Smart contracting in the execution phase – asset tracking in construction using blockchain technology

Wouter van Groesen, w.v.groesen@tue.nl
Eindhoven University of Technology, Eindhoven, The Netherlands

Pieter Pauwels, p.pauwels@tue.nl
Eindhoven University of Technology, Eindhoven, The Netherlands

Abstract

Construction supply chains (CSC) are converging chains in which intransparency, risks and trust issues between parties occur. To increase transparency and efficiency by means of automation, physical asset tracking (PAT), digital asset management (DAM) and distributed ledger technology (DLT) can be combined. We therefore investigate here to what extent geospatial sensors, a jointly managed information system, a private permissioned Ethereum blockchain network and several smart contracts can be combined to aid towards this purpose. A framework is developed and incremented into several interacting prototypes. These prototypes consist of: (1) a QR-code mobile application for asset tracking, (2) a Revit model for parameter centralization, (3) several smart contracts deployed on the Ethereum Goërli network and (4) a dashboard WebApp. Four distinct test scenarios are developed to verify the functionalities of these applications. Test results indicated achievement of desired post-conditions. Hence, (semi) automation of compliance checking activities and tokenized payments is realized.

Keywords: Asset tracking, distributed ledger technology, blockchain, smart contracts, automation.

1 Introduction

The construction industry is considered to be a traditional industry in which multi-party collaborations work together towards the realization of a construction project. In order to do so, construction supply chains are established which encompasses a client, main contractor and a significant number of suppliers. Construction supply chains are: (1) converging in nature, (2) make to order chains in which the client is involved during the entire process, (3) fragmented and (4) temporary (Vrijhoef and Koskela 2000; Behera & Mohanty 2015). Due to divergent goals and mutual interdependencies, risks and trust issues between parties occur. To manage trust issues and hedge risks, collaborations are governed by means of contracts. In order to check whether contracted parties adhere to contract obligations, contracting parties conduct manual compliance checking activities (e.g. checking compliance to planning – 4D). As contracting and contracted parties conduct these compliance checking activities manually and individually, these activities are considered to be inefficient (time-consuming), intransparent and sensitive to errors. Intransparency of compliance checking activities increments the risk of conflict escalation between parties. The risk of conflict escalation is further increased by the chain payment system which is inherently related to the diverging nature of construction supply chains (Ashworth & Perera 2018; Danuri et al 2006). To increase the efficiency and transparency of compliance checking activities and payments in the execution phase, these activities have to be (semi) automated and managed jointly. The combination of physical asset tracking (PAT), digital asset management (DAM) and distributed ledger technology (DLT) as depicted in **Figure 1** can aid towards this purpose (Belle 2017; Heiskanen 2017; Luo et al 2019; Mason 2017; Mason 2019).

In this article, we report research on current practice of PAT and DAM. The state of the art technologies attributed to PAT, DAM and DLT are presented in Section 2. Based on a comparison between current practice and state of the art technologies, we propose a framework in which we combine PAT, DAM and DLT (Section 3). The developed framework is divided into several individual static prototypes which are subsequently incremented to dynamic prototypes. The development process and created prototypes are presented in Section 4. After creation of individual dynamic prototypes, these prototypes are further developed for mutual interaction to enable (semi) automation of compliance checking activities and payments. These prototypes and the interaction between them are later tested in four distinct scenarios based on input from practice to determine their functionality as presented in Section 5, after which a conclusion follows in Section 6.

2 Asset management & tracking

An asset is an item, thing or entity which is of potential or actual value to an organization. Physical assets are considered to be tangible assets whereas digital assets are intangible non-physical assets (Hastings 2015; Ma et al 2014). Digital assets (e.g. design of a wall) can be interrelated to physical assets (the physical wall). Critical assets are considered to be assets which are of significant impact on the achievement of goals for an organization (Braaksma 2016; Guillen et al 2016). During the construction phase of a project, activities related to prefabricated elements can be of high importance due to their presence on the critical path of an execution planning. Prefabricated elements can thus be considered as performance-critical assets during the construction phase (Mason 2019).

2.1 Digital Asset Management

The usability of a BIM model in varying phases is expressed by the level of model development (LOMD). Digital assets which are represented in BIM models as an information system are transferred into physical reality during the construction phase (Vijayeta 2019). Therefore, the transfer from “as-planned” LOMD 400 (including planning & cost data) to “as-built” LOMD 500, is initiated during this phase. To advance towards an updated “as-built” model and thus a (near) real-time update of a digital environment with data from the physical environment, sensor technology should be implemented to efficiently construct a digital twin. Although BIM tools can be considered as a possibility to advance towards the construction and operation of a digital twin, implementational barriers persist in combining this data with IoT data. Development of alternatives to achieve the same goal (e.g. DLT or databases) are available and promising (Lu et al 2019; Boje et al 2020; Grieves 2014; Pučko et al 2017; Shojaei et al 2019).

2.2 Physical Asset Tracking

To (semi) automate physical asset tracking: (1) geospatial (e.g. GPS or RFID chips) and/or (2) LADAR technology (e.g. point cloud) can be used. Geospatial technologies are most suitable for the intended purpose as they enable the collection of data both on and off-site. Thus, it enables the acquisition of data throughout the supply chain to measure the current state of physical assets (e.g. manufactured, delivered and assembled). Most geospatial technologies with the exception of GPS require the operation of a manual scanning procedure (QR codes, barcodes, etc.). Thus, semi-automation of data acquisition can be realized. Although geospatial technologies are suitable for the acquisition of data throughout supply chain stages, it would be undesirable and infeasibly to apply this technology to all construction components (e.g. a tag/sensor for every brick). Therefore, it is argued that geospatial technology is best suited for the tracking of large prefabricated elements combined with LADAR for smaller components. On-site progress tracking by means of LADAR can be exemplified by the comparison of point clouds (resembling the as-built state) with an “as-planned” BIM model (Kopsida et al 2015; Braaksma 2016; Lanko et al 2018; Omar and Nehdi 2016).

2.3 Digital Twin

A digital twin in any form can aid to enable (near) real-time communication between the digital and the physical environment as depicted in **Figure 2**. Contract obligations (planning and costs) should be included in an information system to establish the as-planned state (e.g. LOMD 400 BIM model). Data on the status of physical assets needs to be acquired, communicated and processed into the digital environment while physical activities are conducted (Boje et al 2015; Lu et al 2019; Braaksma 2016). Automated data analysis is required to determine whether a physical asset is compliant or non-compliant to the as-planned state of the twin elements in the digital environment. The result of such a comparison in the digital environment should either result into a reward (financially) or communication (notification) of non-compliance. Corrective actions in the physical environment can be taken based on these notifications of non-compliance. Solutions which enable the comparison between the "as-built" and "as-planned" state and the provision of the appropriate notifications are not readily available. Distributed ledger technology and smart contracts can aid towards this purpose (Belle 2017; Heiskanen 2017; Luo et al 2019; Li et al 2019a-c; Shojaei et al 2019).

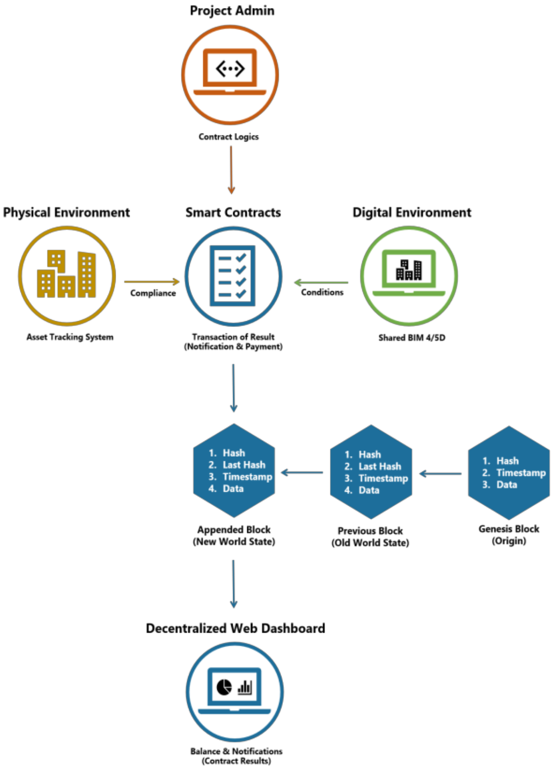


Figure 1. Simplified SC Framework

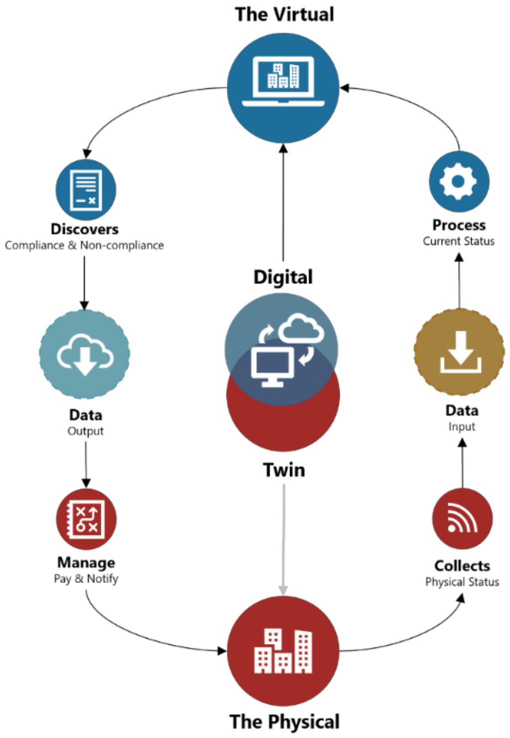


Figure 2. Digital Twin

2.4 Distributed ledger technology

DLT can be defined as a consensus of replicated, shared, and synchronized digital data, geographically spread across multiple sites, countries, or institutions where there is no single entity in control (Scardovi 2016). Nawari et al. (2019) defines distributed ledger technology (DLT) as a digitized, decentralized public ledger of data, assets and all pertinent transactions that have been executed and shared among participants in the network (Nawari et al 2019). DLTs consist of a chain of blocks which contain information and are encrypted through a cryptographic hash function provided by an algorithm. Transactions representing value are grouped within these blocks, are verified and validated through a consensus mechanism within a distributed peer-to-peer network (Turk & Klinc, 2017; Nanayakkara et al 2019). Due to the immutability of the DLT, transactions within the system are secure and can aid to reduce trust issues (Li et al 2019a-c; Belle 2017; Lanko et al 2018). Nakamoto (2019) adds that DLT technology provides certainty marked by complete consensus, provenance, finality, and immutability (Nakamoto

2019). Smart contracts are self-executing contracts which programmed instruction embedded in its code (Szabo 1997; Mason 2017; Mason 2019; Shojaei et al 2019; Wang et al 2017; Kinnaird and Geipel 2017). The Ethereum network is one of the blockchain networks which enables smart contract deployment whereas some other networks do not.

The 'content' of the transaction encoded in a block chain is near to always money (financial transactions). Although it might be feasible to encode transactions of 'information' in a blockchain (e.g. a log of construction progress, without financial transactions), DLT is seldom to never used for that purpose, and it is also rather unfit for that goal. Therefore, also here, we focus eventually on enabling financial transactions, based on confirmation of construction progress.

2.4.1 Ethereum Blockchain

Besides the hardware node on which an Ethereum client runs, the layers of the Ethereum (ETH) blockchain network consist of: (1) infrastructure, (2) data, (3) consensus, (4) network and (5) application. The infrastructure layer consists of nodes (one for each supply chain party) organized in a peer-to-peer (P2P) network. All nodes share a single distributed and decentralized ledger. Each node is related to an externally owned account (EOA) with an unique account address. Data related to each individual address (e.g. balance) is stored in separate account state tries (tree-like structures), which are merged in a single world state trie (Ferretti and D'Angelo 2020; Wood 2014; Baliga 2017). Subsequently, the world state trie at a specific point in time is stored in blocks. These blocks are chained by means of a hashing algorithm which provides immutability and security to the shared ledger. Hence, a single source of truth is created. As consensus between nodes needs to be reached in order to append blocks to the shared ledger, a consensus algorithm is utilized. The Proof of Authority (PoA) consensus algorithm is best suited for built environment collaborations, as power is equalized between nodes independently of node computing power or stake sizes. The consortium-based permission model is the P2P network structure which is best applicable in the built environment. This structure allows all nodes to view transactions and commit transactions to blocks. (Ferretti and D'Angelo 2020; Antonopoulos and Wood 2018; Hileman and Rauchs 2017; Acharya et al 2019).

2.4.2 Application layer

The final layer of the ETH blockchain infrastructure is the application layer, by means of which users interact with the distributed ledger. The application layer is divided between: (5) the execution layer, (6) the oracle layer and (7) the user interface layer. The smart contracts can compare the planned to the real state by means of computations, and they reside within the execution layer. Smart contracts are developed, compiled and deployed subsequently on the blockchain by an EOA. EOAs are able to interact (input values) by means of pre-signed transactions with contract functions (Wood 2014). These transactions can be scheduled for execution in the future by means of the Ethereum Alarm Clock[1]. Based on these inputs and the smart contract computation code, (semi) automated result calculation is realized (payments & notifications). Because blockchains depend on deterministic information and thus requires the attainment of consensus to successfully execute transactions, API integration external to the blockchain is not supported. Because smart contracts in the intended use case requires external "as-built" data, an oracle service is used. An oracle enables integration with external API services by mining and importing the required data into smart contract by means of callbacks (Lo et al 2020). Data on contract parameters (planned date & reward value) and physical asset states are stored in a jointly managed online database to enable the extraction of information by the oracle service. To enable supply chain parties to acquire and process this data into a database, a PAT and contract parameter application have to be utilized. Finally, a dashboard can be used to enable communication of compliance, non-compliance and payments from the smart contract to supply chain parties.

---

[1] https://www.ethereum-alarm-clock.com/

3 Framework & functionalities

In Section 1 we discussed previous work on the combination of the physical and digital environment by means of blockchain. In a simplified framework which is presented in **Figure 1,** we propose a connection between both environments through a smart contract to enable leverage of the advantages of blockchain technology. In Section 2 we have reviewed the state of the art technologies for PAT, DAM and explored DLT. In **Figure 3** we show how we combine a PAT and DAM application with smart contracts on a blockchain. The intended purpose of applying these applications and DLT together, is to enable automated compliance checking by smart contracts. Therefore, the following functionalities should be met: (1) applications enable migration of data onto and from the blockchain, (2) applications have to be usable in multiple industry-based scenario's, (3) smart contracts have to generate a log of activities on the blockchain and (4) a smart contract has to distribute tokens (money) accordingly without human interference.

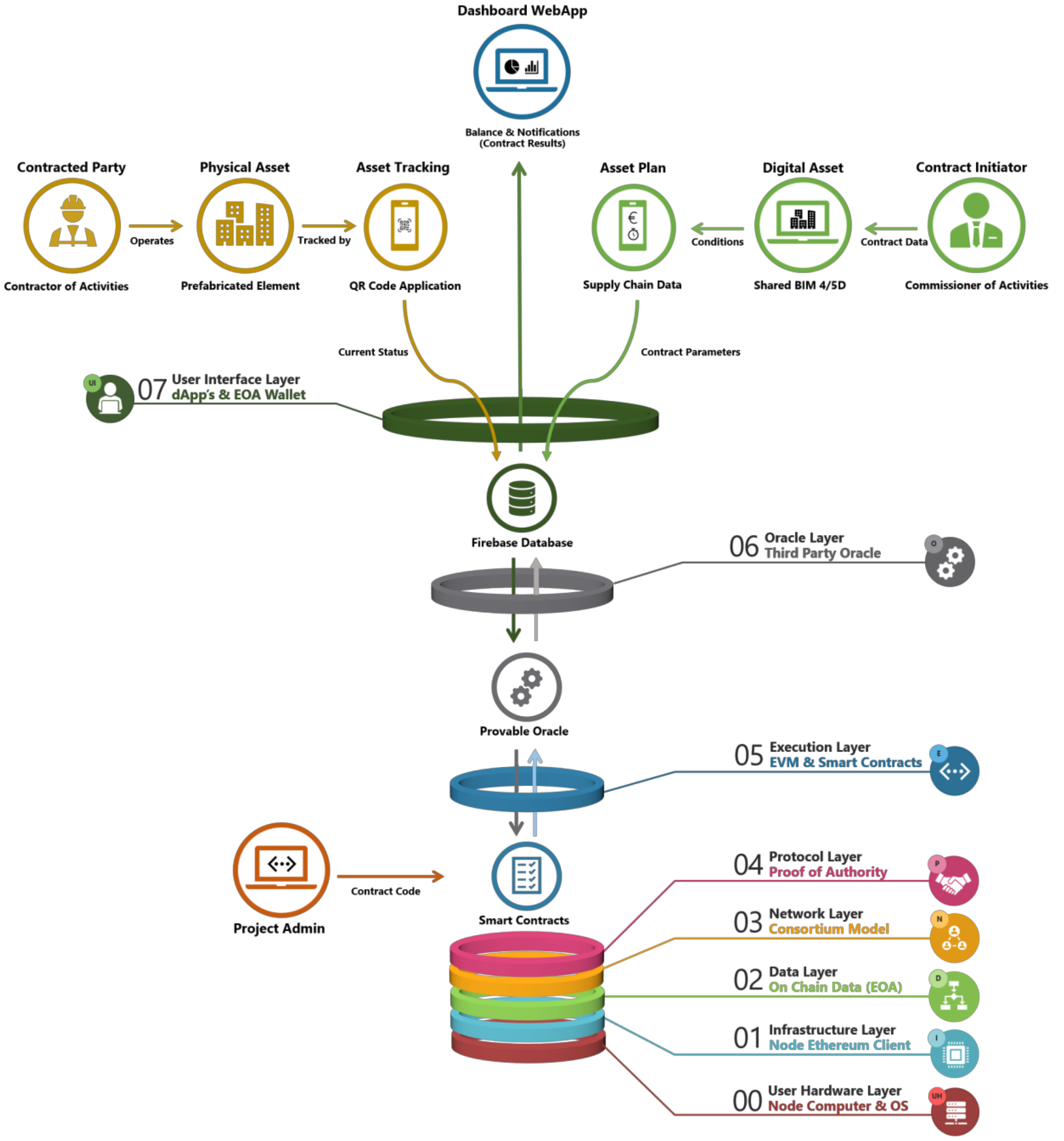


Figure 3. Proposed applications & relations

4 Application Development

To test the combination of the applications which are proposed in Figure 3 and required functionalities, individual components were first developed into individual static prototypes after which dynamic prototypes were developed. After applications were sufficiently tested (Section 5), individual dynamic prototypes were further developed for mutual interaction to enable (semi) automation of compliance checking activities and payments. Finally the developed prototypes and the interaction between them were tested in four distinct scenarios to verify the required functionalities. We thus followed an incremental prototype development (IPD) methodology, as is commonly used for agile software development.

4.1 Asset tracking application

To enable the acquisition of physical asset data, a fully functioning mobile QR-code application was developed. In order to so, a static UI storyboard of the application was created which was internally verified. After sufficient verification and various design iterations, the QR application was coded using the MIT App Inventor application[2]. The dynamic application enabled the capturing of physical asset states by means of scanning asset GUIDs embedded in QR-codes. The PAT application was designed to be used by: (1) manufacturers, (2) sub-contractors, (3) main contractors and (4) clients. Users are enabled to: (1) login, (2) select the desired asset state, (3) scan the assets' QR-code and (4) review the new status of the asset. If non-compliance to quality-related obligations was identified by an actor: (1) selection of a rejection status, (2) capturing of the defect and (3) sharing of the defect was enabled. The UI of the developed PAT application and described functionalities are depicted in **Figure 4**.

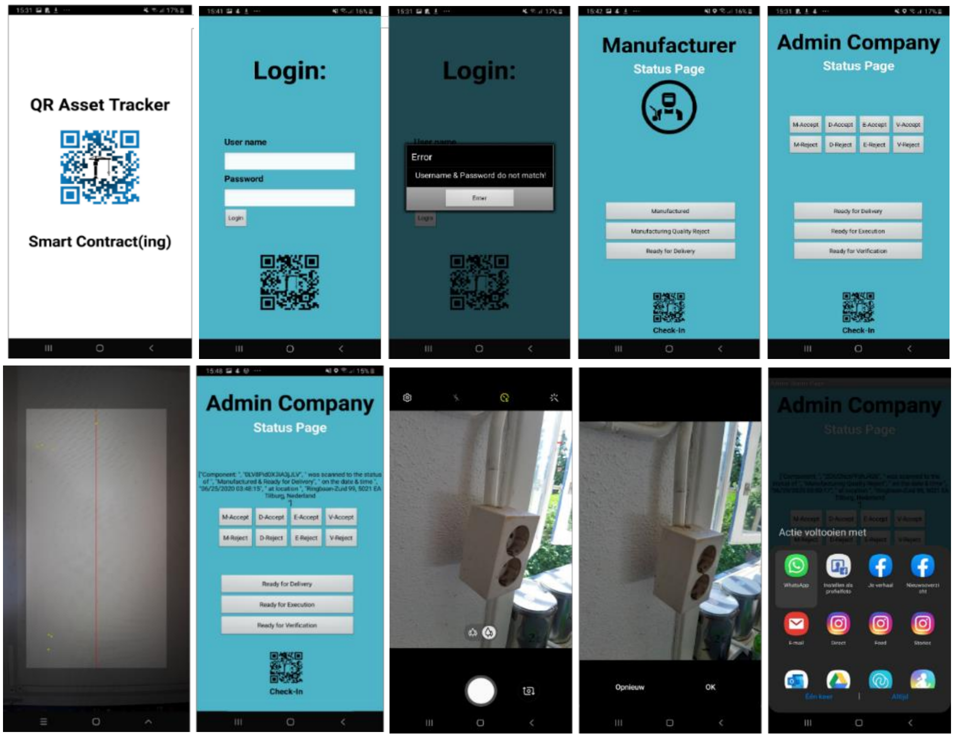


Figure 4. User Interface (UI) of the mobile asset tracking application

4.2 Digital asset plan

To enable comparison of physical asset states with contract obligations by means of a smart contract, the contract has to be provided with parameters. A jointly managed Revit model was used to centralize contract parameters by all contracting supply chain parties. The required contract parameters consist of: (1) planned date for every supply chain step, (2) monetary

---

[2] https://appinventor.mit.edu/

rewards for successful completion of each supply chain step and (3) all actor EOA addresses on the blockchain. These properties were added to objects (assets) in the Revit model which enabled the input and thus centralization of contract parameters .

4.3 Firebase database

Based on the selected status and scanned QR-code, an asset's GUID and corresponding status are directly accumulated in several Google Firebase databases[3]. These databases consist of: (1) a history log which contains a record of all previous asset states and (2) a status log which contains the current state only. The transfer of accumulated contract parameters from Revit to Firebase is conducted by means of: (1) a Revit schedule, (2) a CSV file and (3) a JSON converter. Manually uploading the converted JSON file in the Firebase database results in an obligation log. This routine via a firebase database was chosen to eventually allow direct web-based updates between BIM model and the distributed ledger all in real time, thus avoiding manual file exports and imports. Such a manual workflow would of course defeat the purpose of using a DLT workflow.

4.4 Oracle service

The Provable Oracle Service was utilized due to a dependency of smart contracts on data which resided on the same blockchain as the contract does. Physical asset data and contract parameters were extractable from the Firebase logs by means of an Oracle in a smart contract. The Provable Oracle Service and Firebase URLs were embedded into a smart contract which was coded and compiled in the Ethereum Remix environment. To retrieve data attributed to a specific asset, the asset's GUID was appended to the Firebase URL in the smart contract (URL parameter). Functions which are embedded in the smart contract successfully enable the retrieval and import of all desired data into the DLT environment.

4.5 Smart contract

A second smart contract is coded in the Solidity language on the Ethereum Remix platform which contained the computational capabilities in its code required to compare the "as-planned" state of assets to the "as-built" state. EOAs are created for each actor through Metamask[4]. Due to the utilization of the MetaMask application as an indirect node client, the installation of a full Ethereum node is not required. Based on the (semi) automated comparison between both states, the smart contract identifies: (1) compliance, (2) quality issues or (3) planning-related non-compliance. The smart contract: (1) rewards a specific party with tokens and (2) emits a notification of success if compliance is identified. If non-compliance or quality issues are identified, the smart contract does not transfer tokens and emits an error message.

4.6 Distributed ledger

Transactions with smart contracts by EOAs are registered on an operational blockchain because the development of smart contracts in the Ethereum Remix environment[5] and deployment on the Ethereum Goërli test net. Therefore, transactions with smart contracts, computation results and smart contract notifications are eligible for review on the Goërli Etherscan webpage[6]. An example of the event log on Etherscan which relates to one of the deployed smart contracts is available online[7].

4.7 Dashboard WebApp

Finally, to make the transaction log and information in the blockchain accessible to end users (AEC stakeholders), a dashboard WebApp is constructed and published online to provide insight into smart contract execution results[8]. The dashboard WebApp consists of: (1) a front-end user

---

[3] https://firebase.google.com/

[4] https://metamask.io

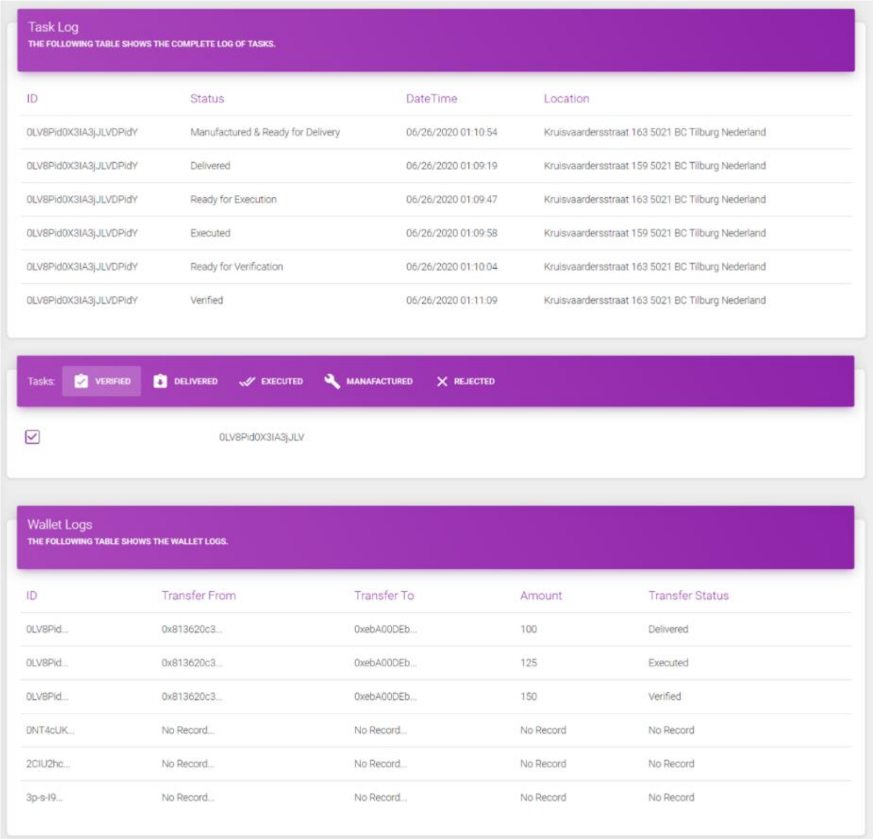
[5] https://remix.ethereum.org/

[6] https://etherscan.io/

[7] https://goerli.etherscan.io/address/0x185ca833440b4a71341fd0b3b3acecb2f420a83e

[8] www.smartcontracting.xyz

interface (2) a back-end which contains computation capabilities similar to the developed smart contracts and (3) a controller which enables interaction between the UI and back-end. Based on the extraction and uploading of JSON files from all Firebase databases, the dashboard displays: (1) a history log, (2) all current element states and (3) an overview of token transfers as depicted in **Figure 5**.



Task Log
THE FOLLOWING TABLE SHOWS THE COMPLETE LOG OF TASKS.

ID	Status	DateTime	Location
0LV8Pid0X3IA3jJLVDPidY	Manufactured & Ready for Delivery	06/26/2020 01:10:54	Kruisvaardersstraat 163 5021 BC Tilburg Nederland
0LV8Pid0X3IA3jJLVDPidY	Delivered	06/26/2020 01:09:19	Kruisvaardersstraat 159 5021 BC Tilburg Nederland
0LV8Pid0X3IA3jJLVDPidY	Ready for Execution	06/26/2020 01:09:47	Kruisvaardersstraat 163 5021 BC Tilburg Nederland
0LV8Pid0X3IA3jJLVDPidY	Executed	06/26/2020 01:09:58	Kruisvaardersstraat 159 5021 BC Tilburg Nederland
0LV8Pid0X3IA3jJLVDPidY	Ready for Verification	06/26/2020 01:10:04	Kruisvaardersstraat 163 5021 BC Tilburg Nederland
0LV8Pid0X3IA3jJLVDPidY	Verified	06/26/2020 01:11:09	Kruisvaardersstraat 163 5021 BC Tilburg Nederland

☑ 0LV8Pid0X3IA3jJLV

Wallet Logs
THE FOLLOWING TABLE SHOWS THE WALLET LOGS.

ID	Transfer From	Transfer To	Amount	Transfer Status
0LV8Pid...	0x813620c3...	0xebA00DEb...	100	Delivered
0LV8Pid...	0x813620c3...	0xebA00DEb...	125	Executed
0LV8Pid...	0x813620c3...	0xebA00DEb...	150	Verified
0NT4cUK...	No Record...	No Record...	No Record	No Record
2CIU2hc...	No Record...	No Record...	No Record	No Record
3p-s-I9...	No Record...	No Record...	No Record	No Record

Figure 5. Populated WebApp dashboard

5 Application testing

To test all individual applications and the interaction between them, several use case test scenarios are drafted. The scenarios are based on actual external PAT pilot data provided by the Royal VolkerWessels company and are described hereafter.

1. Full compliance and thus successful delivery, execution and verification of an asset
2. Non-compliance due to violation of planning-related contract parameters (late delivery)
3. Non-compliance due to violation of quality-related contract parameters (quality reject)
4. Incorrect use of the PAT application and thus a missing status (asset not scanned)

To test each scenario, a fictive design is constructed which contains four prefabricated assets Each asset is attributed with a GUID, contract parameters and a QR-code. Furthermore, four smart contracts are developed which each related to a specific asset. In order to test whether desired results are yielded in each scenario, pre-conditions and post conditions are drafted. These conditions regard the token (money) balance of each actor before and after the execution of all scenarios. During the test: (1) the client EOA possesses all tokens, transacts them into the smart contract and inputs contract parameters. (2) all other actors update physical asset states (e.g. asset delivered) by means of the PAT application, (3) this data is transacted into a smart contract to simulate its behavior in each scenario and (4) after smart contract execution, data is extracted from the chain and uploaded into the dashboard environment.

After the test was executed, the correct amount of tokens was transferred from the client to the manufacturer, sub-contractor and main contractor EOAs. The client EOA did not possess any tokens (money) in the post-conditions. Considering the desired functionalities in Section 3, we found that: (1) external data was successfully migrated into and from smart contracts by means of the developed applications in 3/4 scenarios, (2) the applications permitted usage in multiple practice-based scenarios, (3) an immutable activity log on the blockchain was created by smart contracts and (4) tokens were distributed to the correct parties in 3/4 scenarios without human interference. In the fourth scenario, the WebApp dashboard was not able to cope with a missing asset state (functionality 1). In the second scenario, the manual input of new contract parameters in a smart contract was required after non-compliance (functionality 4).

6 Conclusion & discussion

The yielded results indicate that the acquisition and processing of physical asset data was (semi) automated by means of the PAT application. The comparison of the "as-planned" and "as-built" state including related token transfers and notifications were (semi) automated by means of a smart contract as well. An immutable distributed record of transactions (asset state changes & payments) was created on a blockchain network. The communication of contract parameters into the smart contract was simplified in comparison to current practice. However, the activities which involve the communication of contract parameters into smart contracts allow for significant optimizations. The dashboard WebApp correctly provided insight into assets and contract states, with the exception of the fourth scenario. Hence, increased transparency as well as a decrease of risks and conflicts between parties can be expected. Although (semi) automation was achieved, several components and actions to trigger smart contract execution were identified to be suitable for (semi) automation in the future, which requires a stronger distributed web development approach (real-time communication). Examples of these components are the interactions with smart contract functions by EOA's, which can be scheduled in the future with the Ethereum Alarm Clock.

Previous research regarding DLT in the built environment mainly focused on framework creation, on-site activities, the proposal of a construction log and to some extent contributes to (semi) automation of contract execution (Li et al 2019c; Luo et al 2019; Shojaei et al 2019). The constructed framework and developed prototypes provide a plug and play concept which aids towards increased (semi) automation of compliance checking activities and payments. We developed prototypes and enabled the combination of PAT, DAM and DLT, which can be considered as an essential basis for full automation in the future. We provided insight into the state of the art regarding combining PAT, DAM and DLT. Furthermore, we explored and successfully demonstrated joint application of these technologies in the execution phase of a construction project on the Ethereum network.

For future research four main directions are identified and described hereafter.

1. The developed PAT should be further optimized and further practical verification of its use should be sought. We intend to combine this work with geospatial (RFID) and laser technologies (point cloud) in the future to improve use on construction sites.
2. The conversion process of contract parameters from a Revit model to smart contracts is observed to be inefficient. In future work we intend to optimize this process through development of a contract parameter application.
3. The workflow which is required to efficiently operate the combination of applications requires further automation and simplification. Development of a contract management application, appropriate procedures and direct (real-time) web-based connections are required in future research.
4. To eliminate the requirement of external data and to directly incorporate this data on the blockchain, future research should focus on development of decentralized applications (DApps) which directly reside on the blockchain. Furthermore, exploration of other blockchain networks (e.g. Hyperledger) for construction use-cases should be executed.

Acknowledgements

The authors of this article would like to express their sincere thanks to the Royal VolkerWessels company in the Netherlands for their participation in the expert panel, provision of practical data to construct the required use cases, and verification of results of test-case execution.

References

Acharya, V., Yerrapati, A. E., & Prakash, N. (2019). *Oracle Blockchain Quick Start Guide: A practical approach to implementing blockchain in your enterprise*. Packt Publishing Ltd.

Antonopoulos, A. M., & Wood, G. (2018). *Mastering ethereum: building smart contracts and dapps*. O'reilly Media.

Ashworth, A., & Perera, S. (2018). *Contractual procedures in the construction industry*. Routledge.

Baliga, A. (2017). Understanding blockchain consensus models. *Persistent, 4*, 1-14.

Behera, P., Mohanty, R. P., & Prakash, A. (2015). Understanding construction supply chain management. *Production Planning & Control*, 26(16), 1332-1350.

Belle, I. (2017). The architecture, engineering and construction industry and blockchain technology. *Digital Culture, 2017*, 279-284.

Boje, C., Guerriero, A., Kubicki, S., & Rezgui, Y. (2020). Towards a semantic Construction Digital Twin: Directions for future research. *Automation in Construction*, 114, 103179.

Braaksma, H. H. (2016). Smart Asset Management in the construction sector: A holistic research into utilising the potential of sensor technology.

Danuri, M. M., Munaaim, M. C., Rahman, H. A., & Hanid, M. (2006, November). Late and non-payment issues in the Malaysian Construction Industry-Contractor's perspective. In *International Conference on Construction, Culture, Innovation and Management (CCIM)* (pp. 26-29).

Ferretti, S., & D'Angelo, G. (2020). On the ethereum blockchain structure: A complex networks theory perspective. *Concurrency and Computation: Practice and Experience*, 32(12), e5493.

Grieves, M. (2014). Digital twin: manufacturing excellence through virtual factory replication. *White paper, 1*, 1-7.

Guillen, A. J., Crespo, A., Gómez, J., González-Prida, V., Kobbacy, K., & Shariff, S. (2016). Building Information Modeling as Assest Management Tool. *Ifac-Papersonline*, 49(28), 191-196.

Hastings, N.A.J. (2015). *Physical asset management: With an introduction to ISO55000*. Springer.

Heiskanen, A. (2017). The technology of trust: How the Internet of Things and blockchain could usher in a new era of construction productivity. *Construction Research and Innovation*, 8(2), 66-70.

Hileman, G., & Rauchs, M. (2017). 2017 global blockchain benchmarking study. *Available at SSRN 3040224*.

Kinnaird, C & Geipel, M. (2017). *Blockchain Technology: How the Inventions Behind Bitcoin are Enabling a Network of Trust for the Built Environment*, ARUP, London.

Kopsida, M., Brilakis, I., & Vela, P. A. (2015, October). A review of automated construction progress monitoring and inspection methods. In *Proc. of the 32nd CIB W78 Conference 2015* (pp. 421-431).

Lanko, A., Vatin, N., & Kaklauskas, A. (2018). Application of RFID combined with blockchain technology in logistics of construction materials. In *Matec Web of conferences* (Vol. 170, p. 03032). EDP Sciences.

Li, J., Greenwood, D., & Kassem, M. (2019a). Blockchain in the built environment and construction industry: A systematic review, conceptual models and practical use cases. *Automation in Construction*, 102, 288-307.

Li, J., Greenwood, D., & Kassem, M. (2019b). Blockchain in the construction sector: a socio-technical systems framework for the construction industry. In *Advances in informatics and computing in civil and construction engineering* (pp. 51-57). Springer, Cham.

Li, J., Kassem, M., Ciribini, A. L. C., & Bolpagni, M. (2019c). A proposed approach integrating DLT, BIM, IOT and smart contracts: Demonstration using a simulated installation task. In *International Conference on Smart Infrastructure and Construction 2019 (ICSIC) Driving data-informed decision-making* (pp. 275-282). ICE Publishing.

Lo, S.K., Xu, X., Staples, M. and Yao, L. (2020). Reliability analysis for blockchain oracles. *Computers & Electrical Engineering*, 83, p.106582.

Lu, Q., Xie, X., Heaton, J., Parlikad, A. K., & Schooling, J. (2019, October). From BIM towards digital twin: Strategy and future development for smart asset management. In *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing* (pp. 392-404). Springer, Cham.

Luo, H., Das, M., Wang, J., & Cheng, J. C. P. (2019). Construction payment automation through smart contract-based blockchain framework. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction* (Vol. 36, pp. 1254-1260). IAARC Publications.

Ma, Z., Zhou, L., & Sheng, W. (2014, September). Analysis of the new asset management standard ISO 55000 and PAS 55. In *2014 China International Conference on Electricity Distribution (CICED)* (pp. 1668-1674). IEEE.

Mason, J. (2017). Intelligent contracts and the construction industry. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, 9(3), 04517012.

Mason, J. (2019). BIM fork: Are smart contracts in construction more likely to prosper with or without BIM?. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, 11(4), 02519002.

Nakamoto, S. (2019). *Bitcoin: A peer-to-peer electronic cash system*. Manubot.

Nanayakkara, S., Perera, S., & Senaratne, S. (2019). Stakeholders' Perspective on Blockchain and Smart Contracts Solutions for Construction Supply Chains. *In CIB World Building Congress*.

Nawari, N. O., & Ravindran, S. (2019). Blockchain and the built environment: Potentials and limitations. *Journal of Building Engineering*, 25, 100832.

Omar, T., & Nehdi, M. L. (2016). Data acquisition technologies for construction progress tracking. *Automation in Construction*, 70, 143-155.

Pučko, Z., Vincek, D., Štrukelj, A., & Šuman, N. (2017). Application of 6D building information model (6D BIM) for business-storage building in Slovenia. In *IOP Conference Series: Materials Science and Engineering* (Vol. 245, pp. 1-10).

Scardovi, C. (2016). *Restructuring and innovation in banking (p. 36).* Springer International Publishing.

Shojaei, A., Flood, I., Moud, H. I., Hatami, M., & Zhang, X. (2019). An Implementation of Smart Contracts by Integrating BIM and Blockchain. *In Proceedings of the Future Technologies Conference* (pp. 519-527). Springer, Cham.

Szabo, N., (1997). Formalizing and securing relationships on public networks. *First monday*.

Turk, Ž., & Klinc, R. (2017). Potentials of blockchain technology for construction management. *Procedia engineering*, 196, 638-645.

Vijayeta, M. (2019). Relevance & Applicability of Agile Driven Bim at Conceptual & Design Phases of a Project-A Case Study. *Project management in the emerging world of disruption*, 613.

Vrijhoef, R., & Koskela, L. (2000). The four roles of supply chain management in construction. *European journal of purchasing & supply management*, 6(3-4), 169-178.

Wang, J., Wu, P., Wang, X., & Shou, W. (2017). The outlook of blockchain technology for construction engineering management. *Frontiers of engineering management*, 67-75.

Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper, 151(2014), 1-32