

Article

Automated Payment and Contract Management in the Construction Industry by Integrating Building Information Modeling and Blockchain-Based Smart Contracts

Katharina Sigalov ^{1,*} , Xuling Ye ¹ , Markus König ¹ , Philipp Hagedorn ¹ , Florian Blum ² ,
Benedikt Severin ² , Michael Hettmer ² , Philipp Hückinghaus ² , Jens Wölkerling ³ 
and Dominik Groß ^{4,†} 

- ¹ Chair of Computing in Engineering, Department of Civil and Environmental Engineering, Ruhr-Universität Bochum, 44801 Bochum, Germany; xuling.ye@rub.de (X.Y.); koenig@inf.bi.rub.de (M.K.); philipp.hagedorn-n6v@rub.de (P.H.)
 - ² Chair for Software Engineering, Faculty of Business Administration and Economics, University of Duisburg-Essen, 45127 Essen, Germany; florian.blum@uni-due.de (F.B.); benedikt.severin@uni-due.de (B.S.); michael.hettmer@uni-due.de (M.H.); philipp.hueckinghaus@uni-due.de (P.H.)
 - ³ Adesso SE, Willy-Brandt-Straße 1, 20457 Hamburg, Germany; jens.woelkerling@adesso.de
 - ⁴ Kapellmann und Partner Rechtsanwälte mbB, 40219 Düsseldorf, Germany; dominik.gross@kapellmann.de
- * Correspondence: katharina.sigalov@ruhr-uni-bochum.de
† Current address: District Court of Cologne, Luxemburger Str. 101, 50939 Cologne, Germany.



Citation: Sigalov, K.; Ye, X.; König, M.; Hagedorn, P.; Blum, F.; Severin, B.; Hettmer, M.; Hückinghaus, P.; Wölkerling, J.; Groß, D. Automated Payment and Contract Management in the Construction Industry by Integrating Building Information Modeling and Blockchain-Based Smart Contracts. *Appl. Sci.* **2021**, *11*, 7653. <https://doi.org/10.3390/app11167653>

Academic Editor: Gianluca Lax

Received: 9 July 2021

Accepted: 16 August 2021

Published: 20 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: Construction projects usually involve signing various contracts with specific billing procedures. In practice, dealing with complex contract structures causes significant problems, especially with regard to timely payment and guaranteed cash flow. Furthermore, a lack of transparency leads to a loss of trust. As a result, late or non-payment is a common problem in the construction industry. This paper presents the concept of implementing smart contracts for automated, transparent, and traceable payment processing for construction projects. Automated billing is achieved by combining Building Information Modeling (BIM) approaches with blockchain-based smart contracts. Thereby, parts of traditional construction contracts are transferred to a smart contract. The smart contract is set up using digital BIM-based tender documents and contains all of the relevant data for financial transactions. Once the contracted construction work has been accepted by the client, payments can be made automatically via authorized financial institutions. This paper describes the framework, referred to as BIMcontracts, the container-based data exchange, and the digital contract management workflow. It discusses the industry-specific requirements for blockchain and data storage and explains which technical and software architectural decisions were made. A case study is used to demonstrate the current implementation of the concept.

Keywords: smart contracts; blockchain; automated payment; construction management; construction contracts; BIM; digital contract management

1. Introduction

The construction industry is going through constant changes and is managing the transition from the analog to the digital era. The high complexity and structural fragmentation of the sector, the limited degree of repeatability of construction projects, weak collaboration, and insufficient investment in innovation are the most common reasons for the low level of digitization and productivity [1]. Blockchain is predicted to be a technology that could have an extremely positive force of change in the architectural, engineering, and construction (AEC) industry [2–4]. The potential of blockchain has been recognized in the scientific community and is getting more and more attention [2,3,5,6]. One of the reasonable applications of blockchain in the construction context is the usage of smart

contracts to streamline payment processing [2,7]. Smart contracts can execute the terms of a contract according to mutual agreements in a tamper-proof way. As a part of a construction contract, they can speed up payouts and improve contract management [7,8]. Based on the security, immutability, and traceability of the underlying blockchain technology, it could not only improve the payment process, but also increase trust [2].

Construction projects involve many stakeholders who represent different interests and requirements. The short-term nature of construction contracts, lack of trust, and numerous regulations and laws result in complex contractual structures [8–11]. In combination with a low level of digitization and process automation, lengthy and non-transparent inspection processes, persistent disputes, and long-term payment delays are very prevalent [8,11,12]. Inefficiencies in financial flows and unnecessary bureaucracy therefore bear partial responsibility for the high rate of insolvencies of small and medium-sized enterprises (SMEs) [13].

Building Information Modeling (BIM) is the most prominent example in AEC of how revolutionary new technologies and working methods could be. BIM has a substantial impact on the execution of construction projects and leads to sustainable performance improvements [14]. It opens up a completely new way of working, connecting both the participants and the data across the entire building lifecycle. The increasing use of BIM has already significantly improved both the collaboration and the technical execution of construction projects. However, BIM is still mainly used during the design phase. The construction process is characterized by limited use of end-to-end digital tools and processes [1]. Although the next-generation BIM, such as 5D BIM, has long been recognized as one of the groundbreaking trends for the global construction industry [15], its practical implementation remains rare [16–18]. Five-dimensional BIM is the next level after 4D (scheduling) and includes additional information for model-based cost planning, tendering, and execution. This topic has received much attention in research, and concepts for efficient implementation have been developed [16,18,19]. The need for comprehensive and accurate data, greater collaboration, and the lack of 5D BIM protocols remain open questions regarding the benefits of 5D BIM [16,18]. These issues can be resolved much faster if there are adequate incentives and a better value proposition for all stakeholders. The lack of customer demand and missing use cases are still the biggest barriers to the widespread adoption of BIM [14]. The advantages of blockchain-based smart contracts, such as faster payments and traceability features, have the potential to create these incentives and accelerate digitization. Blockchain, especially in combination with BIM, can act as a generator of data-driven business models and as a facilitator of collaboration in construction [2,7].

Notwithstanding the great scientific interest in this area, there is widespread skepticism about the usability, acceptance, and readiness of the AEC industry for adoption of blockchain solutions [2,3,20]. Regarding this matter, this contribution focuses on the question of how BIM can be combined with blockchain-based smart contracts to reach a practicable and enforceable solution for automated payment between clients and contractors. The proposed framework takes the current working methods, software tools, and contract structures, as well as the existing standards, into account and is designed as a low-threshold system so that SMEs can use it without major technical effort. After a brief introduction to the basic concepts, the paper outlines the latest research on blockchain and smart contracts in the AEC industry and then discusses the known implementation challenges. To ensure proper information exchange, the approach at hand uses the ISO-standardized Information Container for linked Document Delivery (ICDD) for cross-domain-linked nD models (3D models enhanced with costs and payment-relevant data). Therefore, an overview of the literature on the use of standardized linked building models referencing the multi-model approach and the ICDD is also given.

The main part of this paper describes the so-called BIMcontracts framework, the container-based data exchange, and the workflow of managing digital contracts. It points out the industry-specific requirements for the blockchain and data storage and explains which technical and software architectural decisions were made. The general idea is to

establish an ecosystem with formalized interfaces for further extension that can be used to support multiple construction-related use cases. This is done by providing a prototype that shows how blockchain technology can be combined with existing systems and be put to use in a legally valid context.

2. State of the Art

2.1. Blockchain and Smart Contracts

Most prominently, blockchain is known as the technical infrastructure behind cryptocurrencies such as Bitcoin or Ethereum. Blockchain, as a well-known form of Distributed Ledger Technology (DLT) [21] is a technology for providing a tamper-proof and easily verifiable transactional ledger system for storage and program execution. A blockchain groups these transactions into linked data blocks and makes the order and integrity of those blocks easily verifiable [22,23]. This ensures a transparent and tamper-proof environment for exchanging values or executing program code via computer nodes connected to a blockchain network. Therefore, blockchain can be used to introduce a trustless and neutral platform for immutable data storage, exchanging information, saving executable code contracts, and automating payments [24,25]. Blockchain technology is already used, e.g., to support traceability in supply chains [26] where several participants work together and transparency about the shipped goods, the origin of their materials, and immutable records of each handover represent very valuable information. Especially for international exchanges between participants from multiple jurisdictions, a global settlement layer for contracts that can be specified up front can increase the degree of automation and reduce friction [27,28].

To ensure that only valid transactions are included in the blockchain, consensus mechanisms decide who may propose the next block and verify its included transactions. The most prominent consensus mechanisms are Proof of Work (PoW), Proof of Stake (PoS), and Proof of Authority (PoA) [29]. The PoW only allows the first computer node with proof of the correct solution to a computing-intensive puzzle to propose the next block. This procedure serves as the base for security, especially in public blockchains such as Bitcoin, but has the disadvantage of high energy consumption. In PoS, every node has to lock money in the form of cryptocurrencies inside the system, which replaces the need for costly puzzles and, therefore, solves the high energy consumption issue of PoW. The PoA describes a mechanism similar to that of PoS by replacing the validator's financial stake with its reputation, where the validator's identities are known inside the network and faulty behavior leads to a loss of trust. An established PoA mechanism is the Istanbul Byzantine Fault Tolerant (IBFT) based on the works of Castro et al. [30], where an arbitrarily selected node can propose a new block, which is only accepted if a super-majority of all nodes confirm its correctness. This two-level validation provides a trade-off between performance and fault tolerance [31]. A blockchain that limits its access to a controlled user group to protect the information stored on the blockchain is called a consortium blockchain. This type of blockchain offers a trade-off by decreasing a blockchain's decentralization but improving scalability and performance. GoQuorum and Hyperledger Besu are well-known consortium blockchain implementations that still maintain compatibility with smart contracts developed for the public Ethereum chain [31,32].

Smart Contracts (SCs), the software components executed on a blockchain, enforce programmed behavior without any possibility of changing the scripted rules after installation (i.e., "deployment") [23,24]. The users can create and interact with smart contracts by sending a transaction to the blockchain. Once the transaction is evaluated, the data storage is updated according to the codified rules, and the new state is saved by each blockchain node. There are different degrees of decentralization for systems using blockchain technology, and it is not necessary (and, in most cases, not useful) to store everything on the blockchain or have all processes executed as smart contracts. Therefore, a good balance between centralized and decentralized components has to be found when developing smart contracts and the resulting "hybrid" blockchain applications [33–35]. To find this

balance, an application can be split into on-chain and off-chain components for storage and computational aspects [36].

2.2. Blockchain in Construction

A great deal of research has been done on blockchain due to its enormous potential for the construction sector [2–5,20]. The latest literature reviews on AEC-related applications of blockchain identify different research priorities and use cases [2–5,20,37]. Information traceability, the related transparency, and the possibility of creating a single source of truth are crucial benefits for a range of applications in construction. As for many other industries, it can help to improve supply chains with greater security, efficiency, and dependability [38–40].

Furthermore, these features of blockchain technology increase the trust in data and support collaboration, which have a great significance for the implementation of BIM as a working method [9]. Therefore, it is comprehensible that the topic of integrating blockchain with BIM and information management has gained considerable attention in the community [2,4,5,9,20]. Data ownership and rights, responsibilities, and change tracing are of particular importance for project design. The ability of blockchain to securely track changes offers great potential during the design phase by ensuring the incorruptibility of stored information and improving the reliability and trustworthiness of digital collaboration [41,42]. For example, Dounas et al. [43] introduced a BIM-Blockchain framework to optimize the BIM design process. Xue and Lu [44] presented a novel semantic differential transaction approach in order to capture BIM changes and minimize information redundancy in the field of BIM and blockchain integration. After the project planning, data management and storage are important for the entire lifecycle of a building. Numerous contributions discussed possible applications of blockchain during the execution and operation process of construction projects [9,10,45,46]. In that regard, the benefits of using smart contracts and the interrelation with BIM are some of the most promising combinations [2,4,9].

2.3. Smart Contracts and Automated Payment in Construction

In the AEC industry, SCs can be used to automate different kinds of processes. For example, in an approach suggested by Li et al. [41], smart contracts were used to automatically verify the compliance of the digital deliverables provided by Exchange Information Requirements. The payment could then be authorized or denied based on the automatically checked results. However, they did not further discuss how to handle the payment after approval or rejection. Similarly, in the framework proposed by Zhong et al. [46], smart contracts (called chaincode by the Hyperledger Fabric project) were also used for automated compliance checking. These two papers introduced SCs only as pseudocode and need further validation. Another significant use case for the application of SCs is the improvement of supply chains. For example, the chaincode in a blockchain-based information management model built by Wang et al. [40] was used for changing the status of the precast components for a precast supply chain. A further Hyperledger-Fabric-based blockchain framework developed by Sheng et al. [45] used chaincode to create or fill quality information forms, define quality acceptance results, and sign forms for trusted quality information management in construction. However, they only presented the initial smart contract function of “create quality information form”. The smart contract in the framework from Elghaish et al. [47] was proposed to include functions for automatically recording and calculating reimbursed costs, profits, and cost savings for each member of a construction project, especially in integrated project delivery projects. These papers provide ideas for using smart contracts in construction, but only a few of them explain their actual smart contract design.

Smart contracts have proven to be particularly useful for payment automation in the construction industry. As Di et al. [9] pointed out, theoretically, the association between smart contracts and BIM can automate the entire delivery phase, and automatic payment

can be issued via the connection between the information model (i.e., activity progress) and the computational contract. Chong and Diamantopoulos [48] developed a data-flow-diagram framework for integrating BIM, a smart sensor, a blockchain, and a smart contract to address the payment security issues for subcontractors. In their framework, the smart sensor was used to capture the completed sub-tasks, and BIM data were used to validate the completed works. If the tasks and their corresponding transactions were validated, payment was automatically executed via the smart contract. The smart contract design was still in the form of a data-flow diagram, which should be further detailed. Similarly, Hamledari and Fischer [10] introduced a smart-contract-enabled solution in order to autonomously translate on-site reality captures into direct payments between a general contractor and subcontractors. Their smart contract was used to handle payment settlements and transfer lien rights, an idea that should be further detailed and verified. Das et al. [38] proposed a blockchain-based framework to facilitate the execution of interim payments by deploying a logic to automatically initiate, validate, and disburse payments via smart contracts. The proposed framework had a high level of payment security, but it was unfortunately not contextualized in terms of the reality of interim payments in the construction industry. Ahmadiheykhsarmast and Sonmez [8] presented a smart contract payment security system for eliminating or reducing payment issues in construction contracts. An add-on module was developed in the system to transfer the required schedule and cost data to the smart contract through a project management software. Afterwards, the project progress payment amount would be blocked, and the actual payment amount would be paid monthly. However, the add-on module was not connected to the smart contract system, and it was not feasible for clients and contractors to code the conditions in the smart contract.

2.4. Linked Building Models

As a working method, BIM requires close and coordinated collaboration, which increases the need for a shared space where the project team can put, hold, and access information throughout BIM projects [14]. Such a shared space is a Common Data Environment (CDE). According to the ISO 19650-1:2018 standard, a CDE is an “agreed source of information for any given project or asset, for collecting, managing, and disseminating each information container through a managed process” [49]. The integration of technical or spatial part models into a shared coordination model using the Industry Foundation Classes (IFC) data format is supported by most authoring tools and collaboration platforms [50]. Furthermore, the building model and the inherent model elements can be linked to further data, such as documents, plans, schedules, or Bills of Materials (BoMs) [51]. This integration of document-based information can be realized either by referencing the documents using the IFC schema or with a more general approach using a decoupled set of links in a separate file with a predefined schema and structure [51]. However, the extensions of the IFC data schema to integrate cross-domain information into a common shared model do not increase the interoperability between applications [52]. Therefore, a schema for modeling link sets was employed in a generic multi-model approach [52] and the COINS container [53], which both proposed a comprehensive data model comprising a hierarchy of different link types and meta-data descriptions for information models. The efforts of the container developments mentioned here were bundled and published in an international standard called the Information Container for linked Document Delivery (ICDD) [54]. The ICDD supports object-oriented and document-oriented data and supplements the integrated lifecycle information management. It is modeled using Semantic Web technologies recommended by the World Wide Web Consortium (W3C) and enables an extendable structure of the container schema, unique identification of objects, and independent localization of data for the integration of distributed data [53]. The main purpose of the container-based data exchange is to provide a format for linked domain models utilized by several different application cases [52]. These can be used to, e.g., link autonomous

part models to a coordination model, versioned models to an overall model, or create cross-domain models, such as a 4D scheduling model or 5D costing model.

3. Challenges and Applicability of Blockchain and Smart Contracts in AEC

Despite the omnipresence of the blockchain topic and the fast-growing number of articles about the use of smart contracts in the construction context, there is a lack of implementation and easy-to-integrate solutions. Several researchers have also referred to the need for validation and empirical analysis of the applicability regarding the proposed theoretical approaches [2,3,20]. Scott et al. [5] reviewed 21 papers from 2019 to 2020 on blockchain applications in the construction industry. However, less than half of the reviewed papers (38%) discussed technological components that allow for interaction with blockchain applications. It emphasized the skepticism about the usability and readiness of the industry for the adoption of blockchain solutions. Indeed, many issues will need to be addressed before the new technology can be integrated into daily business and reach its full potential. Li et al. [2] analyzed the current state of DLT in the construction sector and presented, beyond the opportunities, an extensive list of identified challenges that concern various social, political, or technical issues. The concept at hand addresses many of these obstacles and tries to propose an adequate solution.

The main idea of the proposed BIMcontracts framework is a gradual and coordinated process of integrating new technologies and organizational changes while considering the current conditions of the construction industry. Some authors emphasized the resistance to change and the unreadiness of the industry for the adoption of blockchain-based solutions [2,3]. However, even if the digitization level is not high enough to take full advantage of them [2], certain digital technologies, processes, and standards have already been established. The research should now use that stage as a springboard for striking concepts that inspire new possibilities. This may help to create new long-term incentives for solving the typical AEC problems with the quicker sharing of information, trust, and collaboration [2]. Clear communication of the benefits—such as faster accounting, quality, control, transparency, and reduced risk of human error through automation—to the target audience is enormously important. The early involvement of experts from the practical field (general contractors, subcontractors, construction law specialists) in the concept phase of BIMcontracts ensures that feasibility and acceptance are checked at an early stage.

To increase acceptance, the solution has to be integrated into the existing systems that people in the AEC industry are familiar with. Many SMEs may not have the resources to invest in new IT infrastructures or new software. Usability, cost issues, complexity, and a lack of skills and know-how are frequently mentioned barriers [2,3,6]. The proposed concept considers a smart contract as a service ecosystem using an API-based approach; it supports the conclusion and execution of the contract and provides pre-programmed and construction-compliant smart contract templates that are easy to configure. The use-case-specific template with all necessary inputs will be deployed on the blockchain. The final users of the system should adjust the desired settings and provide all necessary inputs for the automatic generation and deployment of smart contracts on the blockchain. In this way, neither the coding of smart contracts nor DLT skills are required.

Data protection and data security are other important issues for blockchain applications [2]. To ensure these, this concept proposes the storage of all of the sensitive and important information in off-chain storage. Only secure resources, such as a CDE, that meet all requirements for data storage should be used. For the data stored in a CDE, a hash value is saved within the smart contract as the immutable checksum of the file. The privacy of the data stored on-chain is secured by the private consortium blockchain GoQuorum, which also improves the scalability and performance. The use of the IBFT Proof-of-Authority consensus mechanism [30] brings further benefits, such as limited access to the information on the blockchain.

Cryptocurrencies, which are often associated with blockchain, are also considered problematic. Indeed, there are still many obstacles on the way to the widespread use

of cryptocurrencies as a medium of exchange, such as the lack of regulations, restricted number of possible payment transactions, and fluctuations [2]. However, there is no necessity for the use of cryptocurrencies; therefore, the BIMcontracts concept is based on the usual payment processing involving bank-to-bank transactions.

4. BIMcontracts Approach

4.1. Conditions and Requirements

The BIMcontracts approach is proposed for automated payment and contract management through blockchain-based smart contracts (see Figure 1). The main emphasis is on the combination of BIM with smart contracts and the developed software architecture. In the first step, the focus is, on the one hand, exclusively on the contractual relationships between an owner and a general contractor and on a general contractor and its subcontractor on the other hand. Since all other relevant contractual relationships can be considered similar with respect to the payment process, our findings can be generalized for future applications.

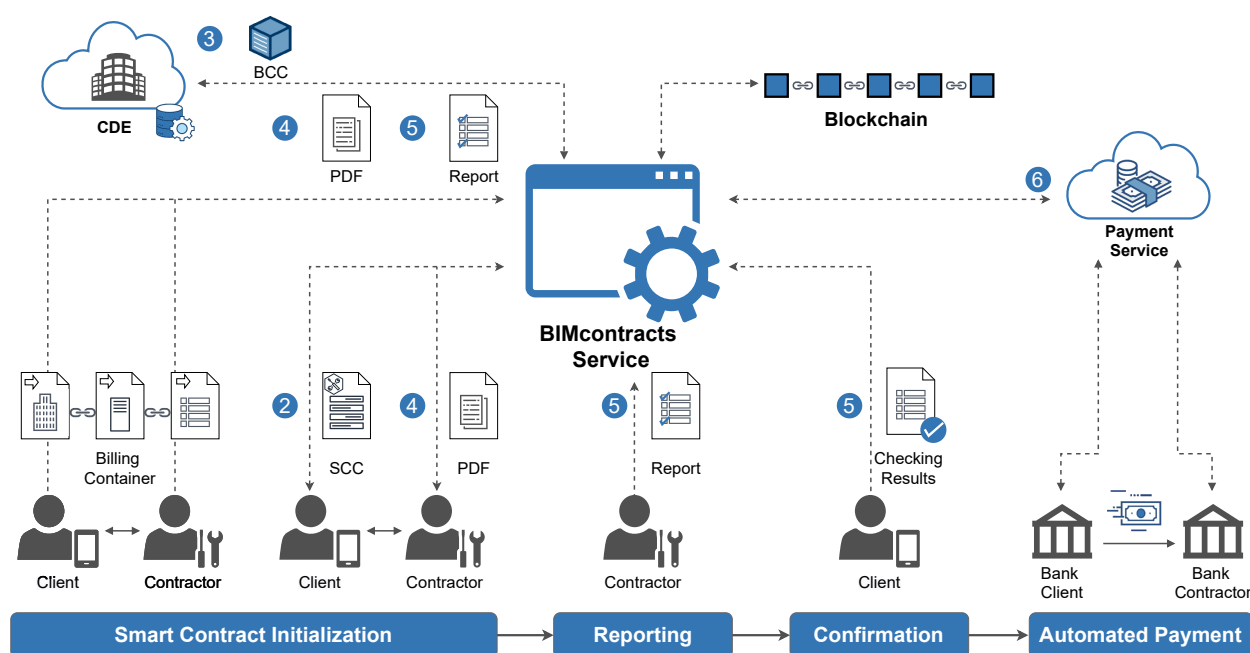


Figure 1. BIMcontracts framework.

The use of a smart contract does not imply the complete digitalization of the contractual relationship between the parties. The parties still conclude a classic paper-based construction contract with digital contract elements. This holds true regardless of whether the parties opt for the application of German law (BGB, VOB/B) or make use of internationally recognized contractual standards (especially FIDIC contracts [55]). The legal issues at hand remain fundamentally the same so that no further differentiation is required for this purpose. Furthermore, the appropriate adjustment of other elements of the contract is an important consideration, but is outside the scope of this paper. To make the smart contract algorithm legally binding, its use must be specified in advance in the paper-based contract. The digital counterpart of the contract is signed via uniquely identifiable blockchain identities.

A prerequisite for the conclusion of the contract is the provision of digital BIM-based tender documents by the client or by the contractor. These documents are agreed upon between the contracting parties during the tendering procedure. The data relevant to the contract should be provided as a digital building model (BIM model) linked with materials, works, and cost in the form of a detailed BoQ using detailed Quantity Take-Offs (QTOs). This information can be understood as a part of a 5D BIM model and can be delivered as an information container.

As already pointed out, it is not efficient to store all of the project-related data on the blockchain. The concept assumes the usage of a CDE in the construction project and treats it as off-chain storage, where it is possible to store any structured (e.g., BIM models, BoQ files) or unstructured (e.g., documents, photos) information. For secure and resource-efficient storage of the information in a blockchain, only checksums in the form of cryptographic hash values of the files stored in the CDE are used. The hash value of a file is generated based on all of the data of the file and the timestamp of the generation time. Once the confirmed file data are modified, a new hash value will be generated that differs from the confirmed hash value. This procedure secures the immutability of the project data stored in the CDE.

4.2. Concept Description

The core of the BIMcontracts system is a so-called billing container that enables the processing of payment-relevant transactions via smart contracts (see Figure 1, ①). In addition to the BIM model and the BoQ, the billing container comprises a billing plan that has been designed to automate transactions [56]. Individual billing units are defined on the basis of the BoQ. If a billing unit is completed and confirmed by the client, the contractor will get the agreed payment of the billing unit. The billing units contain inputs about lump sums, unit prices with quantities, and the completion date. Furthermore, they reference the construction work items and building elements to be completed by linking to both the BIM model and the BoQ. All billing units together compose a billing plan, which also has to be agreed upon between the two contracting parties. Depending on the type of the contract and the underlying service description, billing plans can have different details, from a rough trade-oriented structure with lump sums to the mapping of each quantity from the QTO to the billing unit. The grouping of items from the related BoQ is also possible. All linked digital contractual elements (BIM model, BoQ, and billing plan) can be combined into an information container according to the ICDD standard, as presented in [56].

The workflow of managing smart contracts essentially consists of three phases: the preparation phase, the initialization phase, and the execution phase (Figure 2). The preparation phase primarily includes the arrangement of blockchain identities and registration in the system. The billing container and the paper-based contract are to be coordinated for each contractual relationship during the negotiations and can then be loaded into the BIMcontracts system. Once verified, the execution details of single billing units, control procedures, and the contractual agreement as a whole can be customized as a part of the Smart Contract Configuration (SCC) (Figure 1, ②). The SCC is a machine-readable format of all relevant information and agreed-upon contract terms.

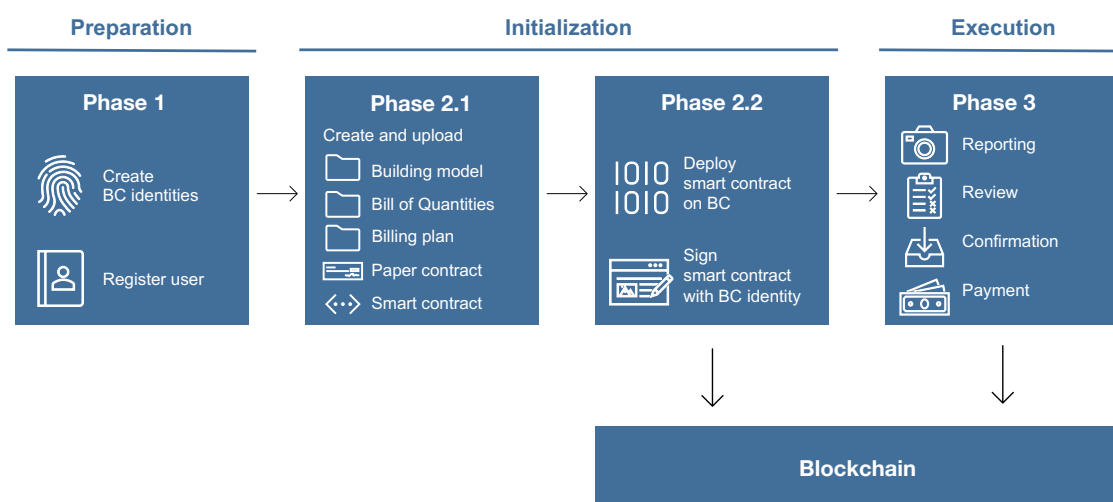


Figure 2. Three phases of the workflow for managing smart contracts.

As soon as an agreement has been reached, the linked digital contractual elements (BIM model, BoQ, and billing plan) and the SCC are exported as an information container, referred to as the BIMcontracts Container (BCC), according to the ICDD standard (Figure 1, ③). Figure 3 shows the file structure of the container. The container is stored off-chain in the CDE. The calculated hash values of both container and internal files are included in the confirmed paper contract to identify the billing basis of the contract. In the next step, a scan of the signed paper-based contract and its hash value are also loaded into the system and stored in the CDE (Figure 1, ④). Afterwards, smart contracts are generated based on the SCC, including the hash values of all documents, and finally, the smart contracts are deployed on the blockchain. The contract conclusion is completed once both contracting parties have signed the smart contracts with their blockchain identities, thus initiating the execution phase (Figure 2). If the client checks and confirms the contractor's work result during the execution, the inspection report will also be stored in the CDE (Figure 1, ⑤), and a direct payment from the client's bank to the contractor's bank will be triggered based on the billing plan data and the smart contract conditions (Figure 1, ⑥). The following sections discuss each phase in detail and introduce the software architecture of the framework.

4.3. Initialization of the Smart Contract

The initialization phase is split into two steps: (2.1) *Linking Billing Plans and Smart Contracts* and (2.2) *Smart Contract Generation and Deployment* (see Figure 2). In these two steps, all required information and contractual details will be gathered, agreed upon, and then converted into a format suitable for saving and executing on a blockchain. The goal of the first step is to upload all digital contractual elements and create the SCC. It consists of three types of data: (i) details about the contractual participants, related companies, and people, (ii) digital contractual elements, each with their hash values, and (iii) the customized execution details, referred to as billing arrangements. This allows the configuration of payment intervals, partial payment options, confirmation and payment deadlines for progress payments, security deposits, and other payment-related conditions. An example for billing arrangements is shown in the case study (see Section 6.3). It can be specified, e.g., that the reporting only occurs in a certain time interval or that partial payments for larger tasks are also enabled above a certain stage of completion (e.g., 50%). Default billing arrangements can be initially set for all billing units, but can still be customized for each unit separately. The resulting SCC serves as the basis for generating and deploying the smart contract in the next step.

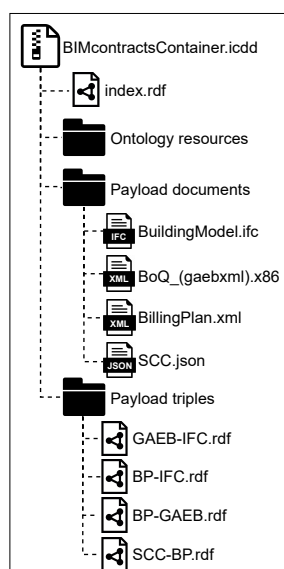


Figure 3. File structure of the BIMcontracts Container (BCC).

To make the system ready for the execution phase, the smart contract must first be created and activated on the blockchain. The SCC is used to generate and customize a specific smart contract based on pre-defined templates. Figure 4 shows a simplified example of the SCC using the JavaScript Object Notation (JSON) format. Each billing unit contains one or many billing unit items representing a single item from the BoQ or a quantity split from the QTO. To track the status of each item of the billing plan (billing unit or billing unit item), reusable code templates are defined and implemented as state machines (see Section 5.2). State machines represent a set of valid states and specific transitions between those states based on pre-defined conditions. While generating smart contracts, each item in the billing plan is mapped to a smart contract representing a certain state machine template.

```

1  {
2    "id": "0xd0e8549b659518b925caa3b0c186b95a063027b4", /ID of contractual agreement/
3    "projectId": "0xf827a8ed7a5b932a89ba4a5908b3d3f92e3cee0d", /ID of Project "Suedufer"/
4    "client": "0x90b2e5d4815f4cff29d7dad131eb5d041f58acbf", /identity of client company/
5    "contractor": "0x5c3fcb83f08b31df5e1cac9559e4e5403f1b3d96", /identity of contractor company/
6    "billOfQuantitiesDocument": {
7      "documentId": "01b3052d366a0212",
8      "version": 1,
9      "fileIds": [
10       "d2b07670794c0ed0"
11     ]
12   },
13   "billingPlan": { /document description/ },
14   "bimModel": { /document description/ },
15   "configurationItems": [
16     { "type": "billingArrangements",
17       "value": "paymentInterval" },
18     { "type": "partialPayment",
19       "value": "false" },
20     { "type": "confirmationInterval",
21       "value": "7" }, /... further items .../
22   ],
23   "billingUnits": [
24     { "id": "BUID_22d465a2",
25       "configurationItems": [
26         { "type": "partialPayment",
27           "value": "true" }
28       ],
29       "items": [
30         { "id": "ID_3311_2896_1",
31           "quantity": 3.063,
32           "unitPrice": 3.75 }
33       ]
34     },
35     { "id": "BUID_e6aa0776",
36       "configurationItems": [
37         { "type": "confirmationInterval",
38           "value": "14" }
39       ],
40       "items": [
41         { "id": "ID_3311_2896_2",
42           "quantity": 7.563,
43           "unitPrice": 3.75 }
44       ]
45     }
46   ]
47 }

```

Figure 4. Example of a smart contract configuration, which is sent to the generator via the REST API.

The overall process for smart contract generation and deployment is shown in Figure 5. First, the contract configuration is created in the Client application and the Management Backend and is then sent to the Blockchain Communication Backend to generate smart contract templates based on the SCC. These templates are converted into raw smart contract deployment transactions, which are signed by the Signing Backend and sent to the blockchain (based on the meta-transaction concept [35] and off-chaining design patterns [36]). The smart contract code to be deployed is prepared with details from the SCC, such as the project ID, client, contractor, and references to all contractual documents. The contract creation on the blockchain returns a public smart contract address that is stored in the internal database of the Blockchain Communication Backend and linked to the corre-

sponding contractual agreement. Both contracting parties must give their agreement to the use of the smart contract for reporting and payment processing. Two steps are required for this. First, the customer initializes the smart contract and, thereby, gives their agreement. Then, the contractor sends a signed approval transaction to the smart contract, officially confirming the bilateral agreement.

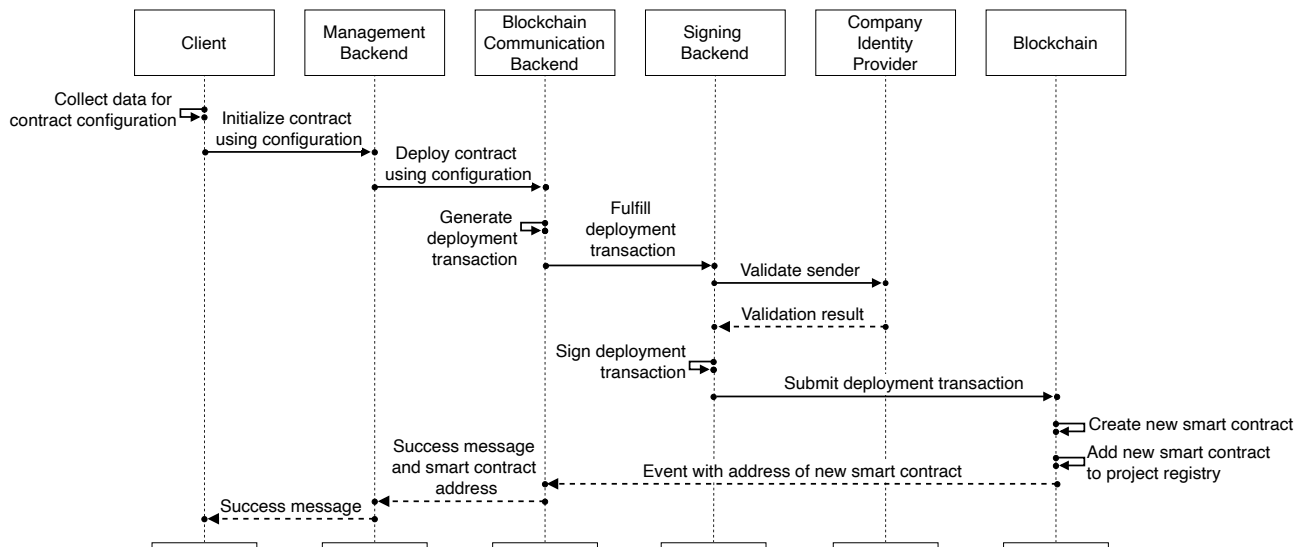


Figure 5. Sequence chart of the generation and deployment of the smart contract.

4.4. Smart Contract in the Execution Phase

After the successful signing of the contract, the system is ready for use, and the execution phase of the construction project begins. The process of the delivery and acceptance of this phase consists of the following main steps (see Figures 1 and 2): (i) reporting the completion of a construction task by the contractor; (ii) checking the results and a notification of the decision (confirmation or rejection) by the client; (iii) automatic triggering of the payment in the case of a (partial) confirmation. Each activity is handled as a transaction and is stored in the blockchain to ensure the validity and traceability of the processes. Thus, the progress of contract fulfillment is continuously and transparently visible to both contracting parties.

During the execution phase, the executed work and the problems, defects, remedying, and repair procedures that occur are documented. Many digital tools already exist to support construction documentation, final acceptance, and issue management. Meanwhile, model-based reporting is already provided by several tools, which are often referred to as field BIM tools [57]. Thus, the concept of BIMcontracts takes into account such model-based reporting and the use of current tools based on open standards. The contractor has the ability to report completion by selecting billing units, items from the BoQ, or even model elements, as the link between a billing schedule, the BoQ, and a BIM model is always provided.

In addition, the contractor should be able to include detailed documentation as evidence, such as measurements in the case of a unit price contract. The documentation can be done by means of text messages, voice messages, or photos and videos. After receiving a completion notification, the client checks within the agreed period whether the work has been carried out without any significant defects. At this point, three scenarios are possible:

1. The construction work is complete and free of defects, so the client confirms and rewards it.
2. Refusal of the confirmation of completion due to existing defects.
3. Partial confirmation of completion with retention.

At the lowest level of complexity is the correctly executed construction process: The client accepts the completion and confirms that the work has been performed without any significant defects. Thus, a payment that is related to the respective billing unit or billing unit element (in the case of an agreed partial payment) is automatically triggered. The security deposit will be subtracted accordingly.

Nevertheless, during the construction, defects may occur. In general, retentions in the case of defects and open work are to be treated in the same way. The billing procedure follows the same logic: The client can either refuse the payment of the reported work or define retention of the amount to be paid. Both refusal and retention due to defects or outstanding remaining work must be justified and documented as precisely as possible by the client. In the event of a refusal, according to the client's opinion, the work has not yet been performed as per the agreement. The contractor can now repeat the work and submit a new notification of completion. A partial confirmation is also possible, which legally corresponds to justified retention of payment due to defects. Through retention, the contractor remains more liquid against a denial of the payment.

5. System Architecture and Technology

The system architecture is outlined in Figure 6 and follows a component-based approach. The general idea of the system architecture is to define software building blocks that are compatible and serve as clearly defined interfaces for further extension or are replaced by existing software tools. The overarching goal of the system is to lower the access barrier for all participants.

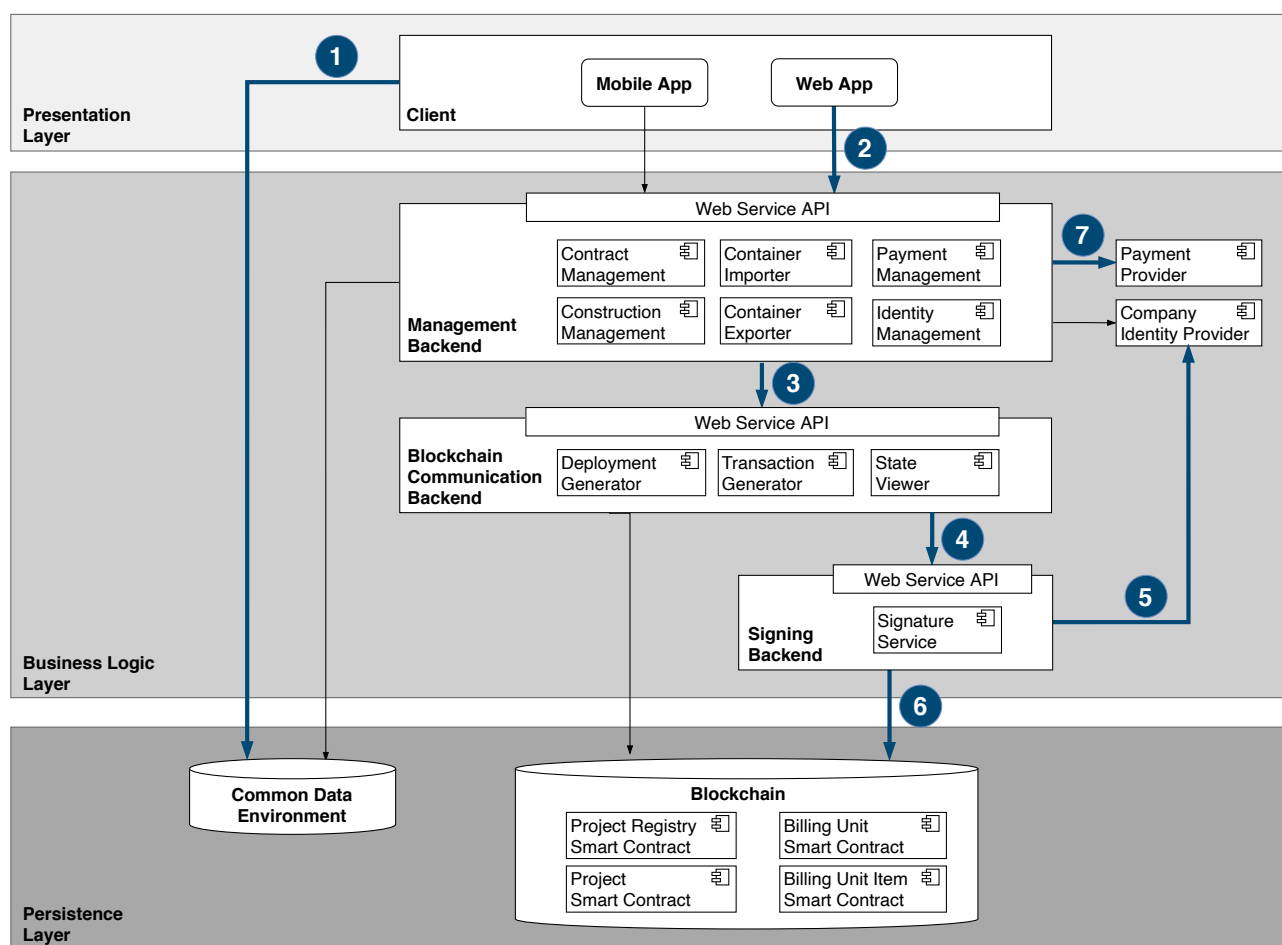


Figure 6. Confirmation process within the system architecture.

This is also the reason for why a mobile and web app was created, and it can be accessed from a browser on any smartphone, tablet, or desktop computer for all interactions with the system. The core of the system is the Management Backend component. It coordinates and encapsulates all interactions between the client, Blockchain Communication Backend, Payment Provider, and Company Identity Provider. The Blockchain Communication Backend combines all blockchain interactions in a dedicated component, which helps to make all connected components independent from technological decisions and implementation details of the underlying blockchain technology. Having a separate component means that the selected blockchain infrastructure can adapt to new requirements regarding data privacy, compliance, and security, change control, and access management processes, or it can switch to a combination of multiple blockchains. The Signing Backend plays an important role regarding control and access management, as it handles the signing identities for each participant in the system.

5.1. Signing Backend

To interact with the system, it is necessary to create and sign transactions that are eventually sent to the blockchain (see Figure 7). Identity and key management is a crucial aspect for any interaction because transactions are signed on behalf of a company. Companies have varying requirements regarding the privacy, visibility, and traceability of their internal processes. While interacting with the system, companies want to preserve their privacy among other project participants. Moreover, companies might require different levels of access to the system from within their own organizations. For example, payment decisions could be restricted to the highest management level in the organization, whereas other decisions could be made by the construction manager on the project. At the same time, some companies require internal traceability of the transaction signing to implement their access and control management processes.

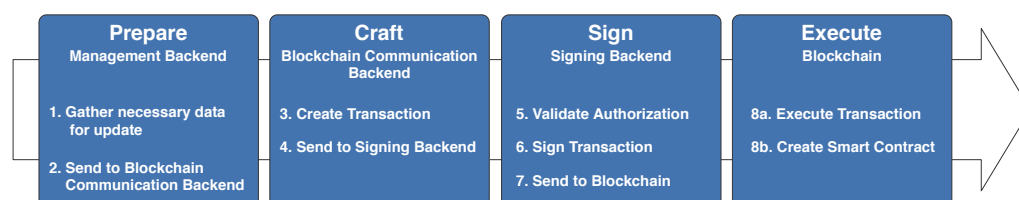


Figure 7. General process for creating and signing transactions.

Companies using the BIMcontracts system have the ability to select between two degrees of autonomy regarding identity and key management functionalities: (i) a centrally provided component managed by the BIMcontracts system or (ii) a self-hosted component within their company-internal data center. This results in a flexible component that is able to address different levels of trust towards the BIMcontracts system. Thus, the system can also be tailored to the specific skills and requirements of the participants.

5.2. Architectural Details and Confirmation Process Example

In this section, we describe an exemplary system walkthrough of the confirmation process during the execution phase of a project, as shown in Figure 6. Consider the following scenario: A contractor finished their construction task, and the client wants to confirm the completion, eventually triggering a payment to the contractor. The contractor has already uploaded a detailed work documentation (text documents, photos) in the mobile or web app. These files are stored in a CDE (Figure 6, ①). The client uploads a review documentation and confirms the completion, thereby sending the review details to the Management Backend (Figure 6, ②). Additional technical information, such as the target smart contract address or the client's digital identity, is added and sent to the Blockchain Communication Backend (Figure 6, ③). The Blockchain Communication Backend crafts a transaction to trigger the review function in the smart contract and sends this transaction to the Signing Backend (Figure 6, ④). In step ⑤, the client's permission is

validated and, in the case of acceptance, the transaction is digitally signed by the client's company. The signed transaction is sent to the blockchain, where the task review results in a smart contract state transition that emits a payout command (Figure 6, ⑥). Finally, in step ⑦, the Management Backend is informed about this command and forwards it to the Payment Provider, leading to a bank transfer.

As mentioned above, process steps that are necessary for payment are virtually represented as state machines. To be able to track the progress of payments as well as the progress of a construction task, both levels of detail are mapped onto separate state machines. Nevertheless, both machines are interconnected, and a change on the task level can automatically trigger a transition on the payment level. Hence, the actual payment state is derived from the underlying task state(s).

As visualized in Figure 8, billing unit items can either start as *Open* if they are mandatory or in the *Not Requested* state in case of a contingent item. Ideally, a contractor claims completion and the client confirm this, which results in a full payment confirmed by the contractor. In the worst case, both parties agree on the cancellation of the billing unit item. If clients find the work faulty or only a partial payment shall be made, both parties must agree thereupon. The result might be full payment, a discounted payment, or even a task cancellation.

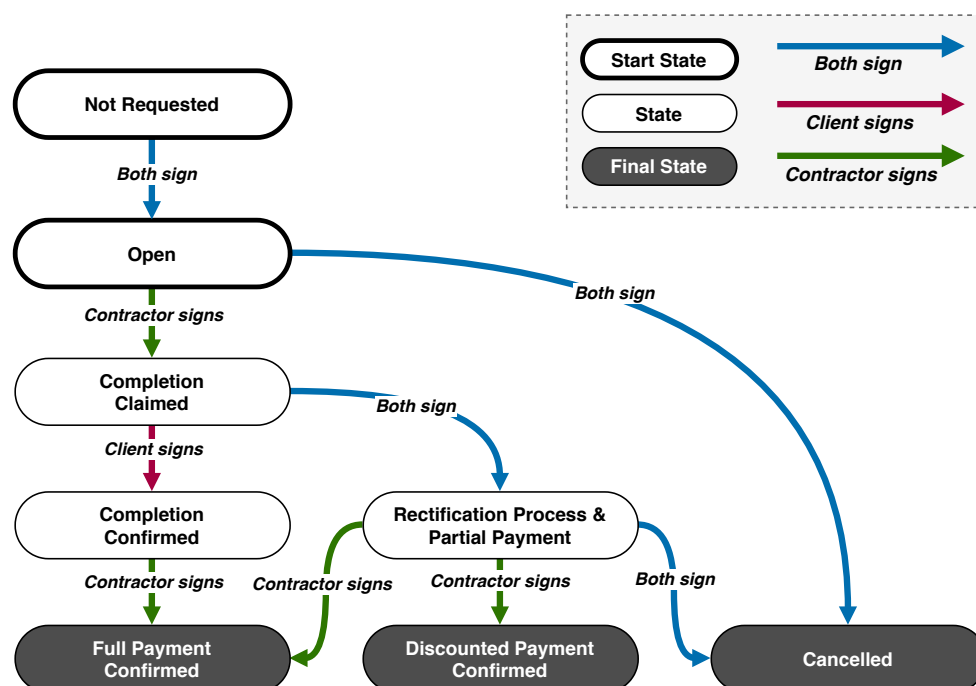


Figure 8. State machine billing unit item.

Figure 9 shows the related billing unit state machine and its possible transitions. Every unit starts as *Unpaid*, and payment confirmations of inferior items can trigger a transition into the *Partial Payment* state. As long as at least one item has not reached a final state, the billing unit remains in the partial payment state. Based on the final states of the items, billing units can either end at *Full Payment* or *Discounted Payment*.

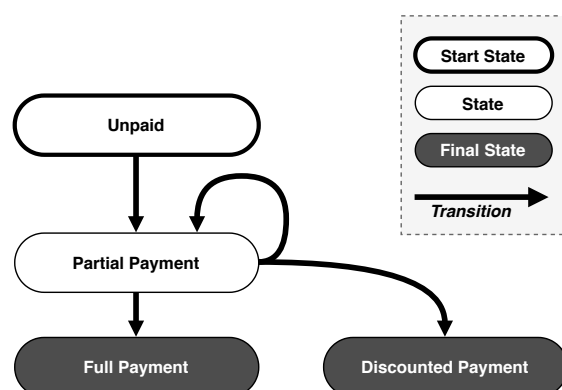


Figure 9. State machine billing unit.

5.3. Blockchain Solution

The BIMcontracts prototype uses the private consortium blockchain GoQuorum by ConsenSys. This blockchain implementation is based on Ethereum and uses a modified version of the default node client Geth. This enables the usage of even more business-application-focused blockchain features while still benefitting from the Ethereum ecosystem and being compatible with other blockchains. GoQuorum introduces a basic access-control mechanism that improves the privacy of the data stored on-chain [32]. Unintended public or unauthorized access is prevented on an infrastructural level. This enhances the privacy of, e.g., competitive information and interactions between parties stored on-chain. Furthermore, GoQuorum offers the possibility of private on-chain interactions that exclude other blockchain participants from accessing sensitive information. This feature can be used in the future to hide not only the content of transactions, but the transactions as a whole [31]. By default, BIMcontracts already only stores references on-chain, and the actual data are stored in an external system (e.g., CDE).

The current Ethereum PoW consensus algorithm requires a large amount of energy and can only process about 15 transactions per second on the mainnet. One of the main goals of BIMcontracts is to reduce the effort and time it takes to process payments. Therefore, BIMcontracts uses the IBFT algorithm [30] as a PoA consensus mechanism. IBFT can process significantly more transactions than PoW by using a less resource-intensive validation scheme. BIMcontracts participants can rely on the finality of transactions, since IBFT guarantees only one possible block at a time, which prevents chain splits (forks). This is one step further towards the intended increased confidence in payment processing and an improved planning security, especially for smaller companies [31,32].

6. Evaluation

6.1. Elaboration of the Evaluation Process

Several application scenarios in the area of building construction have been developed for the systematic derivation of the functions required for the implementation of the BIMcontracts system. Thus, these scenarios form the basis for the methodical development and technological prototyping of the framework. In addition, they allow for the checking of prototypes for the complete and correct mapping of workflows and requirements during the implementation stage. Generally, the scenarios distinguish between two types of contracts and relationships: a lump sum contract between an owner and a general contractor, and a unit price contract between a general contractor and a subcontractor.

At the lowest level of complexity is the undisturbed construction process, in which the contractor performs the contracted service and the client pays for it. Two scenarios have been prioritized in this respect for current development: application scenario 1 (lump sum contract) and application scenario 2 (unit price contract). At a further stage, it should be possible to include the defects recorded during construction in payment processing and automatically map the effects of defect management in application scenario 3 (retentions due to defects during construction). Moreover, an increase in complexity is required if

material supplements (changes in services) are to be included in the partially automated billing process. This is the subject of application scenario 4 (material supplements). Other more complex scenarios are conceivable, but the current focus is on implementing these four stages with the help of the prototype.

For all application scenarios, billing containers are the prerequisite for digital billing. The necessary data (BIM model and BoQ with QTO) are based on a real-life construction project and were determined, prepared, and provided by a construction company for each application scenario. In the next step, billing plans should be defined and information containers should be created on this basis. For this purpose, the prototypical software component presented in [56] was used. Thereby, the initial preparation of the data was completed for the evaluation of the BIMcontracts prototype.

6.2. Development of the Prototype

For the evaluation of the proposed concept, a prototype was developed according to the system architecture defined in Figure 6. The modular system architecture leads to a micro-service implementation of the single components in the business logic layer. These services can be consumed by multiple clients that form the presentation layer, e.g., the proposed mobile and web apps.

The business logic layer employs current technologies, such as Spring Boot, Hibernate, the Java Persistence API, and a Postgres database, to provide functionalities, data persistence, and the API routes for the connectivity to client components. The micro-services are deployed and orchestrated using Docker. The business logic services are implemented with regard to the proposed concept. The backend is secured with the OAuth2 authorization flow as an established authorization method provided through Keycloak. The file-based data persistence is implemented according to the openCDE-conform web service as specified in DIN SPEC 91391-2:2019 [58]. This technical specification defines the container-based exchange of documents and construction-specific metadata via web services. The implementation of an openCDE-conform web service enables the system to store the BCC persistently and access it for information retrieval.

The Blockchain Communication Backend is implemented using a NodeJS server, which offers a REST API via the Express package and is written in TypeScript. Blockchain identities, i.e., private keys for signing transactions, are currently stored in a MongoDB database and accessed via mongoose from the NodeJS server. The underlying blockchain is the consortium variant GoQuorum, which runs on a virtual machine. The NodeJS server accesses the blockchain and creates transactions using the EthersJS package. Smart contracts are developed with the Solidity programming language, and the Hardhat tool suite is used for compiling contracts, running test cases, creating TypeScript interfaces, and deploying to the blockchain.

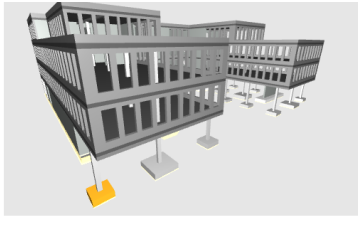
The frontend of the web app (see Figures 10 and 11) was developed using Angular. To provide convenient access to IFC-based information within the system, a viewer for IFC files was implemented in the frontend. Therefore, the XbimWebUI components were used, which are part of the Xbim Toolkit [59]. XbimWebUI offers a WebGL-based viewer as a typescript library. To present IFC files on a website, the models must first be converted into a binary format provided by the Xbim Toolkit, which is also deployed in a separate micro-service. The prototype is still under development and will be validated shortly based on the first two application scenarios. In particular, the interaction of the individual implemented components still needs to be extensively tested and evaluated.

BIMcontracts General Contractor

Contract for Project: Südufer

Contract Overview **Billing Container** Configuration Data

Upload IFC Models



Upload Bill of Quantities

Item Description	Quantity	Unit	Unit Price	Net Amount	Tax	Amount
>1.1.30 PE-Foil d=0.2 mm slip membrane	1,009,847	m²	1,05 €	1.060,34 €	19 %	
>1.1.40 batter board	10	Stck	39,00 €	390,00 €	19 %	
✓1.1.50 in-situ blinding concrete C 12/15	1.019,471	m³	3,75 €	3.823,02 €	19 %	
ID_3311_2896_1	3,063	m²	3,75 €			
ID_3311_2896_2	7,563	m²	3,75 €			
ID_3311_2896_3	3,61	m²	3,75 €			
ID_3311_2896_4	5,29	m²	3,75 €			
ID_3311_2896_5	3,61	m²	3,75 €			
ID_3311_2896_6	3,61	m²	3,75 €			

Upload Billing Model

ID	Description	Quantity	Unit	Unit Price	Net Amount	Planned Completion Date
>I2898	in-situ concrete pad and strip foundation	89,63	m²		1.782,74 €	
>I2897	in-situ, lean concrete fill	1	m³		18,60 €	
✓BUID_22d465a2	ID_3311_2896_1	3,063	m²		11,49 €	31.03.21
	QtySplit in-situ blinding concrete	3,063	m²	3,75 €	11,49 €	
✓BUID_e6aa0776	ID_3311_2896_2	7,563	m²		28,36 €	31.03.21
	QtySplit in-situ blinding concrete	7,563	m²	3,75 €	28,36 €	
>BUID_fe768c9c	ID_3311_2896_3	3,61	m²		13,54 €	23.03.21

Figure 10. Visualization of the imported billing container (linked IFC model, BoQ with QTO and billing plan) in the BIMcontracts application.

BIMcontracts General Contractor

Contract for Project: Südufer

Contract Overview Billing Container **Configuration** Data

Billing Plan

Filter

ID	Short Description	Description	Unit Price	Planned Completion Date	Configuration
BUID_6be62148	in-situ	in-situ base plate, d = 60 cm; in-situ concrete pad and strip foundation; in-situ, lean concrete fill; in-situ blinding concrete C 12/15;	13.383,75 €	30.06.21	
BUID_22d465a2	ID_3311_2896_1	QtySplit in-situ blinding concrete	11,49 €	31.03.21	
BUID_e6aa0776	ID_3311_2896_2	QtySplit in-situ blinding concrete	28,36 €	31.03.21	
BUID_fe768c9c	ID_3311_2896_3	QtySplit in-situ blinding concrete	13,54 €	23.03.21	
BUID_4ab4f80f	ID_3311_2896_4	QtySplit in-situ blinding concrete	19,84 €	23.03.21	
BUID_89c4b9c9	ID_3311_2896_5	QtySplit in-situ blinding concrete	13,54 €	23.03.21	
BUID_2f775935	ID_3311_2896_6	QtySplit in-situ blinding concrete	13,54 €	23.03.21	
BUID_28b4a05a	ID_3311_2896_7	QtySplit in-situ blinding concrete	39,61 €	23.03.21	
BUID_27230fbb	ID_3311_2896_8	QtySplit in-situ blinding concrete	39,61 €	23.03.21	
BUID_bc22988d	ID_3311_2896_9	QtySplit in-situ blinding concrete	13,54 €	23.03.21	

Items per page: 10 1 - 10 of 15

Default Configuration

Installment Payment

Billing Arrangements

☐ Billing Plan

☒ Payment Interval

☐ Minimum Amount

Payment Interval (in days): 14

☒ Partial Payment

Minimal Stage of Completion: 50

Confirmation Completion

☐ Secondary Confirmation Required

Confirmation Interval (in days): 7

Security Deposit (%): 10

Closing Invoice

Figure 11. Listed billing units and configuration of billing arrangements.

6.3. Case Study

This paper highlights a small case study that assumed a unit price contract between a general contractor and a subcontractor (application scenario 2). The subcontractor was contracted for the execution of shell construction work. The four-story office building had a gross floor area of 4500 m², two staircases, and two large roof terraces. The accounting was based on unit prices for the finished construction works. The subcontractor documented the work carried out, e.g., with a quantity survey, which had to be confirmed by the general

contractor. Moreover, it was assumed that both contracting parties were registered to the proposed system and had received their blockchain identities.

As part of the tendering procedure, the execution drawings were provided as a BIM model (Version IFC2x3) with a detailed BoQ (Figure 10). The IFC model contained all necessary semantic and geometric information of the foundations, walls, beams, slabs, stairs, and columns. A high level of detail in the model was a prerequisite for a correct link to the work items. The model differentiated the walls according to the various types of construction and component thicknesses: semi-precast facade elements, prefabricated facade elements, hollow-chamber walls, and cast-in-place concrete walls. The foundation consisted of strip and single foundations, as well as a floor slab with lowerings in certain areas to accommodate the ground structure in the outdoor facilities. The shafts for the building services and elevator underpasses were also created. Under- and overlays, as well as the attic edging, were modeled as beams. If semi-precast elements were involved, they are provided with appropriate properties. The stair landings (precast elements) and intermediate platforms (in situ concrete) were modeled separately. Recesses were provided in the components where necessary; otherwise, they were produced with core drillings. The BoQ contained all relevant details for each item of work, as well as the agreed-upon net price per item, and was linked with the IFC model using unique identifiers. The billing plan had the highest possible level of detail, where each quantity was assigned to the billing unit.

The next step was to specify the execution details for the billing units and other contractual agreements (Figure 11). The billing arrangements were configured so that completed work could be reported and invoiced every 14 days. The client had seven days to check the completion and release it for payment. In addition, a security deposit of 10% per billing unit was agreed upon. Since the billing units were defined at the level of single quantities, a high level of granularity was ensured, and agreement on partial payments was not necessary. This configuration was applied to all billing units; however, an adjustment per billing unit was also possible, as demonstrated in Figure 11. Next, the smart contract could be generated and deployed, as described in detail in the previous sections. As an exemplary smart contract interaction, the client's contract approval and the corresponding transaction are shown in Figure 12. To demonstrate the billing and payment functionalities, it was assumed that the first completion reports would be provided after 14 days from the contractual start date. After a check, the client confirmed the completion, which triggered an automatic payment of 90% of the agreed amount for these billing units at their bank.

Transactions

Showing 18 transactions

Type	Function	Hash	From	To
Contract Call	Unknown	0x01b...89950	0x39F...9004D	0xc39...835FC
Contract Creation	Unknown	0x1a3...ecd8d	0x39F...9004D	0x62b...DBee6
Contract Call	Unknown	0x3c7...d4a2c	0x9F2...198a9	0xa27...a7f82

```

{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "blockHash": "0x40ea2762ec88332bba8c4806519804114673d46a6402259f4e2533e0fb966087",
    "blockNumber": "0x4529e",
    "from": "0x9f25c3aedf9be61dbe0491aad4d82a1ecac198a9",
    "gas": "0xab1c",
    "gasPrice": "0x0",
    "hash": "0x3c7d984ea484b25bc2449b15db18eb96b15336ae82b06e80ee994c585f7d4a2c",
    "input": "0xb8b4f1a0",
    "nonce": "0x0",
    "to": "0xa270f63381e303f370c057d6ab17337f00fa7f82",
    "transactionIndex": "0x0",
    "value": "0x0",
    "v": "0x38",
    "r": "0x23048c7c830a39fc1c58cabb680e1697eefda33240ccd8c49fb8ade6e879c7fd",
    "s": "0x1d01bc5cdef05298d49fd56aa70ca5616f6b9e2b0b8c82dc9e5d281cad9f8c48"
  }
}

```

Sender Address
Transaction ID
Method "signContract"
Smart Contract Address

Figure 12. Smart contract execution in the Blockchain Explorer.

7. Discussion and Outlook

Considering a blockchain-based smart contract framework as a socio-technical system underlines the importance of social, organizational, and technical factors, as well as legal policy [2]. The integration of smart contracts into legally binding construction contracts presents a challenge due to the lack of legal precedents and regulations so far. Semi-automation seems to be a suitable compromise for offering a legally compliant and feasible solution. Construction lawyers accompany the development of the framework and ensure data security and legal compliance. An important issue to deal with is the immutability of the blockchain and the resulting advantages and disadvantages. Storing an immutable copy of the contract in the blockchain can prove beneficial for both parties, as it can prevent disputes about the individual terms of the contract. Furthermore, the immutable copy of the contract can serve as evidence in court. On the other hand, data protection concerns arise whenever personal data are to be stored without the possibility of erasing them. The system needs to be implemented in a way in which General Data Protection Regulation (GDPR) requirements are fulfilled and privacy is preserved. This problem can be mitigated from the start if as few personal data as possible are stored on the blockchain. This supports the idea of preserving privacy within the system by storing all critical files (such as legal contracts, invoices, personal data of participants, and bank details) in a CDE but having a hash value on the blockchain to prove authenticity. The security and confidentiality of transaction information are provided through the proposed use of a private consortium blockchain.

The immutability feature is considered a shortcoming considering the frequent changes and revisions in construction projects [3]. Changes in the contractually agreed scope might require the addition or removal of certain work steps, resulting in an update of the BoQ or the billing plan. The subsequent modification of billing arrangements is also possible if, for example, an additional agreement concerning partial payment needs to be concluded.

Furthermore, global changes that affect the BIM are not rare either, and the implementation of change order management is therefore needed. The BIMcontracts concept provides flexibility through the versioning of the BCC stored in a CDE and, thereby, of its internal files that are affected by the changes. Since these changes are an amendment of the original contract, both contracting parties should sign a new smart contract confirming the agreement on the update. The new hash values of the container and the internal files are calculated and saved within the new smart contract. Since the internal files are stored in a CDE, any changes during the construction project are tracked. Change order management is one of the ongoing implementation steps, and it will be addressed in more detail in future research. The same applies to the implementation of model-based reporting and defect management. As the framework relies on interoperability and openBIM standards and formats, the integration and extension of the BIM Collaboration Format (BCF) are foreseen at this point. This vendor-neutral interface plays an important role in structured project coordination and interdisciplinary collaboration. It allows for the communication of IFC-based issues and openBIM processes and can support use cases in the construction and operation phases.

Certainly, the concept can also be further developed in terms of the final account, different types of construction contracts, and the complexity of the use cases and smart contracts. Since complex scenarios are also possible, e.g., some events can also depend on the status of other billing units or even smart contracts with other subcontractors. The chaining of contractual relationships is an important issue for ongoing enhancement. However, this is not only a technical challenge, but also a legal one. In addition, some restrictions, such as deadline overruns, still have to be accepted to keep the complexity at a manageable level. A 5D BIM model is based on detailed scheduling of the works to be performed. The developed billing plans thus foresee the input of the completion date. Ideally, all dates can be obtained directly from the 4D schedule. Nevertheless, detailed model-based schedules are not in widespread use so far. In the presented concept, the completion date can be considered only as additional input information for the billing units, and the origin of the dates is not of critical importance. For this reason, the non-existence of detailed 4D models is not a hurdle for the use of the system. Contractual agreements of schedule objectives come with additional legal issues. Deadline overruns can trigger contractual penalties if the overrun is caused by the contractor. If, on the other hand, the overrun is not imputable to the contractor, it can constitute a hindrance, thus leading to a modification of the agreed-upon schedule. The deadline overrun should therefore be documented by the system (e.g., through automated notices of hindrance) to benefit from the tamper protection and trustlessness that the blockchain can provide. Due to the complexity of the problem, impacts from missed deadlines are not considered in the concept. However, they certainly represent an important aspect that should be investigated in the future. Cooperation among multiple parties usually causes several challenges, e.g., in agreeing on certain data exchange formats, a common location for saving data, or the permissions of each participant to view and edit these data. A well-planned technological infrastructure is crucial for efficient workflows and has a big impact on how teams collaborate. There are several existing approaches for addressing those challenges, but they are usually created by a single company, hosted on centralized cloud infrastructure, or are not suitable for critical payment processes. Especially with the idea of increasing the degree of automation and digitization in the AEC industry, it becomes clear that trust between the participants plays a big role and that the software components employed need to address this accordingly [41].

The proposed concept investigates automated payment between clients and contractors by considering smart contracts as a service ecosystem that binds to the blockchain and data storage (CDE). This contribution describes the conceptual framework and software architecture and gives a first insight into the current implementation. The concept was already presented to a wide audience consisting of general contractors, IT product providers, and subcontractors and was met with broad approval. The BIMcontracts frame-

work builds on the existing infrastructure and software solutions by trying to integrate the established processes as much as possible. On the other hand, where changes are necessary and improvements are possible, it tries to justify the necessity of and potential for change. This especially concerns the provision of digital BIM-based tender documents and model-based reporting. Use cases that promote collaboration are among the biggest potentials of blockchain and smart contracts alongside initiatives for enhancing transparency, traceability, and automation. Digital transformation is driven by the desire to improve and the courage to change and is merely supported by the adoption of new technologies that help to take the next step towards turning the vision into reality.

Author Contributions: Conceptualization, B.S., D.G., F.B., J.W., K.S., M.K. and X.Y.; methodology, B.S., F.B., J.W., K.S., M.H., M.K., P.H. (Philipp Hückinghaus) and X.Y.; software, B.S., F.B., J.W., M.H., P.H. (Philipp Hagedorn), P.H. (Philipp Hückinghaus) and X.Y.; data curation, B.S., F.B., J.W., P.H. (Philipp Hagedorn), P.H. (Philipp Hückinghaus) and X.Y.; writing—original draft preparation, B.S., F.B., M.H., P.H. (Philipp Hückinghaus), K.S. and X.Y.; writing—review and editing, D.G., J.W., P.H. (Philipp Hagedorn), M.K., K.S. and X.Y.; visualization, B.S., F.B., J.W., P.H. (Philipp Hagedorn), K.S. and X.Y.; supervision, M.K.; project administration, F.B., K.S. and M.K.; funding acquisition, M.K. All authors have read and agreed to the published version of the manuscript.

Funding: The authors gratefully acknowledge the financial support from the German Federal Ministry for Economic Affairs and Energy (BMWi). The BIMcontracts research project is funded within the scope of the *Smarte Datenwirtschaft* technology program run by BMWi (Funding number: 01MD19006). The Ruhr-Universität Bochum, Universität Duisburg-Essen, FREUNDLIEB Bauunternehmung GmbH & Co. KG, Kapellmann und Partner Rechtsanwälte mbB, and adesso SE (consortium leader) are involved in the consortium.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to express their gratitude to FREUNDLIEB Bauunternehmung GmbH & Co. KG for their substantial contribution to the project, comprehensive validation support, and presentation to a wide audience. Special thanks are extended to all employees involved in the preparation of the BIM models and construction data.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Ribeirinho, M.J.; Mischke, J.; Strube, G.; Sjödin, E.; Blanco, J.L.; Palter, R.; Biörck, J.; Rockhill, D.; Andersson, T. *The Next Normal in Construction: How Disruption Is Reshaping the World's Largest Ecosystem*; McKinsey Report; McKinsey & Company: New York, NY, USA, 2020.
2. Li, J.; Greenwood, D.; Kassem, M. Blockchain in the built environment and construction industry: A systematic review, conceptual models and practical use cases. *Autom. Constr.* **2019**, *102*, 288–307. [\[CrossRef\]](#)
3. Kasten, J.E. Engineering and Manufacturing on the Blockchain: A Systematic Review. *IEEE Eng. Manag. Rev.* **2020**, *48*, 31–47. [\[CrossRef\]](#)
4. Perera, S.; Nanayakkara, S.; Rodrigo, M.; Senaratne, S.; Weinand, R. Blockchain technology: Is it hype or real in the construction industry? *J. Ind. Inf. Integr.* **2020**, *17*, 100125. [\[CrossRef\]](#)
5. Scott, D.J.; Broyd, T.; Ma, L. Archival Study of Blockchain Applications in the Construction Industry From Literature Published in 2019 and 2020. In *Proceedings of the ARCOM Doctoral Workshop: Exploring the Mutual Role of BIM, Blockchain and IoT in Changing the Design, Construction and Operation of Built Assets*; Northumbria University: Newcastle upon Tyne, UK, 2020; pp. 17–27.
6. Raslan, A.; Kapogiannis, G.; Cheshmehzangi, A.; Tizani, W.; Towey, D. Blockchain: Future Facilitator of Asset Information Modelling and Management? In *Proceedings of the 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, Madrid, Spain, 13–17 July 2020; pp. 523–528.
7. Hamledari, H.; Fischer, M. Role of Blockchain-Enabled Smart Contracts in Automating Construction Progress Payments. *J. Leg. Aff. Disput. Resolut. Eng. Constr.* **2020**, *13*, 04520038. [\[CrossRef\]](#)
8. Ahmadisheykhsarmast, S.; Sonmez, R. A smart contract system for security of payment of construction contracts. *Autom. Constr.* **2020**, *120*, 103401. [\[CrossRef\]](#)

9. Di Giuda, G.M.; Pattini, G.; Seghezzi, E.; Schievano, M.; Paleari, F. The Construction Contract Execution Through the Integration of Blockchain Technology. In *Digital Transformation of the Design, Construction and Management Processes of the Built Environment*; Springer: Cham, Switzerland, 2020; pp. 27–36.
10. Hamledari, H.; Fischer, M. The Application of Blockchain-Based Crypto Assets for Integrating the Physical and Financial Supply Chains in the Construction & Engineering Industry. *Autom. Constr.* **2021**, *127*, 103711. [\[CrossRef\]](#)
11. McKinsey. *Reinventing Construction: A Route to Higher Productivity*; Technical Report; McKinsey Global Institute: New York, NY, USA, 2017.
12. Durdyev, S.; Hosseini, M.R. Causes of Delays on Construction Projects: A Comprehensive List. *Int. J. Manag. Proj. Bus.* **2018**. [\[CrossRef\]](#)
13. ECSO. *Late Payment in the Construction Sector*; Technical Report; European Construction Sector Observatory: Brussels, Belgium, 2020.
14. NBS. *National BIM Report 2018*; National BIM Report No. 8; RIBA Enterprises Ltd.: London, UK, 2018.
15. Agarwal, R.; Chandrasekaran, S.; Sridhar, M. *Imagining Construction's Digital Future*; McKinsey Report; McKinsey & Company: New York, NY, USA, 2016.
16. Smith, P. Project Cost Management with 5D BIM. *Procedia Soc. Behav. Sci.* **2016**, *226*, 193–200. doi: 10.1016/j.sbspro.2016.06.179. [\[CrossRef\]](#)
17. Schober, K.S.; Hoff, P.; Lecat, A.; de Thieulloy, G.; Siepen, S. *Turning Point for the Construction Industry: The Disruptive Impact of Building Information Modeling (BIM)*; Roland Berger Focus; Roland Berger GmbH: Munich, Germany, 2017.
18. Moses, T.; Heesom, D.; Oloke, D. Implementing 5D BIM on construction projects: contractor perspectives from the UK construction sector. *J. Eng. Des. Technol.* **2020**, *18*, 1867–1888. [\[CrossRef\]](#)
19. Forgues, D.; Iordanova, I.; Valdivieso, F.; Staub-French, S. Rethinking the cost estimating process through 5D BIM: A case study. In Proceedings of the Construction Research Congress 2012: Construction Challenges in a Flat World, West Lafayette, Indiana, 21–23 May 2012; pp. 778–786.
20. Darabseh, M.; Martins, J.P. Risks and Opportunities for Reforming Construction with Blockchain: Bibliometric Study. *Civ. Eng. J.* **2020**, *6*, 1204–1217. [\[CrossRef\]](#)
21. Ølnes, S.; Ubacht, J.; Janssen, M. Blockchain in government: Benefits and implications of distributed ledger technology for information sharing. *Gov. Inf. Q.* **2017**, *34*, 355–364. [\[CrossRef\]](#)
22. Nakamoto, S.; Bitcoin: A peer-to-peer electronic cash system. *Decentralized Bus. Rev.* **2008**, 21260. Available online: <https://nakamotoinstitute.org/bitcoin/> (accessed on 15 May 2021).
23. Antonopoulos, A.M.; Wood, G. *Mastering Ethereum: Building Smart Contracts and DApps*, 1st ed.; O'Reilly Media: Sebastopol, CA, USA, 2018.
24. Xu, X.; Weber, I.; Staples, M. *Architecture for Blockchain Applications*; Springer International Publishing: Berlin/Heidelberg, Germany, 2019.
25. Yaga, D.; Mell, P.; Roby, N.; Scarfone, K. *Blockchain Technology Overview*; Technical Report NIST IR 8202; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2018.
26. Baralla, G.; Pinna, A.; Corrias, G. Ensure traceability in European food supply chain by using a blockchain system. In Proceedings of the 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain, Montreal, QC, Canada, 27 May 2019; pp. 40–47.
27. Walzl, B.; Sillaber, C.; Gallersdörfer, U.; Matthes, F. Blockchains and Smart Contracts: A Threat for the Legal Industry? In *Business Transformation through Blockchain*; Treiblmaier, H., Beck, R., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 287–315.
28. Patel, D.; Shah, K.; Shanbhag, S.; Mistry, V. *Towards Legally Enforceable Smart Contracts*; ICBC 2018, Lecture Notes in Computer Science; Chen, S., Wang, H., Zhang, L.J., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; pp. 153–165.
29. Alsunaidi, S.J.; Alhaidari, F.A. A Survey of Consensus Algorithms for Blockchain Technology. In Proceedings of the 2019 International Conference on Computer and Information Sciences (ICCIS), Sakakah, Saudi Arabia, 10–11 April 2019; pp. 1–6.
30. Castro, M.; Liskov, B. Practical Byzantine fault tolerance. In Proceedings of the Third Symposium on Operating Systems Design and Implementation, New Orleans, LA, USA, 22–25 February 1999; Volume 99, pp. 173–186.
31. Kannengießer, N.; Lins, S.; Dehling, T.; Sunyaev, A. Trade-offs between Distributed Ledger Technology Characteristics. *ACM Comput. Surv.* **2020**, *53*, 1–37. [\[CrossRef\]](#)
32. Tran, N.K.; Babar, M.A. Anatomy, Concept, and Design Space of Blockchain Networks. In Proceedings of the 2020 IEEE International Conference on Software Architecture (ICSA), Salvador, Brazil, 2–6 November 2020; pp. 125–134. [\[CrossRef\]](#)
33. Wessling, F.; Ehmke, C.; Hesenius, M.; Gruhn, V. How Much Blockchain Do You Need? Towards a Concept for Building Hybrid DApp Architectures. In Proceedings of the WETSEB'18: IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB 2018), Gothenburg, Sweden, 27 May–3 June 2018.
34. Wessling, F.; Ehmke, C.; Meyer, O.; Gruhn, V. Towards Blockchain Tactics: Building Hybrid Decentralized Software Architectures. In Proceedings of the 2019 IEEE International Conference on Software Architecture Companion (ICSA-C), Hamburg, Germany, 25–26 March 2019; pp. 234–237.
35. Blum, F.; Severin, B.; Hettmer, M.; Hückinghaus, P.; Gruhn, V. Building Hybrid DApps using Blockchain Tactics—The Meta-Transaction Example. In Proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Sydney, Australia, 3–6 May 2020; pp. 1–5.

36. Eberhardt, J.; Tai, S. On or Off the Blockchain? Insights on Off-Chaining Computation and Data. In *Service-Oriented and Cloud Computing*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2017; pp. 3–15.
37. Yang, R.; Wakefield, R.; Lyu, S.; Jayasuriya, S.; Han, F.; Yi, X.; Yang, X.; Amarasinghe, G.; Chen, S. Public and private blockchain in construction business process and information integration. *Autom. Constr.* **2020**, *118*, 103276. [\[CrossRef\]](#)
38. Das, M.; Luo, H.; Cheng, J.C. Securing interim payments in construction projects through a blockchain-based framework. *Autom. Constr.* **2020**, *118*, 103284. [\[CrossRef\]](#)
39. Rodrigo, M.N.N.; Perera, S.; Senaratne, S.; Jin, X. Potential Application of Blockchain Technology for Embodied Carbon Estimating in Construction Supply Chains. *Buildings* **2020**, *10*, 140. [\[CrossRef\]](#)
40. Wang, Z.; Wang, T.; Hu, H.; Gong, J.; Ren, X.; Xiao, Q. Blockchain-based framework for improving supply chain traceability and information sharing in precast construction. *Autom. Constr.* **2020**, *111*, 103063. [\[CrossRef\]](#)
41. Li, J.; Kassem, M.; Ciribini, A.; Bolpagni, M. A proposed approach integrating DLT, BIM, IOT and smart contracts: Demonstration using a simulated installation task. In *Proceedings of the International Conference on Smart Infrastructure and Construction 2019: Driving Data-Informed Decision-Making*, Cambridge, UK, 8–10 July 2019; pp. 275–282. [\[CrossRef\]](#)
42. Turk, Ž.; Klinc, R. Potentials of blockchain technology for construction management. *Procedia Eng.* **2017**, *196*, 638–645. [\[CrossRef\]](#)
43. Dounas, T.; Lombardi, D.; Jabi, W. Framework for decentralised architectural design BIM and Blockchain integration. *Int. J. Archit. Comput.* **2020**, 1478077120963376. [\[CrossRef\]](#)
44. Xue, F.; Lu, W. A semantic differential transaction approach to minimizing information redundancy for BIM and blockchain integration. *Autom. Constr.* **2020**, *118*, 103270. [\[CrossRef\]](#)
45. Sheng, D.; Ding, L.; Zhong, B.; Love, P.E.; Luo, H.; Chen, J. Construction quality information management with blockchains. *Autom. Constr.* **2020**, *120*, 103373. [\[CrossRef\]](#)
46. Zhong, B.; Wu, H.; Ding, L.; Luo, H.; Luo, Y.; Pan, X. Hyperledger fabric-based consortium blockchain for construction quality information management. *Front. Eng. Manag.* **2020**, *7*, 512–527. [\[CrossRef\]](#)
47. Elghaish, F.; Abrishami, S.; Hosseini, M.R. Integrated project delivery with blockchain: An automated financial system. *Autom. Constr.* **2020**, *114*, 103182. [\[CrossRef\]](#)
48. Chong, H.Y.; Diamantopoulos, A. Integrating advanced technologies to uphold security of payment: Data flow diagram. *Autom. Constr.* **2020**, *114*, 103158. [\[CrossRef\]](#)
49. ISO. *Organization and Digitization of Information about Buildings and Civil Engineering Works, Including Building Information Modelling (BIM)—Information Management Using Building Information Modelling—Part 1: Concepts and Principles*; ISO Standard No. 19650-1; International Organization for Standardization: Geneva, Switzerland, 2018.
50. Borrmann, A.; König, M.; Koch, C.; Beetz, J. *Building Information Modeling: Technology Foundations and Industry Practice*; Springer International Publishing: Cham, Switzerland, 2018.
51. Opitz, F.; Windisch, R.; Scherer, R.J. Integration of Document- and Model-based Building Information for Project Management Support. *Procedia Eng.* **2014**, *85*, 403–411. [\[CrossRef\]](#)
52. Fuchs, S.; Scherer, R.J. Multimodels—Instant nD-modeling using original data. *Autom. Constr.* **2016**, *75*, 22–32. [\[CrossRef\]](#)
53. Hoeber, H.; Alsem, D. Life-cycle information management using open-standard BIM. *Eng. Constr. Archit. Manag.* **2016**, *23*, 696–708. [\[CrossRef\]](#)
54. ISO. *Information Container for Linked Document Delivery: Exchange Specification—Part 1: Container*; ISO Standard No. 21597-1; International Organization for Standardization: Geneva, Switzerland, 2020.
55. Bunni, N.G. *The FIDIC Forms of Contract*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
56. Ye, X.; Sigalov, K.; König, M. Integrating BIM- and cost-included information container with Blockchain for construction automated payment using billing model and smart contracts. In *Proceedings of the International Symposium on Automation and Robotics in Construction*; Osumi, H., Furuya, H., Tateyama, K., Eds.; International Association for Automation and Robotics in Construction (IAARC): Kitakyushu, Japan, 2020; pp. 1388–1395.
57. Lee, G.; Cho, J.; Song, T.; Roh, H.; Jung, J.; Chung, J.; Yong, G.; Jeong, D. *Construction Field Management Using a Popular Text Messenger*; Lecture Notes in Civil Engineering; Springer: Cham, Switzerland, 2021; pp. 971–979.
58. DIN. *Common Data Environments (CDE) for BIM Projects—Function Sets and Open Data Exchange between Platforms of Different Vendors—Part 2: Open Data Exchange with Common Data Environments*; DIN SPEC 91391-2; German Institute for Standardization (DIN): Berlin, Germany, 2019.
59. Lockley, S.; Benghi, C.; Černý, M. Xbim. Essentials: A Library for Interoperable Building Information Applications. *J. Open Source Softw.* **2017**, *2*, 473. [\[CrossRef\]](#)