Journal of Building Engineering 25 (2019) 100832

Contents lists available at ScienceDirect

Journal of Building Engineering

journal homepage: www.elsevier.com/locate/jobe

Blockchain and the built environment: Potentials and limitations

Nawari O. Nawari*, Shriraam Ravindran

University of Florida, College of Design, Construction & Planning, School of Architecture, USA

ARTICLE INFO

Keywords:
Blockchain
BIM
Distributed ledger
Hyperledger fabric
Smart contract
Code-checking

ABSTRACT

Blockchain is a relatively new concept that originated from the first cryptocurrency known as Bitcoin and was soon noted to have a much wider range of applications than just serving as the platform for digital cryptocurrency. A blockchain (BC) is essentially a decentralized ledger that records every transaction made in the network, known as a 'block', the body of which comprises encrypted data of the entire transaction history. The implementation of decentralized technology in any industry would require augmented security, enforce accountability, and could potentially accelerate a shift in workflow dynamics from the current hierarchical structure to a decentralized, cooperative chain of command and affect a cultural and societal change by encouraging trust and transparency. This paper presents an evaluation survey of blockchain technology and its applications in the built environment and examines the potential integration with the BIM process. Moreover, the study explores how employing distributed ledger technology (DLT) could be advantageous in the BIM working environment by reinforcing network security, providing more reliable data storage and management of permissions, and ensuring change tracing and data ownership. The study discusses the basic fundamentals of distributed ledgers, their potential future applications and current advances, and their classification based on inherent characteristics of consensus reaching and permission management. Furthermore, the paper evaluates the potential application of BC technologies in enhancing the framework for automating the construction design review process such as smart contract technologies and Hyperledger Fabric, as well as discussing the pros, cons, and possible future research directions.

1. Introduction

The blockchain is a digitized, decentralized public ledger of data, assets and all pertinent transactions that have been executed and shared among participants in the network. While it is most associated with digital cryptocurrencies such as Bitcoin, blockchain is viewed as an emergent technology that could potentially revolutionize and transform the current digital operational landscapes and business practices of finance, computing, government services, and virtually every existent industry [1]. The chief hypothesis behind blockchain is the creation of a digital distributed consensus, ensuring that data is decentralized among several nodes that hold identical information and that no single actor holds the complete authority of the network. This enables transparency of activity and enhancement of data security. Fig. 1 depicts the general schema of blockchain technology. While initially developed solely for financial transactions with an aim to create a system that enables secure data transfer between two parties without the requirement of an intermediary, the tremendous disruptive potential of blockchain was later evident with the exponentially increasing development of various cryptocurrencies in recent years. By placing emphasis on trust and cooperation between participants, blockchain radically reorganizes existing workflow paths in any organization in which it is implemented, bringing with it a plethora of benefits that include shared learning, instantaneous data exchange, automated contract execution, network security, and improved collaboration.

Fig. 1 illustrates that BC is composed of a linked list of blocks of transactions. Each block within the BC is recognized by a hash, generated using the SHA256 cryptographic hash algorithm on the header of the block. Also, each block references a previous block, through the "previous block hash" field in the block header. In this fashion, each block comprises the hash of its previous block inside its own header. The sequence of hashes linking each block to its predecessor generates a chain going back all the way to the first block ever created, known as the *genesis* block. All the transactions are stored in a schema referred to as the *Merkle Tree*. Such a tree save transaction at leaf nodes, and inner nodes encompass the combined hashes of their immediate subtrees (see Fig. 1). The advancement of BC technology in the finance sector has led to a surge of investors and implementation of newer BC applications and more technological innovations such as automatically executable contracts, which is known as *Smart Contracts*.

[*] Corresponding author.
E-mail addresses: nnawari@ufl.edu (N.O. Nawari), shr1raam@ufl.edu (S. Ravindran).

https://doi.org/10.1016/j.jobe.2019.100832
Received 13 October 2018; Received in revised form 10 June 2019; Accepted 11 June 2019
Available online 14 June 2019
2352-7102/ © 2019 Published by Elsevier Ltd.

Acronyms

ACCC	Automated Code Checking and Compliance
BC	Blockchain
BIM	Building Information Modeling
DLT	Distributed Ledger Technology
DAO	Decentralized Autonomous Organizations
AEC	Architecture, Engineering, and Construction
LOD	Levels of Development
PoS	Proof of Stake
PoW	Proof of Work
BG	Byzantine Generals
PBFT	Practical Byzantine Fault Tolerance
DPOS	Delegated Proof of Stake
MEP	Mechanical, Electrical, and Plumbing
IPD	Integrated Project Delivery
CDE	Common Data Environment
IFC	Industry Foundation Classes
XML	Extensible Markup Language
API	Application Programming Interface
SDK	Software Development Kit
MSP	Membership Service Provider

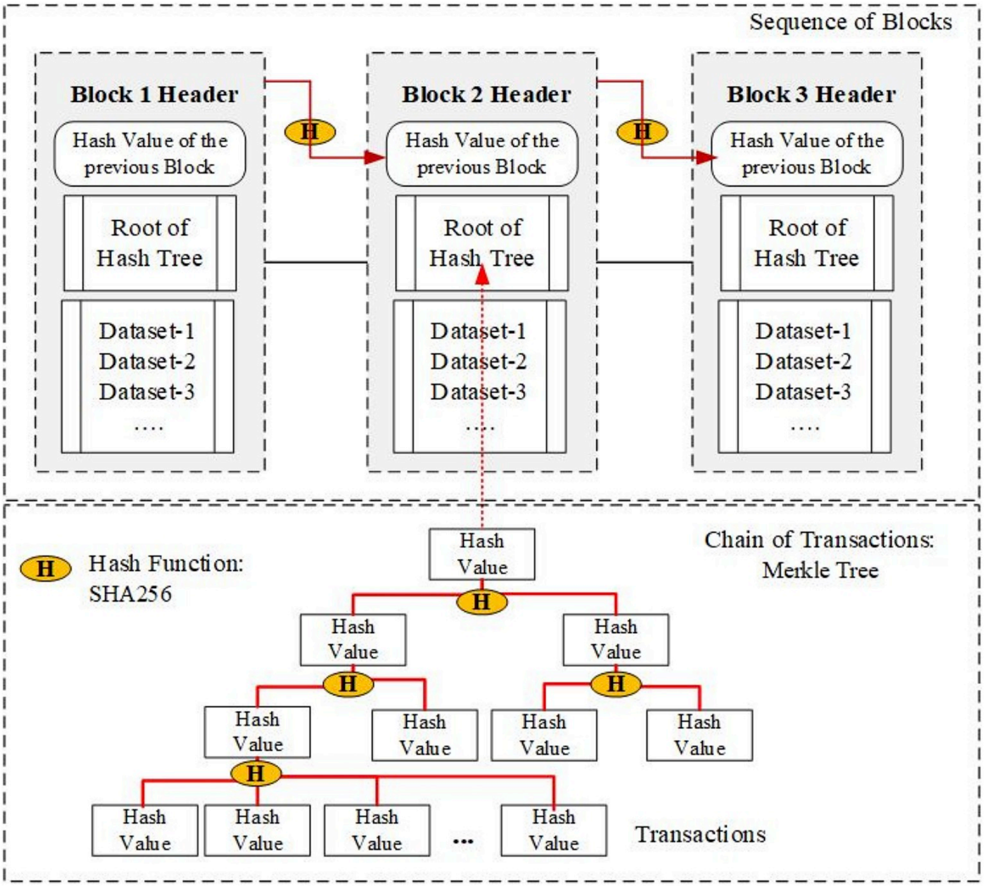

Fig. 1. Overview of the concept of blockchain technology.

2. Statement of purpose

This survey study aims to explore the potential applications of blockchain (BC) in the Architecture, Engineering, and Construction (AEC) industry focusing on Building Information Modeling (BIM) workflow while highlighting its limitations. The advent of BIM software platforms in recent years has been the most important development in the digital transformation of the AEC industry. Its framework and structure center around a collaborative approach to solving problems, monitoring errors, and coordinating tasks between multidisciplinary teams. However, BIM workflow is based on a centralized database (particularly cloud-based) which is often associated with security and accessibility issues (malware injection, online cyber theft, wrapping attach, …etc.) and the risk of losing the quality and integrity of the data transactions [2]. The implementation of BC technology and integrating BC in the BIM environment could solve several of these existing problems as well as offer new areas of applicability. The paper aims to explore the potential applications of BC in BIM processes as well as assess its current development levels and limitations.

3. Objectives

The major objectives of this paper include: (a) to conduct an evaluation survey of BC technology and its applications in general and its relationship to the built environment; (b) to examine the potential integration of BC technology with BIM processes in regard to network security, data storage and management of permissions, and data ownership; (c), to evaluate the potential application of BC technology in enhancing the framework for automating the construction design review process.

4. Methodology

The study is based upon a systematic review of the current body of knowledge, and the retrieval of the required the information from the literature sources. The approach consists of four phases. Fig. 2 depicts

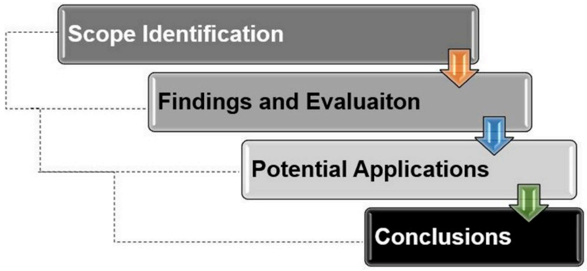

Fig. 2. Overview of the research approach.

the overall method adopted in this research. The first stage focuses on the scope definition and the general survey background on the current state of BC technology and its applications. The second phase determines and filters the literature review relevant to the aim of the study. This includes, classifying the publications according to the main categories: BC related only, BC and BIM, Hyperledger Fabric, and Smart Contracts (Fig. 3). Thirdly, the collected literature is evaluated, and findings are formulated and presented. Finally, conclusions about the potential applications of the emerging BC technology in the AEC industry are presented (see 2).

4.1. Scope identification

The scope identification centers around collecting evidence through extensive literature review. The literature review provides definitions, background, historical development, current knowledge areas, and ongoing research efforts on each relevant area identified. Via Google Scholar and library databases, the relevant papers are identified by keywords. A closer look into the papers' abstracts determined the relevancy of the papers. A preliminary study of research papers suggests that cybersecurity, interoperability, collaboration, and smart contracts are the main areas of emphasis. Extensive literature is collected that address these areas with respect to both BIM and BC. The study suggests that BC, BIM and BC, cybersecurity, and smart contracts are the chief areas defining the scope of this research. Extensive literature is collected that address these areas and are summarized in Tables 1 and 2.

While plenty of publications covering the features, operational mechanisms, and research trends of BC technology, very few research papers are available that focus on areas of BC overlapping with application in the AEC industry. Even fewer papers deal specifically with the application of BC in BIM processes since both areas are relatively new in conception and usage. However, the number of publications addressing these areas are encouraging, as many papers published in the last three years propose the implementation of BC, specifically its feature of smart contracts, to achieve different solutions to the BIM workflow.

5. Findings and evaluations

5.1. History of blockchain

A blockchain is a distributed public ledger that records all transactions or digital events that are executed and shared among the participants in the network [1]. Blockchain was first used in Bitcoin, the first cryptocurrency to achieve widespread circulation and mainstream appeal. Bitcoin was first conceptualized in a whitepaper published in 2008 [8], outlining details required for a protocol to establish a decentralized digital currency that operates on a secure and pseudonymous network of participants. A single 'bitcoin' is essentially a tradeable asset produced by the BC system as a form of payment for its operation and maintenance by miners. Following the success of Bitcoin, the term 'blockchain' was no longer synonymous with digital cryptocurrencies as its vast area of potential applications was evident.

Blockchain 2.0 is an umbrella term denoting these new developments and widening the uses of DLT beyond cryptocurrency. The concept of smart contracts came to realization in 2015, with the release of the second public blockchain called Ethereum by Vitalik Buterin [28]. Created as an alternative to Bitcoin, Ethereum was a breakthrough in that it was a general purpose BC that provides users a flexible, trustless, general platform that can run smart contracts, and develop any conceivable application. Ethereum also features an active developer community involved in building distributed applications (DApps). The radical restructuring of chain of command in the workflow by emphasizing trust and cooperation was described by Buterin [28] as 'Decentralized Autonomous Organizations.' (DAOs)

The next major innovation was the introduction of the 'Proof of Stake' (PoS) method for reaching consensus, which presents several advantages and savings over the current 'Proof of Work' (PoW) method. The PoW is a mechanism that determines the node that writes a block on ledgers using a combination of game theory, cryptography, and incentive engineering [7]. On the other hand, in the PoS, the creator of the block is chosen in a deterministic method, depending on the stake held by the participants.

Currently, every node in the BC processes every transaction. Blockchain scaling is a cutting-edge example of blockchain thinking that can circumvent this process and improve computational speed, while not compromising on security and robustness of the network. This is done by determining the number of necessary computers to validate each transaction and accordingly discretizing the work efficiently. While it is a difficult and ambitious solution to the problem, it is possible to achieve and is viewed as the next great innovation in BC technology. Blockchain 3.0 aims to improve on the capabilities of current blockchain BC with terms to transaction time, scalability, and ease of implementation.

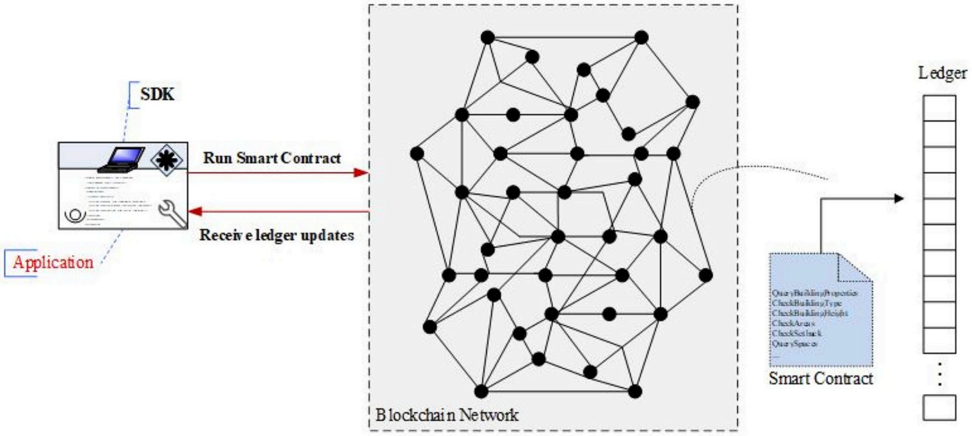

Fig. 3. Overview of blockchain infrastructure.

Table 1
Summary of the scope identification.

Reference Title	Blockchain (BC)	BIM + BC	Cybersecurity	Smart Contracts
[1,3–7]	Concept, functionality, present implementation, mining, transaction protocols		Anti-counterfeiting, privatization of blockchain, selfish mining, risks, blockchain architecture, privatization of blockchain	Framework for DApps, potential business applications
[8]	Bitcoin: concept, mechanism			
Belle, et al., 2017, [9]	Blockchain applications, legal issues with blockchain, barriers in adoption	Intellectual property rights, BIM advantages, BIM workflow scenarios		
[10]; Mason et al., 2017, Hammi, et al., 2017, Mahamadu et al., 2017, [11]	Advantages of using permissioned blockchain, Oracles	Collaboration and trust, data ownership, advantages	Security in transactions, liability protection, data ownership, legal implications	Legal risks in smart contracts
[12]; Seijas, et al., 2016, [13–15]			Security threats in the consensus mechanism	Background, definitions, design landscape, research directions, scripting languages, automated generation
Atkins, et al. [16,17]; Wang et al., 2018, [18–21]	Key concepts and challenges	Adoption levels, benefits, planning, LOD, risks and considerations, emergent DLT framework, shared economy	Professional liability insurance, threats, and advantages	Potential applications, compliance, and enforcement
[22–25]; Androulaki,et al., 2018	Software connector scenarios, DAOs, off-chain data, coordination, Research opportunities		Additional security levels from software connectors, flaws in Hyperledger	Role as blockchain connector and coordination of software, business applications, EVM, Oracles, Hyperledger features, architecture
[26,27]	Scalability, latency, sidechains		Sidechain issues with inter-chain transactions	Sidechain concept, mechanisms
[28]	Current research levels in blockchains		Permission management	Ethereum concept, architecture

5.2. Key features of blockchain architecture

5.2.1. Structure of blockchain

A blockchain is a chain of sequentially arranged *blocks* of discretized, encrypted data from validated transactions. A block consists of a header and block body. The block that precedes the current block is called a *parent block* [7]. The first block of the BC that does not have a parent block is called a *genesis block*). The *Block header* contains. *Block version, Timestamp, Merkle tree root hash, nBits (a* hash equivalent of all transactions), *Nonce (*a 4-byte field that usually starts with zero and increases for every hash calculation); and *Parent block.* The block body contains the actual transaction data. The block size dictates the upper limit for a number of possible transactions and transaction size.

5.2.2. Digital infrastructure

The infrastructure of the BC network can be divided into two layers of code: [29]

- *Fabric layer:* consists of the actual BC code base, communication protocols, public key infrastructure, data structures for database maintenance, smart contract capabilities. Since the BC network is owned and controlled by the developers, the fabric layer cannot be tampered with;
- *Application layer:* contains application logic of smart contracts. It is collectively controlled by the participants who deploy the code onto the BC network when it is operational. Any participant that holds access and control of the deployed code can write the application layer.

5.2.3. Distributed ledgers

Instead of a single central point that is tasked with the monitoring and control of information and operational authority as in the case of a centralized ledger network, a DLT network spreads the computational workload across multiple nodes in a network that can make independent decisions. DLT often incorporate a decentralized consensus mechanism, where all validating nodes in the system run the same agreed upon consensus algorithm [3]. This is a crucial characteristic of BC technology which eliminates the requirement of an intermediary party to validate transactions. Fig. 4 depicts the organizational structure of centralized, distributed, and decentralized ledger networks.

A blockchain consists of discretized 'blocks' of data that is encrypted and arranged in sequence. Each block involves a header and the actual block data. The header contains merely the block number and a reference to the previous block, while the remaining content of the block consists of transaction data. Discretized distribution of data as such enables a decentralized hierarchy, allocating only a portion of ownership or liability to a single party in the network.

BC, being a peer-to-peer system consists of nodes that create connections, keep existing connections active, and disseminate encrypted information while ensuring that every node that receives information forwards it to all its peers. This solves the problem of system failure or shutdown.

BC has several components that involve ownership such as identification, authentication, and authorization to allow access and control over personal data. This is crucial since modifications and exchange of data can be recorded and monitored in a convenient manner. The BC identifies the owner by using hashes.

Hashes are IDs that a BC uses to achieve ownership, that is created by an algorithm that produces a short string from an extensive data set adhering to a set of mathematical rules. A hash is unique, and hence some of their critical intrinsic properties are that they are deterministic, collision-resistant, and pseudorandom. It is crucial that the hashes are irreversible one-way functions.

Table 2
Summary of the literature related to the key concepts.

Key Concepts of the Investigation					
No	Categories	BC	BIM + BC	Cybersecurity	Smart Contracts
1	No. of papers	32	10	19	18
2	Current knowledge	Yes	Yes	Yes	Yes
3	Potential applications	Ongoing	Very limited literature	Yes	Ongoing
4	Limitations	Yes	Very limited literature	Yes	Ongoing

Table 3
Comparison of the different types of permission management in blockchain technology.

Blockchain Privacy	Permissioned Blockchain	Permission-less Blockchain
Privacy	Only permissioned actors have access to the blocks of transactions.	Any participant can read access all blocks of data. This is convenient for a shared database.
Scalability	Permissioned BC can build a simplified PoS model to establish consensus by burning computational cycles.	Uses PoW mechanism to reach consensus.
Fine-Grained Control	Allows fine-grained control by restricting access to few participants.	Not possible to ensure fine-grained control.

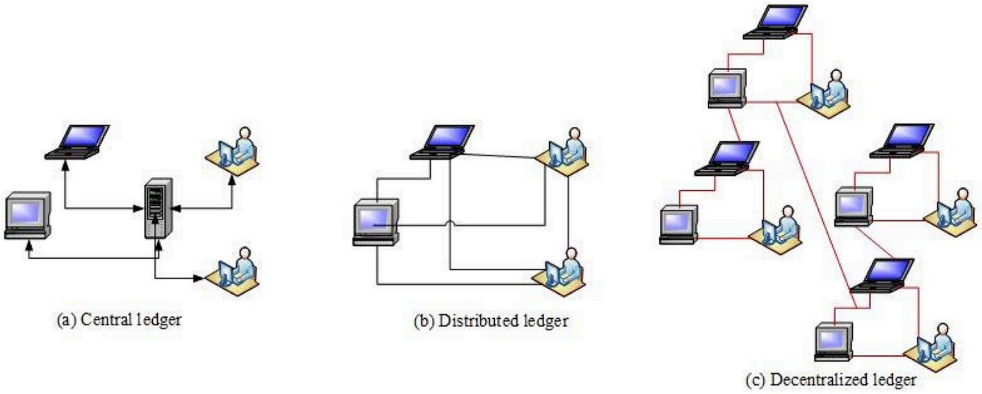

Fig. 4. Types of ledger networks.

5.2.4. Consensus

Members of the network must prove themselves as legitimate members. Thus, reaching a consensus agreement is one of the key features of a distributed technology [3]. A consensus between all participants in the BC network is agreed on prior to the implementation of the BC, and this ensures that the ledger is shared, unchangeable, and immutable throughout its life.

After agreement on the consensus mechanism, the peers execute the consensus protocol to validate transactions, create blocks and hash chains. The ledger is updated and appended in the event of the occurrence of errors, instead of overwriting them. New transactions recorded on the ledger are validated by miners. A block is mined every few minutes. The Byzantine Generals (BG) problem is central to determining consensus in a BC, and all consensus mechanisms are developed with the aim to overcome this issue. The Byzantine General Problem was a security flaw in distributed systems developed before Bitcoin, in which the nodes aim to reach a consensus despite having a faulty component [24]. This increases the possibility of malicious intent or network irregularities. The different mechanisms through which consensus is reached are [7] :

- *Proof of Work:* 'Mining' or the Proof of Work (PoW) mechanism works by determining the node that writes a block on ledgers using a combination of game theory, cryptography, and incentive engineering. The nodes in the network compete to solve a mathematical puzzle (generally a computationally difficult but easily verifiable pattern) to record a transaction. Upon resolving the paradox, consensus is reached by broadcasting the resolved solution to other nodes in the network, thereby ensuring transparency, robustness, and incorruptibility of the network. Consequently, the group with larger total computing power dictates the decision-making and reaching consensus. The two most popular BC systems, Bitcoin and Ethereum operate on PoW mechanism. However, this involves expensive transaction fees, extensive computing tower, and cumbersome mining processes to create new blocks;
- *Proof of Stake:* The creator of the block is chosen in a deterministic method, depending on the stake held by the participant. An algorithm is employed to determine collective decision-making and level of privacy between participants. This mechanism requires the credibility of data, which is denoted by proof of ownership of cryptocurrency coins. If a created block can be validated, the cryptocurrency will be returned to original node as a bonus. This method involves no block rewards but operates solely on transaction fees. It is thus an energy-saving alternative to PoW and presents several economic benefits. Ethereum aims to shift the paradigm by transitioning to a PoS mechanism;
- *Practical Byzantine Fault Tolerance (PBFT):* This algorithm aims to correctly reach enough consensus despite malicious nodes in the system failing or sending out incorrect information. This is a Byzantine agreement consensus method that can tolerate a maximum of 1/3 malicious byzantine replicas. A primary is selected in each round and is responsible for ordering the transaction. The Hyperledger Fabric uses the PBFT algorithm;
- *Delegated Proof of Stake (DPOS):* Stakeholders elect representatives

to validate blocks. Since this mechanism features a relatively small number of nodes, the processing of transactions is quicker. The delegates are authorized to modify the network parameters.

5.2.5. Mining

Mining is the mandatory process of recording transactions on the BC network using the computer's processing power. The subset of nodes in the network that are equipped with special software that validate the transaction operations to be added on the BC is called *miners* [17].

5.2.6. Privacy

Blockchain can address accessibility and visibility of the data securely and efficiently, since the ledger is distributed [3]. It facilitates setting different levels of privacy as every participant is essentially a stakeholder and no single participant has full administrative privilege. Thus, formulating and enforcing the consensus is crucial to the BC operation, with terms to data updates, error-checking and collective decision-making. The selection of which BC to uses depends much on the method of agreement to reach consensus.

Based on privacy, blockchains can be classified as:

- *Permissioned blockchains:* Permissioned BCs restrict the actors that can contribute to the consensus of the system state. Only a restricted set of users have the rights to validate transactions and may also limit access to approved actors who can create smart contracts. Hyperledger Fabric is an example of this permission type.
- *Permission-less or public blockchains*: Blockchains that are permissionless allow any participant creating consensus, as well as smart contracts, and uses the PoW mechanism to reach a consensus. They typically use a native cryptocurrency or none to validate transactions. Bitcoin and Ethereum blockchains are good examples of this type of permission.

The key characteristics of a blockchain network can be summarized in the following: 1) Sequential recording of data transactions, 2) Immutability of data records, 3) Data redundancy, 4) Cryptographic encryption of data, 5) Reaching a consensus mechanism to ensure privacy operations, 6) Decentralization, and 7) Lack of intermediary party to oversee transactions.

5.3. Blockchain data processing example

The blockchain often uses the SHA – 256 algorithm. The SHA – 256 algorithm generates a unique, fixed – size 256 – bit hash. A hash is like a secret code that uses an encryption method that hides data in a way that makes it almost impossible to decrypt without authorization. The hash generated is always the same length. It doesn't matter whether you put one word or an entire book, you will still get the same length for any amount of data entered. If you change one of the letters, then the hash will completely change. The hash appears to be random with no connection to the data entered. It is almost impossible to figure out the original message from the hash unless you know the original message or have a private key. Some examples of hashes generated from different words and phrases are below:

The hash of the term “Building Information Modeling” with upper case B, I and M are: 81737C1EE7C8E875F4BBC397C47CA4CB38A3B4DCD1C376A2861942BEA8B6B538

The hash of the word “building information modeling” with lower cases only is: 987BED3FAD66648E4214EE703523BFB2448E700240B3FDBA9BD689E36F09C063

The hash of the word “BIM” is 4F432A298898B3855F43F28EF616847AE442F1FD10C09C3321A8F4A6B6250877

This is the entire Autodesk Revit Model (ProjectExample.rvt) containing data about a duplex building is: e807d23c1ff8e4ba4aa4542-d35082e28f9f580407ca6031a34bc1eff424fd37a

From the examples above, one can see that it is impossible to tell the input data from the hash generated. It's also clear that a small change, such as changing a letter from uppercase to lower case or adding a space will significantly change the hash generated. Hashing transactions into blocks. The previous examples were to show how the hash generated has no detectable pattern to the length of text or type of data entered.

The examples above don't contain transactions. Thus, the next step is to use transactions and convert them into hashes. After the hashes are generated, they will be linked in a blockchain. The first block in the blockchain is usually known as block 0, also known as the genesis block. This example delineates the concept of generating BC transaction system. Each group of data exchange is converted into a hash, combined with the hash of the previous block and a number known as nonce. The hash is included in the header of the next block linking each new block to the block before it. One can track the transactions from the current block, all the way back to the very first block to understanding what has occurred on the BC.

5.4. Security risks to Blockchain

Blockchain is still susceptible to certain forms of security problems. While it is difficult to compromise a blockchain system, it is important to note that the system is still not completely infallible. Some of the threats to blockchain applications are:

- Double-spending: Two parallel transactions transfer the same data to different recipients, thus creating a new, invalid transaction. This can be prevented by enforcing a PoW consensus, where all participants agree over the order of transactions that have taken place;
- 51% attacks: When a participant controls over 51% of the network, that participant does have a high chance of tampering with the blockchain without any consequences, since he or she controls the majority of the network and thereby gaining more power in dictating the consensus. Thus, smaller systems are more susceptible to attacks since a single participant can gain relatively more control in the blockchain's early stages [30];
- *Inadvertent centralization:* Since security is partly dictated by the participants in the blockchain, it is impossible to stop the weakest participants from transferring assets to a centralized system of exchange. These often occur due to a third-party that has amassed large amounts of assets and is storing them on behalf of the users. Besides, assets in pure blockchains can also be centralized. An example is the cryptocurrency Ripple, large amounts of which are owned by a small number of founders and large multinational organizations.
- *Lack of privacy due to pseudonymity of user: It* is crucial to choose which information is to be made private and which can be accessible to the public. It is also impossible to achieve complete privacy since one can infer the data to a particular user by studying their transaction patterns.
- *Data malleability:* The integrity of digital signatures used to validate transactions cannot be guaranteed. In such a scenario, the hacker would intercept a transaction, modify it, and broadcast it to the network [30].

5.5. Smart contracts

Smart contracts are contracts programmed with the blockchain that automatically executes upon the fulfillment of certain conditions. This removes the requirement of a third-party intermediary for overseeing the transaction in real-time. They are an extension of the BC that can independently enforce rules without requiring manual intervention. Fig. 5 illustrates the concept of smart contract in a blockchain application.

The following is a definition of the concept of Smart Contracts [17]:

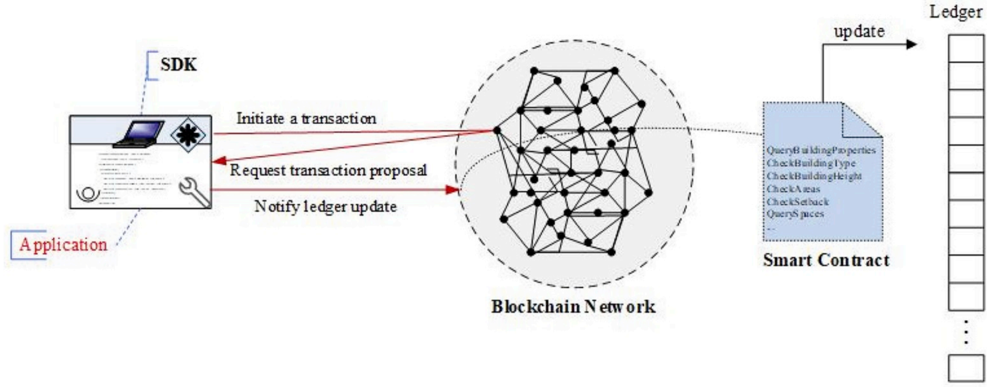

Fig. 5. The concept of a smart contract in a blockchain network.

> "Smart contracts are digital contracts allowing terms contingent on the decentralized consensus that is self-enforcing and tamper-proof through automated execution."

The introduction of smart contracts in blockchains has opened many new possibilities such as complete lifecycle management of legal contracts, automated execution of contracts, and personalization of customizable contracts. In the literature, two different definitions for the term 'smart contracts' are given since the term is used interchangeably for the written code and the binding contracts [12]:

- *Smart contracts codes:* They are software agents fulfilling pre-set obligations and exercising certain rights and may take control of certain assets within a shared leger;
- *Smart legal contracts:* This term focuses on the expression and implementation of the software and encompasses operational aspects and issues pertaining to the composition and interpretation of the contract.

A high-level definition that combines both aspects of the smart contract and based on automation and enforceability is given as [12]:

> "A smart contract is an automatable and enforceable agreement, automatable by a computer, although some parts may require human input and control, enforceable either by legal enforcement of rights and obligations or via tamper-proof execution of computer code."

5.5.1. *Key characteristic features of smart contracts*

A list of the main characteristics of a smart contract generally include:

- *Automation:* Automation is accomplished by linking the legal prose to the smart contract code via parameters that generate instructions regarding the final operational details;
- *Enforceability:* The smart contract code must execute successfully, accurately, and within a reasonable timeframe. A smart legal contract must include legally enforceable obligations and rights that are expressed in complex, time-dependent, sequential, context-sensitive prose. These may also include overriding commitments based on the fulfillment of certain conditions;

In reference to the AEC industry, the application of Smart Contracts can be limited to specific transactions. To achieve wider application of smart contracts, the following issues need to be addressed:

(a) The inherent security vulnerabilities of the inputs to any 'smart' processing of evidence of fulfillment;
(b) to be able to script the contract in a machine-readable fashion that can be reviewed and verified by experts
(c) to employ an upper ontology and reference data model that explicitly and unambiguously define terms, conditions relationships, etc. and can be machine processed.
(d) transaction processing needs to be atomic, i.e., transactions are indivisible and irreducible.

5.5.2. *Scripting smart contracts using DLT*

A script is a bytecode stack-based language that is designed in a manner that its execution is guaranteed to terminate (Seijas et al., 2016). It linearly executes the sequentially coded instructions without backward jumps.

DLT scripting languages can be classified into four categories (Seijas et al., 2016):

- *Turing-completeness*: Scripting languages may be Turing-incomplete as in the case of Bitcoin, or Turing-complete like the Ethereum script. While the Turing-complete scripts can account for looping and recursion, they lose completeness in practice because of runtime bounds placed on execution parameters such as execution time, stack size, etc.;
- *The degree of complexity*: Certain virtual machines feature a low-level computation model that facilitate compiling a higher-level language like Ethereum's Solidity;
- *Finite State Machines:* DLT transactions could be derived by transforming a language such as BPM (Business Process Modelling) languages or finite state machines. Finite state machines are mathematical abstract models of computations that can exist in exactly one of a finite number of states at any given time;
- *Virtualization technology*: These can isolate computations, as is done in Hyperledger with Docker containers.

The emergence of DLT has introduced a secure, reliable, and decentralized methodology for data storage and exchange, which is particularly useful for the execution of smart contracts. Apart from serving as general purpose programming languages, scripting languages also include the provision of randomness, management of anonymity, monitoring transactions, and assigning incentives, among other useful features that can be automated. Although all computations are deterministic in a blockchain, there is always a certain amount of nondeterminism that occurs due to competition between the transactions themselves (Sergey et al., 2017).

5.6. *Permissioned vs permission-less blockchain for BIM*

The performance of a public blockchain is limited and can only process 3–20 transactions per second. The average transaction processing rate is 1.7 transactions per block and the average mining time is 17 s [24]. Public blockchains do not guarantee data privacy as well. As discussed earlier, it is crucial to decide a method of reaching consensus in the BC since it dictates all future mechanisms, functionality, and managing of permissions through the life-cycle of the project. Both permissioned and permission-less BCs have their respective advantages and drawbacks, and the BC for use must be chosen appropriately based on the desired functionality and level of privacy needed. With respect to BIM workflow, there are generally several different parties working simul on one model. In such cases, implementing a permission-less BC could bring positive effects such as improved communication, transparency of work, and presenting opportunities for collaborative design processes. However, the current socio-economic environment of the AEC industry may necessitate tighter management of permissions due to concerns like data theft, conflicting interests, misuse of information, and others arising from the number of third-parties involved in a typical construction project which usually entails high levels of copyrights, budgets and accountability to government bodies and regulatory entities. Hence, employing a permissioned BC is a far more realistic option for most BIM projects.

5.7. Hyperledger Fabric

Hyperledger Fabric is a platform for generating distributed ledger blockchain systems, supported by a modular design, offering an elastic and extensible digital framework, that delivers high levels of confidentiality, and scalability. It is designed to support pluggable implementations of different components and accommodate the complexity and details that exist across the economic ecosystem. The Hyperledger blockchain aims to be a general purpose, enterprise-grade, open-source DLT that features permission management, pluggability, enhanced confidentiality, and consensus mechanism and is developed through a collaborative effort. Hyperledger Fabric is one of the BC projects within Hyperledger. Like other BC technologies, it has a ledger, uses smart contracts, and is a system by which participants manage their transactions.

Hyperledger was founded by the Linux Foundation in 2015 to advance cross-industry BC technologies. It was the first blockchain developed that enabled the development of distributed applications written in standard general-purpose programming languages (Andreoulakis et al., 2018). Presently, the Hyperledger consortium involves IBM, the Linux Foundation, and other organizations that contribute to the BC development and related apps (Seijas et al., 2016). The open source nature of the BC is augmented by the lack of necessity of mining cryptocurrency or expensive computations to validate transactions. The Hyperledger Fabric was the first blockchain developed that enabled the development of distributed applications written in standard general-purpose programming languages [31].

The fundamental differences between Hyperledger Fabric and other BC systems is that it is private and requires permissions. In contrast to an open permission-less system that allows unknown identities to participate in the network (necessitating rules like PoW to authenticate transactions and secure the network), the nodes (members) of a Hyperledger Fabric network join through a trusted Membership Service Provider (MSP). Furthermore, Hyperledger Fabric offers several pluggable options such as ledger data can be stored in multiple formats, consensus mechanisms can be exchanged in and out, and diverse MSPs are supported.

Moreover, Hyperledger Fabric has the ability to create channels, allowing a subgroup of participants in the network to establish a separate ledger of transactions. This is an especially important option for BIM workflow where subcontractors can exchange data within the only subgroup of the network. For example, the structural engineer of record of the project can exchange information with steel connection subcontractors only while still being part of the Hyperledger Fabric network and sharing that transaction with the rest of the nodes. The Hyperledger Fabric has several design components that provide the comprehensive, yet customizable, enterprise BC technology:

- *Assets*: Assets can range from the physical objects (real estate and hardware) to the intangible (BIM models, contracts, and intellectual property). Hyperledger Fabric provides the ability to modify assets using chaincode transactions;
- *Assets*: Assets can range from the physical objects (real estate and hardware) to the intangible (BIM models, contracts, and intellectual property). Hyperledger Fabric provides the ability to modify assets using chaincode transactions;
- *Ledger*: It is comprised of a blockchain to save the immutable, sequenced records in blocks, as well as a state database to preserve the fabric state. There is generally one ledger per channel. Each node sustains a copy of the ledger for each channel of which a node is a member. The shared ledger encodes the entire transaction history for each channel and includes SQL-like query capabilities for efficient processing;
- *Privacy*: Channels enable multi-lateral data exchanges with the high degrees of privacy and confidentiality required by the AEC specific and other regulated industries that exchange data on a shared network. A ledger exists in the scope of a channel - it can be shared across the entire network (assuming every participant is operating on one common channel) - or it can be constrained to only contain a specific set of participants;
- *Security & Membership Services*: Permissioned membership provides a trusted BC network, where participants know that all transactions can be detected and traced by authorized regulators and auditors;
- *Consensus*: It is defined as the full-cycle of verification of the correctness of a set of transactions comprising a block in a distributed ledger system. In Hyperledger Fabric consensus covers the entire transaction flow, from proposal and endorsement to ordering, validation and commitment (see Fig. 6) Hyperledger Fabric has been designed to allow a new application to select a consensus mechanism that best characterizes the relationships that exist between participants in the network;
- *Smart Contracts*: Hyperledger Fabric smart contracts are written in chaincode and are invoked by an application external to the BC when that application needs to interact with the ledger. In most cases, chaincode interacts only with the database component of the ledger, the world state (querying it, for example), and not the transaction log. Chaincode can be implemented in several programming languages. The currently supported chaincode language is Go with support for Java and other languages coming in future releases.

6. Applications of blockchain technology

6.1. Blockchain technology in different sectors

6.1.1. Business and supply chain

Implementing BC technology Implementing BC technology can streamline business transactions by obtaining multiple approvals at once with minimal required supervision, as opposed to the cumbersome process of sequential verification.

6.1.2. Finance

Accountants can manage client accounts in real time upon adoption of BC technology owing to its immutable record-keeping. Thus, one can access the entire history of transactions and not just the most recent activity. Further, adoption BC could be instrumental in accelerating the shift from monthly cycles to instantaneous transactions and accountants [3].

6.1.3. Insurance

Insurance companies can use BC to automate insurance claims, verification of qualifying criteria and execution of privileges. This would be advantageous in saving cost by substantially minimizing the workload by insurance agents who are tasked with manually

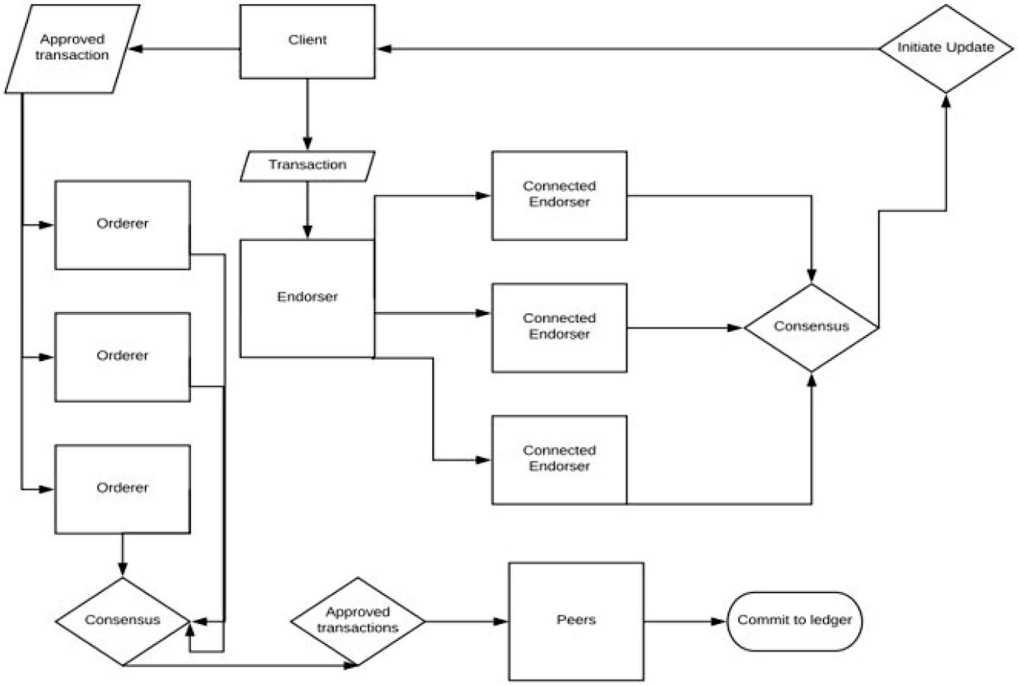

Fig. 6. An illustration of consensus in Hyperledger Fabric.

scrutinizing and cross-referencing insurance claims with factual data, and thus also eliminating fraud and counterfeit claims [3]. BC would also benefit clients by ensuring efficient payouts through instantaneous transactions and providing comparisons between policies for better-informed decision-making [32].

6.1.4. *Government organizations and services*

BC can have broad applications in the public sector in areas such as voting and identification, land registration and monitoring real estate, management of medical records and enabling transparency with non-profit organizations [32].

6.1.5. *Decentralized autonomous organization (DAO)*

DAOs are entire sets of long-term smart contracts consisting of venture capital funds that run on the Ethereum blockchain, and manage decision-making, recordkeeping, and business operations in an organization. DAOs can significantly reduce the requirement of management-level staff (Belle, 2017).

6.1.6. *Oracle*

Oracles or validation oracles are agents that import external states from external sources to evaluate conditions not expressible within blockchains (Xu et al., 2016). Oracles can exist independent of a BC and in its absence, a human arbitrator can periodically inject data to update the variables in a contract. Oracles are necessary because most BCs generally do not allow contracts to query externally sourced data. Oracles can act as an interface between smart contracts and the external data source. Hence, this involves trusting in a third-party organization to send the data [22].

6.2. *AEC industry*

In the following sections, the application of BC technology in the AEC industry is explored, navigating the advantages and potential applications.

6.2.1. *Barriers and obstacles*

While the digitization of many design and engineering processes in the AEC industry has seen advances in recent years, it should be noted that it has the slowest rate of digital transformation, just above agriculture and hunting (Belle et al., 2017). Changes are primarily impeded by the rigid nature of the construction sector, stemming from the current organization of teams and projects, featuring a defragmented collaborative process for a project. This is counterproductive in that, each party may aim to deliver the minimum work order merely, prioritize profit margins, and to minimize liabilities (Belle et al., 2017). Thus, the modern economy favors an adversarial environment where firms would be better incentivized by reducing information exchange between parties. However, it is clear that in the future, data exchange in networks can help to achieve significant savings in cost, time, error, and labor, as well as stimulate collaborative learning.

There is yet clearly inadequate education at the firm level and workforce to present an overall holistic picture of how updating current trends could lead to several advantages. Several organizations have not yet recognized the potential and significance of adopting BIM techniques and conclude that this would, in fact, complicate existing practices while disregarding the long-term benefit and overhead savings. Further, the collaborative design process in BIM could create difficulties in assigning responsibilities and liabilities due to the overlap of roles and responsibilities, ensuring intellectual property protection, risk allocation, privacy, and third-party reliance and software agents [9,33]. The focus of firms on return on investment, the complexity of projects, substantial initial capital, firm reputation, untrained personnel, legal considerations, and government restrictions are other factors that still impede digitization.

6.2.2. *Current advances*

The advent of Building Information Modeling (BIM) has facilitated a collaborative work environment with a single centralized model so that the structure can be analyzed and checked for compliance and can be rectified of errors in the design stage prior to actual construction. Adopting BIM presents a novel way of integrating all pertinent data into a shared digital model with geometric, temporal, financial and asset management dimensions. BIM software like Revit and ArchiCAD also include a plethora of plug-in tools that can simulate and assess several essential aspects of a building structure, such as operational energy analysis, life-cycle costs, occupancy behavior, connection design, interior climatization, building envelope, quantizing ecological footprint, etc. Thus, one can simulate scenarios and focus on one or more specific variables in real-time for every stage in the building life-cycle. This not only avoids unforeseen circumstances and errors in future stages, but dramatically saves time, manual labor and the need for excessive paperwork.

6.2.3. *Blockchain in construction management*

Blockchain technology, while still in its nascent developmental stages, has the potential to accelerate and streamline much of today's design and engineering practices with a multitude of benefit to the firm, individual, industry, clients, and society. The implementation of BC could lead to effective management and utilization of several tools that would drive efficiencies, transform industry culture, and advance futuristic advancements such as [10,46]:

- Building information modeling software for intelligent and collaborative 3D design and modeling;
- Cloud-based technology allowing for real-time creation and coordination of a visualized database and serving as a platform for multi-disciplinary collaboration;
- Smart contracts, a set of coded instructions that can automatically execute upon the fulfillment of certain conditions;
- Reality capture technology allowing verification and conversion of digital assets into real value;
- Managing the Internet of Things (IoT);
- Functionally permissioned BC that facilitates consensus-based collaboration.

BC technology can considerably slash administrative costs, effectively protect Intellectual Property Rights, and eliminate cumbersome paperwork, manual verifications, and contract execution. A potentially new stream of revenue for design professionals could be created by evaluating and selling designs and workflows. This, however, would also include addressing future problems that might arise, drop in quality standards, and continued liability [10].

Building a reputation is an important asset to any organization, which is difficult to quantify and compare. BC can facilitate creation of a registry consisting of past achievements and qualifications, with a view to enable comparison of team constellations and thus aiding in decision-making processes for clients and project managers to select a well-balanced team with varied skill sets, experience, and versatility [10].

6.2.4. *Blockchain and building information modeling (BIM) workflow*

This section particularly concentrates on the potential capabilities when BC technology is integrated with BIM processes. It explores various aspects of existing BIM workflow that could benefit from the implementation of BC technologies.

BIM is at the forefront of digital transformation in the AEC industry, encouraging collaboration and trust, and simplifying data exchange. McGraw-Hill reports that BIM adoption increased to 71% in 2012 from 49% in 2009 [16]. BIM models present a comprehensive design model of the building that can include all aspects of the structure like architecture, structural, and MEP design areas. Further, several built-in plugins in BIM platforms like Revit enable the simulation of external site conditions, geography, weather, as well as carry out energy analysis, building energy modeling, structural analysis, etc. In the future, BIM development will eventually aim to unify all design and analysis tools in one platform.

6.2.4.1. Collaboration. The replacement of a hierarchical organization with an inherently collaborative network could seem to be inevitable due to its effectiveness in isolating faults by tracing alternate paths around these areas. This is one of the key advantages of BIM, making it the current best solution for collaborative creation, data storage, and asset management. Stakeholders can reach a consensus by providing evidence of trust, such as collaboration centered on a shared BIM model, and integration of a distributed decentralized ledger based on BC technology. This cooperation culture change can address issues of unclear responsibilities and liabilities by prioritizing collaboration and sharing risks and rewards among the participants [10]. Integration of BC in BIM software addresses current issues like confidentiality, non-repudiation, traceability, change tracing, provenance tracking and data ownership. This ‘Evidence of Trust’ can be an effective and stable cornerstone for the collaborative design process and the Integrated Project Delivery (IPD) strategies.

Despite the increasing understanding of the significance of BIM, the rate of mainstream utilization and implementation has been relatively slow. The best stage for implementation is between the transaction processing component of the BIM server and its storage functionality [9]. The complete scope of possible applications from adopting BC are not yet fully realized, and it is by no means completely foolproof or at this point. Despite all the advantages, the written code is only as dependable as the accepted collaboration and agreed upon consensus. While varying levels of centralization and establishing a chained or unchained implementation of BC in BIM can yield certain benefits, Turk and Klinc [9] suggest that completely integrating BC in a BIM setting to record all transactions would be the ideal solution to leverage all potential advantages from using a DLT.

Choosing the appropriate type of BC based on privacy levels (public or private) is vital while implementing DLT in the construction sector (Li et al., 2016). This process depends on the suitable consensus mechanism that could be applied, and that would cover all operational necessities of all participants and stakeholders, since it forms the basis of establishing the BC.

A framework proposed in a study (Xu et al., 2018) to choose the appropriate BC depending on application requirements since current development levels of smart contracts necessitate reaching a compromise in a trade-off between scalability, permissions, computational power, and other characteristics. The process of choosing of BC is elaborated below:

Fabric Layer Design Considerations:

- *Scalability:* Larger block sizes will require off-chain data storage to draw on application logic or data, while smaller transactions can be transmitted as scalable metadata on the BC. BIM model data may be too large and might incur high latency rates, and hence using off-chain data is a better option;
- *Consensus mechanism:* PoS, Practical Byzantine Fault Tolerance or PoW may be used depending on the nature of the organization. In the AEC industry involving several parties involved in a joint project, a PoS mechanism can work since each project member has his design contributions to the model to offer as the stake. PBFT is also a suitable mechanism if a slightly decentralized approach is required, and the managers alone can be granted access to reaching consensus.

Application Layer Design Considerations:

- *Data storage:* On-chain data storage uses limited computational power, features limited data storage, and can verify computational data. This is suitable if hashed metadata is adequate for sharing. As discussed earlier, BIM model data might require greater computational power for addition in the BC and hence, off-chain storage may be necessary. This is also suitable if the information is sensitive in nature, and confidentiality is vital. Using off-chain data will also be relatively inexpensive since mining of blocks is reduced;
- *Privacy:* In the case of the AEC industry, a private BC like a permissioned or consortium BC is a better option since it features permission management and ensures data privacy;

- *Single or Multiple chain:* Using a single chain translates to better management of BC and permissions but data management is more difficult. Using multiple chains, on the other hand, can prove useful when information isolation is essential while compromising on chain and permission management. This decision depends on the nature and confidentiality level of the project and participants;
- *External or internal validation oracle:* BIM projects usually involve teams that assemble only for that particular project and hence a human arbitrator who can be tasked with periodically injecting external state into the BC. This may, however, feature incur high latency rates;
- *Permissioned vs. Permission-less blockchain:* Permission-less blockchains cannot preserve data privacy, and all data is visible to all participants on the network. Hence, for a legal contract platform, it is more appropriate to use a permissioned BC that allows developers to grant permission to the participants explicitly. Besides, the information on the BC may require encryption to preserve privacy. In this case, the key needed to be generated and stored off-chain. The BC does not have enough information that can be used by the components without permissions to access the sensitive data.

6.2.4.2. Data ownership. BIM can address the issue of conversion of intrinsic value to digital values by creating added value, coupled with BC's featured to provide reward mechanisms in the form of virtual currencies that hold validity long after the project completion. #AECoin is a recently developed cryptocurrency coin specifically created for design and engineering transactions [10]. It can be used to measure the added intrinsic and intangible value of a physical artifact and accurately calculate rewards earned by the participant by assessing individual/collaborative project contribution over the product life-cycle. This concept of monetizing designs through the life-cycle would ensure a superior outcome and can more effectively motivate engineers and architects to deliver their best efforts by providing proportionate incentives.

6.2.4.3. Cybersecurity in blockchain technology

6.2.4.3.1. Current security levels in BIM. Most industries currently rely on the “security through obscurity” approach to secure engineering, which emphasizes the confidentiality of the implementation and mechanisms of the cybersecurity system. Thus, a small leak of information could potentially endanger the entire network [34]. BIM offers a diverse multifunctional workspace which addresses asset management, performance monitoring, and change management through the life-cycle apart from overlooking the planning, design, and construction phases of the structure. To facilitate continuous collaboration among all parties, BIM makes use of Common Data Environment (CDE), which provides a single repository for project information that is used to collect, manage, and distribute data for multi-disciplinary teams. (IET Cyber Security Consortium Report, 2014). It requires auditing, monitoring, and tracking of data through the CDE, which will develop throughout the project life-cycle. Hence, it is vital to provide proper governance and curation to address information management and uphold data security, quality, and integrity. Since BIM involves complex interactions involving collaborative actions and information exchange between actors, technology, and processes and inter-relationships, it is crucial to consider cyber-security implications, assess current levels of reliability, address current drawbacks, and reinforce security. In contemporary usage, all BIM data is electronically shared across a shared data environment. It is essential for all project members to understand and abide by cybersecurity rules [35,45]. There are lesser trust issues among BIM actors compared to traditional information sharing. However, BIM's complicated collaborative framework creates security issues with terms to data leakage, information theft, and information protection while dealing.

Cybersecurity is “the collection of tools, policies, security concepts, security safeguards, guidelines, risk management approaches, actions, training, best practices, assurance and technologies that can be used to protect the cyber environment and organization and user's assets.” [20] The cyber environment includes the Internet, telecommunication networks, embedded processors, controllers, sensors, data storage and control devices, as well as the information, services, collaborative and business functions that only exist in cyberspace. Cybersecurity threats that can potentially affect BIM workflow and its connected systems could be classified into three categories:

- *External threat agents:* unconnected malicious outsiders, criminal entities attempting to access data for reconnaissance, hackers, intellectual property theft, leak of sensitive or confidential information, malware that can attack BIM database;
- *Internal threat agents*: involved participants who may bear malicious intent, abuse of authorized access to steal, leak, or corrupt information to disrupt BIM operations, human errors like omissions, ignorance, negligence of work;
- *Systems and business failures*: natural causes by extreme weather, interference from animals, storage device failures, poor maintenance of the centralized IT infrastructure, bankruptcies, and business failures.

6.2.4.3.2. Advantages of using blockchain for cybersecurity. The elimination of the requirement of a designated intermediary party to oversee transactions in the network offers several improvements over existent systems concerning cybersecurity levels.

Blockchain networks ensure that no single node in the network has complete access to all the information. Multi-signature (multisig) protection can add another layer of security by authorizing transactions by accepting more than one key. Hackers can gain complete access in the network only if more than 50% of the nodes are compromised [34].

Effective interoperability can be achieved by ensuring identifiability and authentication of participants and establishing a ubiquitous and secure infrastructure that serves as a repository for data storage, as well as a reliable platform that facilitates data exchange subject to required permissibility levels. Critical features of BC such as utilizing secure cryptography, asset sharing, auditing trails of data access and a resilient peer-to-peer network present a novel and promising emergent method to address cybersecurity and collaborative issues. However, it should be noted that the implementation of BC technology does not directly ensure infallibility. Further, BC can enable a secure and transparent method of Proof of Delivery (PoD) with terms to the transport and delivery of physical assets [4].

6.2.4.3.3. Compliance and trust. The immutable, instantaneous, and transparent nature of DLT places emphasis on compliance and trust between all involved parties on the network. The current economy that is inherently adversarial incentivizes the minimal exchange of information between parties involved in the completion of a project intending to protecting one's own interests. The advent of BC technologies is increasingly disruptive to the existing paradigm and will shift work culture in a direction that rewards collaboration and proves advantageous in the design process by facilitating better-informed decision making, collaborative learning, and easier debugging of errors. This change can be accelerated by the ability of BC technology to reward intrinsic value of any data through an #AECoin cryptocurrency [10]. BC can facilitate the creation of a registry comprising of past achievements and qualifications of individuals to enable comparison of team constellations and thus aiding in decision-making processes for clients and project managers to select a versatile and well-balanced team with diverse skill sets and experience.

The BC platforms can primarily serve as recordkeeping tools to record changes in the BIM model throughout the design and construction phases of the structure. Moreover, smart contracts can be programmed to automate awarding or revoking privileges based on the fulfillment of certain terms, as well as store an immutable record of all modifications in the BIM model data, along with other associated information [36]. The tamper-proof append-only nature of the records in a DLT also help to enforce compliance among workers. The transparency of the DLT coupled with the database properties of BIM [36] can introduce an '*evidence of trust'* [10] , which would create a new value proposition for the AEC industry.

6.3. Automation of code compliance checking (ACCC) USING DLT

Regulations are normative text prescribed by governing entities to enforce constraints to design and engineering processes and manufacturing based on existing conditions, and function as the defining text for laws, codes, specifications, standards, etc. Automating code-checking and compliance processes in the AEC industry would benefit the industry, saving time, money, labor, and minimizes the scope for risk and human errors. While much of the decision-making and consideration of the code is dependent on the experience of the reviewers, automation could at least enforce the upper and lower limits and report results instantaneously. Translation of various clauses and statements into computable language presents a major challenge in achieving automation [33,37]. However, following an ideal framework to develop a tool that successfully accounts for all regulations through accurate interpretation of formal language and model data exchange could be pivotal in increasing efficiency and upholding safety standards in an AEC project.

6.3.1. Framework to automate code-checking and compliance (ACCC) processes in BIM

The process tree of an automated rule verification system involves taxonomy development, logic arranging of rules and interpretation in its first phase. This is followed by generating building model data and commencement of information checking. Then, the actual rule compliance verification process is executed, and the results are reported. There are several techniques to translate natural language into a set of rules. There have been developments in Artificial Intelligence, markup document modeling and hypertext to formulate efficient domain knowledge representation techniques. Lau proposed the XML format which draws on standardized IFC model data [37]; Zhang, et al., 2018).

The introduction of the Industry Foundation Classes (IFC) was the most significant data standard for building model schema [42] since it offered an open-source, flexible, and standardized schema that could interact with many supporting technologies. The IFC data proposed for the ACCC framework is represented in the ifcXML format [37,38] using the ifcXML4 specifications [39]. In other words, the ifxXML data will follow these rules and would result in conformance to the ifcXML4 schema definition. The conversion rules are a subset of the configured [40] XML language binding of EXPRESS based schemas and data. Taking advantage of such standardized IFC schema to represent BIM model data, and the automatable and enforceable properties of smart contracts, ACCC can be carried out in a much more effective, secure, and instantaneous method while instantly transmitting results to various parties [42,43]. Using BC technology also negates the requirement of storing all pertinent model checking data in one location.

Clash detection or clash prevention is the most frequently employed compliance-checking domain to validate models owing to its favorable effort-benefit ratio, allowing the project team members to simultaneously detect discrepancies in the model from design, or clash between two or more facets of the building structure [41,44,47]. Most BIM platforms generally provide tools to perform clash detection studies. This technique, however, is limited in that it does not facilitate a comprehensive code review and verification of information-rich content embedded in the BIM objects. A rule-based validation method is thus essential to perform ACCC to process complex building attributes, design specifications, environmental conditions, and other areas that can be sufficiently validated through visual techniques.

The primary objective of developing an efficient framework to automate code-checking and compliance processes is to achieve a computable model with the clear syntax to accurately represent code requirements, to reduce model complexity and develop a unified format to exemplify building regulations and building information modeling. The compliance checking process must be secure and collaborative. The following example demonstrates the potential application of the Hyperledger Fabric in the automating the construction design review process in a more secure and efficient environment.

6.3.1.1. Proposed approaches. The paper proposes new approaches to execute ACCC in BIM platforms using BC technologies. These methods are outlined below:

6.3.1.1.1. Permissioned blockchain with off-chain data storage. Given the multidisciplinary nature of a BIM project team who may belong to different organizations and considering other project participants with varying levels of function and permissions, a permissioned blockchain is most suitable for usage in a BIM environment. Here, this approach considers the Hyperledger Fabric and smart contract platform that relies on the PBFT consensus mechanism. This framework proposes the storage of regulatory texts and BIM model data off-chain and facilitates the chaincode to function as the model checker tool.

- The building codes or regulations upon which the BIM model data is to be checked must be processed into a computable language. A smart contract can be programmed to process the rules from natural language into computable terms. This contract must be defined carefully to account for all clauses, terms, and variables used in the building code. After conversion of the rules, the smart contract generates a second appended smart contract that can now be used by the model checker. If the smart contract capabilities do not support adequate levels of semantic enrichment, the rules are directly expressed in the scripting languages;
- The BIM model data is exported from the platform in IFC format and is converted to the scripting language used by the smart contract platform (Java, in the case of Hyperledger Fabric). The BIM model data file expressible in Java is now generated and utilized as off-chain data;
- A model checker is programmed in the form of another smart contract that can extract information from the BIM model data upon calling and verify them against the translated rules created in the previous file;
- The model checker executes the code-checking process and creates another smart contract where the results are reported.

6.3.1.1.2. External oracles. Governing authorities, legislative bodies, and licensing entities can use external oracles to force an external state into the programmed chaincode. The concerned party can directly force the rules required to validate the building model using the smart contract containing the building model data expressed in a scripting language and can view output results directly. External oracles can also be used to directly append change and updates in existing building codes and regulations that are present in the form of smart contracts.

6.3.1.1.3. Sidechains or cross-chain bridges. A sidechain is a mechanism that can provide a solution to address the scaling issue that is essentially a hierarchy of lower-tier "consensus instances" [27] that can provide a lower degree of decentralization. It is a BC that runs in parallel to the main BC which extends functionality through interoperable BC networks permitting a decentralized method of transactions between the two chains. Fig. 7 illustrate the concept of sidechain data processing framework.

6.3.1.2. Example. BIM model data the and the building code regulations are encrypted using BC. The chaincode (containing a set of key value pairs representing the initial state of the BIM model) is saved on the peer's computers participating in the Hyperledger Fabric and instantiated on the channel. The chaincode contains logic defining

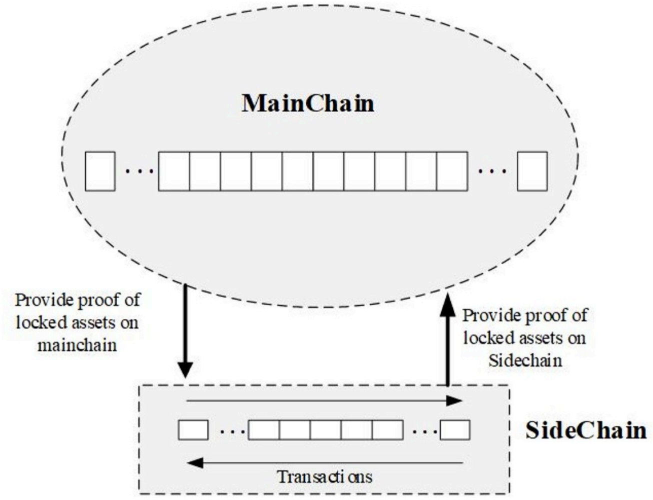
MainChain
Provide proof of locked assets on mainchain
Provide proof of locked assets on Sidechain
SideChain
Transactions

Fig. 7. Sidechains data processing concept.

a set of data exchange instructions and the agreed upon price for the service. An endorsement policy has also been set for this chaincode, stating that both peer A and peer R must endorse any transaction. A 'smart contract' is created using a Turing-complete programming language that consists of information relating to compliance checking of model data against regulations, the order in which it is carried, and the corresponding output. This contract is also discretized and encrypted into blocks. This method is secure, fast, discrete and efficient in that it records every code-checking transaction in the permanent ledger and the information is available for peers viewing.

6.3.1.2.1. Initiates a transaction. Architect A is sending a request to building department to review the design and issue a building permit for the project. The request includes a BIM model for the project. The request targets peer A and peer R, who are respectively representative of A and R. The endorsement policy states that both peers must endorse any transaction, therefore the request is sent to peer A and peer R.

Next step is to construct transaction proposal. This is achieved by leveraging a supported Software Development Kit (SDK) (using Java, Python, …etc.), that uses one of the available API's which generates a transaction proposal. The proposal is a request to invoke a chaincode function so that data can be read and/or written to the ledger. With the help of the Hyperledger Fabric Software Development Kit (SDK) the transaction proposal is transformed into the proper format utilizing the user's cryptographic credentials to produce a unique signature for this transaction proposal (see Fig. 8).

6.3.1.2.2. Verify and execute. The endorsing peers in the channel verify that: (a) the transaction proposal is well formed, (b) it has not been submitted previously (replay-attack protection), (c) the signature is valid (using MSP), and (d) that the submitter (Architect A, in the example) is accurately approved to accomplish the proposed operation on that channel (i.e. the submitter satisfies the channel's Writers policy (The Writing policy specifies at the time of channel creation which user is permitted to submit a transaction to that channel).

The peers consider the transaction request inputs as arguments to the invoked chaincode's function (see Fig. 9). The chaincode is then executed against the current state database to generate transaction results including a response value, read dataset, and write dataset. No updates are made to the ledger at this point. This result, along with the endorsing peer's signature is conveyed back as a “proposal response” to the SDK which parses the information for the application to process.

6.3.1.2.3. Proposal authentication. The application validates the peers' signatures and examines the proposal responses to see if the answers are the correct ones. The chaincode would then process the proposal and send it to respective parties for further processing (see Fig. 10). This can be the case, for instance, when requests for an input from the zoning department is needed to verify the site location of the project. The system is generally built in such a way that even if an application decides not to verify responses or otherwise forwards an unauthorized transaction, the endorsement policy will still be enforced by peers and maintained at the validation phase before writing results to the ledger.

6.3.1.2.4. Assembles replies. The blocks of data are then validated and transmitted to all peers on the channel of the network. The validation process also ensures that there have been no changes to the ledger state and blocks are marked as being valid or invalid.

Then the application will transmit the transaction proposal and response within a “transaction message” to the correct code checking services, which can consist of several chaincodes. The transaction will contain the read/write datasets, the endorsing peers' signatures, and

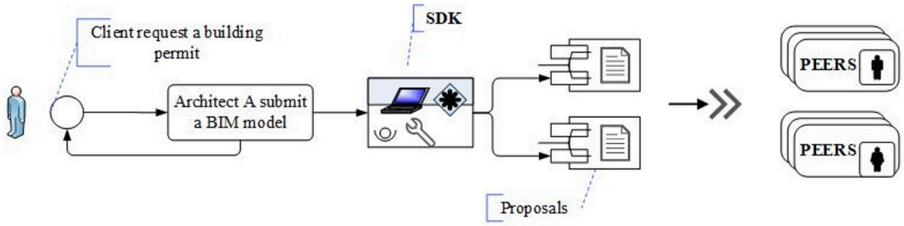
Client request a building permit
SDK
Architect A submit a BIM model
PEERS
PEERS
Proposals

Fig. 8. Data exchange initiation between an architect and reviewers.

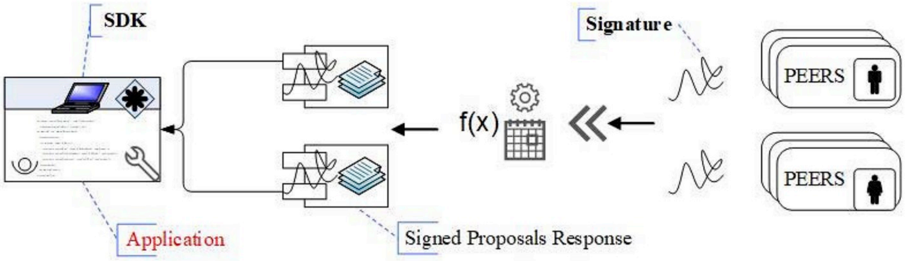

Fig. 9. Response of peers R to the initial request.

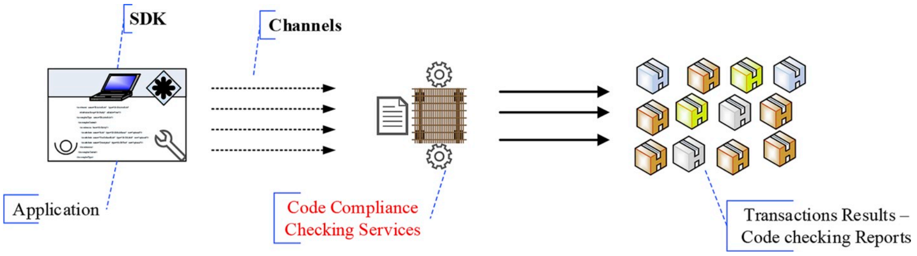
SDK
Channels
Application
Code Compliance Checking Services
Transactions Results – Code checking Reports

Fig. 10. Constructing blocks of the results.

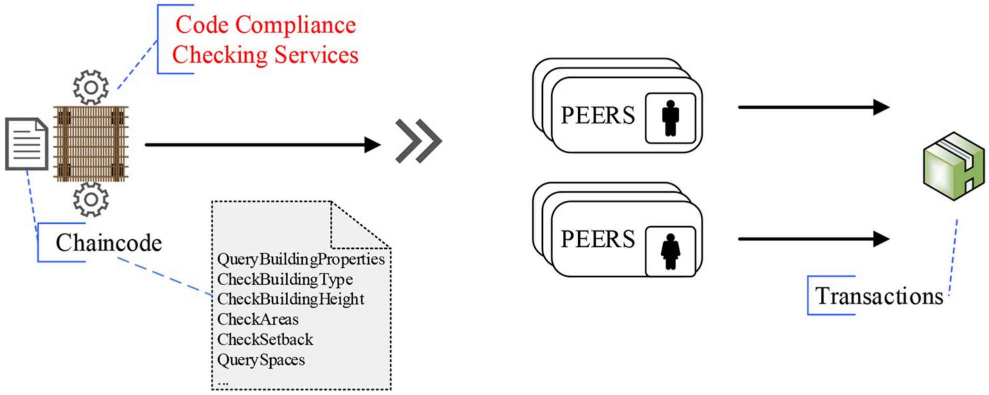

Fig. 11. Constructing blocks of the results.

the Channel identification ID. Then, the application will build up the blocks of the code compliance checking transactions per channel (see Fig. 11).

6.3.1.2.5. Validation and transmission. The blocks of data are then validated and transmitted to all peers on the channel of the network. The validation process also ensures that there have been no changes to the ledger state and blocks are marked as being valid or invalid.

6.3.1.2.6. Updating the database. The blocks will be appended by each peer in the channel for valid blocks and they are written to the current state of the ledger (see Fig. 12). After the transaction is committed to the database, an event is invoked to notify the Architect A that the reply from the building review authority R transaction has been appended to the chain, and available for the Architect to take further actions.

6.4. Summary

When comparing Hyperledger Fabric to permissioned blockchain such as Ethereum, the Fabric is more beneficial because nodes assume different roles and tasks in reaching a consensus, and are classified according to their roles as clients, peers or orderers. This modular architecture of Fabric allows customization into various applications depending on usage. The protocol of Fabric operates through two kinds of peers:

- *Validating peers*: These are nodes tasked with operational functions such as running consensus, verifying transactions, and maintaining the ledger;
- *Non-Validating peers*: These nodes connect the clients that make the transactions to the validating peers.

The client sends the transactions to all its connected endorsers who agree to reach consensus and initiate the update. The client collects the approval of all the endorsers and sends approved transaction to the orderers who then reach consensus on the transaction block and forward it to all the peers holding the ledger. The peers then commit the transaction to the ledger.

Hyperledger Fabric does not require built-in cryptocurrencies since mining is not necessary to reach a consensus. Every transaction is executed (endorsed) only by a subset of peers, allowing for parallel execution and addressing potential non-determinism and scalability. However, the creation of a digital currency is possible with chaincode, a program code that implements the application logic and runs during the execution phase. It serves as the central part of a distributed application in Fabric and may be written by any developer. The chaincode is executed within an environment that is loosely coupled with the rest of peer and support plugins for adding new languages for programming chaincodes.

7. Conclusions

A blockchain is defined as a decentralized database that records every transaction complete in the network, known as a ‘block’. Each block comprises encrypted data of the entire transaction history. The main concepts behind BC technology are the creation of a digital distributed consensus, ensuring that data is decentralized among several nodes in the network that hold identical information and that no single node holds the complete authority of the network. The application of such decentralized technology in any industry would require augmented security, enforce accountability, and could potentially accelerate a shift in workflow dynamics from the current hierarchical structure to a decentralized, cooperative chain of command and effect a cultural and societal change by encouraging trust and transparency. This paper presents a comprehensive survey of BC technology and its applications in the built environment and examines the potential integration with building information modeling (BIM) workflow. The study examines how commissioning distributed ledger technology (DLT) could be advantageous in the BIM working processes by reinforcing network security, providing more reliable data storage and management of permissions, ensuring change tracing and data ownership.

The paper explored the fundamentals of DLT, their potential future applications and current advances, and their classifications based on inherent characteristics of consensus reaching, execution of the smart

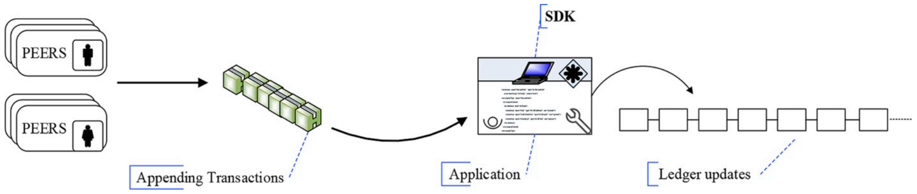

Fig. 12. Updating hyperledger.

contract (chaincode), and permission management. Moreover, the study examines the potential application of BC technology in enhancing the framework for automating the construction design review process. Hyperledger Fabric is a BC technology that can be particularly suited for developing the automation tool for code-checking compliances in BIM design review process due to its ease of programming, flexibility, user-defined smart contract feature, robust security, identity features, and modular architecture with pluggable consensus protocols. The example presented depicts that smart contract technologies available in blockchains like Ethereum and Hyperledger, as well as chaincodes available in Hyperledger Fabrics, provide promising technologies in advancing the security and efficiency of the automation of the Automate Code-Checking and Compliance in the AEC industry. Using Hyperledger Fabric can address many of the current concerns such as data security, change tracing and permission management that arise from using centralized BIM work processes. Future research work will focus on Hyperledger Fabrics applications in enhancing automating code-checking and compliances in the BIM environment.

References

[1] M. Crosby, Nachiappan, P. Pattanayak, S. Verma, V. Kalyanaraman, Blockchain Technology: beyond Bitcoin, Sutardja Center for Entrepreneurship & Technology Technical Report, University of California, Berkley, 2015.

[2] Y. Shen, K. Li, W. Shi, Advanced topics on cloud computing, J. Comput. Syst. Sci. 79 (8) (2013) 2013 https://doi.org/10.1016/j.jcss.2013.02.002.

[3] S. Brakeville, B. Perepa, Blockchain Basics: Introduction to Business Ledgers, Issued by IBM Corporation, 2016.

[4] H. Hasan, K. Salah, Blockchain-based proof of delivery of physical assets with single and multiple transporters, IEEE Access 6 (2018).

[5] D. Puthal, N. Malik, S.P. Mohanty, E. Kougianos, C. Yang, The Blockchain as a Decentralized Security Framework, University of Technology Sydney (UTS), 2017.

[6] M. Swan, Blockchain: Blueprint for a New Economy, O'Reilly Media, Inc., Sebastopol, GA, 2015 Published by.

[7] Z. Zheng, S. Xie, H. Dai, X. Chen, H. Wang, An Overview of blockchain technology: architecture, consensus, and future trends, IEEE 6th International Congress on Big Data 2017, 2017, pp. 557–657.

[8] S. Nakamoto, Bitcoin: A Peer-To-Peer Electronic Cash System, (2008) https://bitcoin.org/bitcoin.pdf , Accessed date: 1 August 2018.

[9] Z. Turk, R. Klinic, Potentials of blockchain technology for construction management, *Creative Construction Conference 2017*, CCC 2017, 19-22 June 2017, Primosten, Croatia, 2017.

[10] M. Mathews, D. Robles, B. Bowe, BIM + Blockchain: a solution to the trust problem in collaboration? CITA BIM Gathering 2017, Nov 2017, 2016.

[11] J. Mason, Intelligent contracts and the construction industry, J. Leg. Aff. Dispute Resolut. Eng. Constr. 9 (3) (2017) 1943–1462.

[12] C.D. Clack, A.V. Bakshi, L. Braine, Smart Contract Templates: Foundations, Design Landscape and Research Directions, Barclays Bank PLC, 2016 2016-2017.

[13] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Gollamudi, G. Gonither, N. Kobeissi, N. Kulatova, A. Rastogi, T. Sibut-Pinote, N. Swamy, S. Zanella-Beguelin, Formal verification of smart contracts, PLAS'16, October 24, 2016, Vienna, Austria, 2016.

[14] M. Dhawan, Analyzing Safety of Smart Contracts, IBM Research, 2016.

[15] C.K. Frantz, From Institutions to Code: towards Automated Generation of Smart Contracts. Workshop on Engineering Collective Adaptive Systems (Ecas), Co-located with SASO 2016, (2016).

[16] J.B. Atkins, A.D. Mendelson, BIM Me UP, Scotty: Navigating Risk in Digital Practice, AIA Trust, 2013 published by the https://www.berkleydp.com/wp-content/uploads/2018/02/BIM-Navigating_Risk_in_Digital_Practice.pdf.

[17] L. Cong, Z. He, Blockchain Disruption and Smart Contracts, Working Paper No. 24399, issued in March 2018, revised in April 2018 NBER, 2018.

[18] J. Li, D. Greenwood, M. Kassem, Blockchain in the construction sector: a socio-technical system framework for the construction industry, CIB W78 2018, 1-3, October, Chicago, Illinois, USA, 2018, pp. 51–58.

[19] M.P. Andersen, J. Kolb, K. Chen, D.E. Culler, R. Katz, Democratizing authority in the built environment, Proceedings of the 4th International Conference on Systems for Energy-Efficient Built Environments – BuildSys '17, November 8-9, 2017, Delft, The Netherlands, 2017, pp. 1–10.

[20] H. Boyes, Building Information Modelling (BIM): Addressing the Cyber Security Issues, Institution of Cyber Security (IET) Consortium Report, London, UK, 2014, p. 2014.

[21] R. Coyne, T. Onabolu, Blockchain for Architects: Challenges from the Shared Economy", Architectural Research Quaterly vol. 21, Cambridge University Press, 2017, pp. 369–374 2018.

[22] M. Bartoletti, L. Pompianu, An Empirical Analysis of Smart Contracts: Platforms, Applications, and Design Patterns, Universitia degli Studi di Cagliari, Cagliari, Italy, 2017 arXiv:1703.06322v1 [cs.CR] 18 Mar 2017.

[23] S. Singh, N. Singh, Blockchain: future of financial and cyber security, 2nd International Conference on Contemporary Computing and Informatics (Ic3i), 2016.

[24] X. Xu, C. Pautasso, L. Zhu, V. Gramoli, A. Ponomarev, S. Chen, The blockchain as a software connector, 13th Working IEEE/IFIP Conference on Software Architecture (WICSA 2017), Venice, Italy, 2017.

[25] J.L. Zhao, S. Fan, J. Yan, Overview of Business Innovations and Research Opportunities in Blockchain and Introduction to the Special Issue*, Financial Innovation 2016*, University of Hong Kong, Hong Kong, 2016.

[26] V. Buterin, Chain Interoperability, (2016) https://static1.squarespace.com/static/55f73743e4b051cfcc0b02cf/t/5886800ecd0f68de303349b1/1485209617040/Chain + Interoperability.pdf.

[27] K. Croman, C. Decker, I. Eyal, A.E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E.G. Sirer, D. Song, R. Wattenhofer, On scaling decentralized blockchains (A position paper), International Conference on Financial Cryptography and Data Security, FC 2016: Financial Cryptography and Data Security, 2016, pp. 106–125.

[28] V. Buterin, Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform, (2014).

[29] P. Glaser, Pervasive decentralization of digital infrastructures: a framework for blockchain enabled system and use case analysis, Proceedings of the 50th Hawaii International Conference on System Sciences (2017), 2017, pp. 1543–1552.

[30] J. Yii-Huumo, D. Ko, S. Choi, S. Park, K. Smolander, Where is current research on blockchain technology? – a systematic review, PLoS One 11 (10) (2016) e0163477, https://doi.org/10.137/journal.pone.016377.

[31] E. Androulaki, C. Cachin, C. Ferris, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sormiotti, C. Stathakopoulou, M. Vukolic, S.W. Cocco, J. Yellick, Hyperledger fabric: a distributed operating system for permissioned blockchains, EuroSys '18, April 23-26, 2018, Porto, Portugal, 2018.

[32] A. McNulty, Blockchain: an Exhaustive Guide to Learning Blockchain Technologies, (2018) ASIN: B079QNPQF6.

[33] C. Eastman, P. Teicholz, R. Sacks, K. Liston, *BIM Handbook: A Guide to Building* Information Modeling for Owners, Managers, Designers, Engineers and Contractors, John Wiley & Sons, 2011.

[34] N. Kshetri, "Blockchain's roles in strengthening cybersecurity and protecting privacy", Telecommun. Pol. 41 (2017) (2017) 1027–1038.

[35] A. Hammi, A. Bouras, Towards safe-BIM curricula based on the integration of cybersecurity and blockchains features, Proceedings of INTED2018 Conference, 5th – 7th March 2018, Valencia, Spain < hal-01737929 > , 2018, pp. 2380–2388.

[36] J. Wang, P. Wu, X. Wang, W. Shou, The outlook of blockchain technology for construction engineering management, Front Eng. Manag. 4 (1) (2017) 67–75 2017.

[37] N.O. Nawari, Building Information Modeling: *Automated Code Checking and Compliance Processes*, CRC Press, Taylor & Francis Group, LLC, Boca Raton, Florida, 2018, p. 2018 Published by.

[38] N.O. Nawari, A generalized adaptive framework for automating design review process: technical principles, *Proc. Of the 35th CIB W78 Conference 2018*, October 1-3, 2018, Chicago, 2018, pp. 405–414.

[39] T. Liebich, M. Weise, ifcXML4 Specification Methodology, buildingSMART International, 2013, http://www.buildingsmart-tech.org/downloads/ifcxml/ifcxml4/ifcxml4_specification_methodology_v1-1.pdf , Accessed date: January 2019.

[40] ISO10303-28:2007, Industrial Automation Systems and Integration – Product Data Representation and Exchange – Part 28: Implementation Methods: XML Representations of EXPRESS Schemas and Data, Using XML Schemas, International Organization for Standardization (ISO), 2007.

[41] V. Getuli, S.M. Ventura, P. Capone, A.L.C. Ciribini, BIM-based code checking for construction health and safety, Creative Construction Conference 2017, CCC 2017, 19-22 June 2017, Primosten, Croatia, 2017.

[42] T. Bloch, M. Katz, R. Sacks, Machine learning approach for automated code compliance checking, 17th International Conference on Computing in Civil and Building Engineering, Tampere, Finland, 2018.

[43] D. Greenwood, S. Lockley, Malsane, J. Matthews, Automated code checking using building information models, The Construction, Building and Real Estate Conference of the Royal Institution of Chartered Surveyors [Held at Dauphine Universitie, Paris, 2-3 September 2010], RICS, London, 2010978-1-84219-618-9.

[44] C.S. Han, J. Kunz, K.H. Law, Making Automated Building Code Checking A Reality, Center for Integrated Facility Engineering, Stanford University, California, 1997 Facility Management Journal (Sep/Oct 1997).

[45] A. Mahamadu, L. Mahdjoubi, C. Booth, Challenges to BIM-cloud integration: implications of security issues on secure collaboration, 2013 IEEE International Conference on Cloud Computing Technology and Science, 2nd-5th Dec, 2013, 2013.

[46] J. Mason, H. Escott, Smart contracts in construction: views and perceptions of stakeholders, Proceedings of FIG Conference, May 2018, 2018.

[47] X. Tan, A. Hammad, P. Fazio, Automated code compliance checking for building envelope design, J. Comput. Civ. Eng. 24 (2) (2010) 203–211. ASCE (March/April 2010).