Modelling blockchain-based Supply Chain applications: Understanding Re-occurring Patterns from Case-studies and Literature Perspectives

Abstract—**Integration of blockchain technologies in supply chains (SC) provides the possibility for improved transparency, efficiency and data interoperability in inter-organizational processes that occur within SC processes. This ensures that SC managers have complete visibility of their multi-organizational business processes, further providing real-time access to data for important business decisions. Still designing, developing, and deploying blockchain applications for various SC operations remains a difficult and complex task, due to the technical requirements of blockchain applications and business requirements of supply chain operations. Hence, the goal of this research is, first, to identify these common components and secondly, to understand the smart contract-based verifications carried on data generated from these blockchain-executed components. To achieve this, a combination of case studies and systematic literature review is used as a research method for identifying common re-occurring patterns in SC applications that are deployed on blockchains. The results of this work will provide input in developing a model-driven framework for building blockchain applications for various SC use cases**

Keywords— blockchain applications, Supply Chains, Modelling Patterns

I. Introduction

Blockchain has shown a potential to enable transparency, improve efficiency and enable data interoperability in inter-organizational processes and collaborations [1]. Blockchain is a peer-to-peer network that consists of blocks of transactions, distributed across the participating nodes, and cryptographically linked to ensure the immutability of data stored [2]. SC processes commonly consist of operations such as procurements, delivery, transportation, invoicing and payments involving multiple parties, stakeholders and organizations [3]. Such processes represent interesting use cases where blockchain and related technology can be applied to improve transparency, interoperability and business efficiency. Hence, ensuring that SC managers have complete visibility of their multi-organizational business processes, further providing real-time access to data for important business decisions. Governance and management of traditional information systems that support collaborative business processes is highly centralized, and the processes are opaque and siloed, thereby preventing data interoperability between collaborating organizations and parties [4].

Still deploying blockchain applications for various SC operations remains a difficult and complex task. Several blockchain types exist, ranging from private, consortium blockchains such as Hyperledger Fabric and public networks such as Ethereum and Polygon [5]. Each of these blockchain networks provides a specific set of computer programs referred to as smart contracts for executing immutable functions on such networks [6]. Smart contracts are commonly written in Solidity, Go, Rust, JavaScript and Typescript programming languages [7,8]. Developing and deploying blockchain applications for SC use cases requires extensive knowledge of business operations within and a strong understanding of blockchain technology stacks for realising such use cases. Thus, there is a need for patterns for defining commonly encountered SC operations that are executed on the blockchain. With the application of model-driven design concepts, such patterns can easily be used to realize smart contract representation of the defined inter-organization business operation in an automated or semi-automated manner [9].

Related work on design patterns and model-driven approaches for the generation of smart contracts are presented as follows. The paper [10] describes an automated generation of smart contracts for the Agric business in the Ethereum blockchain using model-driven design for Agricultural food traceability use cases. The paper [11] proposes a domain-specific design pattern for the automatic generation of smart contract codes. The paper [12] proposes the DAOM framework that describes a modelling process for building blockchain applications for inter-organizational use cases. The research work [13] describes a BPMN-based dynamic modelling system for smart contract generation in building blockchain applications for scheduling movements of physical goods in an SC. The paper [14] describes a solution canvas for building blockchain applications for physical SCs. The canvas specifies technological constraints, decision queries for verifying business conditions such as tracking product status, and information resources for representing assets within the supply chain. Other properties in the canvas specify the users, costs and application objectives. The paper [15] describes a reference model for building blockchain applications in monitoring the logistics of goods in a physical supply chain. The reference architecture covers supply chain operations such as delivery scheduling, collection, delivery, handover and evaluation of the logistic business operations.

Still, there is an absence of design patterns and model-driven approaches for the automated generation of smart contracts for SC use cases such that the commonly occurring objects and components in SC processes are represented with these patterns. While the paper [15] covers some of the components that occur within the supply chain such as product order, delivery schedules, and product handover, however, none of the articles systematically specified the properties of the commonly occurring supply chain components, the on-chain methods that result in a status change of the component on the blockchain and the smart contract verifications that can be used to check specific business conditions captured in the component properties.

The focus of this paper is, first, to identify the common components in SC applications on the blockchain and secondly, to understand their properties such as the on-chain methods and smart contract verifications carried on them. Thus, the results from this paper will provide the basis for further development of model-ling concepts for automated realization and deployment of smart contracts for SC operations that are applicable generally in supply chains. To achieve the objective of this paper, two main research questions are derived to guide the work carried out in this paper. The first research question is What are the common components of blockchain-based SC applications? The second is What are the on-chain methods and smart-contract-based verifications that apply to the identified components? The third question is what is a generalized model for representing blockchain-based SC applications that specify the important components, their properties and relationships. The rest of this paper is structured as follows. Section II provides the literature definitions of blockchain concepts that occur in this paper as well as the research methodology adopted for this paper. Section III answers the first and second research questions by the combination of case studies and literature review results. Section IV provides the answer to the third question by describing a meta-model for representing supply chain applications on the blockchain. Section V provides relevant discussions from this work. Section VI provides the conclusions and future works.

II. Background and Research Method

A. Blockchain Technology

The main blockchain concepts that are contained in this paper include block-chain networks, public key cryptography (PKC), on-chain data storage, smart contracts and token concepts. These concepts are presented in the context of inter-organizational collaborations as represented by various processes in SCs.

Blockchain networks and consensus mechanisms: The blockchain network consists of peers and nodes that represent various organizations that perform functions in blockchain business collaboration execution. Transactions are redundantly recorded on peers representing the organizations in the network such that every peer has the same state of the blockchain [16]. Blockchain networks can be largely classified into private and public networks. The former requires permission to join while anyone can join in the latter [16]. Transactions are validated before they are accepted in the network using various consensus methods. Proof-based consensus methods such as proof of work and proof of stake and their various adaptations are commonly used in public blockchains. Private blockchains commonly use voting-based consensus methods represented by different adaptations of byzantine fault-tolerant systems [17].

PKC: Transactions stored in the blockchain are digitally signed representing the users that executed the function that resulted in the transaction. The PKC consist of key pairs, private and public key. The private keys are used for signing transactions, while the public keys (which are derived from private keys) are used for identification. The user's address in the blockchain is a hash representation of their public keys [18].

Smart contracts: The computer programs that run on the blockchain are generally referred to as smart contracts. Hence, various inter-organizational operations and business logic can be encoded as smart contracts, and executed in a trustable manner without relying on a central authority to manage such collaborations. A specific blockchain application that enables collaboration between business organisations can contain several smart contracts [19].

Cryptocurrencies and tokens: Cryptocurrencies provide incentives for the participants in the blockchain network to perform certain activities such as validating transactions, organizing transactions in blocks and storing them on the blockchain. Users are also required to pay certain fees in cryptocurrency before their transactions are validated and stored on the network. These are common in public networks. Several smart contract applications can exist in a blockchain, hence, tokens are mostly used to provide value for executing functions within a particular blockchain application [20]. Still, other types of tokens can also exist in a blockchain application – tokens that provide governance functionality, non-fungible tokens that provide a unique representation of assets in a blockchain application etc. [20].

B. Research Method

This research follows the combination of case studies and a systematic literature review. Two case studies that outline various use cases of blockchain in different SC operations are used to identify the initial components and verifications that occur in SC-based blockchain applications. These initial results are generated after extensive business understanding workshops in research projects involving partnering organisations. Figure 1 summarizes the steps for this research such that results from related literature work and case studies adopted for this work are combined to generate a model representation of supply chain components and smart contract verifications developing blockchain-based SC applications. The steps applied in this research work are further explained below.

1) Case Descriptions

Here we present the two cases used to generate the initial inputs for this research. The early phases of the projects presented in the case studies involve extensive workshops and interviews to specify the project's initial requirements, hence, specifying the SC components to be stored on the blockchain and the necessary verifications to be carried out on these components.

Case 1: The first case study for this research focuses on the design and implementation of an ordering and invoicing verification system for a concrete delivery SC using blockchain technologies. The project is a research collaboration between an industry partner in this business domain and the Austrian Blockchain Centre (ABC) Research. The main objective of the project is to automate the obligation verification that ensures that the business conditions specified in the customers' orders are duly satisfied by their suppliers. The project furthermore ensures that the customers are not over/under-invoiced by comparing the surcharge business conditions and the actual delivery transactions recorded on-chain.

Case 2: This case focuses on prototyping a blockchain-based SC management system for oil delivery by improving the existing processes. The project is also another research collaboration between industry partners in the oil industry business domain and the ABC Research. The objectives of the project are to explore the application of blockchain technologies in digitizing the purchase-to-stock (P2S) and purchase-to-pay (P2P) processes in fuel procurement. The former covers the ordering, transportation and stocking of petrol in petrol stations, while the latter covers the payment settlements between the organizational entities in the P2S and P2P processes for the purchase and delivery of petrol.

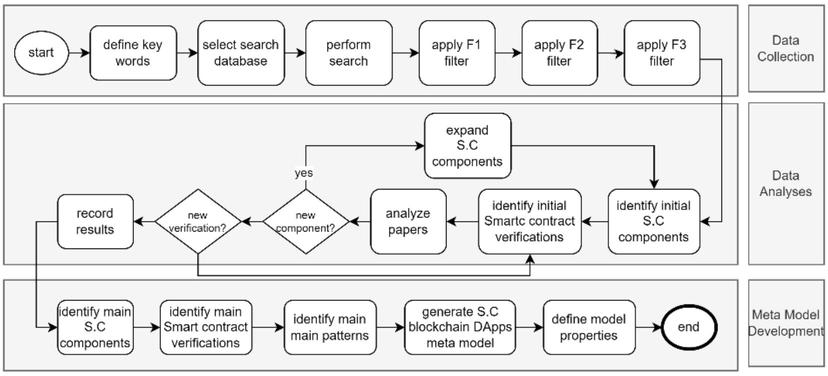

Figure 1: Research method

2) Data Collection from Literature

A systematic literature review is performed in this work to identify patterns in SC applications on the blockchain with the initial results from the case studies. As shown in the first layer of Figure 1, the steps in the literature review include keyword definition, database selection, data search, and application of filters.

Keywords: The search string used in finding the relevant articles is the combination of two strings. The first string contains the following keywords "blockchain + smart contract + logistics" and the second one "blockchain + smart contract + supply chains".

Database: The Scopus database is the selected database for our research. Quality academic publications from different publishing houses are indexed in Scopus, thereby, providing a good representation of the state of the art on the topic for this research.

Data Search: A search is performed with the defined keywords on the Scopus database. The search resulted in 1609 articles, 301 from the first search string and 1,308 from the second string.

Filters: Three filters are applied to generate only relevant papers and eliminate papers that may not provide useful input in the analyses conducted in the later stage of the research.

Filter 1: We applied filtering criteria such as Journal papers that were published from the year 2020 and above. Hence, ensuring that only recent high-quality publications are returned for further analyses. High-quality papers that provide a prototype implementation of blockchain applications are mostly published in journals. Applying filter 1 and removing duplicates resulted in 474 papers.

Filter 2: The purpose of this filter is to identify papers that implemented a prototype SC application on the blockchain. We used the following keys to search the abstract of papers from the previous filtering step "develop", "implement", and "prototype". If any of these keywords appear in the article, then we consider the paper relevant for the next step. The filter resulted in 331 articles.

Filter 3: the purpose of this filter is to identify papers that implemented SC use cases that involve physical products. The following keywords are used in the filtering "develop", "implement", "prototype", "goods", "product", "commodity", "material", and "items". The filter is applied in the abstract of papers from the previous stage, resulting in 129 articles.

3) Data Analyses

As shown in the second layer of Figure 1, analysis is conducted on the collected data to identify patterns in collected papers that capture the supply chain components and verifications performed on them. The SC components and smart contract verifications from the two case studies provided the initial bases for analyzing the papers. If an additional component or smart contract verification is identified in any of the papers, the list of the pattern properties and their occurrence in the other literature works are recorded. The 129 papers from the data collection phase are manually analyzed to categorize them into different SC types, blockchain types, SC components, smart contract verifications and their year of publication. After completing this step, 87 papers were found relevant for the analysis.

4) Model development

The main result of this paper is meta-model development that represents the aggregated SC components and smart contract verifications that be applied in modelling similar blockchain applications. The results of the analyses from the previous stage are presented as graphs representing patterns and trends of the various properties used in the analyses which are then used as inputs for developing the meta-model as shown in the last layer of Figure 1. The meta-model in general outlines all the relevant SC components, their relationships, on-chain methods that are executed on them, and smart-contract verifications performed on them to validate specific business conditions.

III. Patterns in Blockchain-enabled SC Applications

This section provides the main results that are generated by following the research method outlined in the previous section. First, we show in section III.A the results from the case studies which provided inputs for the literature analysis results in section III.B.

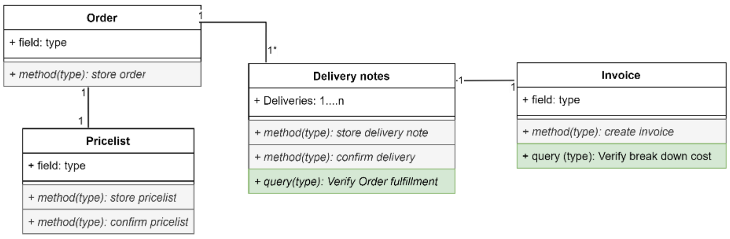

Figure 2: Components and smart contracts verifications in Concrete delivery blockchain application

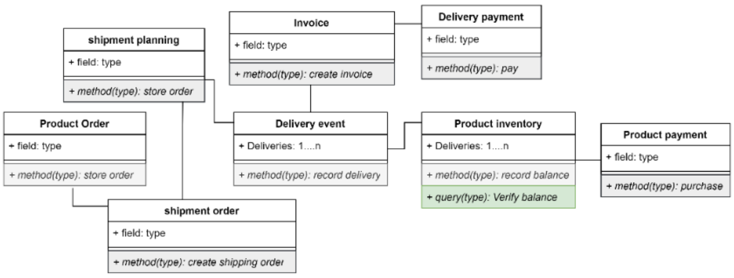

Figure 3: Components and smart contracts verifications in Petrol delivery blockchain application

A. *Initial Results from Case Studies*

Figures 2 and 3 outline the SC chain components that are recorded on the blockchain and the smart contract verifications that occur within the delivery process. The figures are color-coded. The grey property in the SC components captures the methods (functions) that result in the status change of the blockchain. The green property shows built-in queries that verify business conditions by checking the SC. components stored on the blockchain.

1) Concrete delivery case

In this business case of applying blockchain technology in a concrete delivery SC, Figure 2 shows the applicable SC components and smart contracts verification that occur in the delivery process. The product Orders contain customer requirements about a specific product. The Pricelist document specifies the costs for calculating the final price of the delivered product. The Delivery notes outline the actual product deliveries that occur as a result of the order. The invoice is calculated by summarizing the deliveries recorded in the delivery notes.

Two smart contract verifications occur in the concrete delivery SC, the first is to verify the order fulfilment, hence, checking if the product quantity and quality specifications captured in the order document have been fulfilled. The second verification is the breakdown cost, which helps the receiver of the product verify that the final price in the invoices matches the cost conditions within the pricelist and the deliveries that occurred.

2) Petrol delivery case

In the petrol delivery case as captured in Figure 3, the following SC components are stored on the blockchain. The order specifies the amount and the type of product wanted by a receiver, the shipment order and shipment plan help in specifying the delivery details and timeline. The delivery event records the actual deliveries that occur as a result of the product order. The delivery event results in issuing an invoice to the product receiver, which specifies the cost of the product order and shipment. The payment details for the received product are defined in the delivery payment. The product inventory keeps track of the product level at the side of the receiver. Based on the product availability, the consumers purchase a product and the payment details are specified in the product payment.

The goal is to provide transparent and traceable access to product status by the members of the SC. However, only one smart contract verification occurs in the delivery process, which is balance verification which checks the received product against the order while also considering the amount of products that are still available in the inventory storage.

B. *Modelling Patterns from Literature*

The initial results from the case studies provided inputs for analyzing literature to identify patterns of SC. components as well as smart contract verifications that occur in research works of SCs of physical goods deliveries. First, we show the summary of collected data and later, the analyses of the results to identify similar patterns that occur in these works.

1) Data collection results

Table 1 shows the summary of the results obtained by outlining the various SC types that occur in the literature and the blockchain networks the prototyped applications were developed for. Eight (8) SC types including agriculture, automotive, physical commodities, construction, energy, general logistics, healthcare, manufacturing, mining, reverse logistics, online marketplace, and product warehousing services. The blockchain networks in which these applications are prototyped include, Ethereum, Hyperledger fabric, Quorum and custom blockchains. However, some literature works didn't specify the actual blockchains for their prototypes. Table 2 shows the summary SC components and smart contract verifications that occur in these literature works. For each of the relevant research work, the SC components are specified either in the following - conceptual description of the prototype, the sequence diagrams of interactions of the components, pseudo algorithms of the prototype or in the actual smart contract codes of the implemented prototype. The smart contract verifications are qualitatively checked to identify prototypes that implemented interfaces for interacting with on-chain codes to verify specific business conditions.

The complete Table of results representing the collected data used in this research is available on the project link[1] for this research work.

The components include Offer, Purchase Order, Product Creation, Product Listing, Truck Schedule, Shipment order, Transportation notes, Delivery confirmation, Invoices, Payments settlements, and Product Payment. The offer document specifies the conditions intending customer wishes to purchase a product. The supplier can accept or reject an offer. The order captures the product type and quantity a customer wishes to purchase based on agreed price conditions. Product creation involves a series of activities that result in the digitization and identification of a physical

[1] Data link deleted due to double-blind peer review –

product on the blockchain. The product list is a mechanism for showing an intended customer product availability. The truck schedule outlines the shipment planning for product deliveries across various locations. The shipment order specifies the conditions and time frame for delivering an ordered product. Transportation notes record information about a delivered product either in real-time or a summary of delivery transportation. Delivery confirmation captures the activities that result in a customer taking ownership of the ordered product. Invoices specify the costs of product orders and can also include separately, the cost of the shipment. The payment settlements capture the business conditions for making payments within the SC of the product delivery. Lastly, the product payment involves actual payment of the order product either by traditional banking settlement or using cryptocurrency.

In the evaluated literature works, the following possible smart contract verifications were captured in the prototypes - Breakdown costs, Order fulfilment, On-time delivery, Shipment status, Payment confirmation, Quality control, and Authenticity. The breakdown cost verifies the price in the invoice based on the recorded events that occur during the delivery to check if the final price in the invoice is correct. Order fulfilment checks the requirements specified in the order (such as product types, quantity etc.) have been fulfilled after all the deliveries have been confirmed. On-time delivery checks if the product arrived in the specified time frame. The shipment status tracks the state of the shipment (either in real-time or summarized status) to verify if the shipping conditions are satisfied. The payment confirmation verifies if the payment for the condition for a particular order has been fulfilled. The quality control verifies the properties of the received product while considering attributes such as the certificates of the suppliers, raw material locations etc. The authenticity verifies the genuineness of the product by confirming for instance digital signatures associated with the product during its creation.

Table 1: Literature Data Collection Summary – SC Types

SC Type	BC Type	Count of SC Type
Agriculture	Hyperledger fabric	12
	Ethereum	12
	(blank)	1
	Custom	1
Agriculture total		**26**
Automotive	Custom	1
	-	1
Automotive total		**2**
Commodity	Hyperledger fabric	3
	Ethereum	2
	(blank)	1
Commodity total		**6**
Construction	Ethereum	1
Construction total		**1**
Energy	(blank)	1
Energy total		**1**
General logistics	Ethereum	19
	(blank)	3
	Hyperledger fabric	2
General logistics total		**24**
Healthcare	Ethereum	9
	Hyperledger fabric	4
	(blank)	1
Healthcare total		**14**
Manufacturing	Ethereum	3
	Quorum	1
	Hyperledger fabric	1
Manufacturing total		**5**
Market place	Ethereum	2
	Hyperledger fabric	1
Market place total		**3**
Mining	Hyperledger fabric	1
Mining total		**1**
Reverse logistics	Ethereum	2
	(blank)	1
Reverse logistics total		**3**
Warehousing	Ethereum	1
Warehousing total		**1**
Grand total		**87**

Table x: Literature Data Collection Summary – SC Components and Smart Contract Verifications

	Property	**count**
SC components	Offer	3
	Purchase order	21
	Product creation	63
	Product listing	20
	Truck schedule	1
	Shipment order	5
	Transportation notes	45
	Delivery confirmation	44
	Invoices	3
	Payments settlements	22
	Product payment	23
Smart contract verifications	Breakdown costs	1
	Orders fulfilment	6
	On-time delivery	1
	Shipment status	13
	Payment confirmation	5
	Quality control	35
	Authenticity	45

2) Data Analysis Results

The goal of the analysis performed is to identify the main patterns that occur within the analyzed papers, as well as the trends in these patterns. These patterns are based on the properties used in analyzing the paper such as SC types, blockchain types, SC components, smart contract verifications and the year the research work is published.

a) *Patterns in SC types, blockchain types and trends*: Figure 4 shows the SC types that are implemented on the blockchain and the prominent ones that account for the majority of the applications including agriculture, general logistics, healthcare and specific commodity products. Agriculture SCs capture the entire SC of food covering the growing, purchasing and delivery of food items. The healthcare covers SC of drugs (including vaccines) and medical devices. General logistics cover the end-to-end creation, purchase and delivery of generalized goods and services while the commodity SC covers specific products. According to Figure 5, wearable materials such as clothing and luxury accessories account for the majority of specific product SCs that are implemented on the blockchain.

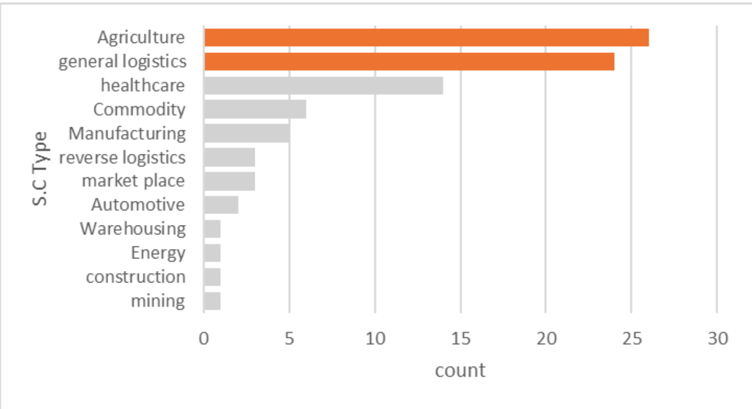

Figure 4: Summary of SC types implemented on the blockchain

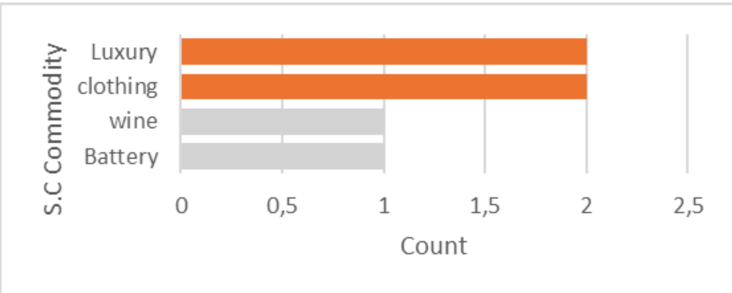

Figure 5: Summary of SC Commodities on Blockchain

For the selected SCs such as Agriculture, general logistics, healthcare and commodity SCs, Figure 6 shows the observable trends that exist during the analysis period of 2018 – 2023. The initial drivers of the SC applications are general logistics applications, however, the rise of agriculture and healthcare applications became noticeable in 2021. Also, initial SC applications on blockchain are not designed for specific commodities, still, from 2020 upwards, entries such as battery, wine, clothing and luxury accessories started occurring in the examined literature works. For the blockchain networks, Ethereum is the noticeable choice for prototyping and deploying SC applications, then, followed by Hyperledger fabric. There is a noticeable trad in custom blockchain use from 2022 to Figure 7.

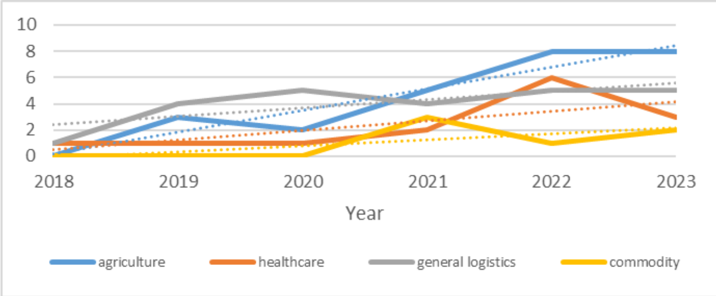

Figure 6: Trends of SC Applications on Blockchain

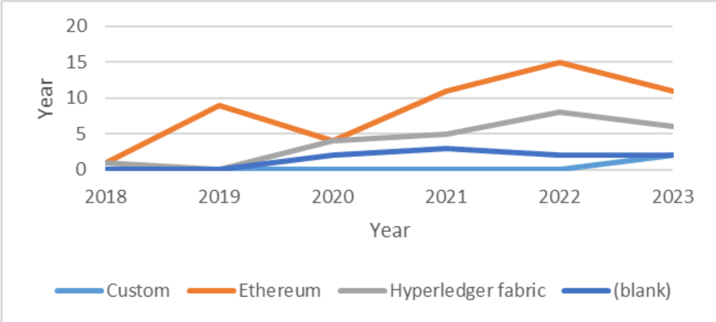

Figure 7: Trends in Blockchain networks for implementing SC Use cases

For the selected SCs such as Agriculture, general logistics, healthcare and commodity SCs, Figure 6 shows the observable trends that exist during the analysis period of 2018 – 2023. The initial drivers of the SC applications are general logistics applications, however, the rise of agriculture and healthcare applications became noticeable in 2021. Also, initial SC applications on blockchain are not designed for specific commodities, still, from 2020 upwards, entries such as battery, wine, clothing and luxury accessories started occurring in the examined literature works. For the blockchain networks, Ethereum is the noticeable choice for prototyping and deploying SC applications, then, followed by Hyperledger fabric. There is a noticeable trad in custom blockchain use from 2022 to Figure 7.

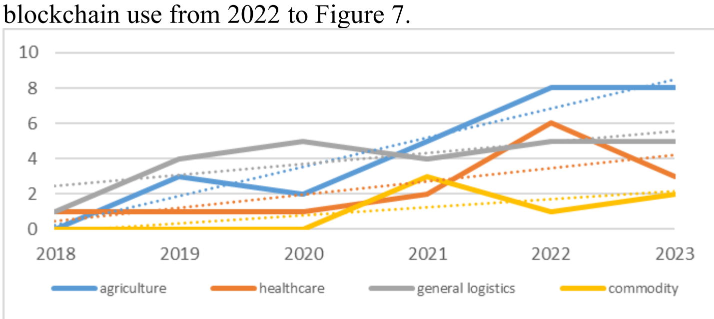

Figure 6: Trends of SC Applications on Blockchain

b) *Common SC. components on blockchain applications*: Figure 8 shows the SC components and their frequency of occurrence in the examined research works while Figure 9 shows SC components occurred in various SC types. The result shows that the main components include product creation, transportation notes, delivery confirmation, product payment, payment settlement conditions, product order and product listing.

Recording information about product creation and the status of product delivery contained in the transportation notes occurred most in the agriculture and general logistics SCs.

Agriculture and healthcare SCs occurred most product delivery confirmations. Although the occurrence frequency of product order, payment settlement and product payment components are minimal, their occurrence is distributed across the main SC types identified in this paper such as agriculture, general logistics, healthcare and commodity SCs.

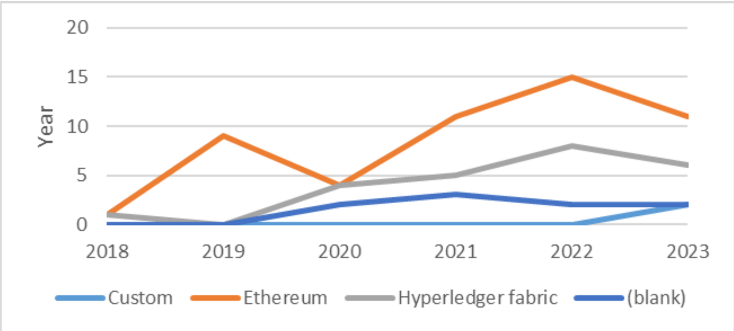

Figure 7: Trends in Blockchain networks for implementing SC use cases

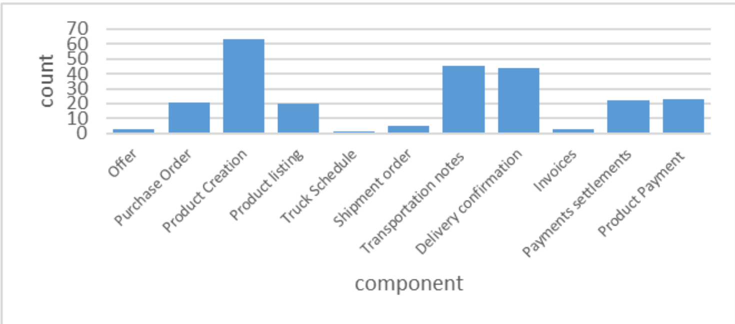

Figure 8: SC Components on Blockchain

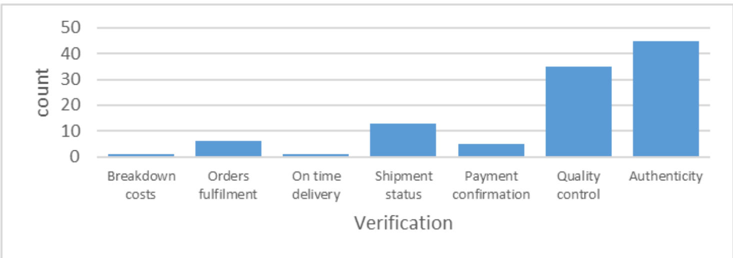

Figure 9: Common Occurring Components in SC Applications on Blockchain

c) *Common Smart contract verifications on SC applications:* Figure 10 shows the smart contract verifications on SC components and their frequency of occurrence, while Figure 11 shows the occurrence of these verifications in various SC types. The result shows that the main verifications include authenticity, quality, shipment status and order fulfilment checks.

By analyzing the checks that occur in various SC types, the result shows that only authenticity verification and quality checks occurred in all four main SC types. Still, tracking of shipment status occurred most in the general logistics SCs.

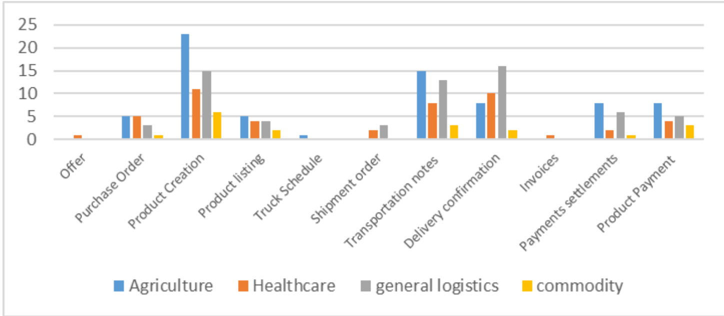

Figure 10: SC Components on SC Application types

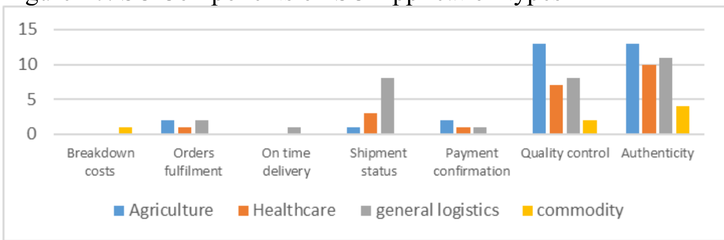

Figure 11: Smart Contract Verifications on SC Applications Types

IV. Model Representation of SC Applications Patterns on Blockchain

Figure 12 shows the model representation of SC components that occur in blockchain applications of various blockchain types. Each component is represented with a class diagram that captures two types of methods; on-chain methods that result in state change of the blockchain (colored grey) and queries for verifying specific business conditions based on the data that is generated from the SC components. The model diagram is generated by combining the results from cases presented in section III.A and results from relevant literature works are presented in section III.B.

A. SC components properties and their relationships

the listed supply chain components in Figure 12 are already explained in the previous section, however, it is still necessary to understand the relationship between these components and additional interfaces (from SC and blockchain) that provide data input to the listed components. The offer component is directly connected to the product list and product order in many-to-many and one-to-many relationships since multiple offers can be generated from different lists of products. Also, many products can be ordered from one single offer. The product list is connected to the inventory interface, ensuring that only available products can be displayed. All the products in the product list are a result of product creation, and a product list can also be empty when no digital representation of the product has been created, necessitating one-to-zero relation between the product list and product creation component. A product can contain information about the suppliers, raw materials, location data etc. which are linked to a unique ID representing the product. IoT sensors can be used as an external interface for collecting data (such as location, raw materials information etc.) about the product created. NFT provides a meaningful blockchain interface for uniquely representing products and the additional product information represented as the NFT meta-data. The product order is connected to the settlement conditions for a specific order. Long-term

business contracts, or specific discounts provided to a customer can represented in the settlement conditions.
The shipment order is connected to the product order in such a way that specifies that a single shipment can contain several product orders. The shipment order is connected to the shipment plan that outlines the timeline/ schedule provided by an internal or third-party logistics (3PL) provider. The shipment order is connected to the transportation notes that record the states of the delivery transportation. The records provided during the product transportation can be provided within specific intervals or summarized data can be recorded once the delivery is completed. For real-time data records of transportation notes, an IoT interface is necessary to provide the streams of data that are stored on-chain. The transportation notes component is connected to the delivery confirmation enabling the receiver to take ownership of the product when the delivery is completed and several delivery notes can be linked to one delivery confirmation. The invoice is connected to either the delivery confirmation or the product order. Some SC require that payment is made immediately product is ordered, while some others require payment to be made on product delivery confirmation (or even at a later date). Multiple delivery confirmations or orders can have one single invoice but multiple invoices cannot be issued for the same product order. Invoices can also be issued separately for product orders and shipment orders. The final price in the invoice and method of payment are specified based on the settlement conditions initially agreed upon by the parties in the supply chain. The invoice is connected to the product payment component and one payment can only be connected to a single invoice. The payment is connected to the cryptocurrency (tokens) interface which is a blockchain-specific component for storing and transferring values between parties.

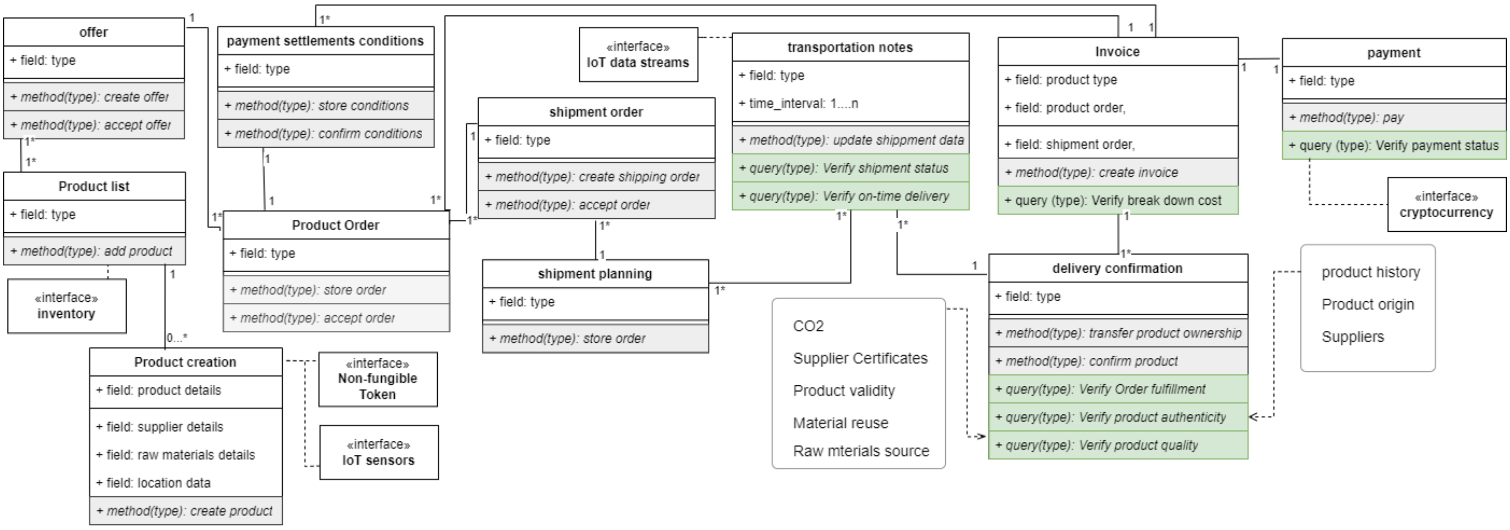

Figure 12: Meta-model of SC applications on blockchain

B. On-chain methods on the components

Two main methods that result in the status change of data stored on the blockchain are available on the SC components listed in Figure 12. The on-chain methods are colour-coded grey in the model figure. The first involves the creation or updating of a particular component on the blockchain while the second is a method to confirm the acceptance of a particular component by the other parties in the supply chain. For components that don't require any confirmation, only the creation or updating method is available on them.

The component that requires confirmation from the other parties in the SC include the offer, Order (product and shipment), payment settlement conditions and delivery confirmation. For instance, an offer made by a purchaser can be accepted or rejected by the supplier. A product Order can be accepted or rejected (for example, if the product is out of stock). A shipment order can be accepted or rejected by a 3PL provider. Settlement conditions require confirmation by both the supplier and purchaser. Lastly delivery confirmation component requires the second party to accept the delivered product in the condition it was delivered in, hence, confirm to take ownership of the received product. The purchase can re-ject the delivered product citing quality or authenticity issues. All the confirmation methods can be implemented as a digital signature that represents the individual or organization the confirmation is required from. Digital signatures are part of the PKC which the blockchain provides.

C. Smart contract verifications

From Figure 12, it is noticeable that verifications are mostly performed on the receiver side to check if the business conditions agreed upon have been fulfilled by the supplier. However, a supplier can also verify if the other parties have fulfilled their payment obligations. The verification queries are colour-coded green in the model figure. On the delivery notes component, a party (usually the purchaser) can query the obligation to check the *shipment status.* If specific shipping conditions (such as temperature, delivery timeline etc.) are specified in the shipping order, the purchase can verify the status of these conditions in real time or a summarized data upon product delivery. Also, delays in shipment (or *on-time delivery*) can be verified from the transportation notes stored on-chain.
Upon receiving the product, a purchase can verify the *order fulfilment, authenticity and quality* of the received product. The query interfaces for these verifications are provided in the delivery confirmation component. The order fulfilment simply checks if what was ordered is correctly delivered in the specified quantity and product types. Authenticity

verification checks the digital signatures associated with the product from when it was created to when it was delivered to the current owner. Hence, historical ownership data can be checked with authenticity verification. Quality verification can be used to check the validity and sustainability properties of the received product such as CO2 emissions, raw materials suppliers certificates and material re-use during the product creation.

In the invoice component, the purchase can verify the *breakdown cost* to check if the final price in the invoice is based on agreed settlement conditions and the actual deliveries that occurred as recorded in the transportation notes. Lastly, the *verify payment status* is contained in the payment component, providing a query interface for a supplier to check if the delivered (or ordered product) has been paid for by the purchaser. The payment status verification can also be queried by the 3PL provider to check if the shipment order has been paid.

V. DISCUSSION

The discussions resulting from this work are centered on three themes, interpretation of our findings, comparison of our result with other similar works and practical implication of our work.

A. Discussions on results interpretation

Public blockchains generally have explorers for viewing transactions in the network. Verifying the transactions and business conditions that resulted in these transactions requires additional technical knowledge. These public transaction viewing portals are not available on permissioned networks. Therefore, application developers must provide specific queries for interacting with the data on the blockchain to verify conditions specified in smart contracts. Our result showed that only 52% of the SC application prototypes on blockchain provided these types of queries although most of the papers claimed to develop software artefacts that enable provenance on the blockchain.

An interesting finding from this work is that Agriculture, general logistics, and healthcare are the main drivers of supply chain applications on the blockchain while being followed by commodity products. Agricultural food products and healthcare drugs are ingested products by humans and, thus, require high levels of transparency in their production and logistics. Hence, it is not surprising that quality control and authenticity verifications are common in these product groups according to Figure 11. In the commodity products, some of the most occurring products are clothing apparel and luxury accessories. These kinds of products have also been greatly impacted by the proliferation of fakes and counterfeit [21,22], and our findings show that smart contract-based authenticity verification is common for this type of supply chain. Tracking the status of the shipment to verify transportation conditions is no surprise the second most important verification on general logistics after authenticity verification.

Our result shows that smart contract verifications such as breakdown costs and on-time delivery are not important business conditions that are verified on the blockchain since their occurrence is rather few and limited to a single use case. Also, the most important supply chain component commonly captured on the blockchain is product creation, followed by transportation notes and delivery confirmation. This is because data captured in the product creation are necessary for performing crucial smart contract checks such as authenticity verification and quality checks. These checks are conducted at the delivery confirmation component when the receiver takes ownership of the delivered physical product. Furthermore, our result shows that the least important SC components that are stored on the blockchain are the offer, shipping order and invoicing components.

From the findings, Ethereum remains the top choice for prototyping blockchain applications in supply chains, followed by Hyperledger Fabric. This contradicts the earlier research [1] that identified Hyperledger as the leading network for deploying blockchain applications for organizational use cases. There are still several privacy, access control and user-management issues in public networks [23]. Since our research didn't differentiate between prototyped proof of concepts (PoC) in comparison to an actual application used by an organization, it is not clear if the result will remain the same.

B. Discussions in comparison of results with similar works

The research works in [13,14 and 15] present model-driven approaches for supply chain applications on the blockchain. The works capture the code-generation aspects such that modelling concepts can be used to realize smart contract codes automatically. While our work presented modelling concepts for realizing blockchain-based SC applications, the current stage of work does the code translation aspects, still, our model represents a more complete and detailed representation of supply chain modelling patterns on blockchain. These research works barely capture the smart contract verifications on on-chain components, the paper [15] included track and trace checks, while [13, 14] included no algorithms for verification of business conditions. Our work has provided a comprehensive list of important SC business conditions that can be verified with smart contracts.

C. Discussions on the practical implications of our work

The most important practical implication of this research is that it provides a generalized basis for modelling and developing supply chain applications on the blockchain. Supply chain analyst can easily design applications for their specific SC use cases by identifying the components that occur in their processes and the relations to other components. Furthermore, analysts can use our meta-model as a template for specifying useful on-chain methods and verification that are relevant to their use cases.

With the increased integration of machine learning and AI in blockchain software engineering, our work also provides a basis for extending the existing works. The research work [24] shows a machine learning application predicting blockchain adoption in a supply chain using Bayesian network analysis. The features used in analyses are categorized into technological factors, organizational factors, and environmental factors. Our work can provide a template for recommending suitable SC operations to deploy on the blockchain based on the result of the ML prediction. It is also possible that advanced AI methodologies can be adopted to identify problematic operations in SC by analyzing the data logs from the supporting information systems [25], while the result of our work will provide the basis for recommending suitable on-chain components for the problematic operations.

VI. Conclusions and Future Work

The objective of this research work is to identify common patterns in blockchain-based SC applications which will provide input for model-driven development that covers most operations in supply chains. To achieve this goal, we outlined three research questions for this work. The first addresses the need to outline common components in blockchain-based SC applications while the second question explores the properties that exist in the identified components such as on-chain methods and smart contract verifications executed on the components. The third seeks to develop a general model representation of supply chain applications on the blockchain, covering the relevant components, their properties and relationships.

A combination of case studies analyses and systematic literature review is applied to identify the common components and smart contract verifications on them, thereby, answering the first and second research questions. The result shows that the relevant components are Offer, Purchase Order, Product Creation, Product Listing, Truck Schedule, Shipment order, Transportation notes, Delivery confirmation, Invoices, Payments settlements and Product Payment. The smart contract verifications on these components include Breakdown costs, Order fulfilment, On-time delivery, Shipment status, Payment confirmation, Quality control and Authenticity Check.

To answer the third research question, a meta-model is used to outline the patterns that exist in blockchain-based SC applications by outlining all the initially identified components and their relations. Furthermore, the meta-model specifies the useful on-chain methods that result in the state change of the SC components and the smart contract verifications that apply to the components.

Limitations and Future Work

Although our research adopted multiple case studies (two business cases) in combination with several high-quality journal articles (87 papers), there is still the risk of generalization which is a common problem for this type of research. For instance, this study excluded conference papers, hence and it is possible that some important patterns could have been missed. Another limitation of this work is a potential subjectivity problem in collecting and interpreting data from various research papers.

Automated code generation is an important aspect of model-driven software development. Hence, the future work from this research is to specify algorithms and smart contract codes that represent various SC components and operations on components. Thereby, providing a no-code alternative for building blockchain applications for various supply chain use cases and operations.

References

[1] Udokwu, Chibuzor, et al. "The state of the art for blockchain-enabled smart-contract applications in the organization." 2018 Ivannikov Ispras Open Conference (ISPRAS). IEEE, 2018.

[2] Saghiri, Ali Mohammad. "Blockchain architecture." Advanced applications of blockchain technology (2020): 161-176.

[3] Ellram, Lisa M., Wendy L. Tate, and Corey Billington. "Understanding and managing the services SC." Journal of SC management 40.3 (2004): 17-32.

[4] Vangen, Siv, John Paul Hayes, and Chris Cornforth. "Governing cross-sector, inter-organizational collaborations." Public Management Review 17.9 (2015): 1237-1260.

[5] Swan, Melanie. Blockchain: Blueprint for a new economy. " O'Reilly Media, Inc.", 2015.

[6] Dixit, Abhishek, et al. "Towards user-centred and legally relevant smart-contract development: A systematic literature review." Journal of Industrial Information Integration 26 (2022): 100314.

[7] Thaman, Minakshi. "Blockchain - Based Framework - A Scientific Tool for Developing a Robust Banking System." Blockchain Technology in Corporate Governance: Transforming Business and Industries (2022): 303-324.

[8] Phelan, Raymond. "An Evaluation on the Performance of Code Generated with WebAssembly Compilers." (2021).

[9] Udokwu, Chibuzor Joseph. "A modelling approach for building blockchain applications that enables trustable inter-organizational collaborations." (2022).

[10] Marchesi, Lodovica, et al. "Automatic generation of ethereum-based smart contracts for agri-food traceability system." IEEE Access 10 (2022): 50363-50383.

[11] Wohrer, Maximilian, and Uwe Zdun. "From domain-specific language to code: Smart contracts and the application of design patterns." IEEE Software 37.5 (2020): 37-42.

[12] Udokwu, Chibuzor, et al. "Implementation and evaluation of the DAOM framework and support tool for designing blockchain decentralized applications." International Journal of Information Technology 13 (2021): 2245-2263.

[13] Dolgui, Alexandre, et al. "Blockchain-oriented dynamic modelling of smart contract design and execution in the supply chain." International Journal of Production Research 58.7 (2020): 2184-2199.

[14] Perboli, Guido, Stefano Musso, and Mariangela Rosano. "Blockchain in logistics and supply chain: A lean approach for designing real-world use cases." Ieee Access 6 (2018): 62018-62028.

[15] Helo, Petri, and Yuqiuge Hao. "Blockchains in operations and supply chains: A model and reference implementation." Computers & Industrial Engineering 136 (2019): 242-251.

[16] Swan, Melanie. Blockchain: Blueprint for a new economy. " O'Reilly Media, Inc.", 2015.

[17] Pahlajani, Sunny, Avinash Kshirsagar, and Vinod Pachghare. "Survey on private blockchain consensus algorithms." 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT). IEEE, 2019.

[18] Xu, Xiwei, et al. "The blockchain as a software connector." 2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA). IEEE, 2016.

[19] Cuccuru, Pierluigi. "Beyond bitcoin: an early overview on smart contracts." International Journal of Law and Information Technology 25.3 (2017): 179-195.

[20] Di Angelo, Monika, and Gernot Salzer. "Tokens, types, and standards: identification and utilization in Ethereum." 2020 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS). IEEE, 2020.

[21] Norum, Pamela S., and Angela Cuno. "Analysis of the demand for counterfeit goods." Journal of Fashion Marketing and Management: An International Journal 15.1 (2011): 27-40.

[22] Udokwu, Chibuzor, et al. "Exerting Qualitative Analytics and Blockchain Requirement-Engineering in Designing and Implementing a Luxury Products Authentication System." Inventions 8.1 (2023): 49.

[23] Udokwu, Chibuzor, Stefan Craß, and Alex Norta. "Authenticating Parties in Blockchain-Enabled Inter-Organizational Processes with Configurable Challenge-Sets." International Conference on Future Data and Security Engineering. Singapore: Springer Nature Singapore, 2023.

[24] Kamble, Sachin S., et al. "A machine learning based approach for predicting blockchain adoption in supply Chain." Technological Forecasting and Social Change 163 (2021): 120465.

[25] Fleig, Christian, Dominik Augenstein, and Alexander Maedche. "Designing a process mining-enabled decision support system for business process standardization in ERP implementation projects." Business Process Management Forum: BPM Forum 2018, Sydney, NSW, Australia, September 9-14, 2018, Proceedings 16. Springer International Publishing, 2018.