sustainability

MDPI

Article

Blockchain-Based Implementation of Building Information Modeling Information Using Hyperledger Composer

Widya Nita Suliyanti * and Riri Fitri Sari

Department of Electrical Engineering, Universitas Indonesia, Depok 16424, Indonesia; riri@ui.ac.id
* Correspondence: widya.nita@ui.ac.id

Abstract: With the advancement of Building Information Modeling (BIM) technology, BIM gains more importance and becomes a prerequisite in building projects. BIM is useful throughout a building lifecycle; from building bid, design, construction, completion, operation, and maintenance to building demolition. However, current information exchange surrounding BIM is still limited and bound to a single participant or organization and is also limited to a particular phase in the building lifecycle. This paper aims to explore BIM information exchange among many parties involved in a secure manner using a blockchain platform throughout the whole building lifecycle. In this research, many parties involved in the building project will be able to recognize one another through deployment of a permissioned blockchain. This information exchange uses Hyperledger Composer, a permissioned blockchain running on a blockchain platform called Hyperledger Fabric. Our experiment shows that BIM information exchange could be further improved. In this study, BIM information exchange can be implemented not only in one building phase but throughout the whole building lifecycle. It also facilitates BIM information exchange among multiple participants in a secure manner via a permissioned blockchain.

Keywords: blockchain; permissioned blockchain; Building Information Modeling; BIM; Hyperledger Composer

Citation: Suliyanti, W.N.; Sari, R.F. Blockchain-Based Implementation of Building Information Modeling Information Using Hyperledger Composer. *Sustainability* **2021**, *13*, 321. https://doi.org/10.3390/su13010321

Received: 30 November 2020
Accepted: 27 December 2020
Published: 31 December 2020

1. Introduction

Building Information Modeling (BIM) is a digital representation of physical and functional characteristics of a facility, creating a shared knowledge resource of information about it and forming a reliable basis for decisions during its lifecycle, from earliest conception to demolition [1,2]. BIM is considered an ideal digital tool for digitally representing the data repository of all information relating to the building lifecycle [3]. BIM is likely to enhance building Life Cycle Assessment (LCA) by managing databases, reducing complexity, improving adaptability to changes, and improving communication and collaboration. LCA is the assembly and estimation of the inputs, outputs, and potential environmental impacts of a building throughout its lifecycle [1]. BIM is also used to calculate Life Cycle Cost (LCC), which is an asset cost or parts of an asset cost throughout a building lifecycle. The LCC should be estimated first to evaluate the economic efficiency of a building, considering all cost elements. In some countries, building projects use BIM software as a requirement. Thus, BIM becomes an essential part of building projects.

BIM saves building-related information throughout a building lifecycle. However, BIM is most often applied at the beginning of design phases, with less use towards the later stages of a building project lifecycle [1–4]. During this later phase, BIM information is communicated to other (external) parties/companies in limited ways through meetings, emails, and reports. Other parties/companies cannot see BIM information directly. Moreover, other parties do not receive all BIM information, and they probably will not be able obtain it at the same time. Building contractors also mostly use BIM internally in a centralized manner.

In a building lifecycle, first, a building owner conducts a building project tender that building contractors participate in by sending building proposals that include BIM information. Then, the building owner will announce the building project winner who will submit a building design proposal to the building owner. Until the building owner is satisfied with the building design, the building contractor winner will keep reiterating the building design. Once the building owner approves the preliminary design, the building contractor will start the building development, and there will be design changes throughout this phase. Then, the building will enter a building completion stage, and building maintenance will start. The building maintenance phase will persist until there is a need to demolish the building. During building maintenance, Facility Management will update the BIM information.

Currently, one potential way to improve BIM information usage efficiency is to overcome the limitation in BIM information exchange so that it is accessible to all parties involved in the project. On the other hand, it also needs to limit the access of some unrelated parties to BIM. We propose that BIM information could be implemented for the entire building lifecycle and stored in a secured permissioned blockchain. This will also provide a way to access BIM information for many involved parties/organizations—such as building designers, contractors, engineers, architects, and demolition companies—in the entire building lifecycle in a secure manner. A blockchain, in the permissioned model, typically establishes identities and forms a consortium [5]. A permissioned blockchain adds a layer of privileges to decide who can participate in the network system; these parties will not have a chance to commit fraud as the management server exposes their identities [6]. In addition, only specific entities are authorized to use, validate, and access a blockchain [7,8]. This paper aims to explore how a permissioned blockchain framework called Hyperledger Composer runs on a blockchain platform called Hyperledger Fabric. It implements BIM information throughout the building lifecycle and allows secured information exchange among multiple parties. The remainder of the paper is organized as follows: Section 2: Material and Method; Section 3: Result; Section 4: Evaluation of Result; Section 5: Conclusion.

2. Material and Method

2.1. Blockchain

A blockchain is a distributed ledger that records all transactions, such as storage and asset transfer, which take place in a safe, chronological, and unalterable way using peer-to-peer networking protocols. It does not depend on a trusted central authority to validate transactions and extend the blockchain [9]. Blockchains provide a transparent, immutable, and consistent data store [10].

Blockchains have the following characteristics:

- Transparency

Data stored on a blockchain are accessible to all participants within the blockchain network [10].

- Decentralization

Blockchain transactions are peer-to-peer, without central authorization. Decentralization promotes data confidentiality by shielding interactions and communication from third parties and encouraging data sovereignty, giving the user control over their data [11–13].

- Persistency

Persistency guarantees a consistent system response of a transaction state. As an illustration: if one network node reports a stable condition, then the other network node should also state as stable when queried and reacted to honestly. Persistency warrants that when a transaction is appended to a block that is located deep in the blockchain of an honest node, this transaction will ultimately be contained in every authentic blockchain node in the network, with high probability [12–14].

- Anonymity

Without revealing real user identities, transactions related to users cannot be linked [12–14]. Transactions are the atomic data structure of a blockchain. Typically, a set of users/participants create a transaction to indicate the transfer of tokens from the sender to the specified receivers. The functionalities of cryptographic hashing and asymmetric encryption are activated to protect the authenticity of a transaction record [15].

2.2. Hyperledger Composer and Hyperledger Fabric

Hyperledger Composer is a Linux Foundation project that is well known and advancing rapidly. It is an open-source development tool that provides a framework to develop blockchains quickly [16,17]. Thus, users can cut down the development time from months to weeks [17]. Hyperledger Composer allows anyone to contribute to the project as one of many benefits.

Hyperledger Composer is a modeling language that is simple and area-specific to model business networks. The model file can use any available development platforms, while JavaScript is used to implement the blockchain transaction logic shown in Figure 1. Hyperledger Composer is a web-based tool that provides support for development, packing, and testing phases. It also supports writing scripts.

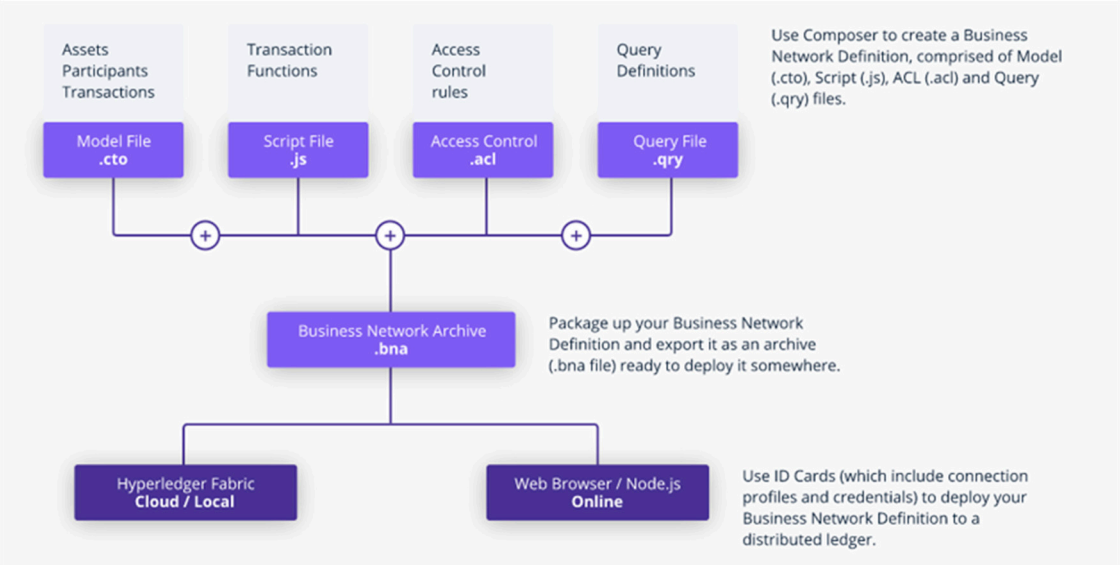

Figure 1. Hyperledger Composer [16].

Hyperledger Composer can contain many Business Network Archives (.bna). Each .bna file consists of a Model file (.cto), a Script file (.js), and an Access Control file (.acl). A Model file specifies blockchain items such as participant, transaction, and asset. A Script file defines the transaction logic, and it is invoked during the submission of a transaction. An Access Control file specifies access rules for business network items [13]. Then, there are two types of Business Network Archive deployments. It can be deployed in Hyperledger Fabric or via Web Browser [16,18]. Hyperledger Composer's framework also provides a flexible development platform, an object-oriented programming language to define Assets, and a JavaScript engine to create Smart Contracts [19].

Hyperledger Composer runs on top of Hyperledger Fabric, which provides developers tools to create permissioned blockchain solutions. Hyperledger Fabric provides an implementation of permissioned blockchains [6,20] and the infrastructure to develop its network. In Hyperledger Fabric, business network participants can specify a policy to validate blockchain transactions by using an alterable blockchain consensus protocol. The business network includes participants who interact with them. Each participant can be associated with a unique identity across several different business networks [17]. In Hyperledger

Fabric, one has to obtain credentials to read the ledger or write in it. The participants that are allowed to read or write to the ledger are called peers. Thus, it is easier to control the transactions on the ledger and it has higher system throughput [21–23]. Hyperledger Fabric secures the interactions among a group of authorized participants to preserve privacy and confidentiality [22]. A Hyperledger Fabric network consists of the following elements:

- Peer

Each network node executes chaincode and receives transactions from external client applications through the Hyperledger Fabric API. Every node's identity on a Hyperledger Fabric network is presented by a membership service provider (MSP) [24].

- Ledger

A ledger is a collection of ordered blocks that contain transactions.

- Channel

Organizations can communicate through a channel that offers isolation and privacy. Each channel consists of only one ledger, and that ledger exists only on that channel.

- Orderers

An orderer gathers transactions from all Hyperledger Fabric channels in a blockchain network. Orderers generate blocks and validate transactions in channels issued by a network administrator.

- Consortiums

A consortium gives organizations the capability to produce agreements on which policies will be followed to validate blocks [19].

2.3. Methods

First, we define the phases in a building lifecycle with BIM attached to each stage. The stages are Bidding, Development, Maintenance, and Demolition. Then, we determine the functional needs, as follows:

- To store BIM information.
- To store BIM status.
- To ensure there is BIM information for each building phase.
- To monitor building status.

Then, in the design stage, we design a workflow for exchanging BIM information among multiple parties involved in each phase of the building lifecycle. The Use Case Diagram, which is one component of the Unified Modeling Language (UML), is implemented to model the behavioral aspect of a system. UML is a modeling language with a semiformal syntax and semantics [25]. It is a suitable development model and is used to capture functional requirements [26]. It supports modeling a wide range of component interaction paradigms and addresses the overall system behavior in a usage scenario [25].

Then, we implement Hyperledger Composer Business Network Archive (BNA) elements that define and determine the capabilities of storing BIM information on the blockchain platform [27]. A BNA file consists of the following parts:

a. Participants

Participants are the involved parties throughout the blockchain network.

b. Assets

Assets are the data that are kept and exchanged in a blockchain network.

c. Transaction

Transactions are the business logic or smart contracts of the system that specify the certain conditions or clauses of the contract.

d. Access Controls

Access Controls determine participants' access to the blockchain network based on their roles in the transaction types executed therein.

3. Result

Figure 2 shows the system flow using blockchains for exchanging BIM information. A Building Contractor submits a building proposal to the Building Owner, and the Building Owner determines the Building Contractor winner. This activity will be verified and recorded in a blockchain along with BIM information. Then, upon the completion of building construction by the Building Contractor, the Building Maintenance (Facility) Manager will take over control from the Building Contractor. It will make the necessary changes to BIM information to keep it updated. Afterward, when the building needs to be demolished, the Facility Manager will hand BIM information over to Demolition Experts. These experts will ask for a building demolition approval from the Building Owner. All these building lifecycle activities will be verified and recorded in the blockchain.

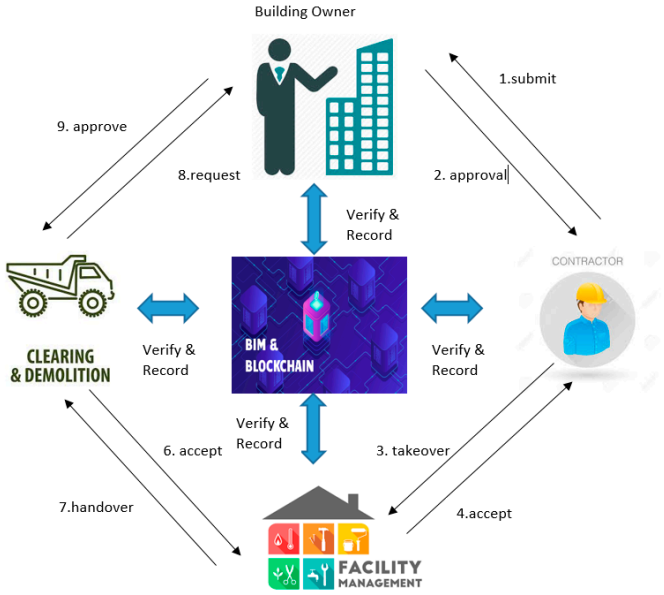

Figure 2. Building information modeling (BIM) workflow on the blockchain platform throughout a building lifecycle.

3.1. Use-Case Diagram

The Use-Case Diagram in Figure 3 is a behavioral model for information systems for exchanging BIM information. The diagram consists of five parties relevant to the building lifecycle, such as: Building Owner, Building Contractor, Building Maintenance Manager, Building Demolition Expert, and Network Admin. Each party has their use case, e.g., the Building Owner can open a building bid and see the bidding details; the Building Contractor can design and construct a building using BIM; the Building Maintenance Manager can maintain the completed building; the Building Demotion Expert can demolish a building upon approval from the Building Owner. A Network Admin can manage all activities and give access to each individual party.

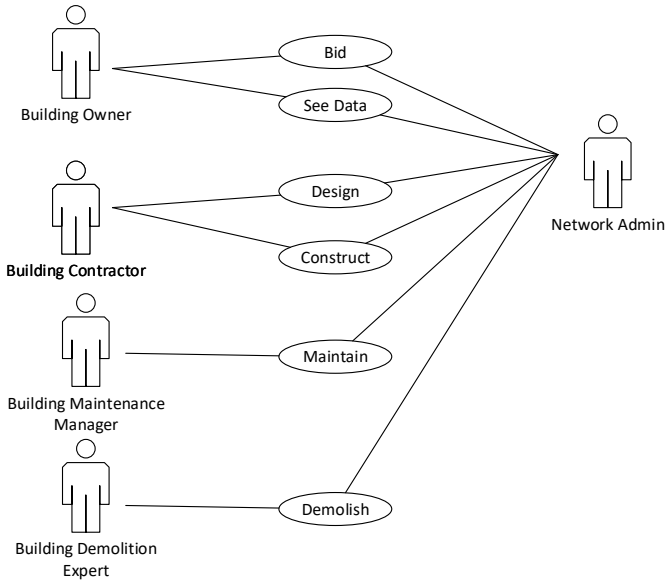

Figure 3. Use-Case Diagram.

3.2. Hyperledger Business Network Archive

After we create the system flow, we create a few components that form the Hyperledger Business Network Archive as follows:

3.2.1. Participant

Table 1 lists the Participants, which refer to parties involved in this information system running on the blockchain network.

Table 1. Participant.

No	Participant
1	Network Admin
2	Building Owner
3	Building Contractor
4	Building Maintenance Manager
5	Building Demolition Expert

A Network Admin governs other parties' privileges and transactions. At the beginning of a building lifecycle, a Building Owner conducts a building bid and can update the bidding information details. A Building Contractor participates in a building bid by sending a proposal, and also participates during building development. Once the building is completed, a Building Maintenance Manager takes over from the Building Contractor. At the end of a building lifecycle, a Building Demolition Expert will receive approval from the Building Owner to demolish the building.

3.2.2. Asset

An Asset is defined as anything that has value. Assets record transaction data. Table 2 shows a list of assets involved throughout a building lifecycle. Bim is an asset that includes the URL location of the Building Information Modeling (BIM) file, while Bid is an asset containing all bid-related information. The Development asset has building construction information. The Maintenance asset contains information once the building is completed and moves to an operational phase. At the end of a building lifecycle, the Demolition asset records building demolition information. The Building asset saves building-related information during each building phase.

Table 2. List of Assets.

No	Asset	Key Fields	No	Asset	Key Fields
1.	Bim	bimId	4.	Development	developmentId
		buildingId			buildingContractorId
		Type			buildingId
		url			Amount
		bimStatus			startDate
		bimDate			Bim
		buildingOwner	5.	Maintenance	maintenanceId
2.	Building	buildingId			buildMaintMgrId
		Name			buildingId
		Address			startDate
		dateCompletion			Bim
		Status	6.	Demolition	demolitionId
		statusDate			buildDemoExpertId
		buildingOwner			buildingId
		buildingContractor			demolitionDate
		buildingMaintenanceManager			Bim
		buildingDemolitionExpert			
3.	Bid	bidId			
		buildingOwnerId			
		buildingContractorId			
		buildingId			
		bidAmount			
		bidDate			
		Bim			

3.2.3. Transaction

A transaction is a process that is done by a business network participant to change the value of an asset. Four transactions cover participants' activities throughout the building lifecycle, e.g., bidBuilding, developBuilding, maintainBuilding, and demolishBuilding as in the below Table 3. The bidBuilding transaction verifies building bid activity. The developBuilding transaction verifies building construction activity. The maintainBuilding transaction verifies building maintenance activity, and the demolishBuilding transaction verifies building demolition activity.

Table 3. List of Transactions.

No	Transaction	Participant	Remarks
1	bidBuilding	Building Owner Building Contractor	This transaction verifies a building bid.
2	developBuilding	Building Owner Building Contractor	This transaction verifies building construction.
3	maintainBuilding	Building Owner Building Maintenance	This transaction verifies building maintenance.
4	demolishBuilding	Building Owner Demolition Expert	This transaction verifies building demolition.

Each of the four transactions above was then implemented using Hyperledger Composer as shown in Figure 4. The bidBuilding transaction involves two participants, Building Owner and Building Contractor, and three assets: Bid, Building, and BIM. The developBuilding transaction consists of two participants, Building Owner and Building Contractor, and three assets: Development, Building, and BIM. The maintainBuilding transaction consists of two participants, BuildingOwner and BuildingMaintenanceManager, and three assets: Maintenance, Building, and BIM. The demolishBuilding transaction consists of two participants, BuildingOwner and BuildingDemolitionExpert, and three Assets: Demolition, Building, and BIM.

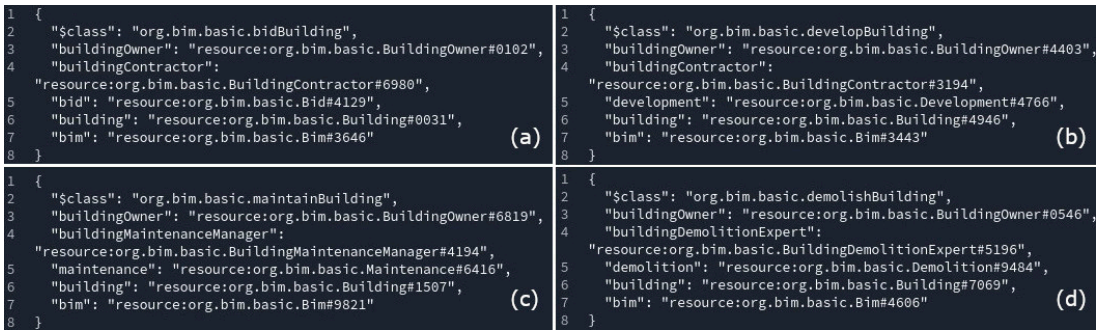

Figure 4. BIM information transactions: (**a**) bidBuilding; (**b**) developBuilding; (**c**) maintainBuilding; (**d**) demolishBuilding.

Thus, throughout the building lifecycle, in each stage, each transaction always includes two valuable Assets, e.g., Building and BIM. BIM information is then stored in the form of an encrypted URL string that points to the location of the BIM file to protect the exact location of the BIM files so users cannot access them directly.

3.2.4. Access Control List

Each participant in Hyperledger Composer has a different privilege in each phase of the building lifecycle. The Hyperledger Composer Access Control List reflects these varied privileges as shown in Table 4. There are four types of privileges, i.e., Create (C), Read (R), Update (U), and Delete (D), given to a Participant to access the Assets. The Create privilege allows a participant to create an Asset; the Read privilege allows a participant to Read Asset information; the Update privilege allows a participant to change the Asset information, and the Delete privilege allows a participant to delete an Asset. There are six Assets involved, i.e., BIM, Building, Bid, Development, Maintenance, and Demolition.

Table 4. Access Control List.

No	Participant	Bim	Building	Bid	Development	Maintenance	Demolition
1	Network Admin	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD
2	Building Owner	CRUD	CRUD	CRUD	R	R	R
3	Building Contractor	CRUD	RU	CRUD	CRUD	R	R
4	Building Maintenance Manager	RU	RU	R	R	CRUD	R
5	Building Demolition Expert	RU	RU	R	R	R	CRUD

3.3. Transaction Processing

When a participant submits a transaction, the participant executes operations on the corresponding assets. In this BIM information implementation, there are four transactions, i.e., bidBuilding, developBuilding, maintainBuilding, and demolishBuilding. During each transaction, there are processes to update related Assets as follows.

3.3.1. bidBuilding Transaction

In this transaction, the Building Owner makes and inputs a building bid (Building Owner ID, Building Contractor ID, Bid ID, Building ID, Bim ID). Then, the Building Owner submits the transaction. When all the parameters are gathered, this transaction calls blockchain transaction processing. Algorithm 1 shows the transaction logic in detail.

Algorithm 1. bidBuilding

Input: Building Owner ID, Building Contractor ID, Bid ID, Building ID, Bim ID

```
Bim ← Bid Bim
Bim Date ← Bid Transaction Date
Building Owner ← Bid Building Owner
Building Contractor ← Bid Building Contractor
If Building Status == "" then
Buiding Status ← "Bidding"
Building Status Date ← Bid Transaction Date
Bim Status ← "Bidding"
else
Error
```

3.3.2. developBuilding Transaction

In this transaction, the Building Owner inputs (Building Owner ID, Building Contractor ID, Development ID, Building ID, Bim ID) and submits the transaction through the API in the system. When all parameters are gathered, this transaction calls blockchain transaction processing. Algorithm 2 shows transaction logic in detail.

Algorithm 2. developBuilding

Input: Building Owner ID, Building Contractor ID, Development ID, Building ID, Bim ID

```
Bim ← Development Bim
Development Start Date ← Development Transaction Date
Building Owner ← Development Building Owner
Building Contractor ← Development Building Contractor
If Building Status == "Bidding" then
Building Status ← "Development"
Building Status Date ← Development Transaction Date
Bim Status ← "Development"
else
Error
```

3.3.3. maintainBuilding Transaction

In this transaction, the Building Maintenance Manager inputs (Building Owner ID, Building Maintenance Manager ID, Building ID, Bim ID) and submits the transaction through the API in the system. When all the parameters are gathered, this transaction calls blockchain transaction processing. Algorithm 3 shows the transaction logic in detail.

Algorithm 3. maintainBuilding

Input: Building Owner ID, Building Maintenance Manager ID, Maintenance ID, Building ID, Bim ID

```
Bim ← Maintenance Bim
Maintenance Start Date ← Maintenance Transaction Date
Building Owner ← Maintenance Building Owner
Building Maintenance Manager ← Maintenance Building Maintenance Manager
If Building Status == "Development" then
Building Status ← "Maintenance"
Building Status Date ← Maintenance Transaction Date
Bim Status ← "Maintenance"
else
Error
```

3.3.4. demolishBuilding Transaction

In this transaction, the Building Demolition Expert inputs (Building Owner ID, Building Demolition Expert ID, Demolition ID, Building ID, Bim ID) and submits the transaction through the API in the system. When all parameters are gathered, this transaction calls blockchain transaction processing. Algorithm 4 shows the transaction logic in detail.

Algorithm 4. demolishBuilding

Input: Building Owner ID, Building Demolition Expert ID, Demolition ID, Building ID, Bim ID

```
Bim ← Demolition Bim
Demolition Date ← Demolition Transaction Date
Building Owner ← Demolition Building Owner
Building Demolition Expert ← Demolition Building Demolition Expert
If Building Status == "Maintenance" then
Building Status ← "Demolition"
Building Status Date ← Demolition Transaction Date
Bim Status ← "Demolition"
else
Error
```

4. Discussion

Previously, parties with BIM information shared the information in limited ways, e.g., emails and meetings, rather than through a systematic approach. These unsecured methods can result in other involved parties not being aware when BIM files are modified. Moreover, parties who have access to BIM files can share BIM information with parties who may be unrelated to the projects. This puts highly confidential BIM information security at risk. Using a permissioned blockchain, BIM information can be securely accessed only by network participants and involved parties throughout the building lifecycle, i.e., bidding, development, maintenance, and demolition. BIM information includes building status, BIM status, and BIM files in the form of encrypted URL links available to blockchain participants. BIM information is recorded in blockchains, which are persistent; this enables highly secure information sharing among participants. It minimizes BIM information leakage while increasing information exchange securely for all participants.

In this study, we implement BIM information usage throughout the building lifecycle using permissioned blockchain technology. We evaluate our framework by employing the Hyperledger Composer business network based on the Hyperledger Fabric blockchain network. We use a virtual environment with Docker version 1.2.10 and Oracle VM Virtual Box version 6.0. The server runs on a local personal computer with Ubuntu 18.04 with 16 GB RAM and 2.20 GHz. We evaluate the performance of Hyperledger Composer by obtaining response times to create a certain number of assets. We create assets from 100 up to 800, with the result shown in Figure 5.

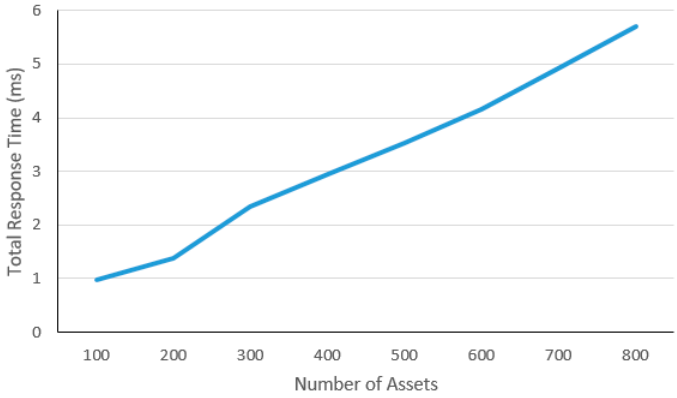

Figure 5. Hyperledger response time vs. number of Assets.

Then, we calculate the average response time of creating an asset on the Hyperledger Composer platform; we obtain the result shown in Figure 6. There is a slight fluctuation of average response times when the number of assets is between 100 and 300, which might be due to network latency. Aside from that, the response time tends to be almost constant for the higher number of assets created.

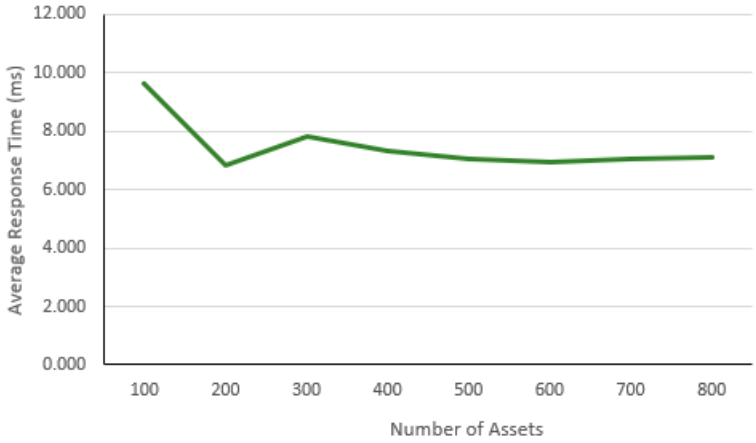

Figure 6. Average Response Time.

In Hyperledger Composer, only registered participants and network admin have access to a blockchain network. This will protect the network from malicious users. Using the decentralized ledger system, participants can easily view each transaction from each phase during a building lifecycle.

5. Conclusions

The purpose of this paper is to explore how BIM information could be included throughout a building lifecycle and exchanged securely by many parties. The following is the main idea of the research:

- Four transactions that correspond to each phase of the building lifecycle (bid, build, maintain, demolish) show that BIM information is always included in each phase. The four transactions are bidBuilding, developBuilding, maintainBuilding, and demolishBuilding.
- For security purposes, BIM information is in the form of encrypted URL links so that involved parties cannot determine the exact location of BIM files.
- Based on experiments to create assets on Hyperledger Composer, the more assets created, the more the average response time tends to be constant.

Author Contributions: Conceptualization, W.N.S. and R.F.S.; methodology, W.N.S.; software, W.N.S.; validation, W.N.S. and R.F.S.; formal analysis, W.N.S.; resources, W.N.S and R.F.S.; data curation, W.N.S.; writing—original draft preparation, W.N.S.; writing—review and editing, R.F.S.; visualization, W.N.S.; supervision, R.F.S. All authors have read and agreed to the published version of the manuscript.

Funding: The authors thank the Ministry of Research and Higher Education of the Republic of Indonesia for financial support for this research under the PTUPT Grant number NKB-2955/UN2.RST/HKP.05.00/2020.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nwodo, M.; Anumba, C.; Asadi, S. BIM-based life cycle assessment and costing of buildings: Current trends and opportunities. *Comput. Civ. Eng.* **2017**, *2017*, 51–59.
2. Abanda, F.H.; Vidalakis, C.; Oti, A.H.; Tah, J.H. A critical analysis of Building Information Modelling systems used in construction projects. *Adv. Eng. Softw.* **2015**, *90*, 183–201. [CrossRef]
3. Marzouk, M.; Azab, S.; Metawie, M. BIM-based approach for optimizing life cycle costs of sustainable buildings. *J. Clean. Prod.* **2018**, *188*, 217–226. [CrossRef]
4. Eadie, R.; Browne, M.; Odeyinka, H.; McKeown, C.; McNiff, S. BIM implementation throughout the UK construction project lifecycle: An analysis. *Autom. Constr.* **2013**, *36*, 145–151. [CrossRef]
5. Cachin, C. Architecture of the hyperledger blockchain fabric. *Workshop Distrib. Cryptocurr. Consens. Ledgers* **2016**, *310*, 4.
6. Rajput, A.R.; Li, Q.; Ahvanooey, M.T.; Masood, I. EACMS: Emergency access control management system for personal health record based on blockchain. *IEEE Access* **2019**, *7*, 84304–84317. [CrossRef]
7. Hasan, H.R.; Salah, K.; Jayaraman, R.; Omar, M.; Yaqoob, I.; Pesic, S.; Taylor, T.; Boscovic, D. A Blockchain-Based Approach for the Creation of Digital Twins. *IEEE Access* **2020**, *8*, 34113–34126. [CrossRef]
8. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference, Porto, Portugal, 23–26 April 2018; pp. 1–15.
9. Hafid, A.; Hafid, A.S.; Samih, M. New Mathematical Model to Analyze Security of Sharding-Based Blockchain Protocols. *IEEE Access* **2019**, *7*, 185447–185457. [CrossRef]
10. Paik, H.-Y.; Xu, X.; Bandara, H.D.; Lee, S.U.; Lo, S.K. Analysis of Data Management in Blockchain-Based Systems: From Architecture to Governance. *IEEE Access* **2019**, *7*, 186091–186107. [CrossRef]
11. De Filippi, P. The interplay between decentralization and privacy: The case of blockchain technologies. *J. Peer Prod.* **2016**, 18–19.
12. Puthal, D.; Malik, N.; Mohanty, S.P.; Kougianos, E.; Das, G. Everything you wanted to know about the blockchain: Its promise, components, processes, and problems. *IEEE Consum. Electron. Mag.* **2018**, *7*, 6–14. [CrossRef]
13. Sinclair, D.; Shahriar, H.; Zhang, C. Security requirement prototyping with hyperledger composer for drug supply chain: A blockchain application. In Proceedings of the 3rd International Conference on Cryptography, Security and Privacy, Kuala Lumpur, Malaysia, 19–21 January 2019; pp. 158–163.
14. Zhang, R.; Xue, R.; Liu, L. Security and privacy on blockchain. *ACM Comput. Surv.* **2019**, *52*, 1–34. [CrossRef]
15. Wang, W.; Hoang, D.T.; Hu, P.; Xiong, Z.; Niyato, D.; Wang, P.; Wen, Y.; Kim, D.I. A survey on consensus mechanisms and mining strategy management in blockchain networks. *IEEE Access* **2019**, *7*, 22328–22370. [CrossRef]
16. Hyperledger Composer. Available online: https://hyperledger.github.io/composer/v0.19/introduction/introduction.html (accessed on 27 May 2020).
17. Hyperledger White Paper Working Group. *An Introduction to Hyperledger*; Linux Foundation: San Fransisco, CA, USA, 2018.
18. Lone, A.H.; Mir, R.N. Forensic-chain: Blockchain based digital forensics chain of custody with PoC in Hyperledger Composer. *Digit. Investig.* **2019**, *28*, 44–55. [CrossRef]
19. Contreras, A.F. Benchmarking of Blockchain Technologies Used in a Decentralized Data Marketplace. Available online: http://oa.upm.es/55775/1/TFG_ALBERTO_FUENTES_CONTRERAS.pdf (accessed on 2 May 2020).
20. Thakkar, P.; Nathan, S.; Viswanathan, B. Performance benchmarking and optimizing hyperledger fabric blockchain platform. In Proceedings of the 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Milwaukee, WI, USA, 25–28 September 2018; pp. 264–276.
21. Benhamouda, F.; Halevi, S.; Halevi, T. Supporting private data on hyperledger fabric with secure multiparty computation. *IBM J. Res. Dev.* **2019**, *63*, 3. [CrossRef]
22. Nasir, Q.; Qasse, I.A.; Talib, M.A.; Nassif, A.B. Performance analysis of hyperledger fabric platforms. *Secur. Commun. Netw.* **2018**, *2018*. [CrossRef]
23. Dhieb, N.; Ghazzai, H.; Besbes, H.; Massoud, Y. A Secure AI-driven Architecture for Automated Insurance Systems: Fraud Detection and Risk Measurement. *IEEE Access* **2020**, *8*, 58546–58558. [CrossRef]
24. Hyperledger. Architecture Explained Read the Docs. 2017. Available online: https://hyperledger-fabric.readthedocs.io/en/release-1.2/archdeep-dive.html (accessed on 3 May 2020).
25. Medvidovic, N.; Rosenblum, D.S.; Redmiles, D.F.; Robbins, J.E. Modeling software architectures in the Unified Modeling Language. *ACM Trans. Softw. Eng. Methodol. TOSEM* **2002**, *11*, 2–57. [CrossRef]
26. Chanda, J.; Kanjilal, A.; Sengupta, S.; Bhattacharya, S. Traceability of requirements and consistency verification of UML use case, activity and Class diagram: A Formal approach. In Proceedings of the 2009 Proceeding of International Conference on Methods and Models in Computer Science (ICM2CS), Delhi, India, 14–15 December 2009; pp. 1–4. [CrossRef]
27. Showkatramani, G.J.; Khatri, N.; Landicho, A.; Layog, D. A Secure Permissioned Blockchain Based System for Trademarks. In Proceedings of the 2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON), Newark, CA, USA, 4–9 April 2019.