2024 European Conference on Computing in Construction
Chania, Crete, Greece
July 15-17, 2024

CHEAPER SMART CONTRACTS FOR THE BUILT ENVIRONMENT? LINKING ON-CHAIN AND OFF-CHAIN IN A BLOCKCHAIN-GOVERNED APPROACH

Jens J. Hunhevicz[1,2], David F. Bucher[2], Kepeng Hong[3], Daniel M. Hall[4], and Catherine De Wolf[2]
[1]EMPA, Dübendorf, Switzerland
[2]ETH Zurich, Zurich, Switzerland
[3]DTU, Kongens Lyngby, Denmark
[4]TU Delft, Delft, The Netherlands

Abstract

The digitization and automation of contracts within the built environment through blockchain has demonstrated potential. Nonetheless, the use of on-chain smart contracts can amount to substantial costs. This study proposes a blockchain-governed approach to individually assess whether and how to use blockchain for different components of contracts. We explain the rationale behind the concept and implement a pilot prototype of a performance-based, blockchain-governed contract. The results promise an alternative and more cost-effective approach to smart or intelligent contracts in the built environment, while still allowing for trusted verification of key contract parts through blockchain.

Introduction

Contracts are important to the collaborative built environment and for interactions between its actors. With the ongoing digital transformation, the digitization and automation of contracts is increasingly being discussed. In particular, blockchain technology promises to enable ”smart legal”, sometimes also termed ”intelligent” contracts (Mason, 2017; McNamara and Sepasgozar, 2021; Allen and Hunn, 2022). Independent of the term, we refer here to the idea to encode contract terms in so-called blockchain smart contracts, which are scripts deployed on a blockchain that can then enforce interaction logic with any blockchain transactions, for example to execute a payment. Blockchain as an immutable ledger of peer-to-peer transactions ensures that contract terms are executed in a transparent and trustworthy manner without the need for a third-party institution. The applications of smart contracts for the built environment are diverse (Li and Kassem, 2021). Examples include the automatic execution of coded terms for contract management (Msawil et al., 2022), automated contracting for construction projects (Gupta and Jha, 2023), or performance-based contracts for lifecycle services based on IoT data (Hunhevicz et al., 2022).

Although a very promising technology for contracting, the use of blockchain smart contracts also creates challenges and risks (Mezquita et al., 2019). Most relevant to this study, their use can amount to substantial costs (Zou et al., 2021; Hunhevicz et al., 2022). As discussed later in this paper, potential solutions are to use a blockchain with low transaction costs or permissioned blockchains. However, we suggest that a more unexplored option is to decouple the functionality of smart contracts and use blockchain only for parts of a contract.

Therefore, we introduce the concept of ”blockchain-governed” contracts in the built environment, and why it could be a way to reduce the costs associated with blockchain-based contract management with reasonable trade-offs. To do so, we first cover the necessary background in the point of departure, then introduce the concept of the chosen approach, and show an exemplary implementation of a performance-based contract. Finally, we discuss our contribution by comparing it to a previous on-chain implementation of a similar smart contract and point out limitations and further research.

Point of Departure

Smart Contract Components

As already mentioned in the introduction, smart contracts encode interaction logic with blockchain transactions. In simple terms, a smart contract performs tasks like *”if this happens on the blockchain, then execute this transaction logic”*. Smart contract can be abstracted into four key components (Hunhevicz et al., 2022), as shown in the upper part of Figure 1 and further described below. Although they can be combined into complex contract constructs, for the scope of this paper, we can focus on these components individually.

Identity: Blockchain identifies users with a pair of public / private keys. The public key serves as an address to identify a blockchain user on the network. The private key acts as a password to sign transaction to prove control of the respective address. Smart contracts require information about which public address is allowed to interact or is affected by the smart contract functionality.

Payment: Smart contracts can initiate payment transactions in native cryptocurrency or another token. They can also hold funds and act as bank accounts. Even if the functionality of a smart contract is not related to finance, smart contract functionality still involves a monetary fee to execute transactions. Therefore, for the scope of this paper, we refer to payments as any use of a token or cryptocurrency, both for contract payments or fees.

Data: Smart contracts can store and act on data. Depending on the blockchain, data is stored through available data primitives, for example, in the case of Ethereum, state vari-

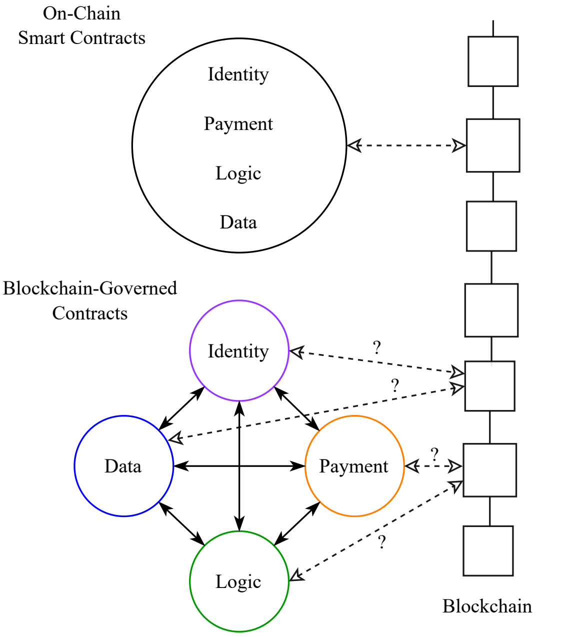


Figure 1: Smart contracts involve identity, payment, logic, and data; a blockchain-governed contract assesses individually whether and how to use blockchain for each category.

ables, arrays, structs, or mappings[1]. To add data to a smart contract, executing a transaction is required to pay the network for storing the data.

Logic: The key part of a smart contract is its logic related to transaction execution and interaction, possibly interacting with and connecting the other described components of identity, payment, and data.

Current Approaches to Smart Contracts

This section reviews current approaches to smart contracts and, where applicable, provides examples of research in the context of the built environment.

On-Chain Smart Contracts

In this paper, we refer to on-chain smart contracts as what is most often meant with the concept of blockchain smart contracts. They include all the above-introduced components in one construct. Once deployed on the blockchain, the logic is deterministic and fully transparent. Ethereum piloted the use of such Turing-complete smart contracts (Buterin, 2014). Several studies in the built environment use on-chain smart contracts, e.g. on Ethereum (Yang et al., 2020; Hamledari and Fischer, 2021; Hunhevicz et al., 2022).

Depending on the used blockchain, the speed of transaction execution can be slow and the associated transaction costs can amount to large sums (Zou et al., 2021). To reduce the costs associated with on-chain contracts, we identified two main options.

First, a blockchain with cheaper transaction costs can be used. For example, new generations of blockchain networks optimize and market themselves for high transaction throughput and low transaction cost, e.g. the Solana blockchain[2]. Moreover, existing blockchain networks, such as Bitcoin and Ethereum, promote the use of second-layer protocols to process transactions faster and cheaper (Gangwal et al., 2023). A potential trade-off to faster transaction can concern the security of the blockchain (Kiayias, 2016). Furthermore, transaction costs can also increase as a function of growing network usage. For now, we are not aware of much research discussing this approach in the built environment. Examples include Naderi et al. (2023) and Scott et al. (2024).

The second option is a permissioned blockchain, also known as a consortium blockchain, e.g. Hyperledger Fabric[3]. This type of blockchain is run by a set of trusted actors who operate the blockchain nodes. Permissioned blockchains offer high throughput and no transaction fees, as the consensus mechanisms are fast and overhead costs are typically paid and shared by the consortium. As a potential downside, permissioned blockchains rely on trusted actors to run the nodes, who have the ability to exclude transactions and users and even shut down the network at their discretion. Furthermore, ensuring the availability of nodes can be challenging over long time periods and changing stakeholders. Nevertheless, due to its project-based nature, a consortium approach is often considered in the built environment (Yang et al., 2020).

Off-chain Approaches to Contracts

A second way to reduce the costs associated with smart contracts involves moving parts of the smart contract off-chain, meaning not stored or implemented on the blockchain. One reason is that the transaction cost of an on-chain smart contract is typically determined by the transaction size and involved computation of a smart contract execution. Especially in the context of networks like Ethereum, moving data off-chain e.g. to external data networks like IPFS[4], is becoming increasingly common, also for applications explored in the built environment (Tao et al., 2021; Adel et al., 2023).

Another reason to research off-chain contracts is that some networks, e.g. Bitcoin, do not support Turing complete smart contracts. Therefore, different approaches are suggested to move data and logic off-chain, e.g. outsourcing to service providers (Wüst et al., 2019), trusted execution environments (TEEs) (Das et al., 2019), or off-chain schemas using blockchain as a state commitment layer and ownership control[5]. We are not yet aware of research in the built environment exploring such contract approaches.

Finally, governance platforms can facilitate trusted process execution through a blockchain-verified and secure semi-automated process (Dursun and Üstündağ, 2021). ”Politea”[6] is an example for such a platform used as a pro-

[1] `https://docs.soliditylang.org/en/latest/internals/layout_in_storage.html`, accessed 30.01.2024

[2] `https://solana.com/`

[3] `https://www.hyperledger.org/projects/fabric`

[4] `https://ipfs.tech/`

[5] `https://rgb-org.github.io/`

[6] `https://proposals.decred.org/`

posal and payment system for the Decred blockchain contractors. Although no formal on-chain logic is utilized, it makes use of timestamps to allow for checking and re-evaluating the process in case of disputes. Timestamping is inherent to blockchains; each block and transaction has a timestamp for clear recognition. Using this mechanisms, data can be attached to a transaction to create a fingerprint for off-chain data as a proof of existence.

Motivation and Scope of This Study

The previous sections showed that there are different approaches to smart contracts. An alternative to on-chain smart contracts is moving parts or all of the contract components off-chain. We see these approaches as a viable alternative for smart or intelligent contracts in the built environment for the following reasons.

Known actors: Construction and the built environment rely on established processes. The decision to use blockchain involves a trade-off between trusting the technical system vs. trusting the involved actors (Hunhevicz and Hall, 2020). If actors are known and accountable, the cost premium of a fully decentralized on-chain smart contract may not be justified.

Time spans: Construction contracts usually span duration of several years of individual project phases, e.g., for design and construction. Even though such contracts can benefit from a blockchain approach, it might not be justified to pay for an on-chain contract that lasts as long as a blockchain exists.

Physical nature: Blockchain-based contracts need to rely on trusted feedback loops of physical processes and products. Although the logic of an on-chain smart contract is trustworthy, wrong input data could jeopardize a correct execution. A partial off-chain approach could be a practical middle ground with reasonable trust and transparency.

Trust in existing systems: The construction industry is known for slow technology adoption. In many cases, the possibility of traditional identity verification, FIAT payments, or the possibility for more data privacy has priority. A blockchain-governed approach could better consider these aspects than a pure on-chain smart contract.

Therefore, this study proposes the concept of ”blockchain-governed” contracts for the built environment, and pilots an off-chain governance system for an exemplary performance-based contract.

Blockchain-Governed Contracts

The term ”blockchain-governed” contract is proposed in this paper for contracts that use blockchain in one of the four components introduced (identity, payment, logic, or data), but at the same time also do not use blockchain with an off-chain approach in at least one category. As shown in Figure 1, such a blockchain-governed approach allows to individually evaluate for use cases whether and how blockchain is used for each component. Table 1 provides non-exhaustive examples of the differences between an on-chain and an off-chain approach for each component.

Proposed Off-Chain Governance System

To illustrate the concept of blockchain-governed contracts in the built environment, we propose an off-chain governance system for a contract in the built environment in Figure 2. The figure is organized in a matrix structure, with the four components of identity, payment, and logic arranged vertically, and the physical built environment, the off-chain governance system, and blockchain (on-chain) arranged horizontally. The proposed system is inspired by the aforementioned Decred governance system, which has been operational since 2017. Therefore, this system is already proven in the context of managing the development

Table 1: Exemplary differences in choosing an on-chain vs. off-chain approach for the four contract components. The proposed concept of blockchain-governed contracts could then use either option, but at least one on-chain, and one off-chain component.

	On-Chain	Off-Chain
Identity	To execute an on-chain transaction, actors need a blockchain address, e.g. for payments, logic execution, or storing data. Each transaction needs to be signed with the corresponding private key.	Traditional credentials such as username and password can be used as means of identification for off-chain contract applications, timestamping, or read-only functionality of blockchain state.
Payment	Cryptocurrency and tokens can be used as a means of payment. Execution guarantee, censorship resistance, and customization of the monetary asset can be advantages of using on-chain payments. Costs refer to network transaction costs.	Traditional third-party payment methods in FIAT currencies such as EUR can be linked to the contract, e.g. via application interfaces (APIs). Benefits include compliance with traditional systems and regulations. Costs relate to service fees.
Logic	On-chain logic provides several benefits, including direct interaction with the blockchain state (such as executing a payment), transparent and verifiable logic, and censorship resistant logic that cannot be changed unnoticed.	Off-chain logic can be transparent and verifiable by using accessible (public or distributed local) code repositories combined with timestamping. This enables the logic to be recalculated and verified. It may also be possible to use other approaches (e.g. TEEs).
Data	Data stored on a blockchain is transparent, immutable, and available as long as the blockchain exists. These characteristics ensure the data cannot be altered or deleted unnoticed.	Moving data to external storage locations can reduce on-chain computation costs. The smart contract can reference the data's location and timestamp to verify its authenticity.

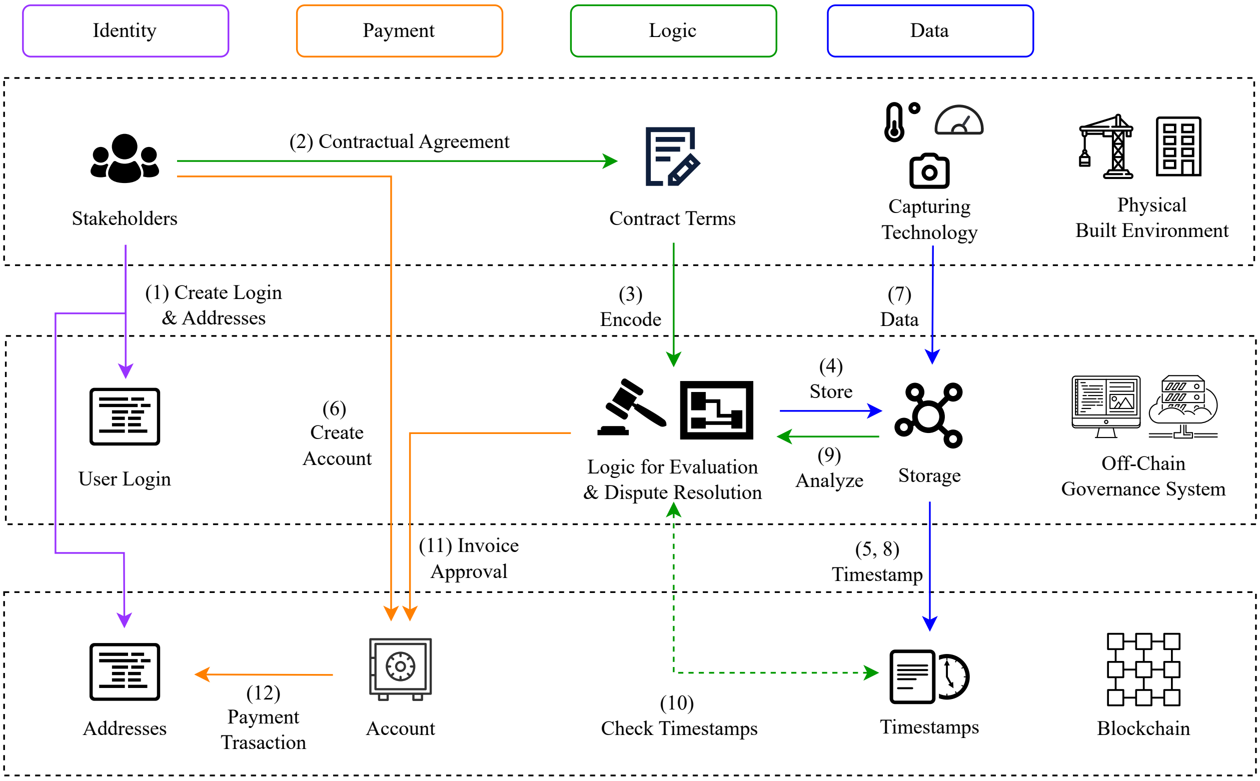


Figure 2: Exemplary blockchain-governed contract system for the built environment.

of the Decred ecosystem, making it a good starting point for investigating blockchain-governed systems in the built environment. In addition, it demonstrates the interaction between the different components using both on- and off-chain approaches.

A typical process would work as follows (see Figure 2): (1) All stakeholders create both login credentials for the governance platform, as well as an address so that the contract payout terms can be defined and encoded. (2) The parties negotiate the contractual agreement. (3) The contract terms are encoded. (4) The contract is stored in the off-chain governance database. (5) The contract is timestamped so that it can later be verified as authentic. (6) The responsible stakeholders create and fund the project account from which the payments will be released. (7) The required data is captured, streamed, and stored off-chain in the project database. All stakeholders should have a local copy, or at least access to the data. (8) The data is periodically timestamped so that it can be verified at a later time based on the locally stored copies. (9) At defined intervals, performance is automatically evaluated based on the input data and contract terms. All results are published transparently to stakeholders. (10) Stakeholders can confirm the evaluation. Only in the case of an error or disagreement, a dispute resolution process would start to verify the correct performance logic and data based on timestamps (dotted line in Figure 2). (11) If there is no disagreement, or after the dispute is resolved, the generated invoices are confirmed and signed by the necessary stakeholders. (12) Payouts are made in cryptocurrency to the defined addresses.

Implementation

We developed a pilot prototype to obtain first insights into the feasibility and challenges of the proposed blockchain-governed contract approach. After introducing the tested process, we outline for each component (identity, payment, logic, and data) the chosen approach for our prototype, in line with the proposed governance system in Figure 2.

Tested Process

The example follows the use case of a performance-based contract that rewards stakeholders for meeting the energy performance targets of operating a building. A similar process was implemented and evaluated in a previous study (Hunhevicz et al., 2022), allowing for a good comparison regarding process, cost, and efficiency. For the interested reader, the referenced study also gives more details on the rationale of performance based contracting.

Figure 3 shows the interaction between the technical parts implemented, as well as the stakeholders. The general idea is that a building owner initiates the use of a blockchain-governed contract to issue automatic payments to a contractor responsible for operating and managing the energy system. The exact logic of the contract is not important for the scope of this work; in this example we used historic temperature data from the NEST building at EMPA Dübendorf, Switzerland[7], issuing payments when the managed temperature stays within a defined range around the set-point temperature.

[7] https://www.empa.ch/web/nest/

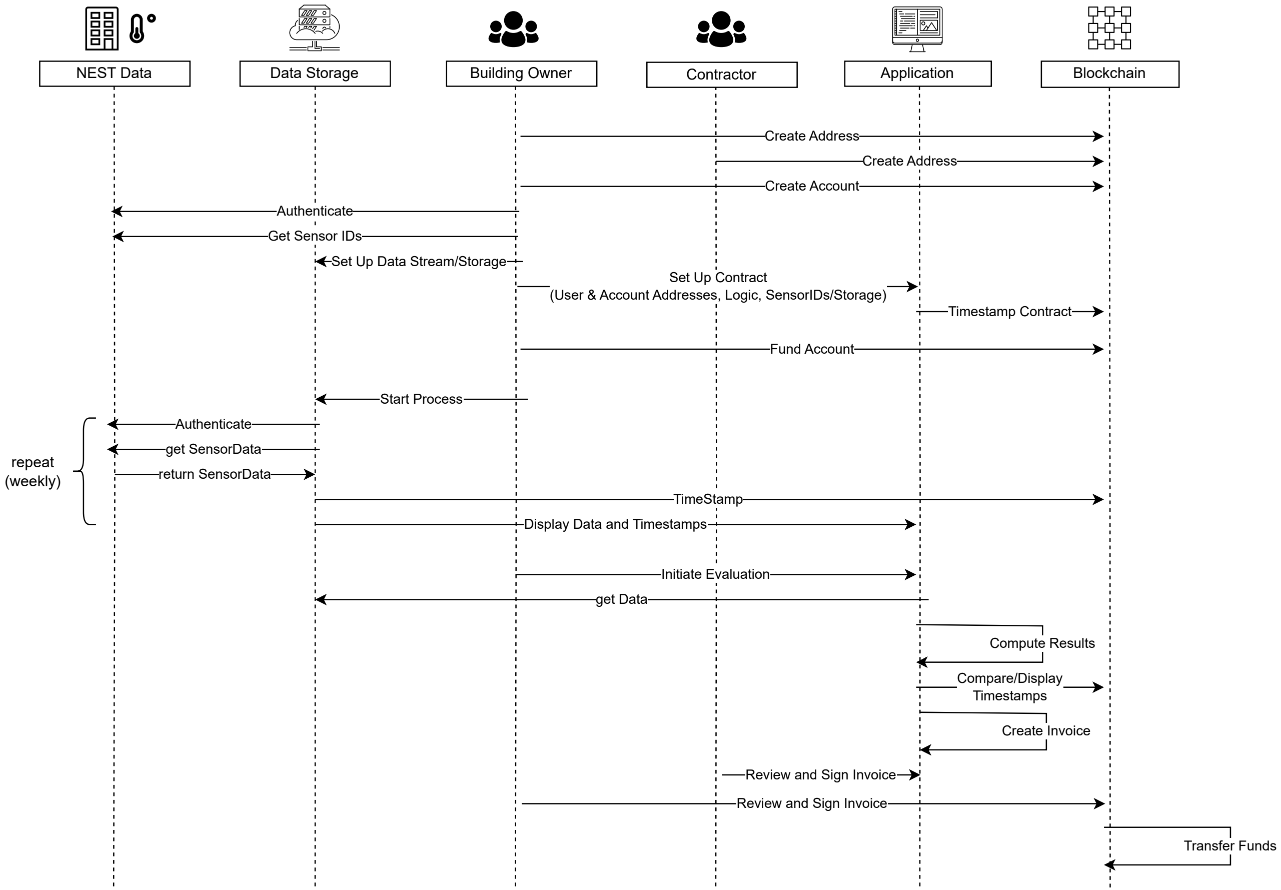


Figure 3: Implemented process of the temperature performance contract. In the tested process, both the owner and the contractor accept the evaluation and sign the invoice transaction.

Logic

The core of the prototype is a web application built with Next.js[8] that manages the main contract governance process. The stakeholders can define a new data set, as well as the contract logic with the main parameters of the contract (see Figure 4). In our case, we defined the sensor data set, the payout address, the set point temperature, the allowed deviation and threshold for penalties, and the payout amount of the performance based contract.

The contract will be time-stamped after its creation, anchoring the hash of the JSON contract file to the Decred blockchain (see Figure 4). We chose dcrtime[9], because the service is open source and freely accessible via API. Dcrtime timestamps approximately every hour, called anchoring. To indicate the status of the process, the application changes from ”not timestamped”, to ”waiting for anchoring time”, to ”timestamped”. Once timestamped, the digest can be retrieved from the application and checked for the timestamp and transaction hash on Timestamply[10]. In Decred's blockchain explorer[11], the transaction details can be displayed.

After a defined evaluation period, the results can be computed for the specified data set and defined contract logic. To make this evaluation trustworthy, the data and results are visualized in the application (see Figure 5). In case there are concerns about the validity of the process, the code and data can be checked against local copies or copies stored in an open source repository, and if needed, even redeployed and recalculated with the timestamped data and contract logic to check for authenticity.

Payment

Payouts are released from a separate project account, which is set up as a multi-sig wallet by the owner and contractor, meaning that both the owner's and contractor's blockchain signatures are required to move funds. This was implemented so that neither party could spend the funds without the other's approval. After calculation of the payouts, the invoice to credit the project account is automatically created based on the specified payout addresses. If both parties agree, they can import the proposed transaction to their respective wallets and sign for execution.

Identity

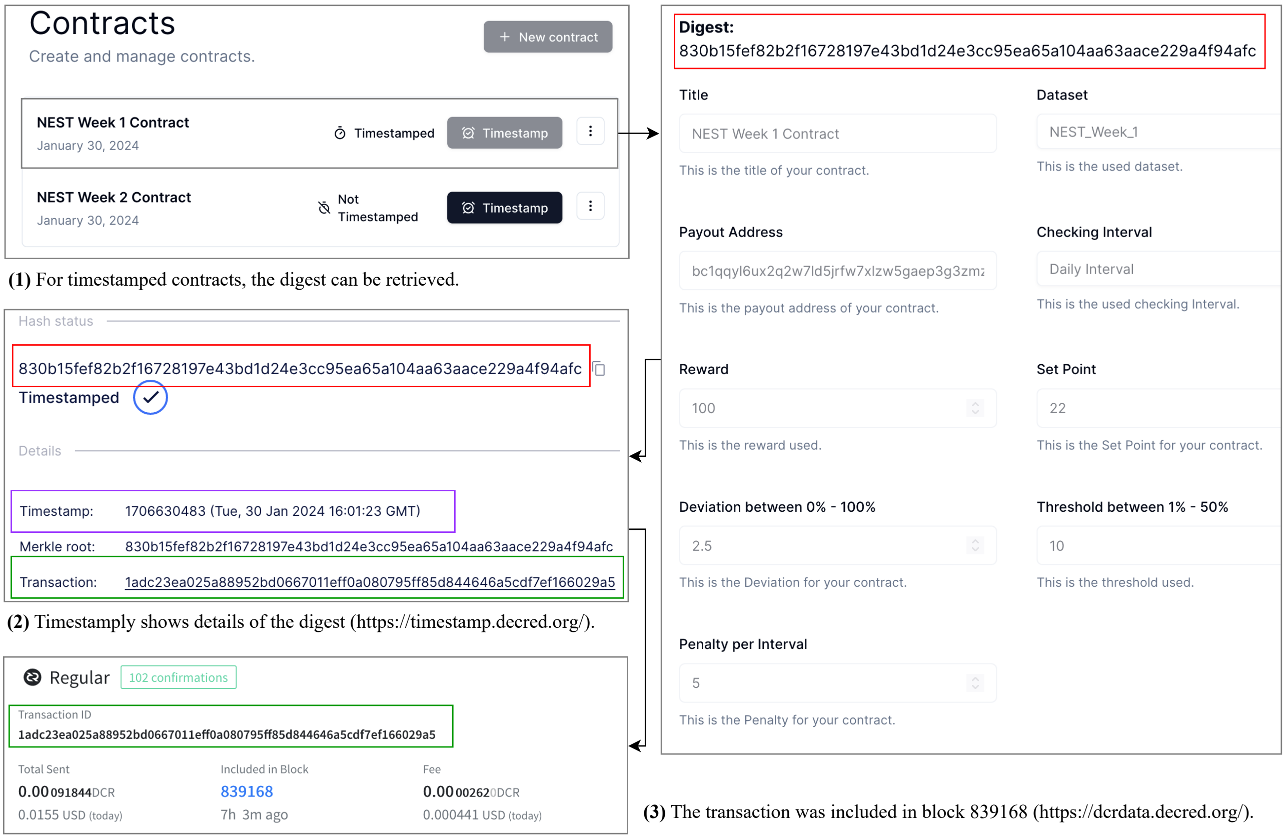
The above described web application requires a standard username and password for identity verification. However, the application could also have used blockchain-

[8] `https://nextjs.org/`

[9] `https://github.com/decred/dcrtime`

[10] `https://timestamp.decred.org`

[11] `https://dcrdata.decred.org`



Contracts

Create and manage contracts.

NEST Week 1 Contract

January 30, 2024

NEST Week 2 Contract

January 30, 2024

(1) For timestamped contracts, the digest can be retrieved.

Hash status

830b15fef82b2f16728197e43bd1d24e3cc95ea65a104aa63aace229a4f94afc

Timestamped ✓

Details

Timestamp: 1706630483 (Tue, 30 Jan 2024 16:01:23 GMT)

Merkle root: 830b15fef82b2f16728197e43bd1d24e3cc95ea65a104aa63aace229a4f94afc

Transaction: 1adc23ea025a88952bd0667011eff0a080795ff85d844646a5cdf7ef166029a5

(2) Timestamply shows details of the digest (https://timestamp.decred.org/).

Regular 102 confirmations

Transaction ID
1adc23ea025a88952bd0667011eff0a080795ff85d844646a5cdf7ef166029a5

Total Sent	Included in Block	Fee
0.00091844DCR	839168	0.00002620DCR
0.0155 USD (today)	7h 3m ago	0.000441 USD (today)

Digest:
830b15fef82b2f16728197e43bd1d24e3cc95ea65a104aa63aace229a4f94afc

Title: NEST Week 1 Contract
This is the title of your contract.

Dataset: NEST_Week_1
This is the used dataset.

Payout Address: bc1qqyl6ux2q2w7ld5jrfw7xlzw5gaep3g3zmz
This is the payout address of your contract.

Checking Interval: Daily Interval
This is the used checking Interval.

Reward: 100
This is the reward used.

Set Point: 22
This is the Set Point for your contract.

Deviation between 0% - 100%: 2.5
This is the Deviation for your contract.

Threshold between 1% - 50%: 10
This is the threshold used.

Penalty per Interval: 5
This is the Penalty for your contract.

(3) The transaction was included in block 839168 (https://dcrdata.decred.org/).

Figure 4: Contracts can be created in the application with the needed parameters for the evaluation and payout. After creation, they can be timestamped to ensure verifiability at a later point.

based access verification (Hunhevicz et al., 2023). This demonstrates that there can be different identity procedures within one blockchain-governed contract.

For the payouts, a blockchain wallet is needed. Since we chose the Bitcoin blockchain for payments, we used the Electrum[12] wallet to create the public/private key pairs, since it allows for easy multi-sig capabilities. The owner and the contractor each need their own wallet.

Data

Data is stored off-chain in a relational SQL database accessed by the application. In our example, the application used historical temperature tiime series of the NEST building from the ehub platform[13] through the available APIs. In addition to the main off-chain storage, the data is timestamped in weekly intervals, as introduced above for the contract logic (see also Figure 4). Therefore, the hashes of the data are stored on-chain on the Decred blockchain.

Discussion and Conclusion

This study is a first attempt to find practical alternatives to pure on-chain smart contracts in the built environment. The proposed decoupling and individual evaluation of whether and how to use blockchain for identity, payment, logic, and data was termed ”blockchain-governed contract”. The then proposed and evaluated exemplary contract governance system combines on- and off-chain approach, mainly using blockchain for proofing authenticity of data through timestamping, as well as payouts.

One of the most interesting consequences of this approach is that it forces an implementer to think about the individual parts of a contract and whether blockchain makes sense. With a complete on-chain approach, the entire contract is usually implemented on-chain, although only a subset would benefit from it. This means that blockchain could be used for only parts of a contract and that different parts of the contract may use different features of blockchain, possibly even different blockchains. For example, timestamping could benefit from the free services offered by e.g. the Decred dcrtime[14] library or Bitcoin's opentimestamps[15], while other features that require on-chain smart contracts could use a fast blockchain layer for high throughput and reasonable transaction costs. A more extensive and systematic analysis on when to use which blockchain would be interesting future research.

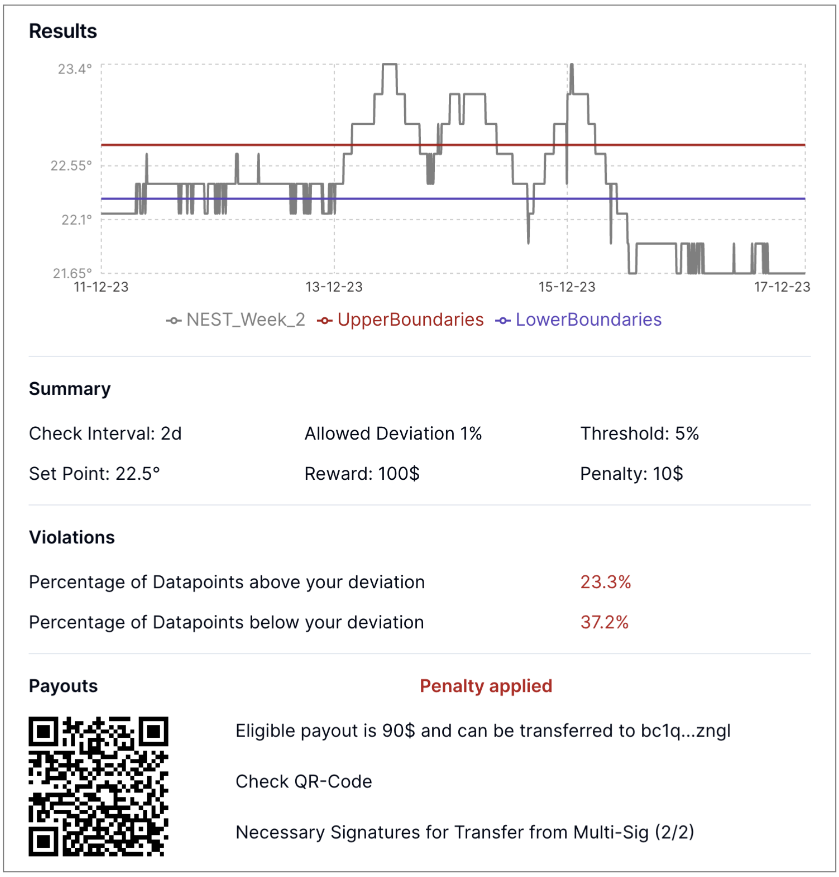
In addition, more work is needed to complete an implementation and validate it in a real-world contracting scenario. One of the missing pieces of the here tested system is the dispute resolution process, in case the parties do not accept the evaluation and payments performed.

[12] https://electrum.org/

[13] https://info.nestcollaboration.ch/wikipediapublic/

[14] https://github.com/decred/dcrtime

[15] https://opentimestamps.org/



Results

Summary

Check Interval: 2d	Allowed Deviation 1%	Threshold: 5%
Set Point: 22.5°	Reward: 100$	Penalty: 10$

Violations

Percentage of Datapoints above your deviation	23.3%
Percentage of Datapoints below your deviation	37.2%

Payouts

Penalty applied

Eligible payout is 90$ and can be transferred to bc1q...zngl

Check QR-Code

Necessary Signatures for Transfer from Multi-Sig (2/2)

Figure 5: Evaluation of data in the application based on the contract terms and data set to calculate payouts. Authenticity of the data and contract can be checked against the timestamps. The invoice is created and needs to be signed by both the contractor and owner.

Other works have already proposed blockchain-based dispute resolution platforms for the built environment (Saygili et al., 2021; Son and Lien, 2022), which could serve as inspiration. We are confident that a suitable mechanism could be designed and implemented. Furthermore, while the proposed system seems like a reasonable approach, it is only one of many possible combinations between on- and off-chain for the different contract components as described in the departure section. Other combinations should be designed and tested for comparison.

A blockchain-governed contract approach appears to offer more flexibility and the potential for cost savings compared to an on-chain smart contract. In addition, we believe it could also improve the usability of smart contracts, since only parts of the contract require a blockchain, which is currently a mostly new technology unfamiliar to most stakeholders in the built environment. Complicated smart contract applications could be avoided. In our case, the proposed governance platform is similar to current web applications, except that payment execution requires a wallet. Another interesting observation is that a blockchain-governed approach requires more manual and human input than a full on-chain smart contract. In this sense, it can be considered less automated then previously proposed smart or intelligent contracts, but is potentially simpler and closer to existing practice. However, the verification process is likely to take longer to resolve than a fully transparent and deterministic on-chain approach, particularly in the event of a dispute. More detailed research is needed to evaluate user experience aspects, the industry requirements and readiness for different blockchain-governed smart contract platforms, as well as a quantitative comparison of cost and performance, e.g. under what circumstances the cost of on-chain smart contracts compared to a blockchain-governed approach is justified. Overall, a blockchain-governed contract approach seems a solution worth exploring in the built environment.

Acknowledgments

We thank Empa for access to the ehub NEST time series data.

References

Adel, K., Elhakeem, A., and Marzouk, M. (2023). Decentralized system for construction projects data management using blockchain and IPFS. Journal of Civil Engineering and Management, 29(4):342–359.

Allen, J. and Hunn, P., editors (2022). Smart Legal Contracts: Computable Law in Theory and Practice. Oxford University Press, 1 edition.

Buterin, V. (2014). Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform.

Das, P., Eckey, L., Frassetto, T., Gens, D., Hostáková, K., Jauernig, P., Faust, S., and Sadeghi, A.-R. (2019). FastKitten: Practical Smart Contracts on Bitcoin.

Dursun, T. and Üstündağ, B. B. (2021). A novel framework for policy based on-chain governance of blockchain networks. Information Processing & Management, 58(4):102556.

Gangwal, A., Gangavalli, H. R., and Thirupathi, A. (2023). A survey of Layer-two blockchain protocols. Journal of Network and Computer Applications, 209:103539.

Gupta, P. and Jha, K. N. (2023). A Decentralized and Automated Contracting System Using a Blockchain-Enabled Network of Stakeholders in Construction Megaprojects. Journal of Management in Engineering, 39(4):04023021.

Hamledari, H. and Fischer, M. (2021). Construction payment automation using blockchain-enabled smart contracts and robotic reality capture technologies. Automation in Construction, 132:103926.

Hunhevicz, J. J., Bucher, D. F., Soman, R. K., Honic, M., Hall, D. M., and De Wolf, C. (2023). Web3-based role and token data access: The case of building material passports. In 2023 European Conference on Computing in Construction and the 40th International CIB W78 Conference.

Hunhevicz, J. J. and Hall, D. M. (2020). Do you need a blockchain in construction? Use case categories and decision framework for DLT design options. Advanced Engineering Informatics, 45(February):101094.

Hunhevicz, J. J., Motie, M., and Hall, D. M. (2022). Digital building twins and blockchain for performance-based (smart) contracts. Automation in Construction, 133:103981.

Kiayias, A. (2016). Speed-Security Tradeoffs in Blockchain Protocols.

Li, J. and Kassem, M. (2021). Applications of distributed ledger technology (DLT) and Blockchain-enabled smart contracts in construction. Automation in Construction, 132:103955.

Mason, J. (2017). Intelligent Contracts and the Construction Industry. Journal of Legal Affairs and Dispute Resolution in Engineering and Construction, 9(3):04517012.

McNamara, A. J. and Sepasgozar, S. M. (2021). Intelligent contract adoption in the construction industry: Concept development. Automation in Construction, 122:103452.

Mezquita, Y., Valdeolmillos, D., González-Briones, A., Prieto, J., and Corchado, J. M. (2019). Legal Aspects and Emerging Risks in the Use of Smart Contracts Based on Blockchain. In Uden, L., Ting, I.-H., and Corchado, J. M., editors, Knowledge Management in Organizations, Communications in Computer and Information Science, pages 525–535, Cham. Springer International Publishing.

Msawil, M., Greenwood, D., and Kassem, M. (2022). A Systematic evaluation of blockchain-enabled contract administration in construction projects. Automation in Construction, 143:104553.

Naderi, H., Shojaei, A., and Ly, R. (2023). Autonomous construction safety incentive mechanism using blockchain-enabled tokens and vision-based techniques. Automation in Construction, 153:104959.

Saygili, M., Mert, I. E., and Tokdemir, O. B. (2021). A decentralized structure to reduce and resolve construction disputes in a hybrid blockchain network. Automation in Construction, page 104056.

Scott, D., Ma, L., and Broyd, T. (2024). Project bank account (PBA) decentralised application for the construction industry. Construction Innovation, ahead-of-print(ahead-of-print).

Son, P. V. H. and Lien, P. N. (2022). Blockchain crowdsourced arbitration in construction project delay resolution. Journal of Science and Technology in Civil Engineering (STCE) - HUCE.

Tao, X., Das, M., Liu, Y., and Cheng, J. C. P. (2021). Distributed common data environment using blockchain and Interplanetary File System for secure BIM-based collaborative design. Automation in Construction, 130:103851.

Wüst, K., Diana, L., Kostiainen, K., Karame, G., Matetic, S., and Capkun, S. (2019). Bitcontracts: Supporting Smart Contracts in Legacy Blockchains.

Yang, R., Wakefield, R., Lyu, S., Jayasuriya, S., Han, F., Yi, X., Yang, X., Amarasinghe, G., and Chen, S. (2020). Public and private blockchain in construction business process and information integration. Automation in Construction, 118:103276.

Zou, W., Lo, D., Kochhar, P. S., Le, X.-B. D., Xia, X., Feng, Y., Chen, Z., and Xu, B. (2021). Smart Contract Development: Challenges and Opportunities. IEEE Transactions on Software Engineering, 47(10):2084–2106.