Received: 9 May 2024 | Revised: 21 June 2024 | Accepted: 29 July 2024 | Published online: 5 August 2024

Journal of Data Science and Intelligent Systems
2024, Vol. 00(00) 1–15
DOI: 10.47852/bonviewJDSIS42023384
BON VIEW PUBLISHING

RESEARCH ARTICLE/REVIEW

Multi-Layer Track & Trace System Landscape with Smart Contract Validation Instances

Markus Rabe[1] and **Henrik Rainer Körsgen**[1,*]

[1] *IT in Production and Logistics, TU Dortmund, Germany*

Abstract: Ideas about blockchain-based applications are soaring, while the realization of existing enterprise architecture landscapes is more cumbersome. This is also valid for applications based on track & trace blockchains. The main impediment to implementing smart contracts with a subsequent financial flow from one blockchain peer to another is trust. Trust issues are prevalent in many business areas. One of the most effective countermeasures is a demonstration of the power of blockchain information systems. By showing the actors of a supply chain how their nodes in a common blockchain are interacting, suspiciousness is traded for knowledge. For this reason, an example of a realistic blockchain application is discussed in detail. The main components are a widely used SAP Business Technology Platform and Hyperledger Firefly. Since multiple layers are part of such an instance, the focus will be on the Hyperledger side. This study presents a notable advancement in the realm of blockchain-enabled track & trace solutions through the use of a novel system architecture for its end-to-end implementation. Its architecture achieves crucial integrations between web3 technologies and established enterprise systems. It offers new functionalities, such as AI integration, while simultaneously acknowledging the inherent challenges and charting potential trajectories for future research.

Keywords: track & trace, web3 technologies, smart contract application, proof of stake concept, logistics enterprise architecture

1. Introduction

Supply chains continue to face challenges such as lack of transparency, inefficiency, and vulnerability to fraud. In the quest to address these issues, blockchain technology has emerged as a transformative solution [1]. Blockchain technology, initially introduced as the underlying framework for cryptocurrencies like Bitcoin, has evolved significantly over time. Its core principles, including decentralization, immutability, transparency, and cryptographic security, make it a promising candidate for supply chain execution [2, 3]. The ability to create a tamper-resistant, decentralized ledger of transactions, and assets offers immense potential to enhance the reliability and transparency of supply chain operations. While being immutable, transparent, and traceable, the blockchain is an ideal technology for building up a sustainable supply chain. Blockchain systems combined with automation and an event-driven approach could boost efficiency by speeding up processes that might usually take days or weeks and save time and resources instead of waiting for manual paperwork. Blockchain's immutable ledgers offer a potential solution against phishing attacks. By verifying the authenticity of information and eliminating the possibility of data tampering, blockchains can significantly reduce the effectiveness of phishing scams, protecting individuals and organizations alike. These are only some of blockchains' potentials.

Despite the growing interest in blockchain applications, including track & trace solutions, integrating them into existing enterprise systems remains a challenge. The problem is that their wide adoption has not occurred yet. The primary obstacle is trust issues within intercompany interactions [4]. Smart contracts regulating financial transactions between blockchain participants seem a feasible solution. The main impediment to implementing smart contracts with a subsequent financial flow from one blockchain peer to another is trust.

The objective of this study is to address the trust gap by demonstrating the power of blockchain information systems (BISs) in a business scenario. Through a detailed example using SAP Business Technology Platform (BTP) and Hyperledger Firefly, the study proposes a novel system architecture for an end-to-end track & trace solutions. This architecture facilitates crucial integrations between web3 technologies and established enterprise systems, paving the way for new AI functionalities. This study presents a notable advancement in the realm of blockchain-enabled track & trace solutions through the use of a novel system architecture for its end-to-end implementation. Next, literature about the interoperability and trust issues connected to the blockchain is provided. The literature review concludes with the challenges of connecting a system of records with a system of engagement. Then, the use case components are presented in the research methodology, followed by a discussion and conclusion chapter.

2. Literature Review

2.1. The interoperability issue

One of the reasons is that blockchain solutions render interoperability across different systems difficult and even more

*Corresponding author: Henrik Rainer Körsgen, IT in Production and Logistics, TU Dortmund, Germany. Email: koersgen@dxk.ch

difficult across enterprise boundaries. The interoperability issue was recognized by the Gartner authors Klappich et al. in 2017. According to a recent evaluation of contemporary smart contract analysis tools, the interoperability issue persists [5]. Although the two findings are similar, the term interoperability is used differently. In the research by Gartner, interoperability concerns the possible blockchain expansion to another organization within the supply chain network. Such an expansion requires stakeholder interoperability, meaning that the blockchain technology needs to be able to connect to different underlying systems. In the evaluation of the smart contract, interoperability pinpoints the difficulty of connecting two different blockchain ecosystems. Standards for establishing compatible blockchain platforms still do not exist [5, 6].

One attempt to design an open interoperability protocol was launched by Toposware, Inc. with Topos. The protocol refrains from utilizing a centralized blockchain or employing consensus mechanisms. The latter needs to be agreed upon by all peers within a blockchain. It is the basis for a uniform transmission of messages within blockchains and their subnets. Instead, their protocol relies on a weak causally reliable broadcast message achieved through a distributed network known as the Transmission Control Engine (TCE). Additionally, the Universal Certificate Interface (UCI) ensures the validity of cross-subnet messages [7]. However, the drawback of Topos is that the required UCI relies on subnets. The latest Hyperledger solutions allow a consortium to split its blockchain into subnets. Messages, tokens, and smart contracts can still be broadcasted to all nodes in the blockchain [8]. By validating trusted transactions, smart contracts can streamline business processes, reduce operational costs, and foster trust in blockchain technology [9]. Hence, smart contract validation offers a powerful tool for secure and automated transactions within blockchain ecosystems. This technology ensures that predefined conditions are met before executing an agreement, minimizing risks associated with fraud and human error [10]. One way to design smart contracts is by means of unified modeling language. Górski [11] presented a smart contract design pattern with a set of interfaces and classes ensuring the electronic circulation of documents [11]. The use of separate channels for each subnet can limit the communication to only those nodes and, therefore, those peers that are part of the subnet at hand. This channel architecture enables private and confidential transactions [12]. Hence, also with Topos, interoperability is limited to a blockchain architecture designed with subnets. Nonetheless, the TCE provides an alternative to the Proof-of-Stake (PoS) concept.

The lack of interoperability arising from the isolated development of blockchain solutions limits the scope of use [6]. Enterprises are bound to agree upon a single blockchain platform or consider blockchain gateway solutions [6]. As a result, one blockchain platform is chosen for the demonstration case.

2.2. The trust issue

In the context of proof, another critical blockchain blocker should be mentioned. One of the reasons for the lack of wide adoption of blockchain technology in supply chains is trust. The PoS consensus was developed as an alternative to Proof-of-Work (PoW) in blockchain technology. In comparison to the PoW, PoS exhibits potential advantages in energy efficiency and scalability. Prominent examples of PoS and PoW are the cryptocurrencies Ethereum and Bitcoin, respectively. Next to both standards PoS and PoW, there also exists the Delegated Proof-of-Stake (DpoS) concept. The distinguishing factor of the DpoS consensus mechanism is that holders of the individual blockchain-based currency can vote for validators. Such a mechanism enhances a decentralized and democratic blockchain structure since validators with the most votes become delegates who validate transactions. The main weakness of the DpoS consensus protocol is that a group of delegates can potentially organize an attack more easily [13]. A voting mechanism is included in the Hyperledger FireFly blockchain. However, in comparison to Lisk, for example, Hyperledger FireFly does not stipulate the requirement of delegates within the blockchain at any point in time. Delegates are nominated by representatives who are chosen by each token holder, a stakeholder of the blockchain. The delegates can vote for representatives via a vote transaction while the blockchain network is running [14]. In conclusion, with the recent hype phase of blockchains, many consensus mechanisms have been developed. Despite the continuously enhanced consensus mechanism and proven security effectiveness of cryptocurrency, lack of trust and arising security issues due to shared resources persist [15].

In practice, most data live off-chain for privacy reasons [16]. The combination of lack of trust and the plausible sensibility of data shared via the BIS might explain the suspiciousness of supply chain actors in a blockchain environment, especially since plenty back-and-forth communications occur between organizations within multi-party blockchain systems. In the blockchain use cases, the actual blockchain development seems simpler than the off-chain part [16]. This will manifest with the rise of artificial intelligence (AI) for business process design. Furthermore, AI-supported developments will speed up track & trace solutions. Ultimately, the key impediment to realizing track & trace solutions is trust barriers among the supply chain network's organizations.

2.3. Connecting an ERP system with a blockchain platform

A successful proof of concept can help to solve the concerns about interoperability and trust concerns. Sislian and Jaegler [17] discovered that integrating Enterprise Resource Planning (ERP) and blockchain technology offers significant advantages for the companies they studied. The combined approach can enhance sustainable corporate performance and facilitate deeper integration within the supply chain [17]. Faccia and Pythagoras [18] spotted that the combination of ERP systems with blockchain platforms results in reduced procurement time and efficiency; increased IT automation and productivity; cost reduction; enhanced security, reliability, and safety; and improved flexibility and quality [18]. Several studies provide a theoretical discussion about the advantages of blockchain technology for product traceability within supply chains [16–18]. For instance, the pre-implementation guide written by Imane et al. gathers fourteen critical decisions that lead toward blockchain-integrated ERP systems [19]. The following three papers underscore the benefit of integrating ERP and blockchain technology. In the case of the Indonesian paper about halal product traceability, a smart contract checks the certificates of all meat processing supply chain stages [20]. The paper about food safety traceability systems highlights in its conclusion that blockchain frameworks ought to be assessed [21]. The input of the above studies is incorporated into the connection of an ERP system with a blockchain platform. An important milestone in the research is to actually apply the frameworks and guidelines that have been published in the realm of track & trace applications based on BIS. This study's novelty lies in its synthesis of key findings from three prior publications, thereby enabling the realization of an end-to-end track & trace system leveraging blockchain technology. The goal is to showcase that an ERP system, a system of records, can be connected to blockchain Hyperledger, a system of engagement. Hereby, the system of records is represented by an SAP S/4HANA Cloud system and the system of engagement is represented by Hyperledger FireFly. The arguments for choosing these two representative systems will be given in the next section in more detail. Establishing a connection between these systems is, however, not sufficient. The exchange of private messaging, tokenization, and sequencing of events needs to be demonstrated to address interoperability and trust issues. The majority of enterprise blockchain use cases require data to be able to sequence events properly in order to deliver these events reliably [16]. Tokenizing physical assets allows the peers in a blockchain to keep track of the movements of the assets.

This can be achieved in six steps:

1. Developing the business logic
2. Modeling the assets in a data model
3. Defining the process orchestration
4. Checking the plausibility and logic of smart contracts
5. Converting smart contracts to application programming interfaces (APIs) that are easy to use
6. Deploying smart contracts

The starting point for these six steps is the digital enterprise architecture (DEA) [22], which helps to develop the business logic of the track & trace solution.

The core of this architecture is formed by the three pillars business, enterprise, and network architecture. Each of these pillars needs to be considered in the track & trace solution. Considering Steps Two and Three of the six-step approach above, they are deep dives into the enterprise architecture pillar. The latter consists of the three building blocks data, technology, and processes (Figure 1). The building block "technology" has already been defined. The Hyperledger FireFly blockchain technology is chosen for the multi-channel track & trace solution. Then, for the second step, the data need to be modeled. This step represents the data building block. More details about the data model will be shared in the next section. For Step Three of the six steps towards creating tokenized physical assets, a process orchestration should be in place. Process orchestration coordinates and sequences diverse business activities, involving both automated and manual tasks. It is supported by the SAP Build Process Automation (SBPA) which is a service feature of SAP BTP. Onto BTP several SAP services can be added. A few of them that are relevant for the demonstration case will be detailed in the outline of the multi-layer blockchain track & trace system sample scenario. Fourth, the plausibility of smart contracts is assured in this example. Smart contracts are used for asset sharing. They permit standardized and protected asset sharing mechanisms enhancing the security level of sharing data among stakeholders [15]. Ideally, the logic of a smart contract should be explainable in one or two sentences.

In the case of the multi-layer track & trace blockchain concept, the logic is that – upon arrival or departure of a transportation unit – payment to the credited party is made.

This payment can be either destined to a forwarder who has shipped the transportation unit at hand or to the supplier of the transportation unit. In simple terms, a smart contract is a contract with a set of underlying rules documented in a computer program. The computer program is stored in the track & trace blockchain and automates the conditional transaction. The program is executed by the nodes that are part of the track & trace blockchain network [13]. The fifth and sixth steps are conducted by means of the Hyperledger FireFly software.

The conversion of smart contracts to APIs is one of the key features of Hyperledger FireFly. As depicted in Figure 2, it entails an API generator that can transform smart contracts written in common programming languages with open-source software (OSS) into an API definition. This API definition can then be integrated with other systems. Another feature of the Hyperledger FireFly system is that it supports building web3 apps. Web3 is the third evolvement step arising from www (static pages web) followed by web2 (social networking web). Web3 stands for ubiquitous computing web. It uses sophisticated intuitive queries that increase the need for data on demand significantly [23]. This in turn means that the computing capabilities need to adapt accordingly.

The deployment of smart contracts will be demonstrated in the following section. Simultaneously, the exchange of tokenized physical assets will be dealt with.

3. Research Methodology

3.1. Reasons for Hyperledger FireFly

Although the integration of web3 technologies with ERP systems is still in a preliminary stage, the development of a practical example showcasing smart contract validation within a multi-layered track & trace landscape is ongoing. This proof of concept is subject to ongoing changes, as the web3 integration to ERP systems is in an early stage of development. Yet, the structured approach to use the DEA to connect web3 technologies with existing enterprise systems can be applied independently of the development stage. This study contributes to the field of Industrial IoT by operationalizing a novel system architecture for blockchain-based track & trace applications. This achievement

Figure 1
Illustration of the digital enterprise architecture [22]

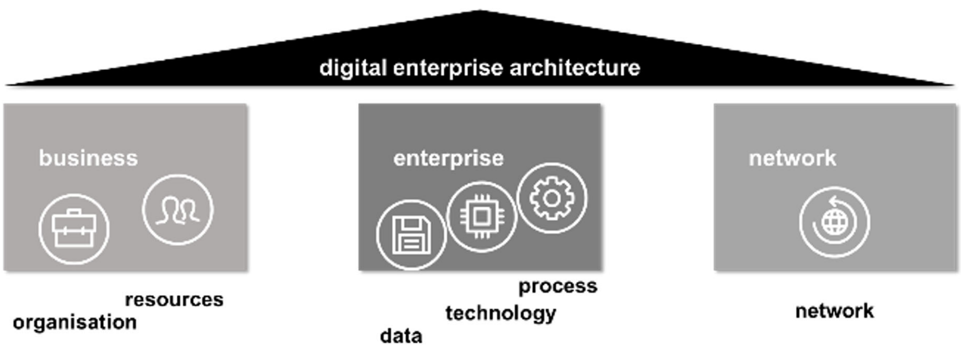
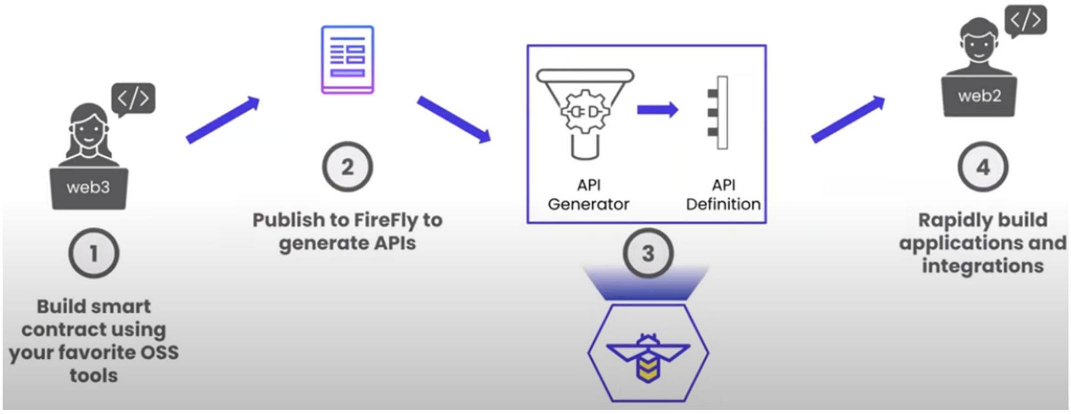


Figure 2
Hyperledger FireFly turns smart contracts into APIs [16]



represents a significant step towards realizing the practical application of this technology within supply chain management. In addition, it encourages to design and implement blockchain technology into logistics enterprise architectures.

For building up an example of a multi-layer track & trace system landscape with a smart contract validation instance, the example of an SAP BTP platform in combination with a Hyperledger FireFly blockchain instance was chosen. SAP is the largest European software company. It focuses on software, applications, and programs, as the abbreviation SAP suggests. The abbreviation BTP stands for BTP. In a complex multi-layer track & trace system landscape setup, it is essential to consider all components to create a successful demonstration case. From a distributed Hyperledger blockchain perspective, FireFly comprises supportive features alleviating interoperability and trust issues. Enterprises can build a complete stack and scale secure web3 applications with Hyperledger FireFly. It is an open-source supernode and gateway to a web3 blockchain ecosystem. A supernode is a set of runtime systems and connectors managed by the blockchain system.

Furthermore, Hyperledger FireFly is a pluggable API orchestration and data layer that integrates into all the different types of decentralized web3 [24].

- It is quick and easy for anyone, with or without prior web3 experience, to get started from scratch.
- It contains TypeScript web sockets and other powerful tools for experienced web3 developers.
- An entire developer stack can be run offline on a local device.

3.2. System setup on Hyperledger FireFly side

The US-based company Kaleido Inc. is marketing a product named "Hyperledger Firefly" for the purpose of connecting legacy systems to web3 applications. This product is provided as OSS. The company claims that it is the first open-source "supernode."

Although simplified, the setup of a Hyperledger Firefly environment requires several steps. One prerequisite is the installation of Docker and Docker Compose. Docker Compose is a tool for defining and running multi-container Docker applications [25]. The container software Docker allows for running the demo on the local device by creating Docker images. Then, the Hyperledger FireFly's CLI package is downloaded and installed. The next step is to set up the Hyperledger FireFly environment. It will run on Docker containers inside a virtual machine. This virtual machine enables the FireFly stack, including one supernode per blockchain peer, to operate in a productive environment. Potentially, every peer can be a different organization. In the simplified demonstration case, three organizations are part of the stack. One peer is the OEM, the second is the OEM's supplier, and the third is the OEM's customer. Once the stack was started, a user interface (UI) and a sandbox were created for each of the three blockchain peers.

3.3. System setup on SAP side

SAP's product portfolio is immense. For the track & trace solution based on SAP, a few prerequisites need to be in place. The SAP system is the enabler for the multi-layer track & trace system landscape rather than the core. For this reason, the validation of the track & trace system landscape is not performed for all the below-mentioned setup steps.

Foremost, a trial or licensed version of SAP BTP is required. In the second step, BTP needs to be linked to the ERP system. In this case, the widely used SAP S/4HANA Cloud system will be added to the BTP global account. For this purpose, a new system needs to be added to the respective system landscape in BTP. The Communication Scenario Group "Eventing Between SAP Cloud Systems" is selected, and the resulting Registration Token must be documented. It is the linking key to create a new extension in the SAP S/4HANA Cloud system. This procedure ensures that the SAP S/4HANA Cloud system is registered in the BTP cockpit. Further steps are the enterprise event enablement where the channel binding is configured. Both an Outbound Topic Binding and an Inbound Topic Binding are maintained. Furthermore, the SAP Event Mesh platform needs to be added to BTP. This can be conducted via the Service Marketplace. As a result, the Event Mash application should be part of the Instances and Subscriptions section of BTP. Regarding the BTP instances, two further entries need to be maintained. One is a Mesh Event service for the SBPA. The other one is an SAP S/4HANA Cloud Extensibility service, which is linked to the ERP system, in this case, the SAP S/4HANA Cloud system. Both instances will also be visible in the Message Clients section of the SAP event Mesh application. Then, a queue is created for the SBPA. While building upon existing SAP S/4HANA Business Events, the most suitable event is the Field Logistics Supplier Item. Its properties are detailed in the annex Table A1. From here, the Field Logistics

Supplier Item is referred to as supplier item. Currently, there are four operations established for the supplier item API: Read, Create, and Update Supplier Item as well as Action ProcessmilestoneEvent are available. The focus is on the operation ProcessmilestoneEvent. With this operation, a web3 technology transaction can be initiated via the track & trace blockchain system. If a supply chain peer in the blockchain has no SAP system access itself, the operations can be triggered by the Hyperledger FireFly instance, too.

3.4. Sample scenario – multi-layer blockchain track & trace system

Having outlined the two main ingredients for the demonstration case of the multi-layer track & trace system landscape with smart contract validation, these ingredients need to be connected. Figure 3 visualizes an exemplary process flow within the system landscape. It shows why the demo is a multi-layer track & trace system landscape.

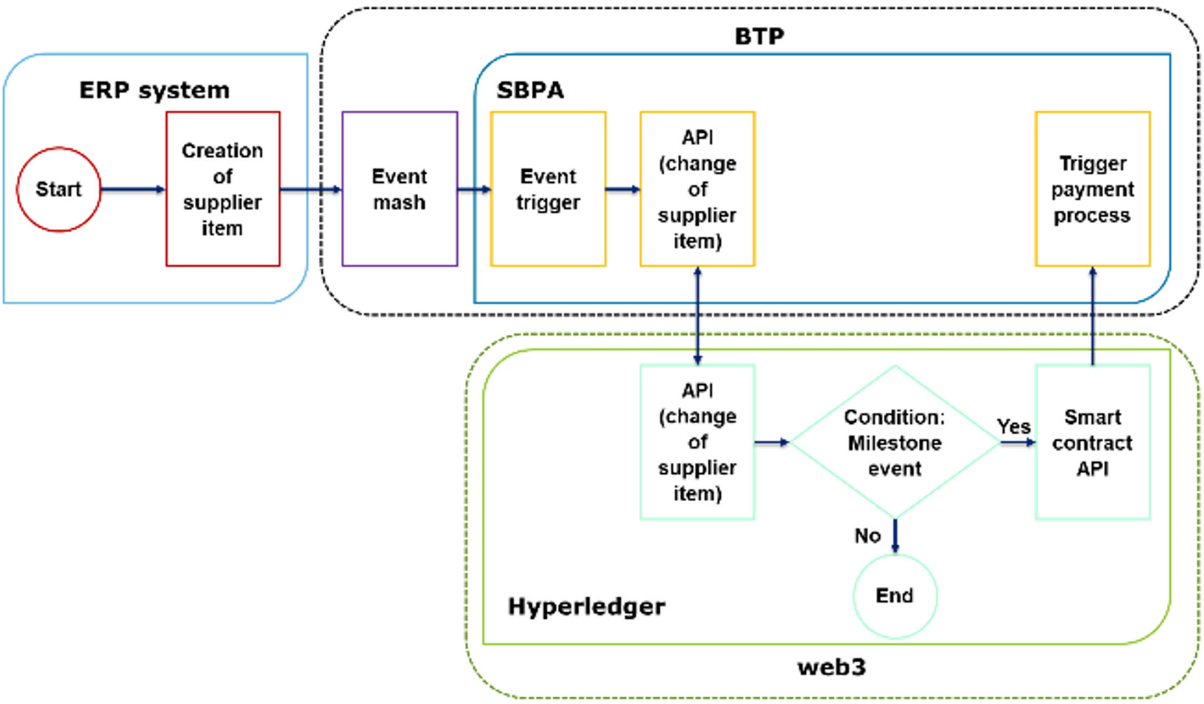
In the top part of Figure 3, the SAP landscape is depicted. On the left-hand side, the process flow starts with a shipping event that is initiated in the S/4HANA system. The assumption is that the OEM runs a widely used ERP system. However, the event mash service allows for connecting non-SAP systems as well. This is the advantage of adding event mash as an intermediary step [26]. The system-sided connections are detailed in the two previous subsections. Hence, the focus is on the sequence of the process steps. In the first place, the supplier item is created. It can originate from a demand placed by the OEM. However, it is not mandatory that the supplier item is created out of the ERP system. The supplier item API includes a "Create Supplier Item" operation. Also, other organizations in the track & trace system landscape can create a supplier item and notify, for example, the OEM's ERP system. This process is part of the diagram in Figure 3 to avoid misunderstanding. In SBPA, the supplier item will experience changes. These changes can either come from the OEM's S/4HANA system or from the Hyperledger Firefly. The Hyperledger can either receive changes in the supplier items by other systems that are connected to the web3 BIS or these changes can be maintained directly in the FireFly stack. For this purpose, Hyperledger FireFly offers a user-friendly UI. The changes can either be the operation Action ProcessmilestoneEvent or the operation Update Supplier Item. The Action ProcessmilestoneEvent constitutes a POST HTTP method, whereas the Update Supplier Item represents a PATCH HTTP method.

In the above sample scenario, the decision point is whether a milestone event has occurred. Thus, if the operation Action ProcessmilestoneEvent was posted via the API integration, a payment process is triggered. If not, for instance, an Update Supplier Item was patched, no follow-up process is triggered. In contrast to the bidirectional arrow for the API interface between the web3 blockchain application and the SAP BTP system, there is only a one-directional link between the smart contract and the trigger payment process. The reason is that the condition for the payment is defined in the Hyperledger stack. There, the involved peers in the web3 track & trace application can verify and approve the changes that are made to the blocks. If the conditions are met in the Hyperledger FireFly, the smart contract details are either shared via the API or via the SBPA, as depicted in Figure 3, or they are transferred to other systems that are connected to the track & trace web3 system.

4. Results

Having outlined the main components of the multi-layer track & trace system landscape, the three components of a system supported by web3 technology are described. In Hyperledger FireFly, they are represented by smart contracts, messages, and tokens.

Figure 3
Sample scenario – track & trace system landscape with smart contract validation instance



4.1. Smart contract exchange

Smart contracts embody executable code within the blockchain system. This program code can be defined with individual functions and conditions as part of the immutable data blocks. When the conditions are fulfilled, the blocks can be exchanged automatically [27]. As outlined in the previous section, this automation can be, for example, the exchange of monetary assets. Besides the transfer of assets, Hyperledger FireFly also allows for secured intercompany communication. A special type of communication is voting, which can be initiated by smart contracts. Hereby, Hyperledger FireFly has a built-in constraint that only one vote can be handed in per voting [16].

Integrating a smart contract into the multi-layer track & trace system landscape involves a series of steps. Initially, the smart contract must be authored in a programming language. Subsequently, after the completion of the writing process, the contract needs to undergo a compilation. This compilation necessitates the definition of a contract interface. In the demonstration case, the contract interface has the title "supplier item contract" and has version 2.0. The definition of the contract can be verified in the Hyperledger FireFly sandbox by running the respective interface format. For the demonstration case, a simplified FireFly interface is chosen. As a result, a message as well as a contract interface are confirmed, a blockchain event is received, and a transaction is submitted (see annex Table B1). This contract interface communicates to the corresponding external systems, which are run by the organizations being part of the blockchain environment. Regarding the demonstration case's system landscape, the Hyperledger is connected to the SBPA via BTP (Figure 3). Next, the contract API needs to be registered. For this purpose, a chaincode and a channel need to be assigned to the previously created contract interface. The chaincode and channel are named asset transfer, both non-technical and technical, and firefly, respectively. The relation between the channel and the chaincode is that the chaincode is executed in the blockchain channel. This relation can be verified on Hyperledger FireFly by running the application code (Figure 4).

The outcome is similar to what has been shown in the previous step (see annex Table B1). The demo stack can be run on the web3 Hyperledger FireFly platform. As a result, the address, where that smart contract lives, is retrievable. With this address, the contract interface can also be defined on any other posting platform, like the SAP Integration Suite. There, the smart contract can be invoked, and distinct functions can be called through restful APIs.

Figure 4
Application code of track & trace blockchain demonstration case

Multi-layer track and trace system landscape

```
const api = await firefly.createContractAPI(
  name: 'asset_transfer',
  interface: {
    name: 'supplier-item-contract',
    version: '2.0',
  },
  location: {
    chaincode: 'asset_transfer',
    channel: 'firefly',
  },
});
return { type: 'message', id: api.message };
```

4.2. Broadcasting messages

Another way to initiate events across system boundaries is by using web3 messages. Hyperledger FireFly enables users to broadcast messages either to all peers in the channel or to broadcast private messages. Upfront, the datatype needs to be defined. This is a prerequisite for all involved parties to broadcast a readable message and validate its JSON schema. If the message structure does not adhere to the predefined JSON schema, an error message hints at the nonconformity. In the demonstration case at hand, the name will be supplier item, the version is 2.0, and the JSON schema is depicted in Figure 5.

This results in a "202 Accepted" server response as displayed in Figure 6. Like the event recording of the contract interface (Annex B Table B1), a batch pin is generated (Figure 7). The information about the data type supplier item (2.0) can also be reviewed in the Hyperledger FireFly UI under the node Off-Chain > Datatypes.

Once the datatype has been created, the organizations in the blockchain can start sending messages to other organizations. The prerequisite is that the organization is represented with at least one node in the blockchain network. The message-issuing organization has either the possibility to broadcast the information about, for example, the arrival of a supplier item, to all blockchain peers or to specifically selected receivers. A party in the blockchain that was not named as a recipient gets the batch pin but cannot decode the hash, because it does not know which other parties are involved. Whether broadcast or private message, the data type can be selected and a tag as well as a topic can be added.

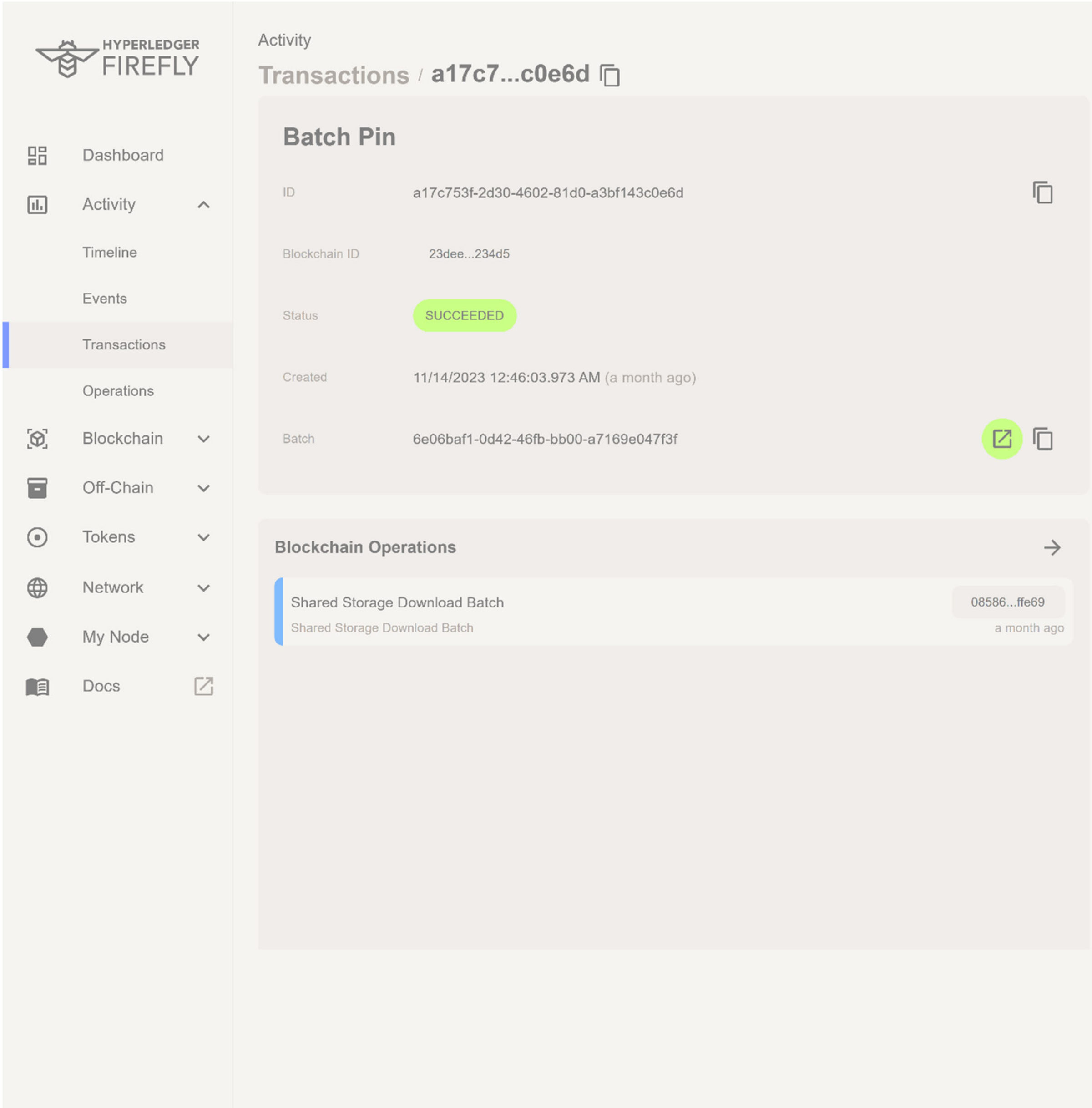
The tag indicates the purpose of the message to the application that processes it. The topic allows for associating the message with an ordered stream of data. Both fields are optional. Continuing with the plain message, the predefined datatype supplier item (2.0) is filled in as shown in Figure 8. The tags "arrival_at_hub" and topic "completed" are added to the message. The Hyperledger FireFly UI records the message details and can download the original message. Above that, the resulting blockchain information, as depicted in Figure 9, can be copied or downloaded. The tag "arrival_at_hub" can also be utilized to trigger the invoice process after receipt of the supplier item without the involvement of an SAP ERP system. This is a simplified alternative to the complex multi-layer system setup as depicted in Figure 3.

Figure 5
JSON schema of track & trace blockchain demonstration case

Multi-layer track and trace system landscape

```
{
  "title": "Field Logistics Supplier Item",
  "type": "object",
  "properties": {
    "FldLogsSuplrItemName": {
      "type": "string",
      "description": "Name of the delivered item"
    },
    "FldLogsSuplrItemSerialNumber": {
      "type": "string",
      "description": "The serial number is the number
      that is assigned by the manufacturer for an
      individual technical object"
    },
    "FldLogsSuplrItmPOItmUniqueID": {
      "type": "string",
      "description": "Item number of delivered item"
    }
  }
}
```

Figure 6
Hyperledger FireFly server response to a new JSON schema (code screenshot)



Yet, combining systems of records with a system of engagement, it is nowadays often required to apply a predefined system landscape. Despite that, filtering or searing an entry in the recorded blockchain transactions can also be accomplished with the SAP system. For investigating the history of blockchain transactions, the foundation is BTP. On top of BTP, the SAP Integration Suite is added to the subaccount which in turn is connected to the BIS. In the SAP Integration Suite, the path Configure APIs > Configure needs to be selected. Next, an API is imported via the respective button. The import works via a JSON file, which can be downloaded from the Hyperledger FireFly dashboard following the path blockchain > API.

In the Integration Suite, the API is listed as an externally managed API. The available API resources are depicted in the annex Figure C1.

4.3. Tokenized certificates

In addition to the smart contracts and the messages, Hyperledger FireFly features tokens. As an example, a purchasing department can use them to issue certificates. After a successful compliance review of a supplier, the original equipment supplier can mint a token to award the respective supplier. This adds value to the supply chain operations, because according to Chapter 2 of the EU-GMP Guidelines, the heads of Quality Assurance, Quality Control, and Production are responsible for monitoring and approving suppliers of materials [28]. Hyperledger FireFly enables attaching data to such a supplier certificate. Additionally, a token can be minted, and optionally a tag can be added. As a result, there is a message ID attached to the mint event that relates to the supplier certificate. In case the supplier does not pass a later audit,

Figure 7
Hyperledger FireFly server response to a new JSON schema – Events (screenshot)

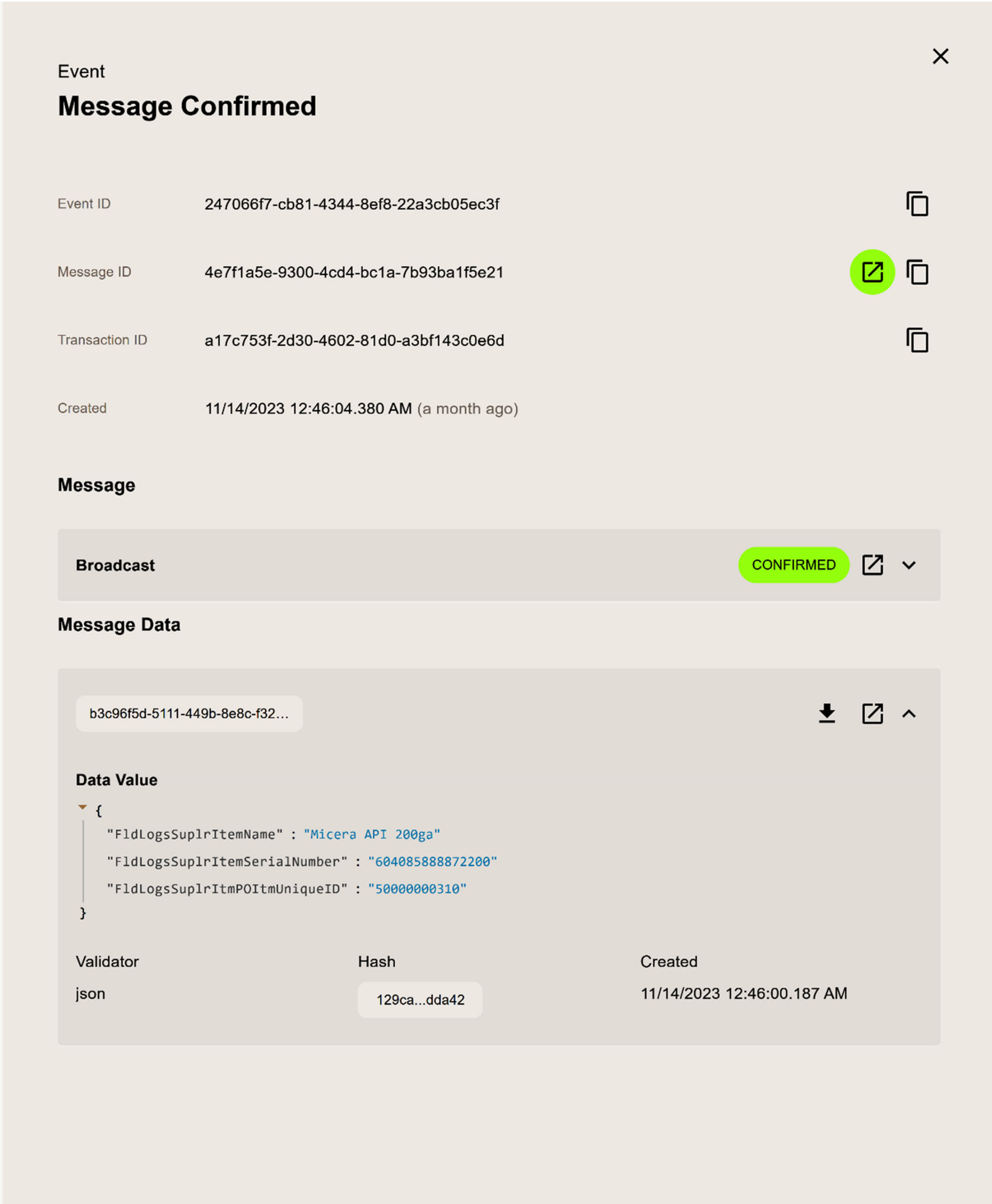


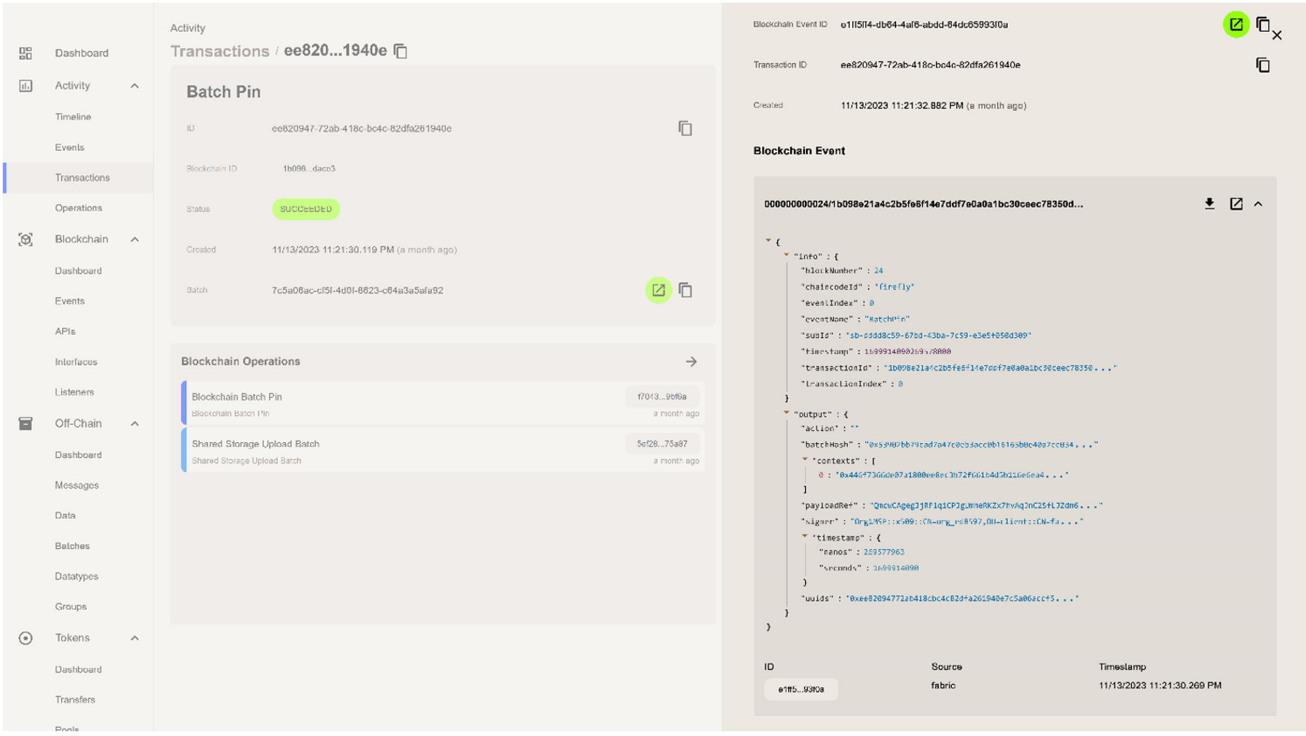
Figure 8
Supplier item message of track & trace blockchain demonstration case

Multi-layer track and trace system landscape


```
{
  "FldLogsSuplrItemName": "Micera API 200ga",
  "FldLogsSuplrItemSerialNumber": "604085888871200",
  "FldLogsSuplrItmPOItmUniqueID": "50000000220"
}
```


a CERTIFI-CATE_REVOKED tagged token can be issued. This will result in a burned token and indicate to the parties in the supply chain that the specific supplier is no longer certified. Next to non-fungible tokens, fungible tokens can be issued with Hyperledger FireFly. They can, for example, serve to award members in the supply chain for their services. These services can comprise products or information. In summary, tokens can be transferred, minted, or burned. Concerning track & trace applications, tokens offer a solution to reward and certification processes.

Figure 9
Message broadcast recording on the blockchain system (screenshot)



4.4. AI integration into blockchain-based track & trace systems

Although Kaleido Inc. currently is not offering any AI features integrated into either FireFly or its Backend-as-a-Service platform, there are several potentials for AI integration. Further exploration of how AI can be integrated into this system for enhanced functionality can be found in the following business areas. AI algorithms can be used to analyze data within the blockchain network and identify anomalies or suspicious activities that might indicate fraud attempts. This can help ensure the integrity and transparency of the track & trace data [29]. AI models can be trained on historical data to predict financial risks related to potential disruptions or delays within the supply chain. This information can be used to optimize logistics and prevent issues before they occur [30]. Concerning data enrichment and insights generation, AI can be utilized to analyze and interpret the vast amount of data collected within the track & trace system. This can generate valuable insights into various aspects of the supply chain, such as product movement patterns, inventory levels, and potential bottlenecks. Finally, AI can be used to personalize the customer experience by providing real-time updates on the status of their orders and proactively addressing any potential issues. This can lead to increased customer satisfaction and loyalty. For leveraging blockchain technology in AI-powered supply chain management, Tsolakis et al. propose a unified framework that captures the key data elements important to continued value delivery. They conclude that the combined implementation of blockchain technology and AI fosters sustainability and data monetization [31].

5. Discussion

The key novel contributions to the advance of knowledge and technical implementations are the system architecture for a multi-layer track & trace system landscape with smart contract validation, the integration of web3 technologies with existing enterprise systems, and the analysis of challenges and future directions. A structured approach to combine an ERP system with a Hyperledger blockchain platform to enable track & trace across system and organizational boundaries is presented. This is one of the first demonstrations of such a system in literature. In case the participating organizations have different blockchain systems in place, Westerkamp and Küpper [32] have developed synchronized smart contracts that enable instant read-only function calls for other applications deployed on the target blockchain. This approach mitigates current limitations in cross-chain communication and facilitates novel forms of interaction between contracts [32]. Furthermore, the paper shows how web3 technologies like smart contracts, messages, and tokens can be integrated with existing enterprise systems like SAP ERP to create new functionalities. This is a significant step forward in the adoption of blockchain technology in supply chain management. From a technological point of view, JSON APIs build the proposed connector to exchange data between different systems in the track & trace landscape. This is a flexible and interoperable approach that can be used with a variety of systems. The key challenges of implementing a multi-layer track & trace system are data ownership, trust, and network architecture.

The research implications are generalizability, improved track & trace applications, and enhanced functionality due to the integration of web3 technologies. Although the focus is on two distinct software providers, the results can be generalized to similar software, promoting broader applicability. By connecting their distinct applications, the interaction between a system of records and a system of engagement becomes more tangible. Thereby, the software provider of both systems can be replaced as long as they can be connected to each other. Integral to the research implication is that there exists a way to exchange track & trace data, originating from a logger device, and connect it to a block. JSON APIs represent the bond between these two different systems. Like other innovative technologies, blockchains in the supply chain undergo constant changes. For instance, SAP launched a Blockchain Business Connector in the third quarter of 2023. It is supposed to integrate web3 technology into existing business processes directly via Event Mash. Equally, novel business processes can feature blockchain values and smart contracts logic. This shows that after the first hype of blockchain applications at the end of the last decade, the plateau of productivity is supposed to arrive soon. As predicted by Furlonger and Kandaswamy in their 2018 Gartner analysis, blockchain and distributed ledger applications had to go through a trough of disillusionment [33]. Now, five years later, blockchain-based track & trace applications with an underlying smart contract validation instance have become more relevant. The use cases have been detailed and the knowledge about the different web3 technologies has become more widespread. One of the action steps is the continuous knowledge sharing of the blockchain web3 technologies. Combined with an outline of how ERP supply chain data can feature, for example, smart contract validation to automate the payment of suppliers, academics and practitioners can tackle the next hurdles. One of them is data ownership in a multi-layer track & trace landscape. Hereby it is indispensable that all organizations agree upon the consensus mechanism, data structure, and smart contract conditions for their blockchain track & trace solution [8]. Without trust in the network's security protocols, participants might hesitate to transact, hindering the network's functionality. Although Cryptography protects data, trust is crucial to ensure participants refrain from distributing data to unauthorized parties. Trust in technology is proven in practical examples [34]. Moreover, trust is enforced with the evolution of new blockchain technologies especially in the area of security [12]. Another hurdle to the successful implementation of a multi-layer track & trace landscape with a smart contract validation instance is the architecture of a blockchain network. In this regard, the type of nodes as well as the use of subnets should be investigated in more detail. In the context of supply chain logistics, the distinction between full knots and light knots must be made. Full knots serve as network nodes storing comprehensive transaction histories, while light knots, such as mobile devices, maintain only origin information. In private blockchains, participants typically have one full knot each, accompanied by light knot connections on mobile devices used in the track & trace operations.

The effective utilization of subnets, or private channels, allows for clear communication and improved efficiency. Layering, where hash values are selectively transmitted to the main net, enhances speed and data visibility. The strategic hosting of full knots internally, within the same firewall as the ERP system, ensures secure data handling. Subnets further enable controlled data sharing and limit payload differentiation, mitigating the risk of unauthorized access. Eventually, qualitative metrics such as processing speed and data visibility need to be balanced. On the one hand off-chain, or to a lesser degree, hybrid on-chain processing increases efficiency by optimally partitioning workloads between ERP and blockchains. On the other hand, all the off-chain processes will not be recorded in the block and visible in the corresponding ledgers. A further quantitative metrics issue to consider is queries that are made in the ERP system for block data. Depending on the number of peers and the number of blocks, it makes sense to apply pre-set filters so that blockchain platforms are not impacted. Future research can examine these practical problems. Innovations such as layered sharding should be considered [35].

6. Conclusion

While the logistics sector awaits a resolute and practical blockchain solution, it is emphasized that blockchain applications thrive within consortiums founded on strong agreements. The consensus mechanism, particularly in private blockchains, relies on PoS consensus mechanisms supported by authority certificates. Organizations in the network can be granted confirmation rights, fostering trust in technology through automation and automatization. Finally, valid data feeding into the blockchain are essential. As the block moves through subsequent full knots, each one verifies the block based on the assumption that the track & trace logger transfers correct values into the blockchain. The network's logic is defined by a smart contract agreed upon, ensuring the confirmation of origin, secure data distribution, and plausibility checks within the logic of the smart contract. This comprehensive approach establishes a robust foundation for the seamless and trustworthy functioning of the blockchain network in the supply chain ecosystem. Yet, while being a concept, several limitations apply. Foremost, the demonstration case is built upon freeware. More complex and, therefore, realistic scenarios can be built with full SAP access. These scenarios can, for example, comprise a supply network with several nodes and completed purchases and sales processes. These processes in turn can entail transportation events, inspections of goods, and payment blocks among other process steps. Instead of triggering a payment process by means of a smart contract, the demonstration case exchanges a web3 event to the BTP (Figure 3). Once released to the authors, the process can be validated by means of the novel Blockchain Business Connector. Also, instead of Firefly a Caliper or any other suitable Hyperledger software could be used. For instance, Caliper was used in a recent study by Valencia-Payan et al. [10] where a self-updating smart contract for supply chain traceability with data validation is described [10]. The combination of a BIS with AI is another subject that deserves further future research attention. Integrating AI solutions into smart contracts opens a wide array of enhanced use cases not only in the realm of track & trace. Zhang et al. [36] highlight that blockchain integration does not only add value to track & trace but also enhances the trustworthiness of AI-based applications [36]. In a recent study, the potential applications of blockchain and AI in enhancing supply chain coordination and mitigating associated disruptions are investigated [37]. This line of research warrants further investigation.

Conclusively, the conceptual paper leads the path from academic frameworks to real applications. The exchange of web3 events simulating the arrival of a purchase order item at a logistics hub paves the way for a multi-layer track & trace system landscape with smart contract validation instances. The research proposes a novel and simplified system architecture for track & trace using blockchain. The key finding is that blockchain with smart contract validation is possible with a few system components and that it improves track & trace applications. Integrating web3 technologies with established ERP systems fosters broader adoption of blockchain solutions within supply chain management. However, challenges like data ownership and network architecture need to be addressed for successful implementation.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in SAP at https://help.sap.com/docs/SAP_S4HANA_ON-PREMISE/671fe27096664c2c82ac6d2cabb38dea/74c6b169bfb74953b07a66aa248cd9aa.html?locale=en-US and https://help.sap.com/docs/SAP_S4HANA_ON-PREMISE/671fe27096664c2c82ac6d2cabb38dea/02b0ed279b5b430f9f48ad41c7e37f4f.html?locale=en-US

References

[1] Sunmola, F. T., Burgess, P., & Tan, A. (2021). Building blocks for blockchain adoption in digital transformation of sustainable supply chains. *Procedia Manufacturing*, *55*, 513–520. https://doi.org/10.1016/j.promfg.2021.10.070

[2] Mondal, S., Wijewardena, K. P., Karuppuswami, S., Kriti, N., Kumar, D., & Chahal P. (2019). Blockchain inspired RFID-based information architecture for food supply chain. *IEEE Internet of Things Journal*, *6*(3), 5803–5813. http://doi.org/10.1109/MNET.021.2000442

[3] Hasan, H. R., Salah, K., Jayaraman, R., Yaqoob, I., & Omar, M. (2020). Blockchain architectures for physical internet: A vision, features, requirements, and applications. *IEEE Network*, *35*(2), 174–181. http://doi.org/10.1109/MNET.021.2000442

[4] Barbosa, R., Santos, R., & Novais, P. (2023). Trust-based negotiation in multiagent systems: A systematic review. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, 133–144. https://doi.org/10.1007/978-3-031-37593-4_11

[5] Wang, G., Wang, Q., & Chen, S. (2023). Exploring blockchains interoperability: A systematic survey. *ACM Computing Surveys*, *55*(13s), 1–38. http://doi.org/10.1145/3582882

[6] Kotey, S. D., Tchao, E. T., Ahmed, A. R., Agbemenu, A. S., Nunoo-Mensah, H., Sikora, A., . . . , & Keelson, E. (2023). Blockchain interoperability: The state of heterogenous blockchain-to-blockchain communication. *IET Communications*, *17*(8), 891–914. https://doi.org/10.1049/cmu2.12594

[7] Gauthier, T., Dan, S., Hadji, M., Del Pozzo, A., & Amoussou-Guenou, Y. (2022). Topos: A secure, trustless, and decentralized interoperability protocol. *arXiv Preprint*: *2206.03481*.

[8] Körsgen, H., & Rabe, M. (2023). Blockchain subnets for track & trace within the logistics system landscape. In *International Conference on Innovative Intelligent Industrial Production and Logistics*, 165–181.

[9] Zheng, Z., Xie, S., Dai, H.-N., Chen, W., Chen, X., Weng, J., & Imran, M. (2020). An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems*, *105*, 475–491. https://doi.org/10.1016/j.future.2019.12.019

[10] Valencia-Payan, C., Griol, D., & Carlos Corrales, J. (2024). Blockchain self-update smart contract for supply chain traceability with data validation. *Logic Journal of the IGPL*. https://doi.org/10.1093/jigpal/jzae047

[11] Górski, T. (2021). The 1+ 5 architectural views model in designing blockchain and IT system integration solutions. *Symmetry*, *13*(11), 2000. https://doi.org/10.3390/sym13112000

[12] Gamal, I., Abdel-Galil, H., & Ghalwash, A. (2022). Osmotic computing enhancement based on blockchain approach. *Journal of Theoretical and Applied Information Technology*, *100*(13), 4643–4652.

[13] Copigneaux, B., Vlasov, N., Bani, E., Tcholtchev, N., Lämmel, P., Fuenfzig, M., & Frazzani, S. (2020). Blockchain for supply chains and international trade. https://doi.org/10.2861/957600

[14] Wicaksana, A., & Wira, J. C. (2022). Security analysis of private blockchain implementation for digital diploma. *International Journal of Innovative Computing, Information and Control*, *18*(5), 1601–1615. https://doi.org/10.24507/ijicic.18.05.1601

[15] Barenji, A. V., & Montreuil, B. (2022). Open logistics: Blockchain-enabled trusted hyperconnected logistics platform. *Sensors*, *22*(13), 4699. https://doi.org/10.3390/s22134699

[16] Lalamentik, A., & Cerveny, S. (2022). *Build sustainable supply chains with hyperledger FireFly and Kaleido*. Retreived from: https://www.youtube.com/watch?v=pGnYlUW5aLI

[17] Sislian, L., & Jaegler, A. (2022). Linkage of blockchain to enterprise resource planning systems for improving sustainable performance. *Business Strategy and the Environment*, *31*(3), 737–750. https://doi.org/10.1002/bse.2914

[18] Faccia, A., & Pythagoras, P. (2021). Blockchain, enterprise resource planning (ERP) and accounting information systems (AIS): Research on e-procurement and system integration. *Applied Sciences*, *11*(15), 6792. https://doi.org/10.3390/app11156792

[19] Imane, L., Noureddine, M., Driss, S., & Hanane, L. Y. (2023). Towards blockchain-integrated enterprise resource planning: A pre-implementation guide. *Computers*, *13*(1), 11. https://doi.org/10.3390/computers13010011

[20] Kusnadi, A., Arkeman, Y., Syamsu, K., & Wijaya, S. H. (2023). Designing Halal product traceability system using UML and integration of blockchain with ERP. *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, *9*(1), 29–41. https://doi.org/10.26594/register.v9i1.3045

[21] Singh, A., Gutub, A., Nayyar, A., & Khan, M. K. (2023). Redefining food safety traceability system through blockchain: Findings, challenges and open issues. *Multimedia Tools and Applications*, *82*(14), 21243–21277. https://doi.org/10.1007/s11042-022-14006-4

[22] Rabe, M., & Körsgen, H. (2023). Track and trace based on the digital enterprise architecture concept. In *2023 IEEE International Conference on Engineering, Technology and Innovation*, 1–7. https://doi.org/10.1109/ICE/ITMC58018.2023.10332433

[23] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645–1660. https://doi.org/10.1016/j.future.2013.01.010

[24] Hyperledger FireFly. (n.d.). *Orchestration engine*. Retrieved from: https://hyperledger.github.io/firefly/latest/overview/key_components/orchestration_engine/

[25] Bhat, S. (2022). Understanding docker compose. In S. Bhat (Ed.), *Practical docker with python* (pp. 165–198). Apress. https://doi.org/10.1007/978-1-4842-7815-4_7

[26] Acharya, W. *Business event triggers in SAP build process automation for SAP S/4HANA cloud: The customer return (BKP) use case*. Retrieved from: https://blogs.sap.com/2023/08/30/business-event-triggers-in-sap-build-process-automation-for-sap-s-4hana-cloud-the-customer-return-bkp-use-case/

[27] Hinckeldeyn, J. (2019). *Blockchaintechnologie in der supply chain. Einführung und Anwendungsbeispiele*. Germany: Springer.
[28] ECA Academy. (n.d.). *What are the GMP requirements for supplier qualification?* Retrieved from: https://www.gmp-compliance.org/gmp-news/what-are-the-gmp-requirements-for-supplier-qualification
[29] Mohammed, M. A., Boujelben, M., & Abid, M. (2023). A novel approach for fraud detection in blockchain-based healthcare networks using machine learning. *Future Internet*, *15*(8), 250.
[30] Natanelov, V., Cao, S., Foth, M., & Dulleck, U. (2022). Blockchain smart contracts for supply chain finance: Mapping the innovation potential in Australia-China beef supply chains. *Journal of Industrial Information Integration*, *30*, 100389. https://doi.org/10.1016/j.jii.2022.100389
[31] Tsolakis, N., Schumacher, R., Dora, M., & Kumar, M. (2023). Artificial intelligence and blockchain implementation in supply chains: A pathway to sustainability and data monetisation? *Annals of Operations Research*, *327*(1), 157–210. https://doi.org/10.1007/s10479-022-04785-2
[32] Westerkamp, M., & Kupper, A. (2022). SmartSync: Cross-blockchain smart contract interaction and synchronization. In *2022 IEEE International Conference on Blockchain and Cryptocurrency,* 1–9. https://doi.org/10.1109/ICBC54727.2022.9805524
[33] Hawlitschek, F., Notheisen, B., & Teubner, T. (2020). A 2020 perspective on "The limits of trust-free systems: A literature review on blockchain technology and trust in the sharing economy." *Electronic Commerce Research and Applications*, *40*, 100935. https://doi.org/10.1016/j.elerap.2020.100935
[34] Álvarez-Díaz, N., Herrera-Joancomartí, J., & Caballero-Gil, P. (2017). Smart contracts based on blockchain for logistics management. In *Proceedings of the 1st International Conference on Internet of Things and Machine Learning*, 1–8. https://doi.org/10.1145/3109761.3158384
[35] Hong, Z., Guo, S., & Li, P. (2022). Scaling blockchain via layered sharding. *IEEE Journal on Selected Areas in Communications*, *40*(12), 3575–3588. https://doi.org/10.1109/JSAC.2022.3213350
[36] Zhang, P., Ding, S., & Zhao, Q. (2024). Exploiting blockchain to make AI trustworthy: A software development lifecycle view. *ACM Computing Surveys*, *56*(7), 1–31. https://doi.org/10.1145/3614424
[37] Li, Z., Xu, H., & Lyu, R. (2024). Effectiveness analysis of the data-driven strategy of AI chips supply chain considering blockchain traceability with capacity constraints. *Computers & Industrial Engineering*, *189*, 109947. https://doi.org/10.1016/j.cie.2024.109947

How to Cite: Rabe, M., & Körsgen, H. R. (2024). Multi-Layer Track & Trace System Landscape with Smart Contract Validation Instances. *Journal of Data Science and Intelligent Systems*. https://doi.org/10.47852/bonviewJDSIS42023384

Appendix

Table A1
Properties of supplier item

Data field	Description	Field status
FldLogsSuplrItemName	Supplier item description	Mandatory
FldLogsSuplrItemSerialNumber	Supplier item serial number	Mandatory
FldLogsSuplrItmPOItmUniqueID	Concatenated field representing purchase order and purchase order item	Mandatory
FldLogsContainerID	Container identifier	Optional
FldLogsExtWorkerIdentifier	Supplier Item External Identifier	Optional
FldLogsSuplrItemDngrsGdsCat	Dangerous goods identifier – concatenated field representing dangerous goods class and compliance requirement	Optional
FldLogsSuplrItemGrossVolume	Supplier item volume	Optional
FldLogsSuplrItemHeight	Supplier item height	Optional
FldLogsSuplrItemLength	Supplier item length	Optional
FldLogsSuplrItemMeasureUnit	Unit of measure of the dimension	Optional
FldLogsSuplrItemPlndDelivDate	Supplier item planned delivery date	Optional
FldLogsSuplrItemQuantityUnit	Unit of measure of the quantity	Optional
FldLogsSuplrItemType	Supplier item type	Optional
FldLogsSuplrItemWidth	Supplier item width	Optional
FldLogsSuplrItmGrossVolumeUnit	Unit of measure for volume	Optional
FldLogsSuplrParentItemUUID	Supplier item parent identifier	Optional
FldLogsSupplierItemSource	Supplier Item Origin	Optional
ItemIsDangerousGood	Flag representing dangerous goods	Optional
ProductGrossWeight	Supplier item gross weight	Optional
ProductWeightUnit	Unit of measure for weight	Optional
EWMStorageBin	Storage bin identifier	Read-only
FldLogsActlSrvcEndDate	Actual service end date	Read-only
FldLogsActlSrvcStartDate	Actual service start date	Read-only
FldLogsIsWorkOrderOnHold	Flag representing work order on hold	Read-only
FldLogsLoadingPointID	Supplier item loading point identifier	Read-only
FldLogsPlndSrvcEndDate	Supplier item service end date	Read-only
FldLogsPlndSrvcStartDate	Supplier item service start date	Read-only
FldLogsRemotePlant	Remote plant to which supplier item has to be transported	Read-only
FldLogsStorageLocationID	Supplier item storage location identifier	Read-only
FldLogsSuplrBasePlant	Base plant to which supplier item to be received	Read-only
FldLogsSuplrItemQuantity	Supplier item quantity	Read-only
FldLogsSuplrItemRateUnit	Unit of measure of the rate	Read-only
FldLogsSuplrItemStatusID	Status identifier for supplier item	Read-only
FldLogsSuplrItemUUID	Supplier item identifier	Read-only
FldLogsSuplrItemVislInspCode	Supplier item visual inspection code	Read-only
FldLogsSuplrItmChangeDateTime	An administrative field, which holds the last updated date and time of the supplier item entry in the system	Read-only
FldLogsSuplrItmGRDocID	Concatenated field representing material document and year	Read-only
FldLogsTranspContainerID	Transport container identifier	Read-only
FldLogsUnloadingPointID	Supplier item unloading point identifier	Read-only
MaintenanceOrder	Maintenance order identifier	Read-only
PreferredSupplier	External supplier identifier who will provide the service	Read-only
Project	Project ID	Read-only
ProjectDemand	Project demand ID	Read-only
PurchaseContract	Purchase contract identifier	Read-only
PurchaseRequisition	Purchase requisition identifier	Read-only
PurgDocNetPriceAmount	Supplier Item Rate	Read-only
FldLogsSuplrItmPackggTypeCode	Supplier item package type	Supplier item type

Table B1
Verification of Hyperledger FireFly contract interface definition

Data field	Description
Message Confirmed	Time: 11:11:42.908 PM Message ID: 500932ca-a1a9-4c6c-8515-ff5707c9153f Message Type: definition Tag: ff_define_ffi Topic: 164c7e2586371b4fdbcc187a27f1eb6a81a754682d7399765bfcdaf87a251fd0
Contract Interface Confirmed	Time: 11:11:42.908 PM Interface ID: 42158394-493a-4cd5-8277-14ebd627af19 Name: supplier-item-contract Version: 2.0
Blockchain Event Received	Time: 11:11:42.850 PM Blockchain Event ID: e7d90911-96e0-44d7-bd92-2894c8ea41c2 Name: BatchPin Protocol ID: 000000000013/72c5d9bffaf44b34f3dd6f684803ce66dbd2f0aa66c0a84387627dfd2cffba54
Transaction Submitted	Time: 11:11:39.546 PM Transaction ID: b1a614e4-db44-45ef-a3d9-56464a00ef36 Type: batch_pin
Message Confirmed	Time: 11:11:42.908 PM Message ID: 500932ca-a1a9-4c6c-8515-ff5707c9153f Message Type: definition Tag: ff_define_ffi Topic: 164c7e2586371b4fdbcc187a27f1eb6a81a754682d7399765bfcdaf87a251fd0
Contract Interface Confirmed	Time: 11:11:42.908 PM Interface ID: 42158394-493a-4cd5-8277-14ebd627af19 Name: supplier-item-contract Version: 2.0
Blockchain Event Received	Time: 11:11:42.850 PM Blockchain Event ID: e7d90911-96e0-44d7-bd92-2894c8ea41c2 Name: BatchPin Protocol ID: 000000000013/72c5d9bffaf44b34f3dd6f684803ce66dbd2f0aa66c0a84387627dfd2cffba54
Transaction Submitted	Time: 11:11:39.546 PM Transaction ID: b1a614e4-db44-45ef-a3d9-56464a00ef36 Type: batch_pin
Message Confirmed	Time: 11:11:42.908 PM Message ID: 500932ca-a1a9-4c6c-8515-ff5707c9153f Message Type: definition Tag: ff_define_ffi Topic: 164c7e2586371b4fdbcc187a27f1eb6a81a754682d7399765bfcdaf87a251fd0
Contract Interface Confirmed	Time: 11:11:42.908 PM Interface ID: 42158394-493a-4cd5-8277-14ebd627af19 Name: supplier-item-contract Version: 2.0
Blockchain Event Received	Time: 11:11:42.850 PM Blockchain Event ID: e7d90911-96e0-44d7-bd92-2894c8ea41c2 Name: BatchPin Protocol ID: 000000000013/72c5d9bffaf44b34f3dd6f684803ce66dbd2f0aa66c0a84387627dfd2cffba54
Transaction Submitted	Time: 11:11:39.546 PM Transaction ID: b1a614e4-db44-45ef-a3d9-56464a00ef36 Type: batch_pin

Table C1
View resources in SAP Integration Suite for API supplier item (screenshot cutout)