International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: www.tandfonline.com/journals/tprs20

Smart contracting for production supply in shared manufacturing: a practical simulation approach

Larissa Krämer, Patrick Stuckmann-Blumenstein, Pascal Kaiser, Michael Henke, Moritz Roidl & Alice Kirchheim

To cite this article: Larissa Krämer, Patrick Stuckmann-Blumenstein, Pascal Kaiser, Michael Henke, Moritz Roidl & Alice Kirchheim (2025) Smart contracting for production supply in shared manufacturing: a practical simulation approach, International Journal of Production Research, 63:14, 5142-5168, DOI: 10.1080/00207543.2024.2430449

To link to this article: https://doi.org/10.1080/00207543.2024.2430449

INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH
2025, VOL. 63, NO. 14, 5142–5168
https://doi.org/10.1080/00207543.2024.2430449

Smart contracting for production supply in shared manufacturing: a practical simulation approach

Larissa Krämer [a], Patrick Stuckmann-Blumenstein [b], Pascal Kaiser [a], Michael Henke [b], Moritz Roidl [a] and Alice Kirchheim [a]

[a]Chair of Material Handling and Warehousing, TU Dortmund University, Dortmund, Germany; [b]Chair of Enterprise Logistics, TU Dortmund University, Dortmund, Germany

ABSTRACT

Enhancing transparency in production processes, especially in shared manufacturing, relies heavily on sharing data. Information asymmetries and coordination problems between parties with conflicting interests pose a challenge in this multi-stakeholder interaction. Blockchain technology with smart contracting can be a solution due to its immutable data and decentralised data storage features. Designing and executing blockchain in industrial applications is a highly intricate task that requires extensive testing, expertise, and proficiency. This paper is the first to propose a holistic simulation model for evaluating the impact of smart contracting on shared manufacturing, including a novel approach to simulated smart contracting in time-lapse for Ethereum-based networks. The introduced model guides the design and implementation process of blockchain applications in shared manufacturing to address this challenge. A systematic literature review establishes ten design process requirements and ten smart contract functions. The implementation is developed based on the design benchmarks of three Ethereum-based frameworks to investigate the simulation model's respective feasibility and scalability. The simulation model validation demonstrates our approach's suitability for simulating smart contracting in shared manufacturing within a hybrid production. It enables fast and scalable simulations, offering an innovative approach to extensively testing blockchain applications before their introduction to ongoing industrial operations.

ARTICLE HISTORY
Received 1 March 2024
Accepted 4 November 2024

KEYWORDS
Shared manufacturing; digital twin; blockchain; smart contract; simulation; hybrid production

SUSTAINABLE DEVELOPMENT GOALS
SDG 9: Industry; innovation andinfrastructure

1. Introduction

Data sharing among companies is a crucial aspect of Industry 4.0 that increases supply chain efficiency (Camarinha-Matos, Fornasiero, and Afsarmanesh 2017). However, understanding the supply chain as a value-creation network, especially during times of crisis and interruptions, is a complex and multifaceted task (Durugbo and Al-Balushi 2023). The lack of transparency and complexity make it challenging to identify specific approaches for improving multi-stakeholder interaction, particularly at the level of shared data. It is reasonable to focus on manufacturing and production supply to overcome this challenge, as there are clearly defined inputs and outputs, work steps and available data that can be shared with stakeholders such as suppliers or customers. This area influences the entire value chain from purchasing to distribution (Bányai 2021). According to a study conducted by the World Economic Forum, data sharing for manufacturing improvements is valued at over $100 billion, and 72% of the 996 surveyed manufacturing managers consider data sharing with other manufacturers as crucial for success (Forum and Group 2020).

Exchanging data within a value network enables new business models, such as shared manufacturing, in which idle production resources such as machines are shared within a network (Yu et al. 2020). This approach opens up previously untapped economic potential and enables companies to benefit from greater flexibility and lower fixed costs (Liu, Liu, and Wei 2021). Sharing resources allows companies to achieve economies of scale while reducing production's environmental and social impact through optimal utilisation of machine resources (Sontag 2000). However, information asymmetries and coordination problems between several players with conflicting interests or inconsistent behaviour pose a challenge for shared manufacturing (Gong, Zhang, and Alharithi 2022; Ioannou and Demirel 2022). Information asymmetries can, for instance, arise in production data regarding the status of a work-in-progress product and associated costs due to insufficient accurate information.

CONTACT Larissa Krämer larissa.kraemer@tu-dortmund.de

These challenges can be addressed by digital twins and blockchain technology (Gong, Zhang, and Alharithi 2022). A digital twin is a virtual representation of a physical system in the digital world (Park, Easwaran, and Andalam 2019). Its integration leads to transparent insights, reducing coordination problems and information gaps (Kritzinger et al. 2018). Data is provided through blockchain, allowing immutable and transparent transactions without intermediaries using distributed trust (Treiblmaier 2019). Microservices on the blockchain, called smart contracts, enable automated, irreversible, and trusted transactions such as financial transactions (Ioannou and Demirel 2022). In the above example, a smart contract can document a product's preceding and current work steps and can even be used to instantly transfer money for provided services between machines or companies according to predefined rules. Combining blockchain and digital twins enables the financial flow to be integrated into the information and material flow. Process data, including financial data, can be shared among all stakeholders decentralised while maintaining one single source of truth and increasing end-to-end transparency (Gong, Zhang, and Alharithi 2022; Kshetri 2018).

While this potential has been recognised in literature, few concepts on blockchain in a digital twin in shared manufacturing exist. None of them consider a holistic simulation model to facilitate the process of introducing smart contracting in industrial applications.

Against that backdrop, this contribution targets the following research questions:

(1) How can we simulate the impact of smart contracting on the digital twin of shared manufacturing to develop a decision-making aid for multiple stakeholders considering the identified requirements?
(2) How can smart contracting for Ethereum-based networks be simulated in shared manufacturing to enable simulations in time-lapse and with less technical effort?

We introduce a novel concept to simulate the impact of smart contracting for production supply in shared manufacturing based on requirements derived from literature. We demonstrate our approach using Ethereum-based test networks. Additionally, we present an innovative method to simulate smart contracting for Ethereum-based frameworks in time-lapse to facilitate decision processes regarding smart contracting in shared manufacturing. Our simulation experiments provide empirical evidence on the scalability of smart contracting in shared manufacturing for Ethereum-based frameworks. This work demonstrates the feasibility of blockchain in a digital twin in shared manufacturing. Managers are enabled to evaluate and assess Ethereum blockchain usage for similar industrial use cases before implementation in terms of the resulting costs and feasibility.

This paper uses the design science research (DSR) methodology as a procedural model for developing a method artefact (Hevner et al. 2004; Peffers et al. 2007). The approach is divided into six activities (Peffers et al. 2007) and applied to our contribution as displayed in Figure 1. Through the integration of the DSR process, a rigorous process in terms of construction and evaluation is ensured (Hevner et al. 2004). We adopt a problem-centred approach by discussing the problem space in Section 1. The second activity, covered in Section 2, explores objectives and potential solutions through a systematic literature review (SLR) for related work. In Section 3, we delve deeper into the design and development process by addressing the results from the SLR and creating the artefact. The demonstration in activity four proves that the artefact can solve a subsection of the problem space using simulation and experiments. In this contribution, Section 4 presents the implementation and validation through a simulation model, which includes the design of experiments and experiments for several scenarios. Activity 5, covered in Section 5, evaluates the research outcomes, discusses implications and gives perspectives for future work. Finally, the artefact, its effectiveness, and the problem itself are communicated through this contribution.

2. Fundamentals and related work

This chapter explains the fundamentals of shared manufacturing, digital twins, and blockchain relevant to this contribution. Based on this, requirements for the simulation model are derived from literature. These requirements are the foundation for the concept, the simulation model and experiments. The state-of-the-art literature is researched with these requirements to deduce the research gap addressed in this contribution.

2.1. Fundamentals

Shared manufacturing refers to sharing capacities in production systems (Yu et al. 2020). It is a business-to-business (B2B) production process promoting value creation for multiple stakeholders, and facilitating the emergence of innovative business models (Klarin and Suseno 2021). Internet platforms unite producers and consumers by offering idle manufacturing resources and matching them to consumers' tasks (Liu, Liu, and Wei 2021). In shared manufacturing, companies that are unknown to each other and do not share a common

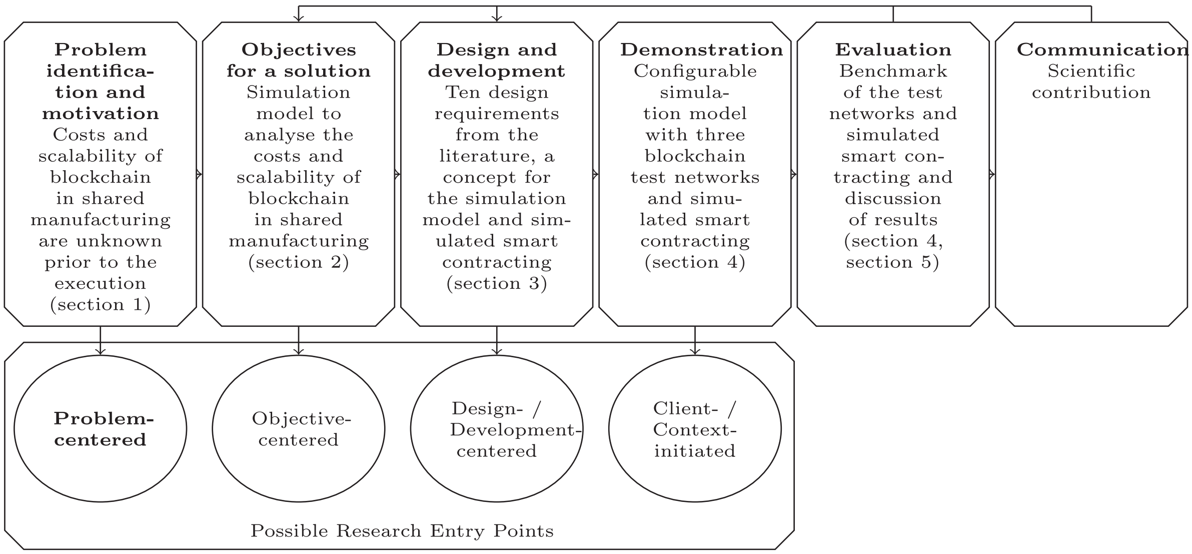


Figure 1. The approach of DSR based on Peffers et al. (2007) for this contribution.

foundation of trust interact (Rouzbahani and Taghiyareh 2022) and might face the disclosure of crucial information to unwanted third parties (Müller et al. 2022). Another challenge is the lack of transparency and data on the shop floor level to generate knowledge about overcapacity or undercapacity (IPRI and IPH 2021). For this, a highly digitised production to handle the process of data sharing and to give precise information about capacity to the sharing platform is required (Yu et al. 2020). Additionally, companies in shared manufacturing face an increasing variety of product models and designs along with decreasing quantities per variant (Koren and Shpitalni 2010). A hybrid production is suitable for addressing this, as it can realise flexible layouts, combining an assembly line and a matrix production (Kaiser et al. 2022) to integrate the advantages of both systems: high assembly line efficiency with adaptability and flexibility of the matrix system (Göppert et al. 2021). In a matrix production, workstations are arranged in groups with similar tasks and tools (Schmidtke, Rettmann, and Behrendt 2021) and positioned parallel to one another in several rows. The **production supply** is an essential logistics task in a matrix production due to the flexible and adaptable system. The production supply comprises two stages. In the first stage, production orders are assigned to available workstations (Bányai 2021). In the second stage, the material transport to and between the workstations is executed by automated guided vehicles (AGVs), enabling flexible transport routes (Schmidtke, Rettmann, and Behrendt 2021).

To facilitate data sharing in shared manufacturing and tap into the monetary potential presented in the introduced study of Forum and Group (2020), a digital twin is a viable solution. The term *digital twin* was introduced by Grieves in 2003 and describes the precise representation of a physical object or system in the digital (cyber) world (Grieves 2015). Physical and digital systems can affect each other via a bilateral information flow, which leads to a merge of the physical system and its digital representation (Grieves and Vickers 2017; Grieves 2015; Kritzinger et al. 2018). A vital purpose of a digital twin is the 2D or 3D visualisation of production processes to enable their observability (Park, Easwaran, and Andalam 2019). Digital twins are also applied for data analysis and simulation of scenarios to reduce costs, predict failures, and prepare for unexpected events in manufacturing (Barricelli, Casiraghi, and Fogli 2019). A digital twin is highly dependent on reliable data from the physical world, whereas the physical system depends on the correct input from the digital world Schweiger and Barth (2023). Besides, security, such as device authorisation, is critical in manufacturing. Manipulated data of the physical system influences the digital twin due to the interplay between physical and digital systems (Park, Easwaran, and Andalam 2019). Since a trustworthy and secure layer is required to operate a digital twin properly, Blockchain could be a feasible solution (Suhail et al. 2022).

Blockchain is a distributed and practically immutable database maintained by a decentralised network (Nakamoto 2008). It is subordinated to the Distributed Ledger Technologies (DLT) (Kannengießer et al. 2019). In this peer-to-peer network, every participant has a copy of the stake (Morabito 2017). The data is stored sequentially in cryptographically linked blocks and validated by consensus mechanisms (Morabito 2017). Blockchains are divided into three categories: Public blockchains are decentralised and accessible to everyone, while private blockchains can be restricted to authorised participants (Zheng et al. 2017). Consortium blockchains are partially decentralised and governed by a group of organisations (Zheng et al. 2017).

The properties of a blockchain regarding throughput, computing power, and security depend on its consensus mechanism. This mechanism determines which participant may add another block. Proof of Work (PoW) requires network participants, called miners, to expend a certain amount of computing power to solve a cryptographic puzzle associated with a block of transactions. The first miner to solve the puzzle is granted the right to validate and append the block to the blockchain (Nakamoto 2008). Proof of Authority (PoA) requires network participants, called validators, to prove their authority and trustworthiness by fulfilling certain conditions and demonstrating their long-term commitment to maintaining the blockchain (De Angelis et al. 2017).

Some blockchain frameworks, such as Ethereum (Ethereum-Foundation 2024), include *smart contracts*. These microservices are implemented and automatically executed on the blockchain if predefined requirements are met (Buterin 2014). As a smart contract code is executed on every network node, the processing consumes fees, known as gas in the Ethereum framework, representing the computing power required (Glaser 2017).

Ethereum is used in the following due to its comprehensive programming language, Solidity, and its application popularity, ensuring genericness (Ranganthan et al. 2018). Ethereum's commitment to open standards, its well-established ecosystem, and its compatibility with other blockchain frameworks enhance the security of smart contracts and facilitate interoperability between various simulation systems and environments (Onwubiko et al. 2023). This interoperability fosters collaboration and knowledge sharing among stakeholders. Furthermore, developments in Ethereum are expected to significantly enhance its scalability and throughput, bolstering the network's ability to support large-scale simulations with increased transaction throughput and reduced latency (Onwubiko et al. 2023; Ranganthan et al. 2018).

For large-scale and complex blockchain applications, *blockchain simulators* have emerged, facilitating testing configurations and implementations at low costs (Paulavicius, Grigaitis, and Filatovas 2021). An approach to simulating blockchains was developed by Alharby and van Moorsel (2020). The proposed system distinguishes between an incentive layer, a consensus layer, and a network layer as basic modules for the simulation. The model simulates the throughput of networks, mined blocks not attached to the chain, and blocks added to the main chain per day (Alharby and van Moorsel 2020).

Since all data stored on a blockchain is traceable, logged, and retrievable, blockchain participants can trust the information in the blockchain, even if they do not know or trust each other (Hawlitschek 2019). This mechanism is the underlying main driver of blockchain in SCM for creating a fully transparent, immutable information and material flow with an integrated financial flow in real-time, which can be traced and monitored by all stakeholders (Gong, Zhang, and Alharithi 2022; Gürpinar, Henke, and Ashraf 2024; Kshetri 2018). Smart contracts enable substituting paper-based documents with digital documents and automated processes on a blockchain (Kshetri 2018), enabling faster transactions and cost reduction (Guo, Zhang, and Zhang 2023; Kshetri 2018) and giving easier access to financing (Gong, Zhang, and Alharithi 2022). In manufacturing, smart contracting can carry out payment or replenishment orders, realise 'pay-per-use' models (Ranganthan et al. 2018) and provide a distributed and immutable record of transactions (Teslya and Smirnov 2018). A related application uses blockchain to keep records of registered and authorised devices to increase security (Reyna et al. 2018).

Combining blockchain in SCM and manufacturing with a digital twin enables a complete and tamper-proof history of all system states stored on the blockchain (Raj 2021). It supports sound decision-making and profound process analysis and provides reliable data, for instance, for predicting failures or peak loads (Ding et al. 2019). Via blockchain and a digital twin, data exchange within production and between external stakeholders can be standardised and simplified (Guo, Zhang, and Zhang 2023), allowing for data management across a digital twin's life cycle (Suhail et al. 2022).

In terms of *underlying research theories*, smart contracting in shared manufacturing is located in microeconomics, focussing on individual markets in which products and services are bought and sold (Wiese 2021). A microeconomic theory applied to this market mechanism is the principal-agent theory (Jensen and Meckling 1979), which focuses on the interaction between a principal and an agent before (ex-ante) and after (ex-post) reaching an agreement. The principal hires an agent to perform a task on its behalf. The shared manufacturing environment allows stakeholders to act as both principals and agents, depending on their role as producers, consumers or prosumers. The principal has to rely on the agent's information about the manufacturing process without complete knowledge or control and cannot ensure the correct fulfilment directly. If the principal and the agent aim at maximising their utility, conflicting interests can arise. The core of this is reflected in the principal-agent problem, which can occur due to an information asymmetry between a principal and its agent (Halldorsson et al. 2007). Information asymmetries can arise from a lack of transparency, actuality, and accuracy in production data, such as production status, and financial data, such as costs. One solution to this problem is smart contracting, which increases transparency and ensures immutable data (Treiblmaier 2018).

Connected to the principal-agent theory is the transaction cost theory (Williamson 1987), which assumes that every transaction has costs. These can include costs for forming new contracts, bargaining costs, or costs of fulfilment and enforcement (Halldorsson et al. 2007). The theory assumes that organisations aim to minimise these costs, which occur due to bounded rationality and opportunism (Williamson 1987). Bounded rationality refers to environmental uncertainty caused by limited information and limited capacity to understand the available information. In shared manufacturing, intra- and inter-organisational transaction costs originate, for instance, from matching producers and consumers, i.e. forming new contracts, and from sharing data, processing customer orders, and financial transactions, i.e. costs of fulfilment and enforcement. Information asymmetries resulting from the principal-agent problem cause bounded rationality for stakeholders in shared manufacturing, increasing transaction costs.

2.2. Design requirements

To identify related work, we first conduct a systematic literature review (SLR) in accordance to Brocke et al. (2009) to identify requirements for a simulation model to research the impact of smart contracting on shared manufacturing and its digital twin. The detailed approach can be found in the appendix in Table A1. The identified literature was categorised and synthesised, as depicted in Table 1. The sequence of requirements (R1...R10) is not sorted. Shared manufacturing and the digital twin share many requirements, especially regarding adaptability (R1) and data acquisition (R3), sharing (R4), storage (R6), analysis (R8), and configurability (R9). An open mindset (R10) is a requirement derived explicitly from shared manufacturing, while a human-machine interface (HMI) (R7) is essential for the digital twin. While we recognise R10 as an essential key requirement for implementing such a system in industry, it is not suitable to consider the human mindset in our simulation, and we exclude it from further considerations.

R1 Adaptability. In shared manufacturing, it is beneficial for producers to offer production capacities on configurable machines arranged in a flexible layout (IPRI and IPH 2021; Müller et al. 2022). Suitable machines can either have a wide range of applications to be adaptable to different production tasks, or they can be very specified with low usage in a single production (IPRI and IPH 2021). Flexibility is also important for the digital twin regarding scalability, modularity, and extensibility to adapt the digital twin to changing physical conditions and to enable different configurations (Bellavista et al. 2023; Durão et al. 2018).

R2 Fidelity. The precise reflection of the physical entity in the virtual entity is essential for a digital twin (Javaid, Haleem, and Suman 2023). While the virtual entity should mirror the changes in the physical entity regarding changes of status or events, modifications occurring in the virtual entity should also be passed on to the

Table 1. Design requirements derived from the literature.

		Shared manufacturing				Digital twin			
Requirements		Yu et al. (2020)	IPRI and IPH (2021)	Müller et al. (2022)	Stuckmann-Blumenstein et al. (2024)	Durão et al. (2018)	Bellavista et al. (2023)	Schweiger and Barth (2023)	Javaid, Haleem, and Suman (2023)
R1	Adaptability / Flexibility	✓	✓	✓	✓	✓	✓	✓	✓
R2	Fidelity / Reflection	✓	X	✓	X	✓	✓	✓	✓
R3	Data acquisition	✓	✓	X	✓	✓	✓	✓	✓
R4	Connectivity / Interoperability / M2M / Data sharing	✓	✓	X	✓	✓	✓	✓	✓
R5	Data privacy / Data security / Availability	✓	✓	✓	✓	✓	✓	X	X
R6	Memorisation / Process documentation / Data storage	X	✓	✓	X	✓	✓	X	✓
R7	HMI / Process Visualisation / GUI	X	X	X	X	✓	✓	✓	✓
R8	Process automation / Data analysis	✓	✓	✓	X	✓	✓	X	✓
R9	Configurability / Prediction / Optimisation	X	✓	✓	✓	✓	X	✓	✓
R10	Mindset / Human role	X	X	✓	✓	X	X	X	X

physical entity (Bellavista et al. 2023). Synchronisation can be near real-time or in defined periods (Schweiger and Barth 2023). In shared manufacturing, a reflection of the physical world in a digital twin is also proposed regarding physical assets to be virtualised (Yu et al. 2020) such as a virtual inventory (Müller et al. 2022). A high fidelity can reduce information asymmetry between consumers and producers in shared manufacturing, addressing the principal-agent problem.

R3 Data acquisition. To enable calculations of capacities and prices and to provide up-to-date information for the process documentation, automated data collection, for instance via sensors, is required for shared manufacturing and a digital twin (Javaid, Haleem, and Suman 2023; Tao, Zhang, and Nee 2019; Yu et al. 2020). An accurate data acquisition increases the available information for multiple stakeholders, targeting the problem of bounded rationality and decreasing transaction costs (Williamson 1987). In the digital twin, this requires a bidirectional automated information flow between the virtual and physical entity, allowing the digital twin's virtual entity to operate autonomously without human input (Stuckmann-Blumenstein et al. 2024; Valk et al. 2022).

R4 Connectivity. Data has to be shared with and by all stakeholders to leverage the potential of shared manufacturing (Stuckmann-Blumenstein et al. 2024). Open communication with the physical entity, other digital twins, a platform service, or between a digital twin and domain experts is key (Tao, Zhang, and Nee 2019; Yu et al. 2020) to reduce information asymmetry and hinder opportunistic behaviour (Halldorsson et al. 2007). A digital twin requires interoperability and must include handling raw and preprocessed data from different sources and systems, such as the physical entity and supplementary systems (Schweiger and Barth 2023; Tao, Zhang, and Nee 2019; Valk et al. 2022). Standard interfaces such as RESTful APIs can extend the functions of the physical entity (Bellavista et al. 2023).

R5 Data privacy and security. Private data, not essential for shared manufacturing, should not be disclosed (Müller et al. 2022; Yu et al. 2020). Companies often fear a coopetition situation in which their sensible data is exposed to unknown companies or competitors (IPRI and IPH 2021; Müller et al. 2022). Data security enhanced by suitable governance and technological aspects is vital to achieve successful shared manufacturing (Stuckmann-Blumenstein et al. 2024) and digital twins (Durão et al. 2018). Data should be secure but accessible and always available in a digital twin (Bellavista et al. 2023).

R6 Memorisation. Tamper-proof and accessible data storage is crucial for stakeholders to enable reliable, trustful processes. It is also relevant for process visualisation and data analysis (Barricelli, Casiraghi, and Fogli 2019) in the digital twin (Bellavista et al. 2023; Valk et al. 2022). Documented processes can be usage times, failures, maintenance operations, or ownership, called historical static data (Barricelli, Casiraghi, and Fogli 2019; Javaid, Haleem, and Suman 2023), while descriptive static data contains characteristics of the physical system (Barricelli, Casiraghi, and Fogli 2019).

R7 Process visualisation. The clear virtual visualisation of processes allows to increase observability for users with an HMI (Schweiger and Barth 2023; Valk et al. 2022), addressing bounded rationality (Williamson 1987). A 3D visualisation of process data enables the monitoring of processes without being physically involved (Grieves 2015). To manage the abundance of available data in an automated production, an accessible and easy-to-operate Graphic User Interface (GUI) with underlying data analysis is required (Tao, Zhang, and Nee 2019). A GUI can show machine or product characteristics and historical data or issue warnings to facilitate monitoring processes and reduce reaction times to failures or unwanted process executions.

R8 Data analysis. To manage the process visualisation and to handle the amount of data generated by automated processes, data analysis is required (Barricelli, Casiraghi, and Fogli 2019; Valk et al. 2022). In shared manufacturing, this is essential for matching producers and consumers (Müller et al. 2022; Yu et al. 2020). Data analysis in a digital twin can monitor the physical entity of a running digital twin or predict the system's behaviour in what-if scenarios (Javaid, Haleem, and Suman 2023).

R9 Configurability. To reduce uncertainties in shared manufacturing, configurability of the virtual entity is required (Barricelli, Casiraghi, and Fogli 2019; IPRI and IPH 2021; Müller et al. 2022). Configurable features should allow for extensive testing of what-if-scenarios (Barricelli, Casiraghi, and Fogli 2019; Javaid, Haleem, and Suman 2023) to forecast or optimise the system behaviour (Stuckmann-Blumenstein et al. 2024; Valk et al. 2022). Configurations can be used to test different scenarios to decrease transaction costs. Furthermore, the virtual entity's parameters should allow adjustments to mirror the physical entity's characteristics (Bellavista et al. 2023; Tao, Zhang, and Nee 2019).

2.3. Related work and research gap

We conducted another systematic literature review to identify related work based on the presented requirements, leading to the results displayed in Table 2. The methodology is displayed in the appendix in Table A2.

Only five contributions cover the topics of shared manufacturing, digital twins and blockchain. Krämer

Table 2. Related work and fulfilled requirements.

Literature	Manufacturing / Production	Sharing / Collaboration	Digital Twin	Blockchain	Content	R1	R2	R3	R4	R5	R6	R7	R8	R9
Krämer et al. (2022)	✓	✓	✓	Ethereum	Design and case study of a blockchain-based digital twin for cyber-physical production systems	✓	✓	X	✓	✓	✓	✓	✓	X
Kuruppuarachchi, Rea, and McGibney (2023)	✓	✓	✓	No specific framework	Architecture and case study for collaborative ecosystems based on digital twin; blockchain is addressed as a design choice	X	X	X	✓	✓	✓	X	X	✓
Onaji et al. (2022)	✓	✓	✓	X	Review of and conceptual framework for a digital twin for collaborative manufacturing in Industry 4.0	✓	✓	✓	✓	X	✓	✓	X	✓
Shen et al. (2023)	✓	✓	✓	X	Architecture and implementation of digital twin-based smart manufacturing and a framework for online interactive simulation	X	✓	✓	✓	X	✓	✓	✓	✓
Wang et al. (2021)	✓	✓	✓	X	Architecture and case study for resource allocation in shared manufacturing, including a digital twin-based service model	X	✓	✓	✓	X	✓	✓	✓	✓
Hasan et al. (2020)	✓	X	✓	Ethereum	Proposal for and implementation of the creation of a digital twin with blockchain, including a security and cost analysis	✓	✓	✓	✓	✓	✓	✓	✓	✓
Krämer et al. (2023)	✓	X	✓	Ethereum	Concept and implementation of a reverse digital twin of a hybrid production with smart contracting	✓	✓	✓	✓	✓	✓	✓	✓	X
Liu et al. (2023)	✓	✓	✓	Based on the gossip protocol	Approach for and simulation of blockchain-based data sharing between digital twins in manufacturing	✓	✓	✓	✓	✓	✓	X	✓	✓
Nielsen, Ribeiro da Silva, and Yu (2020)	✓	✓	✓	Ethereum	Concept and case study for a digital twin of a matrix production using tokens to represent assets	✓	✓	✓	✓	✓	✓	X	✓	✓
Onwubiko et al. (2023)	✓	X	✓	Ethereum	Concept and implementation of a digital twin with blockchain in the context of Industry 5.0	X	✓	✓	✓	✓	✓	X	✓	✓
Sahal et al. (2021)	✓	X	✓	No specific framework	Concept and case study for the collaboration of digital twins using blockchain and predictive analysis	X	✓	✓	✓	✓	✓	✓	✓	✓
Suhail et al. (2022)	✓	X	✓	No specific framework	Review of the state-of-the-art of blockchain-based digital twins and proposal of a framework	X	✓	✓	✓	✓	✓	✓	✓	✓
Tao et al. (2022)	✓	✓	✓	No specific framework	Digital twin- and blockchain-based mechanism for service collaboration on the Industrial Internet Platform in manufacturing	X	✓	✓	✓	✓	✓	X	✓	✓
Yaqoob et al. (2020)	✓	X	✓	No specific framework	Survey on blockchain and digital twins in manufacturing	X	✓	✓	✓	✓	✓	✓	✓	✓

et al. (2023) present a concept and a case study of a blockchain-based digital twin for a cyber-physical production system using Ethereum. However, they neglect R3 and R9 by not building an extensive simulation or including simulated blockchains. Kuruppuarachchi, Rea, and McGibney (2023) focus on a digital twin for shared manufacturing and only cover blockchain as a design choice, but lack several requirements and do not implement any solutions. Liu et al. (2023) present an approach for blockchain-based data sharing between several digital twins in manufacturing and implement a simulation model. However, they do not consider a visualisation of the digital twin and do not focus on production processes but on exchanging product information between digital twins. Nielsen, Ribeiro da Silva, and Yu (2020) consider a digital twin for a flexible matrix production using Ethereum-based tokens to represent physical assets. While considering most requirements except visualisation, they focus specifically on tokens, do not implement their solution, or execute any testing. Tao et al. (2022) propose a digital twin- and blockchain-based mechanism for service collaboration on the Industrial Internet Platform in manufacturing. They do not consider a flexible manufacturing system or visualisation of the digital twin and do not implement their solution. Hasan et al. (2020) build a digital twin with blockchain, including all requirements of a digital twin, but they do not consider the sharing context.

All the contributions address connectivity (R4) and memorisation (R6), while adaptability (R1), data privacy and security (R5), and process visualisation (R7) are only partially addressed.

The reviewed literature shows that most scientific contributions have focussed on the potential of blockchain-based digital twins in (shared) manufacturing without any practical implementation. Only a few studies have included small-scale implementations with test networks. No research has been conducted to address the requirements R1 to R9 in a digital twin in shared manufacturing with smart contracts in an exhaustive concept and implementation, including benchmarks of different blockchain frameworks. Additionally, none consider using simulated smart contracting in this field to facilitate planning processes and design decisions. While blockchain simulators on their own have emerged, they are not considered in the context of manufacturing. Most existing blockchain simulators focus on PoW-based blockchains, need to be better maintained, have a complex structure, and need to consider newer consensus mechanisms and blockchain frameworks (Paulavicius, Grigaitis, and Filatovas 2021). These aspects make them unsuitable for a digital twin's design and planning processes in shared manufacturing. Thus, a simulation approach to facilitate the design decisions of smart contracting implementation and to predict the impact of smart contracting in manufacturing has yet to be developed. The development follows the DSR approach presented in Section 1 in Figure 1.

3. Addressing the research gap

This section presents a digital twin-based simulation model design for a shared manufacturing system with smart contracting. The design is developed based on the requirements identified in the previous Subsection 2.2 and the underlying research theories in Subsection 2.1.

R1 Adaptability. A hybrid production system with workstations is suitable to ensure an adaptable shared manufacturing system with flexible routing of incoming orders in the simulation. The workstations must be equipped to execute at least two different work steps, and at least two workstations can execute each step. Mobile robots execute the transport between the workstations for routing flexibility. A production schedule must include incoming jobs. The virtual entity of the digital twin must be adaptable to the changing manufacturing system regarding scalability, i.e. the number of workstations or mobile robots.

R2 Fidelity. All relevant resources of the shared manufacturing system, such as workstations, robots, or bins, must be virtually represented and updated to ensure the high fidelity of the digital twin, including near-real-time updates of the positions of all objects and their status and events. The physical entity must also be able to react to changes in the virtual entity.

R3 Data acquisition. To enable high fidelity, a motion-tracking system can acquire position data with high accuracy. At the same time, status updates can be sent between the virtual and physical entities via interfaces such as MQTT. The digital twin should have a bilateral connection to the blockchain to be able to acquire data from the blockchain.

R4 Connectivity. Automated scheduling is an effective way to process incoming orders, reducing costs and enabling high throughput. Due to the required computing power, this must be handled outside a blockchain. Blockchain can reduce information asymmetry and increase observability for most other data sharing in shared manufacturing, addressing the principal-agent problem. Smart contracts can reduce the administrative and manual work in shared manufacturing. Table 3 shows the design of three critical smart contract functions to integrate the financial flow into shared manufacturing via micropayments. The first function registers a work step with the smart contract, while the second one handles the payments of the costs associated with the work step

Table 3. Design of the smart contract functions.

Function	Conditions	Input	Output
Register the order with the smart contract	• Sender must be authorised • Provided order information must be complete • Resource for the order execution must be authorised	Order information, resource identification	Unique service number
Pay the order amount from resource A into the smart contract	• Sender must be authorised • Unique service number must exist • Unique service number must not have been used • Funds of the sender must be sufficient	Unique service number	Status of financial transaction
Pay the order amount from the smart contract to resource B	• Sender must be authorised • Sender must be a registered resource for the order • Unique service number must exist • Amount of the order must have been paid into the smart contract • Costs of the order must not be settled • Funds of the smart contract must be sufficient	Unique service number	Status of financial transaction

into the smart contract. The third function releases the money to a resource once a service has been fulfilled. This approach guarantees multiple stakeholders' connectivity to the financial processes and the virtual entity. Further connectivity must be established to the physical entity to acquire data for the virtual entity of the digital twin.

R5 Data privacy and security. Due to the blockchain's decentralised and distributed data storage, it is crucial to exercise due diligence when sharing data with multiple stakeholders. The design considers this requirement by sharing only anonymised data with the blockchain and selecting multiple blockchain frameworks with different characteristics, including public and consortium frameworks. The transaction costs of a blockchain interaction have to be considered, as modifying actions can have significant costs depending on the design of the smart contract and the blockchain framework. Data related to payment processes is critical to store on a blockchain (Krämer et al. 2021). The smart contract must consider several conditions to ensure data security and create reliable transactions. Senders and receivers of transactions should be registered with the smart contract to prevent unauthorised access to the system. Double spending on orders must be prevented by setting certain conditions, and funds for involved resources on the blockchain must be sufficient.

R6 Memorisation. The cryptographic principles of a blockchain enable immutable and accessible data storage. Machine usage and associated costs play an important role in process documentation in shared manufacturing. The system must track and store this data on a blockchain. Historical data, such as failures, maintenance operations, or usage times, can be written on a blockchain or documented solely in the digital twin.

R7 Process visualisation. Resources are virtually represented in a detailed 3D or a more straightforward 2D visualisation to increase production and observability of material supply. Transactions documented on the blockchain and in the digital twin must be visualised for users to increase transparency and accessibility of data. The visualisation comprises the process documentation and historical data for each resource, including its current task and key figures of the smart contract, including its account balance. Other key figures, such as machine utilisation or total covered distance for mobile robots, can be included. The GUI must be intuitive for all stakeholders, independent of their technical knowledge. It should be easy to operate and understand, given that multiple stakeholders interact with it in shared manufacturing.

R8 Data analysis. To monitor the impact of what-if scenarios during run-time, the key figures described in the visualisation must be calculated or called from the blockchain. Key figures can be calculated in the backend, such as the total transaction costs of each resource, the revenue of each resource and its utilisation. These figures allow for data analysis at first glance. Data can also be called from the blockchain, such as a transaction history or the account balance. Further analysis can be executed based on key figures saved during and evaluated after the simulation run of a what-if scenario, such as costs, validation times, or machine-specific key figures.

R9 Configurability. The simulation model should run in a digital twin mode with a visualisation comprising the physical and virtual entities of the digital twin and their interfaces to physical resources. As the physical entity limits the digital twin mode regarding the number of resources and the space in the production environment, this mode should be extended by a pure simulation mode with a scalable layout and without physical components. This mode tests configuration features to simulate scenarios before execution. The simulation should include different blockchain frameworks. Smart contracting should be executable on a blockchain test network to test features without generating substantial costs and to gain realistic validation times and transaction costs.

Blockchain test networks require real-time execution, making them unsuitable for complex production

Algorithm 1 Simulated smart contracting

Data:
$B = \{B_1, B_2, \ldots, B_n\} \wedge B_n = (B_g, B_t, B_i) \wedge n \in \mathbb{N}$ ▷ Blocks $T = (T_c, T_i)$ ▷ Transaction
$Q = \{T_1, T_2, \ldots, T_m\} \wedge (m \in \mathbb{N} \vee Q = \emptyset)$ ▷ Transaction queue
$B_i = \{T_1, T_2, \ldots, T_r\} \wedge (r \in \mathbb{N} \vee B_i = \emptyset)$ ▷ Transactions stored in blocks

loop every d seconds ▷ Delay between blocks
  $n \leftarrow n + 1$ ▷ Create new block
  $B_n.B_t \leftarrow currentTime$ ▷ Block gets timestamp
  for $T_m \in Q$ **do** ▷ Loop through transaction queue
    if $B_n.B_g + T_m.T_c \leq B_{g,max}$ **then** ▷ Check capacity of block
      $B_n.B_g \leftarrow T_m.T_c$ ▷ Add transaction costs to block
      $B_n.B_i \leftarrow T_m.T_i$ ▷ Add transaction information to block
      $Q \leftarrow Q \setminus T_m$ ▷ Remove transaction from queue
    else
      break ▷ Wait for new block
    end if
  end for
end loop

processes. Thus, simulated smart contracting should be included. To gain realistic results, the simulated smart contracting must closely mirror the behaviour of the blockchain test networks in terms of gas costs, validation times, and scalability. Algorithm 1 presents a novel concept for simulated smart contracting for Ethereum-based frameworks. The algorithm is used for state-changing functions, while read functions are unaffected by time delays and do not need to be simulated. Blocks B are created every d seconds, creating a blockchain-like structure. Each block B_n has a maximum capacity $B_{g,\max}$, measured in gas units. Each transaction consists of costs T_c and information T_i. When a transaction is received, it is stored in the queue Q, comprising T_m transactions. The gas consumed for a block B_g results from $\sum_{T_c \in B_g}$. Information from $T_m.T_i$ is stored in $B_n.B_i$. Block number n and timestamp B_t identify a block. T_m is then removed from Q and stored in a block.

4. Demonstration and validation

We simulate a use case to validate our design and analyse transaction costs and validation times. The use case is based on previous work by Krämer et al. (2023), who implemented a digital twin of a shared manufacturing system in their research lab.

4.1. Scenario

Shared manufacturing is mirrored by incoming orders from companies sharing the production space. Other manufacturers can provide resources; for instance, workstations and mobile robots can be rented without belonging to the company providing the production space. The material for assembly is supplied in a consignment warehouse, where the material belongs to the supplier until it is removed from the warehouse.

The layout for our scenario comprises three areas. On the left-hand side is an incoming assembly line feeding work in progress (WIP) to the matrix production system. Task robots wait in a queue until they pick up an incoming cart and transport it to the matrix production. Within the matrix production, the carts with WIP products are transported from workstation to workstation according to a production schedule. Waiting time in between is bridged in buffer positions at the workstation to ensure that a workstation is not blocked for other task robots. Once every work step of an order has been completed, a task robot transports its cart to the right-hand side and transfers it in the correct sequence according to the production schedule to the outgoing assembly line. The task robot returns and queues up.

An automated small parts warehouse provides material for the assembly tasks at the workstations. Pick robots wait in a queue until they pick up a cart with material at the retrieval points. The production schedule determines which retrieval point is selected for which cart. The pick robot transports its cart to the workstation according to the production schedule, waits for a few seconds for the material to be removed, and then returns to a scheduled storage point. The empty cart is returned to the small parts warehouse, and the pick robot queues up to await a new task.

4.2. Implementation and technical details

This section describes how the simulation model implemented in the game engine software Unity[1] (version 2020.3.27f1) is composed based on the requirements R1 to R9. We chose Unity due to its flexibility in implementing interfaces and visualising the scenario.

To achieve adaptability (**R1**) in the matrix production, we set the required minimum of workstations to four. Mobile robots move in the production area without predefined tracks. They calculate their shortest path in Unity based on the A∗-Algorithm (Hart, Nilsson, and Raphael 1968) using the Unity AI Navigation Package (Unity-Technologies 2023). The number of tasks and pick robots corresponds to the number of workstations times two to allow for a constant production supply.

Distances between workstations and transfer points are calculated to determine the production schedule and costs for each work step, assuming an underlying pay-per-use model. The work of Kaiser et al. (2022) is adapted to calculate the production schedule. Input parameters for the algorithm are the previously described properties of the production and the sequence of the incoming and outgoing assembly lines. The schedule is calculated for a batch of the following five jobs at a time, considering the previous jobs, and the following calculation is triggered by the end of the previous calculation. The batch size and the calculation trigger are adjustable, increasing flexibility and allowing a short-term calculation of the production schedule as long as the calculation is finished before a resource starts.

Only data on the production schedule and associated costs relevant to payment processes is shared on the blockchain (**R4**). Each job comprises several operations and requires several resources, such as a workstation for assembly, a pick robot for material transport, and a task robot for the WIP product. Workers' data is not stored on-chain due to data privacy (**R5**).

To perform a comparative analysis of different Ethereum-based frameworks (**R8**), we set up three test networks: a private permissionless PoW to represent the Ethereum 1.0 Mainnet, a private and permissionless PoA for the current Ethereum 2.0 Mainnet, and a private enterprise class consortium PoA network to reflect a solution for the enterprise environment. The detailed configuration of each blockchain test network is explained in the appendix in Table A3.

Our simulation model keeps track of resource-specific data during run-time and enables the evaluation of recorded data post-run-time (**R6**). During run-time, the account balance and the current task are displayed, including a history of previous tasks. Key figures, such as transaction costs, are calculated and documented. The model records more data off-chain for extensive analysis, including the payer, the payee of a transaction and the smart contract function. Task costs are documented, and all transaction costs are separated by estimated and used gas units, the gas price, and the delta in the account balance resulting from a transaction. Time-related key figures are documented, such as the service's start and end times and the blockchain-specific validation times of transactions.

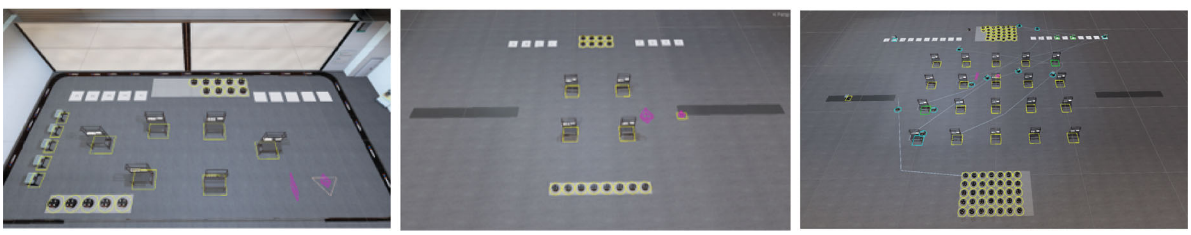
The simulation model includes ten smart contract functions as shown in Table 4. The main functions are modifying functions for payment processes: Adding a service to the blockchain (I), reserving it (II), and settling it (III) after a performed service. The input parameters and emitted event data show which data and data type is required to execute the functions. The main functions are built according to the conditions defined in Section 3. Auxiliary functions support the main functions. To register a resource, its account's public key, designation, and category are required in auxiliary function (IV) due to data security (**R5**). To check with the main functions if a resource is registered, auxiliary function (V) is called. If a resource is no longer available or permitted, its access is deleted by auxiliary function (VI). Auxiliary functions (VII) and (VIII) check the balance of accounts and the smart contract. Auxiliary function (IX) resets the balance of the smart contract. It is necessary if simulation runs are manually interrupted and money paid into the smart contract is not disbursed automatically.

An HMI visualises the manufacturing system and the starting menu of our simulation (**R7**). Detailed information on the HMI and implemented Graphic User Interface (GUI) are in the appendix in Figure A1. Visualisation features support understanding the observability and current utilisation of resources by defined shapes and a colour scheme. A dynamic dashboard enables user interaction with the simulation model during run-time.

Key figures such as the account balance are taken from the blockchain in ether and offset against the exchange rate euro to ether. The other key figures are calculated in Unity based on data from the blockchain. Each amount transferred to the blockchain with smart contract main function (II) comprises the costs of a task and estimated transaction costs. For mobile robots, costs are calculated based on transportation time, depending on the covered distance; for workstations, based on the processing time; and for carts, based on the material costs. Transaction costs are gas units for a transaction multiplied by the gas price for each gas unit. To compare gas fees, we fix the gas price at 1 gwei. As the actual gas units depend on the input parameters and the network load, they are only estimated before execution. As the transaction costs are

Table 4. Implemented smart contract functions.

No.	Function	Type	Sender	Input parameters	Emitted event data
			Main functions		
(I)	`AddService`	Modify	SchedulingManager	`(uint jobId, string operationId, address resource, uint costs)`	`(uint serviceNumber)`
(II)	`ReserveService`	Modify	SchedulingManager	`(uint serviceNumber)`	`(uint serviceNumber, bool successful)`
(III)	`SettleService`	Modify	Resource	`(uint serviceNumber)`	`(uint serviceNumber, bool successful)`
			Auxiliary functions		
(IV)	`AddUser`	Modify	SchedlingManager	`(address resource, string designation, string category)`	`(address resource, string designation, string category)`
(V)	`IsUserRegistered`	View	SchedulingManager	`(address resource)`	`(bool exists)`
(VI)	`DeleteUser`	Modify	SchedulingManager	`(address resource)`	`(address resource, address deletedByResource)`
(VII)	`AccountBalance`	View	Resource	`()`	`(uint256 balance)`
(VIII)	`ContractBalance`	View	SchedulingManager	`()`	`(uint256 balance)`
(IX)	`ResetSmartContract`	Modify	SchedulingManager	`()`	`(bool successful)`
(X)	`ResetBalance`	Modify	SchedulingManager	`()`	`(bool successful)`



(a) Physical model (b) Layout with four workstations (c) Layout with 20 workstations

Figure 2. Physical model and scaled layouts of the hybrid production in the simulation model (screenshots from the Unity model).

burned during the process, the corresponding resource only receives the task costs.

With configuration options, the user determines if physical resources should be included as shown in Figure 2(a) or if the simulation model should run in a scalable stand-alone mode (**R9**) as displayed in Figure 2(b,c). The simulation model allows for a configuration of the production layout from four to 20 workstations, with the production layout automatically adjusting to the selected number. Several interfaces to the physical entity are integrated to realise the bilateral information flow between the physical and virtual entities (**R3, R4**) and to ensure sufficient fidelity of the simulation model (**R2**). An exemplary implementation of the digital twin with its physical entity in a research lab is demonstrated in Krämer et al. (2023) and will not be discussed here. The .NET library Nethereum[2] connects the model to the blockchain. The user can choose between three blockchain test networks and simulated smart contracting. Simulated Smart Contracting is made by instantiating the described pseudo Algorithm 1 and created with C# in Unity 2020.3.27f1. The structure and mode of operation are entirely derived from the pseudo algorithm.

4.3. Design of experiments

To test the implementation, we evaluate how the system's scalability impacts the execution time and required gas units, and conduct experiments on the blockchain test networks to benchmark them. We investigate how accurately the simulated smart contracting substitutes the blockchain test networks regarding gas units and validation times. To guarantee the accuracy of the evaluated system, we conduct a preliminary study and evaluate how accurately the production schedule is executed within the simulation model.

To gain comparable results, we choose five fixed configurations A to E for all studies as shown in Table 5. A production schedule is calculated for each configuration. All configurations A to E are simulated with each blockchain test network Ethereum PoW, Ethereum PoA, and Quorum BFT. For each work step, five services are added to the blockchain: one for a pick robot, its cart

Table 5. Configuration of five scenarios for the experiments.

Scenario	Number of							Simulation duration per run [seconds]
	Work-stations	Task robots	Pick robots	Pick carts	Retrieval / Storage points	Jobs	Simulation runs	
A	4	8	8	12	4/4	100	1	40187
B	8	16	16	24	6/6	90	3	19317
C	12	24	24	36	7/7	70	5	10551
D	16	32	32	48	8/8	65	6	10548
E	20	40	40	60	10/10	60	11	8089

(material), and a workstation, and two for a task robot for the workstation and its buffer position. Each service is added, reserved, and settled on the blockchain with the presented main functions. The auxiliary functions are not considered as they are either view functions without delays or costs, or they are modify functions only called once for each resource, such as registering.

The five scenarios are tested with the following key figures for each test network:

- Gas units per transaction: Each transaction causes an individual number of gas units, depending on the smart contract function, its input and the system load.
- Gas units per block: They comprise the sum of gas units of all transactions written into a specific block. Considering the gas limit of a block, this figure is essential to analyse if all transactions are processed or if scaling issues arise.
- Transactions per block: This figure measures how many transactions a block comprises, is essential to study scaling effects, and shows how balanced the system load is regarding incoming requests. As each modifying transaction needs gas units, the gas units and transactions per block loosely correlate.
- Validation time per transaction: This is the period from sending the Unity request to receiving a receipt for a successful blockchain transaction in Unity. This figure is vital for correctly executing the production schedule.

Simulation experiments are terminating with a defined end, or non-terminating, reaching a steady state after a transient phase (Wenzel 2018). Our simulation model is a non-terminating system as the production system has to pass the transient phase by filling the workstations and can then run for an infinite amount of time, depending on incoming orders. To determine the length and number of simulation runs, two methods are standard: In the replicate-and-delete method, many shorter simulation runs are performed in which the transient phase is removed from the data. The batch-means method involves one long simulation run in which the transient phase is cut off, and the remaining simulation time is divided into equal segments called batches (Wenzel 2018). We use the Chernoff Bound to determine the number of batches from which we can gain statistically reliable results.

Definition 4.1 (Chernoff Bound for sample complexity): Let $\epsilon \in (0, 1)$ be the accuracy and $(1 - \delta) \in (0, 1)$ be the confidence level, then the one-sided additive Chernoff bound for sample complexity is defined as follows, where N is the number of needed samples (Calafiore, Dabbene, and Tempo 2011):

$$N \geq \frac{1}{2\epsilon^2} \ln \frac{2}{\delta}$$

For our simulation experiments, we choose $\epsilon = 0.1$ and $\delta = 0.05$ and require at least 184.444 samples per experiment, i.e. 185 samples. Thus, our results stay within 10% from the result of an infinite number of samples with a confidence of 95%.

The transient phase must be excluded from the samples to avoid distortion of the results. We determine the transient phase based on the production schedules. For each scenario, the system load is steady if all workstations are occupied, i.e. when scenario A's fourth job enters production. The transient phase towards the end of the simulation run is determined as an analogue, i.e. for scenario A, job 96 has to leave the system. The duration of the transient phase is translated into specific blocks in a scenario. We choose at least 40 samples for each batch, i.e. 40 blocks per batch. Depending on the production schedule and the duration of the transient phase, several simulation runs are required for each scenario to gain the same number of samples per batch. We resort to the replicate-and-delete method in combination with the batch-means method, resulting in the number of simulation runs displayed in the last column of Table 5. The simulated smart contracting is tested with the same scenarios and key figures as the test networks to gain comparable results.

All experiments with the simulation model are conducted on a computer with sufficient resources. At the same time, the blockchain test networks run on a separate server to avoid performance distortions by CPU-intensive consensus mechanisms such as PoW. The appendix in Table A4 gives a detailed overview of their specifications.

Table 6. Results of the pre-study.

	Start deviation				End deviation				Tardiness			
Scenario	Mean	Median	Max	Min	Mean	Median	Max	Min	Mean	Median	Max	Min
A	−0.08 s	−0.08 s	0.05 s	−0.16 s	44.86 s	26.05 s	190.05 s	−243.76 s	−4.32 s	−5.47 s	0.77 s	−29.81 s
B	−0.07 s	−0.06 s	0.18 s	−0.32 s	41.26 s	22.75 s	188.32 s	−151.34 s	−3.96 s	−3.87 s	0.79 s	−64.41 s
C	−0.04 s	−0.07 s	2.99 s	−1.56 s	43.82 s	24.25 s	190.81 s	−59.72 s	−4.28 s	−4.2 s	1.13 s	−35.8 s
D	−0.08 s	−0.07 s	0.16 s	−0.31 s	44.48 s	25.93 s	190.24 s	−31.29 s	−3.69 s	−3.49 s	1.73 s	−50.16 s
E	−0.06 s	−0.08 s	8.99 s	−7.11 s	41.69 s	22.57 s	191.15 s	−123.87 s	−3.93 s	−3.53 s	1.8 s	−38.8 s

4.4. Experiments and evaluation

We present the results of the production schedule's pre-study, followed by the experiments with the blockchain test networks and the simulated smart contracting.

4.4.1. Pre-study: production schedules

This pre-study aims to analyse the consistency of the simulation model execution times with the calculated times, to reveal the influence of the test networks on the production schedule, and to show necessary adjustments to the calculation algorithm. For the analysis, we use three time stamps: the arrival time of the task robots at the workstation and the start and end times of the pick robots. They are based on the production schedule and the time stamps of the simulation run. Indicators for each line of the production schedule are the following:

- The start/end deviation is the difference between the pick robots' calculated start/end time and the actual start/end time in the simulation model.
- The tardiness is the difference between the calculated arrival time at the workstations and the actual arrival time of the task robots.

A positive value indicates the simulation model's early start or arrival, and a negative value indicates a late start or arrival. Table 6 shows each indicator and statistical value.

The mean start deviation range is −0.04 s to −0.08 s due to slight variations in the computation time, which may be disregarded. The pick robots in the simulation start at the calculated times. While the start deviation is negligible, the mean value for the end deviation ranges from 41.26 s to 44.86 s, and outliers fall between −243.76 s and 191.15 s. These values suggest that the pick robots complete their tasks in the simulation earlier than calculated. The calculation uses static values for the path, and to avoid any delays that affect subsequent tasks, a buffer time is added to the calculated time. The outliers with negative values result from robot deadlocks, causing late arrivals. This problem arises from the collision avoidance in Unity, which cannot solve deadlocks in rare cases. The mean value of the tardiness is between −3.69 s and −4.32 s, meaning that the task robots arrive at the workstations several seconds late. Thus, the production schedule algorithm should consider the robot's roughly four-second delay. Outliers up to 64.41 s are also a result of the path planning algorithm.

4.4.2. Test networks

The measurement values of the experiments with the blockchain test networks and simulated smart contracting are displayed in Table 7. While all scenarios provide data for Ethereum PoA and Quorum BFT, we faced difficulties gathering data for Ethereum PoW due to its consensus mechanism. We could gather data from 49 batches, which shows that Ethereum PoW values deviate more than Ethereum PoA values from scenario A. Scenarios B to E faced scalability issues and could not provide valid data. Significant values and comparisons of the experiments are explained in the following.

While addService consumes the most *gas units per transaction* for all three test networks, reserveService consumes about 60% less, and settleService 67% less. AddService is the most expensive function due to the high number of input variables, while the other two functions only transfer money. The measurement values show that the deviation increases with higher-scaled scenarios for all smart contract functions of the test networks due to higher input and system load variability. While addService and reserveService show lower standard deviations of up to 32 gas units, settleService has comparatively high standard deviations of up to 487 gas units. This deviation results from an optional feature in the settleService function that allows for a financial remittance if the robots' paths differ from those calculated for the production schedule. The consumed gas units also differ depending on the blockchain framework. Compared to Ethereum PoA, Quorum BFT consumes less gas for each smart contract function due to the consensus mechanism and the consortium framework. AddService consumes about 0.5% less, while reserveService and settleService consume about 14% and 11% less. Regarding Ethereum PoW, one speciality is that while the measurement values look similar to Ethereum PoA, some transactions did not fit into a block as the gas usage increased to fees above the gas limit per block. This resulted in a few

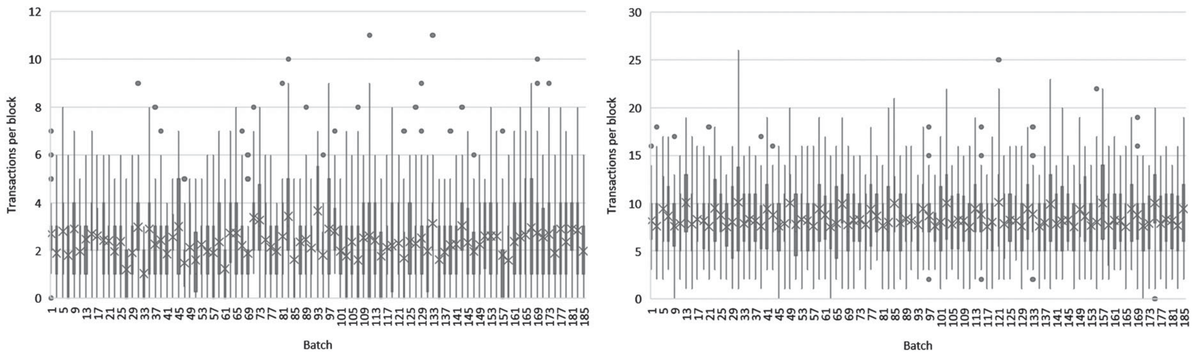
Table 7. Measurement values for five scenarios and three blockchain test networks and simulated smart contracting.

Key figures	Batch values	Ethereum PoA					Ethereum PoW				
		Scenario					Scenario				
		A	B	C	D	E	A	B	C	D	E
Transactions per block	Mean	2.30	4.46	6.91	6.46	8.24	6.20	–	–	–	–
	Median	2.35	4.43	6.75	6.28	8.10	6.32	–	–	–	–
	Minimum	0.75	2.85	5.08	5.00	6.95	2.04	–	–	–	–
	Maximum	3.66	6.15	9.32	8.65	10.13	10.36	–	–	–	–
	Standard deviation	0.52	0.68	0.95	0.84	0.82	1.89	–	–	–	–
Gas units per block	Mean	172410.00	334839.96	518712.15	485349.98	618788.24	492802.47	–	–	–	–
	Median	174555.85	333234.65	519943.00	471695.95	602961.80	467598.25	–	–	–	–
	Minimum	54012.35	220381.85	385848.15	369702.05	507304.73	147292.70	–	–	–	–
	Maximum	271165.66	466456.90	701661.90	659170.21	763536.50	1023042.73	–	–	–	–
	Standard deviation	39486.12	51806.00	69659.59	65195.29	64562.05	180293.88	–	–	–	–
Gas units per transaction (addService)	Mean	129675.46	129675.39	129675.05	129674.83	129675.03	129675.55	–	–	–	–
	Median	129675.54	129675.43	129675.11	129674.85	129675.05	129675.47	–	–	–	–
	Minimum	129671.33	129673.83	129673.60	129673.00	129674.09	129673.64	–	–	–	–
	Maximum	129677.14	129676.47	129676.38	129675.77	129675.76	129677.08	–	–	–	–
	Standard deviation	0.81	0.48	0.50	0.45	0.32	0.76	–	–	–	–
Gas units per transaction (reserveService)	Mean	52299.68	52301.31	52309.34	52307.93	52309.62	52299.96	–	–	–	–
	Median	52300.00	52300.00	52311.89	52311.88	52311.90	52300.00	–	–	–	–
	Minimum	52288.00	52299.68	52299.84	52299.81	52299.87	52299.48	–	–	–	–
	Maximum	52300.00	52312.00	52312.00	52312.00	52312.00	52300.00	–	–	–	–
	Standard deviation	1.76	3.82	4.95	5.68	4.76	0.13	–	–	–	–
Gas units per transaction (settleService)	Mean	42959.03	43076.65	43016.02	43266.65	43167.38	43240.85	–	–	–	–
	Median	42930.63	43057.33	43026.02	43225.53	43120.33	43102.73	–	–	–	–
	Minimum	42540.00	42551.74	42552.00	42851.46	42551.87	42551.45	–	–	–	–
	Maximum	44236.44	43958.14	43528.73	43946.02	43674.87	44388.12	–	–	–	–
	Standard deviation	385.13	295.97	209.72	238.66	246.90	487.07	–	–	–	–
Validation time per transaction [seconds]	Mean	4.12	4.24	4.00	4.13	3.97	13.64	–	–	–	–
	Median	4.11	4.38	3.89	4.16	3.89	13.60	–	–	–	–
	Minimum	3.38	3.61	3.56	3.62	3.50	8.39	–	–	–	–
	Maximum	4.65	4.76	4.70	4.68	4.71	23.01	–	–	–	–
	Standard deviation	0.20	0.33	0.28	0.25	0.26	2.97	–	–	–	–

(continued)

Table 7. Continued

Key figures	Batch values	Ethereum PoA					Ethereum PoW				
		Scenario					Scenario				
		A	B	C	D	E	A	B	C	D	E
Transactions per block	Mean	2.30	4.46	6.91	6.47	8.24	2.30	4.46	6.91	6.47	8.24
	Median	2.33	4.45	6.75	6.28	8.10	2.35	4.43	6.83	6.25	8.08
	Minimum	0.72	2.88	5.10	5.03	6.90	0.76	3.00	5.25	5.08	6.90
	Maximum	3.63	6.15	9.33	8.58	10.18	3.63	6.18	9.43	8.60	10.18
	Standard deviation	0.52	0.68	0.95	0.84	0.81	0.52	0.67	0.93	0.82	0.83
Gas units per block	Mean	162888.96	316031.89	490185.72	458923.00	584826.61	172543.28	334728.01	518659.93	485246.80	619060.10
	Median	165903.10	314022.05	489613.60	447084.00	570039.50	176429.78	330427.73	515634.83	474000.70	603307.80
	Minimum	49956.67	209904.65	364020.55	350672.87	479074.25	55575.37	225691.43	386466.20	376529.45	505928.10
	Maximum	262976.49	440567.40	668372.90	618865.25	726839.08	275955.13	467038.00	707675.60	654897.90	764174.13
	Standard deviation	37777.65	49033.29	66205.19	62420.94	61015.67	40189.19	51488.32	68861.62	64891.78	64996.37
Gas units per transaction (addService)	Mean	128984.47	128983.58	128982.21	128980.97	128982.07	129675.00	129675.00	129675.00	129675.00	129675.00
	Median	128984.76	128983.68	128982.55	128981.06	128981.82	129675.00	129675.00	129675.00	129675.00	129675.00
	Minimum	128962.44	128974.59	128974.08	128971.33	128977.13	129675.00	129675.00	129675.00	129675.00	129675.00
	Maximum	128994.95	128988.86	128988.00	128985.84	128986.24	129675.00	129675.00	129675.00	129675.00	129675.00
	Standard deviation	4.29	2.72	2.61	2.45	1.70	0.00	0.00	0.00	0.00	0.00
Gas units per transaction (reserveService)	Mean	45111.78	45147.38	45162.36	45165.03	45150.84	52300.00	52300.00	52300.00	52300.00	52300.00
	Median	45112.00	45174.88	45176.00	45176.00	45175.40	52300.00	52300.00	52300.00	52300.00	52300.00
	Minimum	45108.80	45110.48	45111.15	45111.09	45111.35	52300.00	52300.00	52300.00	52300.00	52300.00
	Maximum	45112.00	45176.00	45176.00	45176.00	45176.00	52300.00	52300.00	52300.00	52300.00	52300.00
	Standard deviation	0.63	31.87	26.02	23.97	31.29	0.00	0.00	0.00	0.00	0.00
Gas units per transaction (settleService)	Mean	38426.90	38536.60	38524.56	38710.94	38629.93	43099.00	43099.00	43099.00	43099.00	43099.00
	Median	38420.00	38520.41	38538.02	38692.00	38618.29	43099.00	43099.00	43099.00	43099.00	43099.00
	Minimum	38161.44	38164.00	38164.00	38373.38	38163.29	43099.00	43099.00	43099.00	43099.00	43099.00
	Maximum	39172.76	39239.58	38879.96	39190.91	39055.32	43099.00	43099.00	43099.00	43099.00	43099.00
	Standard deviation	255.30	198.89	150.58	159.93	180.23	0.00	0.00	0.00	0.00	0.00
Validation time per transaction [seconds]	Mean	4.74	4.77	4.85	4.76	4.60	4.03	4.09	4.03	4.01	4.10
	Median	4.74	4.79	4.90	4.75	4.49	4.03	4.10	4.03	4.02	4.12
	Minimum	4.16	4.20	4.32	4.27	4.13	3.75	3.71	3.78	3.78	3.79
	Maximum	5.32	5.38	5.31	5.32	5.29	4.32	4.34	4.34	4.26	4.30
	Standard deviation	0.24	0.32	0.22	0.28	0.29	0.12	0.11	0.11	0.08	0.12



(a) Scenario A (b) Scenario E

Figure 3. Transactions per block in the Ethereum PoA test network.

transactions that did not go through. The measurement values in Table 7 do not consider these transactions.

The *transactions per block* increase with the scalability of the scenarios. For Ethereum PoA and Quorum BFT, the numbers are similar and increase from a mean of 2.30 transactions per block in scenario A to 8.25 transactions per block in scenario E. Other statistical values behave similarly. As the numbers refer to average batch values, they must be supplemented by the specific consideration of values within the batches. Figure 3 displays Ethereum PoA transactions per block in a boxplot diagram for each batch to visualise the variability within a batch. For scenario A, at least one block has no transactions per batch, as transactions are issued according to the production schedule. An empty block is still produced if no transactions occur. Across all batches, the maximum number of recorded transactions fluctuates between 3 and 11 transactions per block for this scenario. In the highest-scaled scenario E with 20 workstations, the maximum is 26 transactions per block, while the minimum number of transactions per block differs between 0 and 5 transactions. Quorum behaves similarly for all scenarios as Figure 5(b) shows. The transactions per block are nearly identical, with a difference of −0.12% for scenario A and 0.19% for scenario E.

In contrast to Ethereum PoA and Quorum BFT, for Ethereum PoW, the average number of transactions in scenario A is thrice as high as the other two test networks as shown in Figure 5(a). Fewer blocks are mined in the same period due to an irregular block time, on average with 12.63 s 2.53 times higher than the fixed block time of the other test networks. Fewer blocks are mined simultaneously, resulting in 270% more transactions per block than Ethereum PoA in scenario A due to the same underlying production schedule.

The *gas units per block* correlate with the transactions per block, considering the variability of gas units for a specific transaction as shown in Figure 4. The scattering of the measurement values increases with the number of transactions included in one block. The standard deviation increases 57.69% for transactions per block and 63.51% for gas units per block from scenario A to scenario E in the Ethereum PoA test network. The higher deviation in scenario E in the number of transactions results from more tasks according to the production schedule compared to scenario A, leading to a higher deviation in the gas unit values. Quorum scales similar to Ethereum PoA as shown in Figure 5(b), but consumes less gas. For instance, in scenario A, Quorum consumes about 6% less gas on average. Ethereum PoW has more transactions and gas units per block due to the irregular block time displayed in Figure 5(a). Both key figures still show a correlation between the number of transactions and gas units per block.

To avoid scalability issues, the maximum gas values should be below the gas limit of a block. Ethereum PoA needs slightly more gas units than Quorum, and the systems behave similarly; thus, we will only examine Ethereum PoA in detail. As some blocks in scenario A do not comprise any transactions, the minimum amount of gas units for blocks in each batch is zero, as displayed in Figure 6. The maximum number of gas units per block across all batches is 1132276 gas units, which is below the target of 15 million gas units per block. The system has not reached its scalability limit for four workstations, and all incoming transactions are processed. For the higher-scaled scenario E, the maximum gas units per block are 2369188 gas units, which is still lower than the target of 15 million gas units, ensuring that the scalability of the test network is sufficient for the presented use case. For Ethereum PoW, the higher gas units per block amount to a maximum of 4596638 gas units per block in any evaluated batch for scenario A. This value is much higher than the maximum gas units for Ethereum PoA, even for scenario E, but still below the target and limit of gas units of a block.

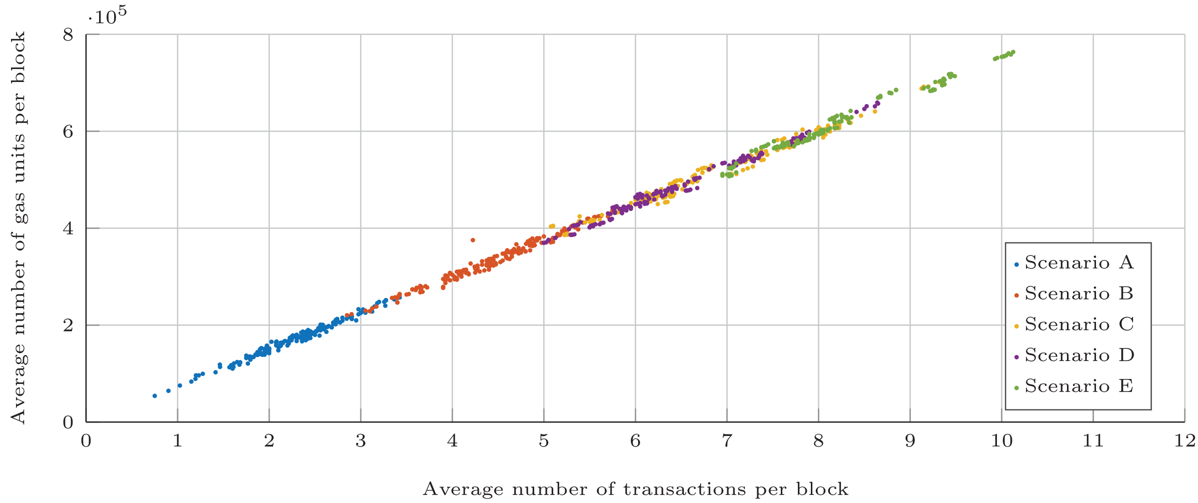


Figure 4. Transactions and gas units per block in the Ethereum PoA test network.

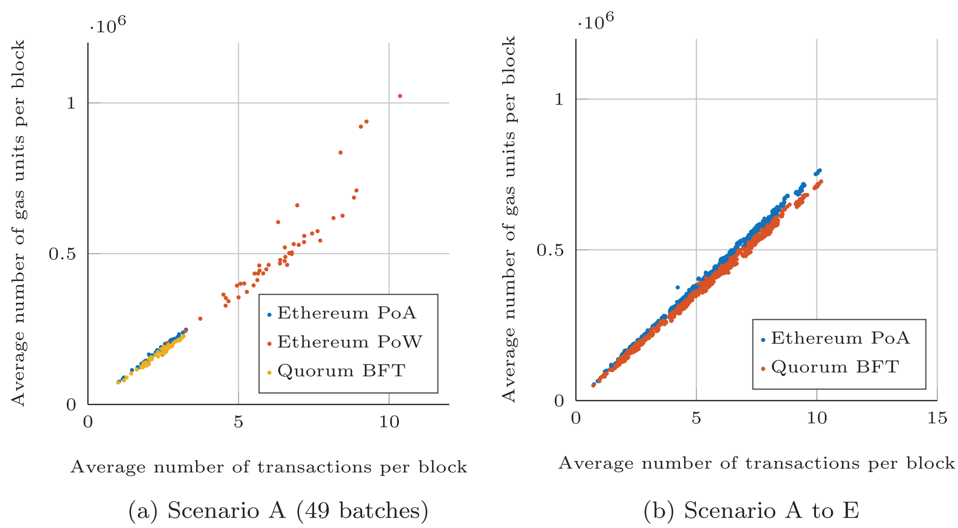


Figure 5. Number of transactions and corresponding gas units per block.

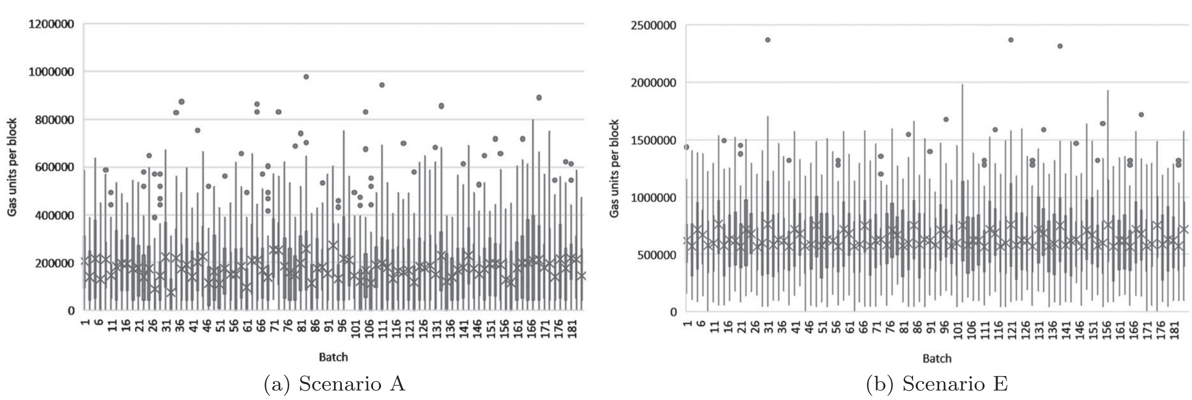


Figure 6. Gas units per block in the Ethereum PoA test network.

Figure 7(a) displays the *validation time per transaction*. As Ethereum PoA and Quorum BFT have fixed block times of 5 s, transactions are validated within a maximum of 5 s. Some maximum values are slightly above 5 s, which results from the measurement, including the processing time via the interface between Unity and the blockchain test network. Still, a mean of 2.5 s would have been expectable. The validation time depends on the

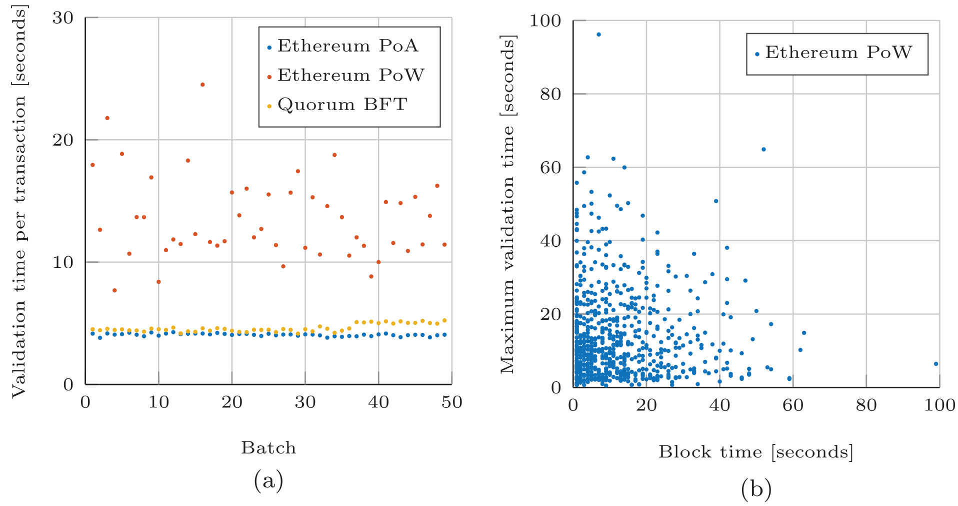


Figure 7. Validation time per transaction in Scenario A.

time of entry of a new task into the system. A mean of 2.5 s would require an equal distribution of incoming transactions. As our production schedule often triggers transactions with a temporal difference of 5 s, the values differs from the expected average.

While Ethereum PoA and Quorum BFT have average validation times around their block time, for Ethereum PoW, the validation times are about thrice as high (331.07% compared to PoA and 287.76% compared to QBFT). The standard deviation of 2.97 s across all batches in scenario A is also higher than for Ethereum PoA (0.20 s) or Quorum BFT (0.24 s). The unpredictable behaviour of the validation time is expected from a PoW consensus mechanism in which a specific validation time cannot be guaranteed due to the irregular block time. The block time fluctuates around an average of 12.63 s. The maximum values of block times in batches are between 22 s and 99 s, showing the wide scattering of measurement values with a standard deviation of 12.23 s across all batches for the block time. Figure 7(b) shows that multiple blocks contain transactions that have a longer validation time than block time, for instance, block 222344 with 7 s block time and a transaction with 96.19 s validation time. This imbalance means that the transaction was sent to the blockchain way before the block it was included in was created, which indicates scalability problems during the simulation. Not every incoming transaction could be processed immediately in the next block.

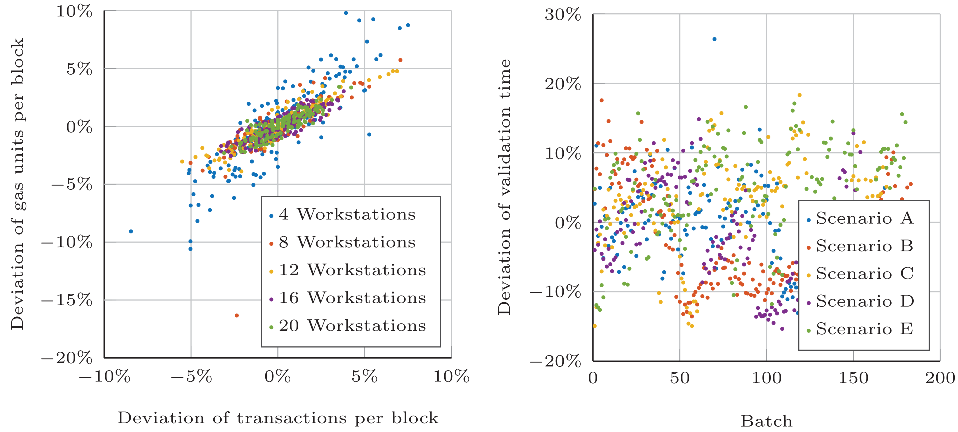
The evaluated data shows that Ethereum PoW is unsuitable for shared manufacturing due to the high uncertainty regarding validation times, consumed gas units and scalability. Fast replanning in a defined time horizon is not possible, and it cannot be guaranteed that all transactions go through. This behaviour is critical for manufacturing systems regarding the efficiency of processes, the error rate, and the trustworthiness of data. As it counteracts the purpose of smart contracting in shared manufacturing, we will not further consider Ethereum PoW. The Ethereum PoA and Quorum BFT evaluations show that both test networks behave similarly with slight deviations. While Ethereum PoA is slightly faster, Quorum BFT consumes fewer gas units per transaction. For further development, we decided to apply simulated smart contracting to Ethereum PoA, which is more prevalent in applications. The following simulation can be adapted to represent Quorum BFT as well.

4.4.3. Simulated smart contracting of ethereum poA

The following evaluation shows that the simulated smart contracting is functional and generates similar outcomes to the Ethereum PoA test network.

The *gas units per transaction* for the simulated smart contracting are based on the average value of Ethereum PoA across all scenarios. Using a BigInteger value to keep the simulated smart contracting compatible with the test networks, the values are rounded to the next integer. For addService, this amounts to 129675 gas units, for reserveService to 52300 gas units and settleService to 43099 gas units. To simplify the simulated smart contracting, deviations of gas units within one function are neglected in our simulated smart contracting.

The measurement values show that the simulated *transactions per block* are similar to the test network as displayed in Figure 8(a). On average, transactions deviate 0.07% across all scenarios, and gas units deviate 0.03% from the test network. The average and mean values and the standard deviation are identical in scenario A. Only minimum and maximum values show slight deviations of



(a) Transactions and gas units per block in scenario A to E

(b) Validation time per transaction in scenario A to E

Figure 8. Deviation of simulated smart contracting from the Ethereum PoA test network.

+1.72% and −0.93%. For single batches, a maximum of 8.46% for transactions and 16.34% for gas units exists. For *gas units per block*, the deviation can be higher than for transactions due to the static gas units for each smart contract function. The results of the other scenarios are similar. The data of transactions and gas units per block show a similar correlation as the Ethereum PoA test network.

The average *validation time per transaction* is below the block time of 5 s and that the values are similar to those of the test network. The percentual deviation of the simulated values and the values from the test network is displayed in Figure 8(b). No validation time of a batch deviates more than 26.37 %, with average values for scenarios fluctuating between −4.04% for scenario B and 3.11 % for scenario E, which equal −0.17 s and 0.12 s.

5. Conclusion, implications and outlook

The contribution concludes by summarising the work, pointing out managerial insights and giving an outlook for future research.

5.1. Conclusion and theoretical advancements

To select an appropriate blockchain framework and ensure efficient execution of manufacturing tasks, simulating a production with smart contracting prior to implementation is crucial. This contribution introduced a novel notion to simulate the impact of smart contracting on the production supply in shared manufacturing. Ethereum-based test networks exemplified the approach. We proposed an innovative method of simulated smart contracting, which allows us to analyse the effects of smart contracting in time-lapse in the simulation. Empirical evidence was presented based on simulation experiments, demonstrating the scalability and feasibility of smart contracting in this context. This approach contributes to overcoming information asymmetries and reducing the principal-agent problem.

The concept development followed the design science research approach. We identified ten requirements based on the existing literature, including adaptability, connectivity, and configurability, for a digital twin in shared manufacturing. In doing this, we extend the theoretical literature presented in Subsection 2.2 in Subsection 1 concerning the design of a digital twin in shared manufacturing. To address the requirements, blockchain and smart contracting were introduced by developing ten smart contracting functions, of which three main functions handle the financial flow of an order from reservation to payment, and seven auxiliary functions support the data handling.

Implementing the concept created a configurable simulation model in Unity with the blockchain frameworks Ethereum PoA, Ethereum PoW, and Quorum BFT. The benchmark revealed that Ethereum PoA and Quorum BFT processed all incoming transactions reliably and within a predefined period due to the stable test network setup and the fixed block time of 5 s. Both frameworks scale well in shared manufacturing. In contrast, Ethereum PoW provided highly unreliable block and validation times, higher gas costs and scalability issues due to the PoW consensus. We do not recommend Ethereum PoW for a productive manufacturing environment. Unpredictable validation times may disturb the correct execution of schedules and reduce the production and supply chain efficiency. The implemented concept and conducted experiments serve as an extension to the existing literature analysed in Table 2 by providing a holistic simulation model addressing all identified requirements, by defining indicators for the performance measurement of blockchain frameworks, and by giving benchmark values of different blockchain frameworks.

We further demonstrated that simulated smart contracting gives results with deviations below 5% compared to the Ethereum PoA test network. Simulated smart contracting allows for predictive and precise calculations of blockchain applications before execution and enables the simulation in time-lapse. This innovative method enriches existing literature on blockchain simulators by centring on a user's facilitated simulation modelling of Ethereum-based processes. It provides a simple application and focuses on the impact of blockchain rather than on its complex inner workings. We show that the behaviour of smart contracting in a blockchain framework can be simulated if the characteristics of the underlying framework are known and extensive tests are carried out.

5.2. Managerial implications

This contribution focuses on the production supply for shared manufacturing within the broader supply chain and logistics management framework. Our approach offers valuable managerial insights into enhancing the efficiency of shared manufacturing by improving transparency and reducing information asymmetries among multiple stakeholders. It further sheds light on the principal-agent problem and transaction costs in shared manufacturing.

For industry, we provide insights into how smart contracting in a digital twin in shared manufacturing can be simulated and which aspects must be considered. This contribution identifies vital requirements for a digital twin for the production supply in shared manufacturing and points out how blockchain and smart contracting address these requirements. Further managerial insights into how such a system can be set up and implemented technically are described and validated in experiments.

Additionally, this work contributes insights into the feasibility of several popular blockchain frameworks for production supply in shared manufacturing, including scalability, validation times, and associated costs. Ethereum PoA and Quorum BFT scaled reasonably well for our use case and did not encounter scalability issues. All blocks could hold the required transactions, and a block's gas limit was not exceeded, while Ethereum PoW proved unsuitable.

For industrial applications, this successful scaling in the simulation model offers benefits such as optimised resource allocation, risk mitigation, cost reduction and avoidance, and adaptability to changing conditions. These benefits improve operational efficiency and competitive advantage in dynamic industrial environments. However, scaling effects became visible in all implemented scenarios, which might become critical in high-scaled industrial use cases. Managing and processing large volumes of data generated by the simulation model within Ethereum's blockchain architecture constraints can pose challenges. Ensuring efficient data storage, retrieval, and processing while maintaining data integrity and security becomes increasingly complex as the simulation model grows in scale. Decision-makers in the industry should define how many transactions with which scope might arise in their manufacturing system using a simulation model. Closely related are the decisions on which information to share with multiple stakeholders and which characteristics a blockchain framework should have, such as access rights. Our work demonstrates significant key figures for managers, which can be used as benchmarks for suitable blockchain frameworks.

To facilitate the testing of blockchain frameworks, we contribute to science and industry the simulated smart contracting of Ethereum-based blockchain frameworks in time-lapse. It factors scalability and costs without blockchain test networks. We evaluate and benchmark this novel approach against a test network in our simulation, providing insights into how to set up such simulated smart contracting without detailed expertise on blockchain test networks or main networks. For managers, this approach provides insights into how to evaluate Ethereum blockchain usage for comparable industrial applications prior to implementation, considering both the associated costs and feasibility.

5.3. Outlook

Based on the findings of our work, we propose transferring these insights to the broader context of supply chain management. While we identified requirements for the model and smart contract functions, the design features might vary, be prioritised and be limited for specific use cases, requiring further research. We suggest investigating how well the analysed frameworks scale under a high system load, near the scalability limit. Doing so can evaluate how the system behaves in a higher-scaled supply chain context with multiple stakeholders and how the system is affected if the scalability limit is exceeded. The exhaustion of the scaling limit leads to delayed transaction processing, which can even lead to congestion and non-processing. It may also hinder the simulation model's seamless expansion to accommodate large-scale industrial operations. We suggest examining how the test networks perform compared to productive networks and analysing further solutions regarding scalability and costs. Further investigation might address how the simulated smart contracting can be refined, for instance, regarding gas consumption, to enable even more precise predictions of the impact of smart contracting on shared manufacturing.

Notes

1. https://unity.com/
2. https://nethereum.com/
3. https://ethereum.org/en/developers/docs/consensus-mechanisms/pow/mining-algorithms/ethash/.
4. https://docs.goquorum.consensys.net/configure-and-manage/configure/consensus-protocols/clique.
5. https://docs.goquorum.consensys.net/configure-and-manage/configure/consensus-protocols/qbft.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The work presented in this paper was funded by the Deutsche Forschungsgemeinschaft (DFG) – 276879186/GRK2193.

Notes on contributors

Larissa Krämer, M.Sc. is a research assistant at the Chair of Material Handling and Warehousing (FLW) at TU Dortmund University. She studied both industrial engineering and logistics at TU Dortmund University. Her research focuses on blockchain technology and smart contracting in intralogistics and digital twins. She is a member of the Research Training Group 'Adaption Intelligence of Factories in a Dynamic and Complex Environment' (GRK2193).

Patrick Stuckmann-Blumenstein, M.Sc. is a research assistant at the Chair of Enterprise Logistics (LFO) at TU Dortmund University. He studied industrial engineering at TU Dortmund University. His research focuses on supply chain optimisation and smart contracting.

Pascal Kaiser, M.Sc. is a research assistant at the Chair of Material Handling and Warehousing (FLW) at TU Dortmund University. He studied logistics at TU Dortmund University. His research focuses on scheduling in a hybrid production using answer set programming.

Univ.-Prof. Dr. Habil. Dr. H. C. Michael Henke is Director of the Fraunhofer Institute for Material Flow and Logistics IML and holds the Chair of Enterprise Logistics (LFO) in the Faculty of Mechanical Engineering at TU Dortmund University. He is also Adjunct Professor for Supply Chain Management at the School of Business and Management at Lappeenranta University of Technology in Finland and received an 'Honorary Doctorate of Science in Economics and Business Administration' from them.

His research interests include the management of Industry 4.0 and platform economy, blockchain and smart contracts, financial supply chain management, supply chain risk management, purchasing, logistics and supply chain management.

Dr.-Ing. Moritz Roidl is the Chief Engineer at the Chair of Material Handling and Warehousing at TU Dortmund University. He received his Dr. (PhD) from TU Dortmund University. His research focuses on cyber-physical twins in intralogistics. He studied computer science.

Univ.-Prof. Dr.-Ing. Alice Kirchheim is Director of the Fraunhofer Institute for Material Flow and Logistics IML and holds the Chair of Material Handling and Warehousing (FLW) in the Faculty of Mechnical Engineering at TU Dortmund University. Her major research field is the automation of logistics processes.

Data availability statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Larissa Krämer http://orcid.org/0000-0003-1620-3083
Patrick Stuckmann-Blumenstein http://orcid.org/0000-0001-5613-2844
Pascal Kaiser http://orcid.org/0000-0002-5701-6025
Michael Henke http://orcid.org/0000-0002-5848-9940
Moritz Roidl http://orcid.org/0000-0001-7551-9163
Alice Kirchheim http://orcid.org/0000-0002-8529-425X

References

Alharby, Maher, and Aad van Moorsel. 2020. "BlockSim: An Extensible Simulation Tool for Blockchain Systems." *Frontiers in Blockchain* 3:28. https://doi.org/10.3389/fbloc.2020.00028.

Bányai, Tamás. 2021. "Optimization of Material Supply in Smart Manufacturing Environment: A Metaheuristic Approach for Matrix Production." *Machines* 9:220. https://doi.org/10.3390/machines9100220.

Barricelli, Barbara Rita, Elena Casiraghi, and Daniela Fogli. 2019. "A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications." *IEEE Access* 7:167653–167671. https://doi.org/10.1109/Access.6287639.

Bellavista, Paolo, Nicola Bicocchi, Mattia Fogli, Carlo Giannelli, Marco Mamei, and Marco Picone. 2023. "Requirements and Design Patterns for Adaptive, Autonomous, and Context-Aware Digital Twins in Industry 4.0 Digital Factories." *Computers in Industry* 149:103918. https://doi.org/10.1016/j.compind.2023.103918.

Brocke, Jan vom, Alexander Simons, Björn Niehaves, Kai Riemer, Ralf Plattfaut, and Anne Cleven. 2009. "Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process." In *ECIS 2009 Proceedings*, Vol. 161, 2206–2217. Verona, Italy.

Buterin, Vitalik. 2014. "A Next Generation Smart Contract & Decentralized Application Platform." Whitepaper. Accessed December 11, 2023. https://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf.

Calafiore, Giuseppe C., Fabrizio Dabbene, and Roberto Tempo. 2011. "Research on Probabilistic Methods for Control System Design." *Automatica* 47 (7): 1279–1293. https://doi.org/10.1016/j.automatica.2011.02.029.

Camarinha-Matos, Luis M., Rosanna Fornasiero, and Hamideh Afsarmanesh. 2017. "Collaborative Networks as a Core Enabler of Industry 4.0." In *Collaboration in a Data-Rich World*, IFIP Advances in Information and Communication Technology, 3–17. Cham: Springer International Publishing.

ConsenSys-GoQuorum. 2023. Accessed November 29, 2023. https://docs.goquorum.consensys.net/configure-and-manage/configure/consensus-protocols/qbft.

De Angelis, Stefano, Leonardo Aniello, Federico Lombardi, Andrea Margheri, and V. Sassone. 2017. "PBFT vs Proof-of-Authority: Applying the CAP Theorem to Permissioned Blockchain." In *Proceedings of the Italian Conference on Cybersecurity*. Venice, Italy.

Ding, Kai, Felix T. S. Chan, Xudong Zhang, Guanghui Zhou, and Fuqiang Zhang. 2019. "Defining a Digital Twin-based Cyber-Physical Production System for Autonomous Manufacturing in Smart Shop Floors." *International Journal of Production Research* 57 (20): 6315–6334. https://doi.org/10.1080/00207543.2019.1566661.

Durão, Luiz, Sebastian Haag, Reiner Anderl, Klaus Schützer, and Eduardo Zancul. 2018. "Digital Twin Requirements in the Context of Industry 4.0." In *Product Lifecycle Management*, 204–214. Turin, Italy.

Durugbo, Christopher M., and Zainab Al-Balushi. 2023. "Supply Chain Management in times of Crisis: A Systematic Review." *Management Review Quarterly* 73 (3): 1179–1235. https://doi.org/10.1007/s11301-022-00272-x.

Ethereum-Foundation. 2024. Accessed January 31, 2024. https://ethereum.org/en/.

Glaser, Florian. 2017. "Pervasive Decentralisation of Digital Infrastructures: A Framework for Blockchain enabled System and Use Case Analysis." In *Proceedings of the 50th Hawaii International Conference on System Sciences*. Hawaii.

Gong, Yu, Yun Zhang, and Mohammed Alharithi. 2022. "Supply Chain Finance and Blockchain in Operations Management: A Literature Review." *Sustainability* 14 (20): 13450. https://doi.org/10.3390/su142013450.

Göppert, Amon, Esben Schukat, Peter Burggräf, and Robert H. Schmitt. 2021. "Agile Hybrid Assembly Systems: Bridging the Gap Between Line and Matrix Configurations." In *Advances in Automotive Production Technology – Theory and Application. Stuttgart Conference on Automotive Production (SCAP2020)*, 3–11. Berlin: Springer Berlin Heidelberg.

Grieves, Michael. 2015. "Digital Twin: Manufacutring Excellence through Virtual Factory Representation." Whitepaper. Accessed October 10, 2023. https://www.researchgate.net/publication/275211047_Digital_Twin_Manufacturing_Excellence_through_Virtual_Factory_Replication.

Grieves, Michael, and John Vickers. 2017. "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems." In *Transdisciplinary Perspectives on Complex Systems*, 85–113. Springer International Publishing.

Guo, Xin, Geng Zhang, and Yingfeng Zhang. 2023. "A Comprehensive Review of Blockchain Technology-Enabled Smart Manufacturing: A Framework, Challenges and Future Research Directions." *Sensors* 23:155. https://doi.org/10.3390/s23010155.

Gürpinar, Tan, Michael Henke, and Riad Ashraf. 2024. "Integrating Blockchain Technology in Supply Chain Management – A Process Model with Evidence from Current Implementation Projects." In *Proceedings of the 57th Hawaii International Conference on System Sciences*, Vol. 57. Hawaii.

Halldorsson, Arni, Herbert Kotzab, J. H. Mikkola, and Tage Skjøtt-Larsen. 2007. "Complementary Theories to Supply Chain Management." *Supply Chain Management: An International Journal* 12 (4): 284–296. https://doi.org/10.1108/13598540710759808.

Hart, Peter E., Nils J. Nilsson, and Bertram Raphael. 1968. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths." *IEEE Transactions on Systems Science and Cybernetics*4:100–107. https://doi.org/10.1109/TSSC.1968.300136.

Hasan, Haya R., Khaled Salah, Raja Jayaraman, Mohammed Omar, Ibrar Yaqoob, Sasa Pesic, Todd Taylor, and Dragan Boscovic. 2020. "A Blockchain-Based Approach for the Creation of Digital Twins." *IEEE Access* 8:34113–34126. https://doi.org/10.1109/Access.6287639.

Hawlitschek, Florian. 2019. "Trust in the Sharing Economy: A Behavioral Perspective on Peer-to-Peer Markets." Repository KITopen.

Hevner, Alan R., Salvatore T. March, Jinsoo Park, and Sudha Ram. 2004. "Design Science in Information Systems Research." *Management Information Systems Quarterly* 28 (1): 75–105. https://doi.org/10.2307/25148625.

International Performance Research Institute, and 2021. "KapShare – Leitfaden: Befähigung von KMU zum branchenübergreifenden Sharing von Produktionskapazitäten mittels digitaler Plattformen." Technical Report. Accessed November 15, 2023. https://www.iph-hannover.de/_media/files/downloads/Projekt_KapShare_Leitfaden.pdf.

Ioannou, Ilias, and Guven Demirel. 2022. "Blockchain and Supply Chain Finance: A Critical Literature Review at the Intersection of Operations, Finance and Law." *Journal of Banking and Financial Technology* 6 (1): 83–107. https://doi.org/10.1007/s42786-022-00040-1.

Javaid, Mohd, Abid Haleem, and Rajiv Suman. 2023. "Digital Twin Applications toward Industry 4.0: A Review." *Cognitive Robotics* 3:71–92. https://doi.org/10.1016/j.cogr.2023.04.003.

Jensen, Michael C., and William H. Meckling. 1979. "Theory of the Firm: Managerial Behavior, Agency Costs, and Ownership Structure." In *Economics Social Institutions: Insights* *from the Conferences on Analysis & Ideology*, Rochester Studies in Economics and Policy Issues, 163–231. Rochester, NY: Springer Netherlands.

Kaiser, Pascal, Andre Thevapalan, Christopher Reining, Moritz Roidl, Gabriele Kern-Isberner, and Michael ten Hompel. 2022. "Hybrid Production: Enabled by Controlling the Output Sequence of A Matrix Production Using Answer Set Programming." *Procedia CIRP* 107:1305–1310. https://doi.org/10.1016/j.procir.2022.05.149.

Kannengießer, Niclas, Sebastian Lins, Tobias Dehling, and Ali Sunyaev. 2019. "What Does Not Fit Can Be Made to Fit! Trade-Offs in Distributed Ledger Technology Designs." In *Proceedings of the 52nd Hawaii International Conference on System Sciences*. Hawaii.

Klarin, Anton, and Yuliani Suseno. 2021. "A State-of-the-Art Review of the Sharing Economy: Scientometric Mapping of the Scholarship." *Journal of Business Research* 126:250–262. https://doi.org/10.1016/j.jbusres.2020.12.063.

Koren, Yoram, and Moshe Shpitalni. 2010. "Design of Reconfigurable Manufacturing Systems." *Journal of Manufacturing Systems* 29 (4): 130–141. https://doi.org/10.1016/j.jmsy.2011.01.001.

Krämer, Larissa, Rico Ahlbäumer, Jens Leveling, Peter Detzner, Marius Brehler, and Michael ten Hompel. 2021. "Towards a Concept for Blockchain-based Cyber-Physical Production Systems." *LogisticsJournal* 1–10. https://www.logistics-journal.de/archive/proceedings/2021/5423.

Krämer, Larissa, Nick Große, Patrick Stuckmann-Blumenstein, Rico Ahlbäumer, Michael Henke, and Michael ten Hompel. 2022. "Designing a Blockchain-Based Digital Twin for Cyber-Physical Production Systems." In *Proceedings of the Conference on Production Systems and Logistics: CPSL 2022*, 141–150. Hannover: publish-Ing.

Krämer, Larissa, Pascal Kaiser, Jonathan Kajewski, and Moritz Roidl. 2023. "Reversing the Digital Twin – Smart-Contracting in Hybrid Production." In *IEEE International Conference on Omni-Layer Intelligent Systems*. Berlin, Germany.

Kritzinger, Werner, Matthias Karner, Georg Traar, Jan Henjes, and Wilfried Sihn. 2018. "Digital Twin in Manufacturing: A Categorical Literature Review and Classification." *IFAC-PapersOnLine*51:1016–1022. https://doi.org/10.1016/j.ifacol.2018.08.474.

Kshetri, Nir. 2018. "Blockchain's Roles in Meeting Key Supply Chain Management Objectives." *International Journal of Information Management* 39:80–89. https://doi.org/10.1016/j.ijinfomgt.2017.12.005.

Kuruppuarachchi, Pasindu Manisha, Susan Rea, and Alan McGibney. 2023. "Trusted and Secure Composite Digital Twin Architecture for Collaborative Ecosystems." *IET Collaborative Intelligent Manufacturing* 5 (1): e12070. https://doi.org/10.1049/cim2.v5.1.

Liu, Peng, Caiyun Liu, and Xiaoling Wei. 2021. "Optimal Allocation of Shared Manufacturing Resources Based on Bilevel Programming." *Discrete Dynamics in Nature and Society* 2021: 1–15.

Liu, Shimin, Yuqian Lu, Jie Li, Xingwang Shen, Xuemin Sun, and Jinsong Bao. 2023. "A Blockchain-based Interactive Approach between Digital Twin-based Manufacturing Systems." *Computers & Industrial Engineering* 175:108827. https://doi.org/10.1016/j.cie.2022.108827.

Morabito, Vincenzo. 2017. "Business Innovation Through Blockchain: The B3 Perspective."

Müller, Maren, Fabian Schüler, Malte Stonis, and Peter Nyhuis. 2022. "Development of a Method for Decision Support on Participation in Capacity Sharing for Manufacturing SMEs." In *Proceedings of the Conference on Production Systems and Logistics: CPSL 2022*, 411–423. Hannover, Germany: publish-Ing.

Nakamoto, Satoshi. 2008. "Bitcoin: A Peer-to-Peer Electronic Cash System." SSRN Electronic Journal.

Nielsen, Christian Petersson, Elias Ribeiro da Silva, and Fei Yu. 2020. "Digital Twins and Blockchain – Proof of Concept." *Procedia CIRP* 93:251–255. https://doi.org/10.1016/j.procir.2020.04.104.

Onaji, Igiri, Divya Tiwari, Payam Soulatiantork, Boyang Song, and Ashutosh Tiwari. 2022. "Digital Twin in Manufacturing: Conceptual Framework and Case Studies." *International Journal of Computer Integrated Manufacturing* 35 (8): 831–858. https://doi.org/10.1080/0951192X.2022.2027014.

Onwubiko, Austine, Raman Singh, Shahid Awan, Zeeshan Pervez, and Naeem Ramzan. 2023. "Enabling Trust and Security in Digital Twin Management: A Blockchain-Based Approach with Ethereum and IPFS." *Sensors* 23 (14): 6641. https://doi.org/10.3390/s23146641.

Park, Heejong, Arvind Easwaran, and Sidharta Andalam. 2019. "Challenges in Digital Twin Development for Cyber-Physical Production Systems." In *Cyber Physical Systems. Model-Based Design*, 28–48. Springer International Publishing.

Paulavicius, Remigijus, Saulius Grigaitis, and Ernestas Filatovas. 2021. "A Systematic Review and Empirical Analysis of Blockchain Simulators." *IEEE Access* 9:38010–38028. https://doi.org/10.1109/Access.6287639.

Peffers, Ken, Tuure Tuunanen, Marcus Rothenberger, and S. Chatterjee. 2007. "A Design Science Research Methodology for Information Systems Research." *Journal of Management Information Systems*24 (3): 45–77. https://doi.org/10.2753/MIS0742-1222240302.

Raj, Pethuru. 2021. "Empowering Digital Twins with Blockchain." In *Advances in Computers*, Vol. 121, 267–283. Elsevier.

Ranganthan, Vishnu Prasad, Ram Dantu, Aditya Paul, Paula Mears, and Kirill Morozov. 2018. "A Decentralized Marketplace Application on the Ethereum Blockchain." In *IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, 90–97. Philadelphia, PA, USA.

Reyna, Ana, Cristian Martín, Jaime Chen, Enrique Soler, and Manuel Díaz. 2018. "On Blockchain and Its Integration with IoT. Challenges and Opportunities." *Future Generation Computer Systems* 88:173–190. https://doi.org/10.1016/j.future.2018.05.046.

Rouzbahani, Amin, and Fattaneh Taghiyareh. 2022. "A Trust-Aware Task Allocation Method Based on Blockchain for The Internet of Things." In *8th International Conference on Web Research (ICWR)*, 156–161. Tehran, Iran.

Sahal, Radhya, Saeed H. Alsamhi, Kenneth N. Brown, Donna O'Shea, Conor McCarthy, and Mohsen Guizani. 2021. "Blockchain-Empowered Digital Twins Collaboration: Smart Transportation Use Case." *Machines* 9 (9): 193. https://doi.org/10.3390/machines9090193.

Schmidtke, Niels, Alina Rettmann, and Fabian Behrendt. 2021. "Matrix Production Systems – Requirements and Influences on Logistics Planning for Decentralized Production

Structures." In *54th Hawaii International Conference on System Sciences 2021. Proceedings*, 1665–1674. Hawaii.
Schweiger, Lukas, and Linard Barth. 2023. "Properties and Characteristics of Digital Twins: Review of Industrial Definitions." *SN Computer Science* 4:436. https://doi.org/10.1007/s42979-023-01937-4.
Shen, Bingqing, Xirui Xiong, Han Yu, Pan Hu, Lihong Jiang, and Hongming Cai. 2023. "Online Interactive Simulation with Digital Twins for Collaborative Manufacturing." In *2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, 462–467. Shanghai, China.
Sonntag, Viki. 2000. "Sustainability – in Light of Competitiveness." *Ecological Economics* 34 (1): 101–113. https://doi.org/10.1016/S0921-8009(00)00141-5.
Stuckmann-Blumenstein, Patrick, Dominik Bons, Nick Große, and Lana Benkhoff. 2024. "Problems and Potentials of Shared Manufacturing in the Context of Industrial Ecosystems: A Bibliometric Analysis." In *Proceedings of the 57th Hawaii International Conference on System Sciences*. Hawaii.
Suhail, Sabah, Rasheed Hussain, Raja Jurdak, Alma Oracevic, Khaled Salah, Choong Seon Hong, and Raimundas Matulevičius. 2022. "Blockchain-Based Digital Twins: Research Trends, Issues, and Future Challenges." *ACM Computing Surveys* 54:1–34. https://doi.org/10.1145/3517189.
Tao, Fei, Yongping Zhang, Ying Cheng, Jiawei Ren, Dongxu Wang, Qinglin Qi, and Pei Li. 2022. "Digital Twin and Blockchain Enhanced Smart Manufacturing Service Collaboration and Management." *Journal of Manufacturing Systems* 62:903–914. https://doi.org/10.1016/j.jmsy.2020.11.008.
Tao, Fei, Meng Zhang, and A. Y. C. Nee. 2019. *Digital Twin Driven Smart Manufacturing*. Academic Press.
Teslya, Nikolay, and Alexander Smirnov. 2018. "Blockchain-based Framework for Ontology-Oriented Robots' Coalition Formation in Cyberphysical Systems." In *MATEC Web of Conferences*. St. Petersburg, Russia.
Treiblmaier, Horst. 2018. "The Impact of the Blockchain on the Supply Chain: A Theory-based Research Framework and a Call for Action." *Supply Chain Management: An International Journal* 23 (6): 545–559. https://doi.org/10.1108/SCM-01-2018-0029.
Treiblmaier, Horst. 2019. "Toward More Rigorous Blockchain Research: Recommendations for Writing Blockchain Case Studies." *Frontiers in Blockchain* 2: 1–15. https://doi.org/10.3389/fbloc.2019.00003.
Unity-Technologies. 2023. "Unity AI Navigation Package." Accessed December 6, 2023. https://github.com/Unity-Technologies/NavMeshComponents.
Valk, Hendrik van der, Hendrik Haße, Frederik Möller, and Boris Otto. 2022. "Archetypes of Digital Twins." *Business & Information Systems Engineering* 64 (3): 375–391. https://doi.org/10.1007/s12599-021-00727-7.
Wang, Gang, Geng Zhang, Xin Guo, and Yingfeng Zhang. 2021. "Digital Twin-Driven Service Model and Optimal Allocation of Manufacturing Resources in Shared Manufacturing." *Journal of Manufacturing Systems* 59:165–179. https://doi.org/10.1016/j.jmsy.2021.02.008.
Wenzel, Sigrid. 2018. "Simulation Logistischer Systeme." In *Modellierung Logistischer Systeme*, edited by Horst Tempelmeier, Fachwissen Logistik, 1–34. Köln, Germany: Springer.
Wiese, Harald. 2021. *Advanced Microeconomics*. Wiesbaden, Germany: Springer Fachmedien Wiesbaden.
Williamson, Oliver E. 1987. "Transaction Cost Economics: The Comparative Contracting Perspective." *Journal of Economic Behavior & Organization* 8 (4): 617–625. https://doi.org/10.1016/0167-2681(87)90038-2.
Wood, Gavin. 2023. "Ethereum: A Secure Decentralised Generalised Transaction Ledger." Whitepaper. Accessed February 27, 2024. https://ethereum.github.io/yellowpaper/paper.pdf.
World Economic Forum, and 2020. "Share to Gain: Unlocking Data Value in Manufacturing." Whitepaper. Boston Consulting Group. Accessed December 15, 2023. https://www3.weforum.org/docs/WEF_Share_to_Gain_Report.pdf.
Yaqoob, Ibrar, Khaled Salah, Mueen Uddin, Raja Jayaraman, Mohammed Omar, and Muhammad Imran. 2020. "Blockchain for Digital Twins: Recent Advances and Future Research Challenges." *IEEE Network* 34 (5): 290–298. https://doi.org/10.1109/MNET.65.
Yu, Chunyang, Xun Xu, Shiqiang Yu, Zhiqian Sang, Cheng Yang, and Xuanlin Jiang. 2020. "Shared Manufacturing in the Sharing Economy: Concept, Definition and Service Operations." *Computers & Industrial Engineering* 146:106602. https://doi.org/10.1016/j.cie.2020.106602.
Zheng, Zibin, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. 2017. "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends." In *2017 IEEE International Congress on Big Data*, 557–564. Honolulu, HI: IEEE.

Appendices

Appendix 1. Systematic Literature Reviews

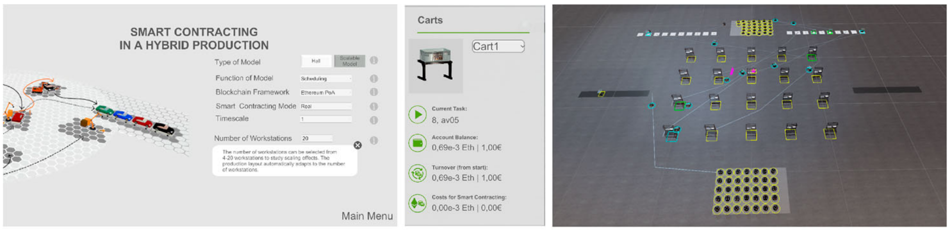
The systematic literature reviews (SLR) follow the approach of Brocke et al. (2009). Two searches were conducted on the database Scopus with the search strings depicted in Table A1 for the requirements. To gain precise and manageable results, several restrictions were imposed on the search string, such as the publication year, the source type, the document type and the language. The searches yielded 82 and 114 unfiltered results, which we filtered for titles relevant to the requirements. We then checked the abstracts and the full texts, resulting in one and three suitable publications. Backward- and forward searches based on these publications revealed further publications, resulting in 4 publications for each topic.

For the related work, we conducted the similar SLR in Table A2 on Scopus. We did not consider any restrictions for shared manufacturing and digital twin due to the already low results with 35 publications. The second string was restricted, similar to the first SLR. The search resulted in five and ten publications for related work.

Appendix 2. Supplementary technical details

A.1 Graphic user interface

The user can set start configurations in the GUI using buttons and dropdown menus as shown in Figure A1(a). When selecting the configuration, the model automatically checks whether the desired options can be combined and issues a warning if necessary. Help buttons provide information on the options. Once the user has started the simulation model, a 3D model



(a) Configuration options (b) Key figures (c) Colour scheme

Figure A1. Visualisation (GUI) of the simulation model (screenshots from Unity model).

Table A1. SLR methodology for design requirements for shared manufacturing and digital twin.

Literature	SM	DT
Database	Scopus	Scopus
Unfiltered results	82	114
Fitting title	33	35
Fitting abstract	8	13
Fitting full text	1	2
Results from backward / forward search	3	2
Total number	4	4

Search string	
Shared Manufacturing (SM)	Digital Twin (DT)
TITLE-ABS-KEY ((‘shar∗ manufact∗’ OR ‘collab∗ manfact∗’ OR ‘shar∗ product∗’) AND (‘review’ OR ‘definition’ OR ‘concept∗’)) AND PUBYEAR > 2016 AND PUBYEAR < 2025 AND (LIMIT-TO (SRCTYPE, ‘j’) OR LIMIT-TO (SRCTYPE, ‘p’)) AND (LIMIT-TO (DOCTYPE, ‘cp’) OR LIMIT-TO (DOCTYPE, ‘ar’)) AND (LIMIT-TO (LANGUAGE, ‘English’))	TITLE-ABS-KEY (‘digital twin∗’ AND ‘requirement∗’ AND ‘review’) AND PUBYEAR > 2017 AND PUBYEAR < 2025 AND (LIMIT-TO (SRCTYPE, ‘j’) OR LIMIT-TO (SRCTYPE, ‘p’)) AND (LIMIT-TO (DOCTYPE, ‘cp’) OR LIMIT-TO (DOCTYPE, ‘re’) OR LIMIT-TO (DOCTYPE, ‘ar’)) AND (LIMIT-TO (LANGUAGE, ‘English’))

Table A2. SLR methodology for related work.

Literature	SMDT	MDTB
Database	Scopus	Scopus
Unfiltered results	35	153
Fitting title	20	49
Fitting abstract	11	21
Fitting full text	5	10
Total number	5	10

Search string	
Shared manufacturing and digital twin (SMDT)	Manufacturing, digital twin and blockchain (MDTB)
TITLE-ABS-KEY ((‘Shar∗ Manufact∗’ OR ‘Shar∗ Product∗’ OR ‘Collab∗ Manufact∗’ OR ‘Collab∗ Product∗’) AND ‘Digital∗ Twin∗’)	TITLE-ABS-KEY ((‘Manufact∗’ OR ‘Product∗’) AND ‘Digital∗ Twin∗’ AND ‘Blockchain∗’) AND PUBYEAR > 2016 AND PUBYEAR < 2025 AND (LIMIT-TO (LANGUAGE, ‘English’)) AND (LIMIT-TO (DOCTYPE, ‘ar’) OR LIMIT-TO (DOCTYPE, ‘cp’) OR LIMIT-TO (DOCTYPE, ‘re’)) AND (LIMIT-TO (SRCTYPE, ‘j’) OR LIMIT-TO (SRCTYPE, ‘p’))

of the hybrid production is created according to the configuration. During the simulation, an interactive dashboard provides information on the smart contract and on each resource as shown in Figure A1(b) to increase accessibility.

Observability is increased by a colour scheme visualising the current utilisation of resources as shown in Figure A1(c). Yellow shapes represent idle resources waiting for an order, while green resources are reserved for a task and are waiting to begin. Resources currently engaged with a task are shown in cyan.

A.2 Configuration of the blockchain test networks

We used the configurations summarised in Table A3 for the blockchain test networks.

Ethash[3] is chosen as a proof of work (PoW) mechanism because it played a fundamental role in Ethereum 1.0, allowing to analyse a framework whose characteristics are closely related to the preceding version of the Ethereum Mainnet. This consensus mechanism has a high level of security but scalability limitations and significant energy consumption. Clique[4] as a proof of authority (PoA) mechanism reflects the current Ethereum Mainnet. In the enterprise environment, PoA is depicted by Quorum Byzantine Fault Tolerance (QBFT[5]), a consensus mechanism designed to provide robust security alongside exceptional throughput capabilities. Clique and Ethash use gas fees as public frameworks, while the gas fees in the consortium QBFT are zero due to a zero gas price. For each test network, we choose four nodes, which is the minimum number required for the QBFT system to be Byzantine fault-tolerant (ConsenSys-GoQuorum 2023). All test networks and resources' accounts are prefunded. The block time determines the average period for creating a new block. For Ethereum PoA

Table A3. Specifications of the blockchain testnets.

Specification	Ethereum PoA	Ethereum PoW	Quorum BFT
Consensus mechanism	Clique	Ethash	QBFT
Type of mechanism	Proof of Authority (PoA)	Proof of Work (PoW)	Proof of Authority (PoA)
Gas fee	yes	yes	no
Purpose	public	public	private Enterprise
Nodes	4	4	4
Prefunded	yes	yes	yes
Block time	5 s	Variable, 5 s target	5 s
Target block size	1.5×10^7 gas	1.5×10^7 gas	1.5×10^7 gas
Maximum block size	3×10^7 gas	3×10^7 gas	3×10^7 gas
Gas limit	8×10^6 gas	8×10^6 gas	8×10^6 gas
Difficulty	0×1	0×1	0×1
Epoch	30,000	–	30,000

and Quorum BFT, a steady block time can be configured. We set this block time to 5 s. A longer block time could cause scalability issues, while a shorter block time would cause a more significant load on the network. For the Ethereum PoW test network, a 5 s block time is targeted. The target block size, called gas limit, determines the maximum amount of gas that should be spent on one block, while the maximum block size is defined as the maximum amount of gas that can be included in one block (Wood 2023). The maximum block size is similar to that of the Ethereum Mainnet (Ethereum-Foundation 2024). The target block size is half the maximum block size (Wood 2023). To ensure that most transactions go through in Ethash, we set the gas limit of each transaction so high that at least two transactions still fit into a block. The target mining difficulty determines the difficulty of the cryptographic puzzle and impacts how often a new block is created (Wood 2023). We set the difficulty to 0x1, the lowest available difficulty, to gain maximum throughput for Ethereum PoW. Ethereum PoA and Quorum BFT do not have mining, so their difficulty does not influence the block time. It is still required due to formal reasons stated in Wood (2023) and can influence the order of selecting a proposed block. The network configuration is made in the genesis files of the depicted networks. The epoch represents the number of blocks to be completed to create new committees of validators. We set the number to 30,000, similar to the Ethereum Mainnet (Ethereum-Foundation 2024).

Table A4. Specifications of the computer and the server used for the experiments.

Component	Computer	Server
Operating System	Windows 10	Ubuntu 22.04.2 LTS
Central Processing Unit (CPU)	AMD EPYC 7302P	AMD EPYC 7401P
Graphics card	NVIDIA RTX A4000	2x NVIDIA GeForce RTX 2080 Ti
Random Access Memory (RAM)	20 GB RAM	64 GB RAM

A.3 Specifications of the computer and server

We used a computer and server with the specifications displayed in Table A4. The simulation model ran on the computer with Windows 10 as the operating system and the given performance specifications, while the server with Ubuntu 22.04.2 LTS was responsible for running the blockchain test networks to avoid performance distortion.