Smart Contract: Is it Really Smart in Construction?

Liupengfei Wu(✉), Weisheng Lu, Rui Zhao, and Fan Xue

Department of Real Estate and Construction, University of Hong Kong, Hong Kong, China
u3545425@connect.hku.hk

Abstract. A smart contract is a protocol that can self-execute when predefined conditions are met. This new technology is considered destructive and can transfer the construction industry. In Blockchain 2.0, the combined use of blockchain and smart contracts allows users to express business logic to achieve more advanced transactions. This research aims to critically analyze the challenges, progresses, and benefits of smart contracts in construction through a systematic literature review to address whether it is smart. The findings suggested that numerous progress had been made to address the challenges of smart contracts. Besides, the benefits of smart contracts have attracted the construction industry. The research findings can open the avenue for researchers and construction practitioners to understand the impacts of the salient features of smart contracts and determine appropriate application areas.

Keywords: Smart contracts · Blockchain · Construction industry · Self-execution · Disruptive technology

1 Introduction

In recent years, blockchain has attracted widespread attention from researchers and practitioners. A blockchain is a distributed database with cryptography and endorsement, without a trusted third party [1]. Thus, transactions can be realized cheaply and swiftly. The cryptography of blockchains also ensure trust-building. Therefore, attackers almost impossible to tamper with recorded transactions in blockchains, and all historical transaction records are traceable [1].

Blockchain technology is driving smart contracts, which Nick Szabo initially proposed in the 1990s [2]. A smart contract is a protocol that can self-execute when preset conditions are satisfied. Smart contracts are essentially coupled with blockchains. On the contrary, traditional contracts may need to be completed in a centralized mode by a trusted third party, leading to time-consuming process and high financial costs. The coupled use of blockchain with smart contracts satisfies the fair beliefs of contemporary society, where efficient transactions and trust perform an essential part. However, the construction industry was listed as one of the lowest sectors to have employed engineering informatics during the third industrial revolution [1]. Therefore, comparable to the digital revolution in other industries, there is a puzzle about whether smart contracts are smart in construction.

H. Guo et al. (Eds.): CRIOCM 2021, LNOR, pp. 751–759, 2022.
https://doi.org/10.1007/978-981-19-5256-2_59

Research on smart contracts will help understand the impacts of salient features and determine appropriate application areas. Initially, in Blockchain 1.0, Blockchain was applied as the fundamental technology of cryptocurrency. In Blockchain 2.0, the coupled use of Blockchain and smart contracts enables users to code more advanced business logic to achieve more automated transactions [2]. The advantages of smart contracts include but are not limited to automated processes, high accuracy, trust-free (because there is no third party, so there is no need to trust individuals), and reduced costs [3]. Smart contracts are considered to be disruptive to many global industries including construction [3].

This study discussed the basics of blockchain and smart contracts, building up the research question of whether the smart contract is really smart in construction. The rest of the paper is organized as follows: Sect. 2 provides the research methodology. The next Sect. 3 reviews the basics of blockchain and smart contracts. The Sect. 4 identifies the challenges and progresses of smart contracts. Subsequently, Sect. 5 highlights recognized benefits of smart contracts in the construction industry. Section 6 concludes the study.

2 Research Methodology

The research methodology of this article includes four parts, as shown in Fig. 1. A systematic literature review was adopted, and the limitation was set to filter the publications relevant to blockchain from the Google Scholar. Next, a screening procedure was performed to determine the publications that are entirely concentrated on smart contracts and technological aspects. The literature review was conducted on 15 conference papers and 7 journal papers. Then, the collected and finalized publications were

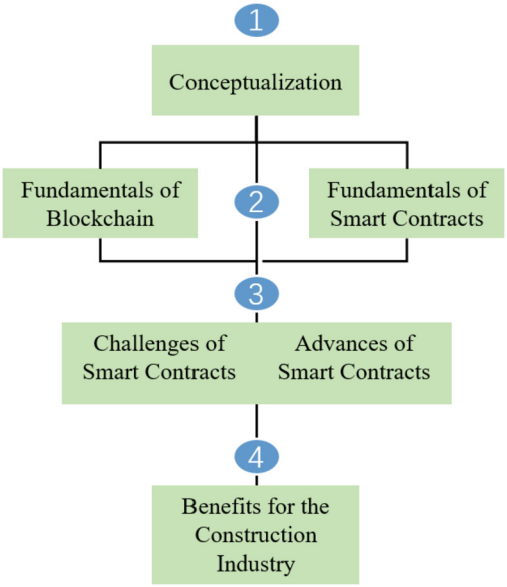

Fig. 1. Research methodology

critically reviewed. A similar review was conducted by [1] to assess the published academic articles on blockchain technology and have determined the application areas of blockchains.

Figure 1 demonstrates that the conceptualization part occurred where finalized publications were grouped into various categories. Next, the basics of blockchain and smart contracts were determined. This progresses to the interpreting of challenges and progresses of smart contracts. After that, an analysis was conducted to determine smart contracts' potential benefits in construction.

3 Literature Review

3.1 Blockchain Basics

A blockchain can be defined as a distributed database in which all transactions are immutable after recording. As shown in Fig. 2 below, a blockchain is an incessantly expanding chain of blocks. The three fundamental technologies that support blockchain functions are cryptography, distributed databases, and consensus mechanisms [1]. The consensus mechanisms are developed to help blockchain network participants endorse the correctness of transactions [3]. Representative algorithms are Proof of Work (PoW) and Crash Fault Tolerance (CFT), and each algorithm has its own merits and shortcomings.

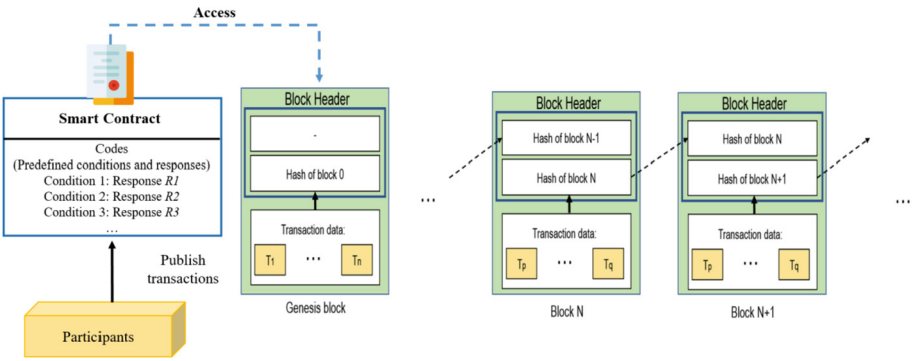

Fig. 2. Smart contract and blockchain

The blockchain database is composed of many ledgers, which are distributed in different places in a shared manner [3]. The distributed database is realized through a decentralized network, in which peer-to-peer transactions can be conducted without the participation of a third party. Blockchain ensures the immutability of transaction data through a hash algorithm [2]. Any block carries the hash of the current block and the hash of the previous block [1]. Therefore, if an attacker wants to tamper with the transaction data of the block, the hash pointers of all the blocks on the chain will also change.

3.2 Smart Contract Basics

Smart contracts can be observed as a major advancement in blockchains. A smart contract is a digital protocol that self-executes the responses when preset conditions are satisfied [2]. As demonstrated in Fig. 2, a smart contract is composed by key two parts, namely preset conditions and responses [3].

Blockchain is the enabler of smart contracts. Smart contracts are combined with blockchains to automatically execute the processes in the blockchain network [2]. The logical links between contract terms are processed in logical flow in the protocols (e.g., if-then statement) [2]. After the contract statement is completed, the smart contract will mark it as a transaction and save it in the blockchain. In addition, smart contracts can ensure reasonable user control and contract self-execution [2]. For example, the project owner and the main contractor agree on financial penalties for breach of the progress contract. If the main contractor violates the contract, the corresponding financial penalties will be automatically deducted from the main contract's deposit.

The lifecycle of a smart contract can be divided into four phases, as shown in Fig. 3. The first phase is initialization. The parties involved must reach an agreement, and the lawyers will help draft the preliminary contract agreement. Then, the software developer will encode the agreement as digital protocols. Each smart contract has to be designed, implemented, and validated. This phase is iterative, because an agreement cannot be reached without a few rounds of negotiation. Besides, this phase involves numerous participants, such as project owners, legal representatives, and software developers.

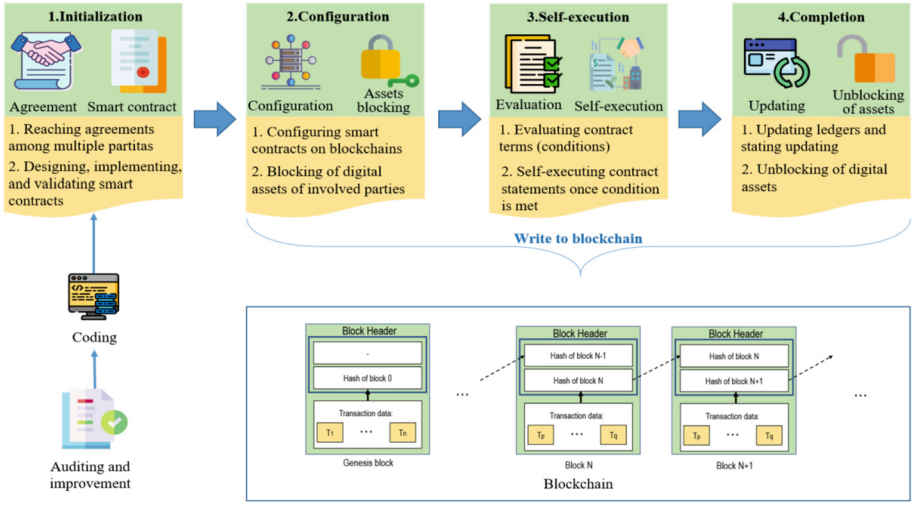

Fig. 3. A lifecycle of a smart contract

The second phase is the configuration of smart contracts. After the participants validate the smart contract, the software developer can configure it to the blockchain platform. Because of its immutability, blockchain can provide a secure environment for smart contracts. Also, the relevant participants' digital assets defined in smart contracts

are blocked by blocking the corresponding e-wallets [2]. The third phase is the self-execution. After the configuration of smart contracts, the agreement conditions must be evaluated. When a preset condition is met, the responses will be carried out automatically. As a result, a transaction will be endorsed and recorded in the blockchains. The last phase is the completion of smart contracts. After execution, the transaction records will be updated in the ledgers of the blockchain, and the digital assets will be transferred to the corresponding parties. Therefore, the digital assets of the participants can be unlocked. Then, the smart contract can complete the lifecycle.

The configuration, self-execution, and completion phases must feed data to blockchains, as demonstrated in Fig. 3. This is because smart contracts are configured on blockchains, and transaction execution and recording are all done in the blockchain.

4 Challenges and Progresses

4.1 Challenges of Smart Contract

Despite the myriad promises of smart contracts, there are still some challenges to be solved. According to the four phases of the smart contract lifecycle, the identified challenges are divided into four categories. An overview of the latest progresses in addressing these challenges is also provided. Table 1 outlines the identified challenges and progresses.

Table 1. Challenges of smart contracts

Phases	Challenges	Progresses
Initialization	Readability	• Recover source code [4] • Human readable code [5] • Human readable execution [6]
	Risk vector	• Re-entry [7] • Block randomness [8] • Overcharging [9]
Configuration	Correctness	• Bytecode analysis [10] • Source code analysis [11] • Machine learning based analysis [12]
	Dynamic control flow	• Graph based analysis [13] • Path-searching [14] • Execution environment [15]
Self-Execution	Smart oracle	• Third-party involved [16] • Reputational incentive mechanism [17]
	Transaction-ordering dependence	• Sequential execution [18] • Predefining contract [19]

(*continued*)

Table 1. (*continued*)

Phases	Challenges	Progresses
	Efficiency	• Execution serialization [20] • Inspection of contract [21]
Completion	Privacy	• Privacy [22]
	Scam	• Ponzi scheme [23] • Honeypot [24]

In the initialization phase, readability may pose a challenge to users. Smart contracts are mainly coded in computer programs using languages such as Go. Then, the software engineers will compile the smart contract. Therefore, the program can have numerous arrangements of codes. Making programs readable in every arrange is one of the existing challenges reported from the collected literature. Also, there are several risk vectors associated with smart contracts. For example, the re-entry issue allows discontinuous functions to be securely called again, and attackers may utilize this imperfection to take digital assets.

In the configuration phase, the correctness of contract remains as a challenge. Due to the blockchains' immutability, a smart contract is immutable after it is configured on blockchains. However, examining the correctness can be a challenge because of the difficulty of forming smart contracts. Smart contracts can interact with each other, so designing a dynamic control process to ensure a reliable execution environment is also a challenge.

In the self-execution phase, how to determine and use the oracle to ensure the authenticity of the information from the off-chain world is a challenge. Moreover, current smart contracts cannot always send transactions to the ordering node to pack them in the correct order. When configuring a large number of smart contracts on the platform, it is also a challenge to ensure the efficiency of the smart contracts running at the same time.

In the last phase, ensuring privacy can be a challenge. Existing blockchain solutions are lack of consideration for privacy, as they report all recorded transactions to network participants. Therefore, anyone in the network can use smart contracts to invoke private data. As a novel technology, smart contracts are also exposed scams.

4.2 Progresses of Smart Contract

Recent progresses in smart contracts are summarized in Table 1 above. The latest progresses in readability challenges include source code recovery, human-readable code, and human-readable execution. [4] proposed a reverse engineering tool so that the hex-encoded contract can be converted into human-readable pseudo-codes. [5] demonstrated an automatic analysis system that can turn the human-readable agreement into programmable programs. [6] showed an intermediate level language to offer compliers with high-level information. Progresses in minimizing the risk vectors have also been found in the literature. For example, using named states allows consistent checks for condition transitions and verification, thereby minimizing re-entry issues [7], applying

delay function to produce block randomness [8], and adopting GasReduce to detect gas-costly patterns [9]. During the configuration phase, bytecode [10], source code [11] and machine learning-based analysis [12] are discovered to ensure the correctness of the contracts. Graph-based analysis [13], path-searching [14], and execution environment [15] are measures to solve dynamic control flow vulnerabilities.

Recent progresses for oracle include using third-party to scrape data from a reliable source and feed those data to smart contracts [16]. A reputational-based incentive mechanism was also found for solving the oracle issue [17]. Sequential execution [18] and predefining contract [19] are two progresses in Transaction-ordering dependence. [20] used execution serialization (a method based on Software Transactional Memory) to run smart contracts concurrently. [21] demonstrated a method named "Inspection of contract" to allow users to revise initial smart contracts without redeploying them. At the completion stage, [22] demonstrated a decentralized smart contract system with a privacy protection mechanism. Progresses have also been made for detecting scams related to the Ponzi scheme [23] and Honeypot [24].

5 Benefits of Using Smart Contracts in Construction

Smart contracts have a wide range of potential benefits in construction [25]. Firstly, smart contracts can bring accuracy to the construction industry. If the terms and conditions of construction contracts are accurately written on smart contracts, the execution and supervision of the conditions will be very accurate. Secondly, smart contracts can help enhance transparency. Every payment, transaction, interaction, and execution can be coded on smart contracts, making the construction- related processes transparent. Thirdly, smart contracts can help risk management. Self-executing smart contracts can reduce the complexity of construction procurement, thereby minimizing the risk of delayed payment and reducing disputes. Fourthly, smart contracts can facilitate compliance checks. Combined with construction standards, smart contracts can help stakeholders automatically check compliance. Finally, smart contracts can reduce construction costs by eliminating middlemen and administrators in certain processes.

6 Conclusions

Introduction of Blockchain 2.0 formed a trend that fascinates construction stakeholders to use smart contracts due to its potential benefits. A smart contract is a protocol that can self-execute when predefined conditions are met. This paper critically reviews the extensive existing literature on smart contracts and their challenges, progresses, and benefits in the construction industry. The literature findings indicated that the smart contract is smart as the next disruptive technology in construction. The results of this study can help researchers and construction practitioners understand the impact of the distinctive features of smart contracts and determine appropriate application areas.

References

1. Perera, S., Nanayakkara, S., Rodrigo, M.N.N., Senaratne, S., Weinand, R.: Blockchain technology: is it hype or real in the construction industry? J. Indust. Inf. Integr. 100125 (2020)
2. Zheng, Z., et al.: An overview on smart contracts: Challenges, progresses and platforms. Futur. Gener. Comput. Syst. **105**, 475–491 (2020)
3. Ahmadisheykhsarmast, S., Sonmez, R.: A smart contract system for security of payment of construction contracts. Autom. Constr. **120**, 103401 (2020)
4. Zhou, Y., Kumar, D., Bakshi, S., Mason, J., Miller, A., Bailey, M.: Erays: Reverse engineering ethereum's opaque smart contracts. In: 27th {USENIX} Security Symposium {USENIX} Security 2018, pp. 1371–1385 (2018)
5. Frantz, C.K., Nowostawski, M.: From institutions to code: towards auto- mated generation of smart contracts. In: Proceedings of IEEE International Workshops on Foundations and Applications of Self Systems, pp. 210–215. IEEE (2016)
6. Lattner, C., LLVM, V.: A compilation framework for lifelong program analysis & transformation. In: Proceedings of the International Symposium on Code Generation and Optimization: Feedback-Directed and Runtime Optimization, vol. 75. IEEE Computer Society (2004)
7. Liu, C., Liu, H., Cao, Z., Chen, Z., Chen, B., ReGuard, B.: Finding reentrancy bugs in smart contracts. In: Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings, pp. 65–68. ACM (2018)
8. Bünz, B., Goldfeder, S., Bonneau, J.: Proofs-of-delay and randomness beacons in ethereum. In: IEEE Security and Privacy on the blockchain (IEEE S&B) (2017)
9. Chen, T., Li, X., Luo, X., Zhang, X.: Under-optimized smart contracts devour your money. In: Proceedings of 24th International Conference on Software Analysis, Evolution and Reengineering, SANER, pp. 442–446 (2017)
10. Brent, L., et al.: Vandal: A scalable security analysis framework for smart contracts (2018). arXiv:1809.03981
11. Dhawan, M.: Analyzing safety of smart contracts. In: Proceedings of the Conference: Network and Distributed System Security Symposium, San Diego, CA, USA, pp. 16–17 (2017)
12. Tann, W.J.W., Han, X.J., Gupta, S.S., Ong, Y.S.: Towards safer smart contracts: A sequence learning approach to detecting security threats (2018). arXiv:1811.06632
13. Fröwis, M., Böhme, R.: In code we trust? In: GarciaAlfaro, J., NavarroArribas, G., Hartenstein, H., HerreraJoancomartí, J. (eds.) ESORICS/DPM/CBT -2017. LNCS, vol. 10436, pp. 357–372. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67816-0_20
14. Nikolić, I., Kolluri, A., Sergey, I., Saxena, P., Hobor, A.: Finding the greedy, prodigal, and suicidal contracts at scale. In: Proceedings of the 34th Annual Computer Security Applications Conference, pp. 653–663. ACM (2018)
15. Fu, Y., Ren, M., Ma, F., Jiang, Y., Shi, H., Sun, J.: EVMFUZZ: differential fuzz testing of Ethereum virtual machine (2019). arXiv:1903.08483
16. Zhang, F., Cecchetti, E., Croman, K., Juels, A., Shi, E.: Town crier: an authenticated data feed for smart contracts. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 270–282. ACM (2016)
17. Huang, C., et al.: Repchain: a reputation based secure, fast and high incentive blockchain system via sharding. IEEE Internet Things J. **8**(6), 4291–4304 (2020)
18. Mavridou, A., Laszka, A.: Designing secure Ethereum smart contracts: a finite state machine based approach. In: Meiklejohn, S., Sako, K. (eds.) FC 2018. LNCS, vol. 10957, pp. 523–540. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-662-58387-6_28
19. Natoli, C., Gramoli, V.: The blockchain anomaly. In: 2016 IEEE 15th International Symposium on Network Computing and Applications (NCA), pp. 310–317. IEEE (2016)

20. Dickerson, T., Gazzillo, P., Herlihy, M., Koskinen, E.: Adding concurrency to smart contracts. Distrib. Comput. **33**(3–4), 209–225 (2019). https://doi.org/10.1007/s00446-019-00357-z
21. Bragagnolo, S., Rocha, H., Denker, M., Ducasse, S.: SmartInspect: solidity smart contract inspector. In: 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE), pp. 9–18. IEEE (2018)
22. Kosba, A., Miller, A., Shi, E., Wen, Z., Papamanthou, C.: Hawk: the blockchain model of cryptography and privacy-preserving smart contracts. In: 2016 IEEE Symposium on Security and Privacy (SP), pp. 839–858. IEEE (2016)
23. Bartoletti, M., Carta, S., Cimoli, T., Saia, R.: Dissecting Ponzi schemes on Ethereum: identification, analysis, and impact. Futur. Gener. Comput. Syst. **102**, 259–277 (2020)
24. Torres, C.F., Steichen, M.: The art of the scam: demystifying honeypots in Ethereum smart contracts. In: 28th {USENIX} Security Symposium ({USENIX} Security 19). King Abdullah University of Science and Technology, 1591–1607 (2019)
25. Penzes, B., Kirkup, A., Gage, C., Dravai, T., Colmer, M.: Blockchain technology in the construction industry: Digital Transformation for High Productivity. Institution of Civil Engineers, London, UK (2018)