor, and the supply chain network will need to be traced more. What the customer will see will not be actual data, but they will become able to see the data that the auditor will transfer. Additionally, when new companies join the network, we need to create new channels, which can lead to scalability problems. In Hyperledger Fabric, every channel peer must maintain a copy of the network. Especially auditors who might need to join multiple channels must keep the number of channels copied.

7.2 Confidentiality & Identity Mixer

We considered utilizing the Identity Mixer feature from Hyperledger Fabric to create trackable carbon tokens while concealing the identities of the token creators. This functionality of the Identity Mixer could significantly enrich system confidentiality. However, due to the continuous development of this feature and its current compatibility only with Java chaincodes, its integration proved not feasible.

ntuition

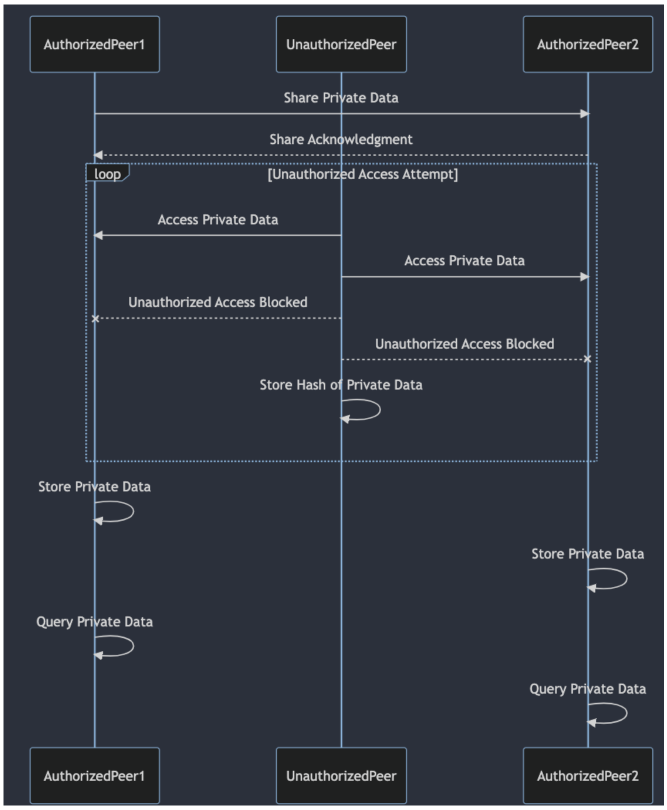


Figure 6: Sequence diagram for private data collection

7.3 Confidentiality & Privacy Private Data Collections

Private data collection, a vital feature in Hyperledger Fabric, was used to achieve confidentiality in the blockchain. Private data collection is a very robust and secure system in Hyperledger Fabric to ensure companies confidentiality. To use private data collections in the smart contract needs to be modified according to business logic; these modifications must include the information that organizations will be allowed to see these private data. In sections of code where inserting and reading from private databases, there should be some access control to determine who can read-write actions for these collections.

When a write happens to private data collections, private data will be stored in the private database of the peers, not in channels. Channels will only contain a hash of private data, allowing only proof of these data's existence. When a function involves private data, the transaction proposal will be sent to endorsing peers of the private data collection. The client application will send a transaction that involves private data to the ordering service, but the ordering service cannot see the private data; instead, they can work with the hash of this data[29]. The peers of the authorized organizations will involve in the validation of transactions to commit the ledger. Other organizations of the channel will be able to access this data.

In our system design, we have different private data collections for organizations that make trade, and for shipping, we also have various private data collections. Using Fablo, we created private data collections by writing a yaml or json file. We specified which organizations will have read-write access to this collection in these files. For instance, when an asset is created and subsequently traded among parties, only the originating organization and the company involved in the transaction can disclose this data. Thus, by utilizing Hyperledger Fabric's private data collections, we successfully established a confidential trading environment to preserve substantial traceability features within the system.

7.4 TLS/SSL Certificates

Currently, the client faces difficulties connecting to the network peer due to TLS/SSL certificate issues arising from the peer deployment in docker containers. Although this issue is yet to be resolved, the overall concept remains valid and has been demonstrated as feasible in other projects [38]. Furthermore, the client is missing some functionality, for example a method to create recipes. Thus it is not possible to exclusively use the client as an interface to the supply chain network.

8 Conclusion

Blockchain has a remarkable impact on today's technology. Blockchain technology has revolutionized various industries, and one of its prominent applications is in supply chain networks. The implementation of blockchain in supply chains has brought remarkable benefits. In today's digital era, many companies are striving for digitalization in their operations. Blockchain technology emerges as a robust and reliable solution for these companies, offering enhanced security, robustness, and efficiency in various business processes.

Utilizing the power of blockchain, supply chain networks in the automotive industry have become a crucial use case, and we wanted to utilize blockchain power for this project. Detailed analysis is necessary for the automotive industry due to extensive trading between parties. Confidentiality among participants must be maintained without losing product traceability. By leveraging blockchain, we can achieve secure and transparent transactions while preserving the integrity of the supply chain. To accomplish this, we analyzed which information needed to be visible to customers. We eliminated sensitive data that users do not reveal.

In our project to create a supply chain network, we are exploring different approaches to track the environmental impact of goods. One promising approach is the tokenization of carbon footprints. By tokenizing carbon footprints, we can create a transparent and traceable system that accurately measures and records the environmental friendliness of the goods produced and traded in our supply chain network. However, we also consider traders' confidentiality, so we must hide the token creator's identity without losing traceability.

For this purpose, we utilized the potential of private data collections to uphold trading confidentiality while preserving traceability. Our current network has a single channel in which all transaction data and business logic is stored. By using private data collections, we can hide users' sensitive data.

In supply chain networks, the immutability of records is a critical issue, and blockchain technology provides a solution by ensuring data cannot be altered. Moreover, the transparency of smart contracts allows parties who lack trust to engage in secure transactions. We facilitated safe transactions between raw material suppliers and car manufacturers using this blockchain feature.

In the context of the car manufacturing process, the issue of document forgery poses a significant problem. Blockchain offers a promising solution to combat such crimes effectively. In traditional scenarios, auditors rely on paper documents for their decisions, which are susceptible to manipulation without regular monitoring. However, leveraging blockchain technology eliminates the need for paper documents, replacing them with digitized and reliable data. This ensures that auditors can access reliable information anytime, minimizing the risk of document tampering.

In future work establishing a more detailed sustainability metric is a crucial objective for our system. At present, we quantify this in kilograms of CO2 emissions, which, while accurate, only sometimes translates into meaningful information for consumers. As we refine our system, we aim to include more comprehensive indicators to show a product's environmental impact. This way, we hope to provide consumers with a clear and understandable sustainability score for each product.

9 Acronyms

ESG Environmen- tal, Social and Governance

GHG Greenhouse Gas

ERP Enterprise Resource Planning

PDC Private Data Collection

References

[1] K. Abeliotis, C. Koniari, and E. Sardianou, "The profile of the green consumer in greece," *International Journal of Consumer Studies*, vol. 34, no. 2, pp. 153–160, 2010.

[2] S. M. Harris, "Does sustainability sell? market responses to sustainability certification," *Manage-*

ment of Environmental Quality: An International Journal*, vol. 18, no. 1, pp. 50–60, 2007.

[3] P. McDonagh and A. Prothero, "Sustainability marketing research: Past, present and future," *Journal of Marketing Management*, vol. 30, no. 11-12, pp. 1186–1219, 2014.

[4] S. Romani, S. Grappi, and R. P. Bagozzi, "Corporate socially responsible initiatives and their effects on consumption of green products," *Journal of Business Ethics*, vol. 135, pp. 253–264, 2016.

[5] M. J. Polonsky, "Transformative green marketing: Impediments and opportunities," *Journal of business research*, vol. 64, no. 12, pp. 1311–1319, 2011.

[6] B. Siebenhüner and M. Arnold, "Organizational learning to manage sustainable development," *Business strategy and the environment*, vol. 16, no. 5, pp. 339–353, 2007.

[7] O. Chkanikova and R. Sroufe, "Third-party sustainability certifications in food retailing: Certification design from a sustainable supply chain management perspective," *Journal of Cleaner Production*, vol. 282, p. 124344, 2021.

[8] J. Wang, J. Tao, and M. Chu, "Behind the label: Chinese consumers' trust in food certification and the effect of perceived quality on purchase intention," *Food Control*, vol. 108, p. 106825, 2020.

[9] "fortiss, dlt decision framework." https://dlt.fortiss-demo.org/. accessed: 23.05.2023.

[10] S. Bose, "Evolution of esg reporting frameworks," *Values at Work: Sustainable Investing and ESG Reporting*, pp. 13–33, 2020.

[11] G. R. Initiative, "Gri 305: Emissionen 2016," 2016.

[12] G. G. Protocol, "Greenhouse gas protocol," *Sector Toolsets for Iron and Steel-Guidance Document*, 2011.

[13] M. E. Sotos, "Ghg protocol scope 2 guidance," 2015.

[14] G. G. Protocol, "Corporate value chain (scope 3) accounting and reporting standard," *World Resources Institute and World Business Council for Sustainable Development, Washington, DC*, 2011.

[15] "Greenhouse gas protocol." https://ghgprotocol.org/. Accessed: 2023-07-14.

[16] X. Liu, H. Wu, W. Wu, Y. Fu, and G. Q. Huang, "Blockchain-enabled esg reporting framework for sustainable supply chain," in *Sustainable Design and Manufacturing 2020: Proceedings of the 7th International Conference on Sustainable Design and Manufacturing (KES-SDM 2020)*, pp. 403–413, Springer, 2021.

[17] T. K. Agrawal, V. Kumar, R. Pal, L. Wang, and Y. Chen, "Blockchain-based framework for supply chain traceability: A case example of textile and clothing industry," *Computers & industrial engineering*, vol. 154, p. 107130, 2021.

[18] N. Kshetri, "Blockchain systems and ethical sourcing in the mineral and metal industry: a multiple case study," *The International Journal of Logistics Management*, vol. 33, no. 1, pp. 1–27, 2022.

[19] C. Ma, X. Kong, Q. Lan, and Z. Zhou, "The privacy protection mechanism of hyperledger fabric and its application in supply chain finance," *Cybersecurity*, vol. 2, no. 1, pp. 1–9, 2019.

[20] D. Lu, P. Moreno-Sanchez, P. Mitra, K. Feldman, J. Fodale, J. Kosofsky, and A. Kate, "Toward privacy-aware traceability for automotive supply chains," *SAE International Journal of Transportation Cybersecurity and Privacy*, vol. 4, no. 11-04-02-0004, pp. 61–82, 2021.

[21] J. Wessel, A. Turetskyy, O. Wojahn, T. Abraham, and C. Herrmann, "Ontology-based traceability system for interoperable data acquisition in battery cell manufacturing," *Procedia CIRP*, vol. 104, pp. 1215–1220, 2021.

[22] J. Wessel, A. Schoo, A. Kwade, and C. Herrmann, "Traceability in battery cell production," *Energy Technology*, p. 2200911, 2022.

[23] H. M. Kim and M. Laskowski, "Toward an ontology-driven blockchain design for supply-chain provenance," *Intelligent Systems in Accounting, Finance and Management*, vol. 25, no. 1, pp. 18–27, 2018.

[24] K. Seidenfad, T. Wagner, R. Hrestic, and U. Lechner, "Demonstrating feasibility of blockchain-driven carbon accounting–a design study and demonstrator," in *International Conference on Innovations for Community Services*, pp. 28–46, Springer, 2022.

[25] S. Chen, "Blockchain mechanism for tracking ghg emissions through supply chain," *Available at SSRN 4082449*, 2022.

[26] "Ibm supply chain intelligence suite: Food trust." https://www.ibm.com/de-de/products/supply-chain-intelligence-suite/food-trust. Accessed: 2023-07-15.

[27] "The rcs global group." https://www.rcsglobal.com. Accessed: 2023-07-15.

[28] "Volvo cars to implement blockchain traceability of cobalt used in electric car batteries."

https://www.media.volvocars.com/global/en-gb/media/pressreleases/260242/volvo-cars-to-implement-blockchain-traceability-of-cobalt-used-in-electric-car-batteries. Accessed: 2023-07-15.

[29] "Private data." https://hyperledger-fabric.readthedocs.io/en/release-2.2/private-data/private-data.html/. Accessed: 2023-07-14.

[30] "Msp implementation with identity mixer." https://hyperledger-fabric.readthedocs.io/en/release-2.5/idemix.html?highlight=Identity%20Mixing. Accessed: 2023-07-16.

[31] M. Q. Nguyen, D. Loghin, and T. T. A. Dinh, "Understanding the scalability of hyperledger fabric," *arXiv preprint arXiv:2107.09886*, 2021.

[32] "Smart contracts and chaincode." https://hyperledger-fabric.readthedocs.io/en/release-2.2/smartcontract/smartcontract.html/. Accessed: 2023-07-14.

[33] "Hyperledger-labs fablo." https://github.com/hyperledger-labs/fablo/. Accessed: 2023-07-14.

[34] "Hyperledger-labs fablo rest." https://github.com/fablo-io/fablo-rest/. Accessed: 2023-07-14.

[35] H.-P. Kriegel, P. Kröger, and A. Zimek, "Outlier detection techniques," *Tutorial at KDD*, vol. 10, pp. 1–76, 2010.

[36] "Expressjs." https://expressjs.com. Accessed: 2023-07-17.

[37] "Fabric gateway." https://github.com/zalando-incubator/fabric-gateway. Accessed: 2023-07-17.

[38] "Asset transfer basic sample." https://github.com/hyperledger/fabric-samples/tree/main/asset-transfer-basic. Accessed: 2023-07-17.