2023 European Conference on Computing in Construction
40th International CIB W78 Conference
Heraklion, Crete, Greece
July 10-12, 2023

# DESIGN DIMENSIONS FOR BLOCKCHAIN ORACLES IN THE AEC INDUSTRY

Theodoros Dounas[1*], Jens Hunhevicz[2,3], and Brandon S. Byers[2]
[1]Robert Gordon University, Aberdeen, UK
[2]ETH Zurich, Zurich, Switzerland
[3]EMPA, Dübendorf, Switzerland

*Corresponding author: t.dounas@rgu.ac.uk

## Abstract

The literature on blockchain for AEC often has limited discussion on the design of the blockchain Oracle, the entity that ensures trusted information input from the real world. This work aims to illuminate an often unaddressed topic of how to best design Oracles for application to the AEC industry. To answer this, the paper reviews the state-of-the-art of existing Oracle taxonomies, literature on blockchain for AEC that implement Oracles, and proposes Oracle design dimensions to frame future blockchain work. The work can help researchers and practitioners to think about relevant dimensions when designing and discussing Oracles for AEC use cases.

## Introduction

Oracles are a third-party service or process that provides information to blockchains about the exterior environment and the real world. In many cases Oracles act as a two-way bridge, passing information to and from a blockchain to the external environment. Oracles enable connections to the level of the block generation, but more often to the level of the smart contract. This allows the execution of smart contracts depending on a real-world event taking place, for example starting a process based on whether a particular weather event has taken place. Due to their nature blockchains are treated as trustworthy sources, a feature that shifts the question of whether one can trust the data onto the Oracle, rather than the blockchain (Figure 1).

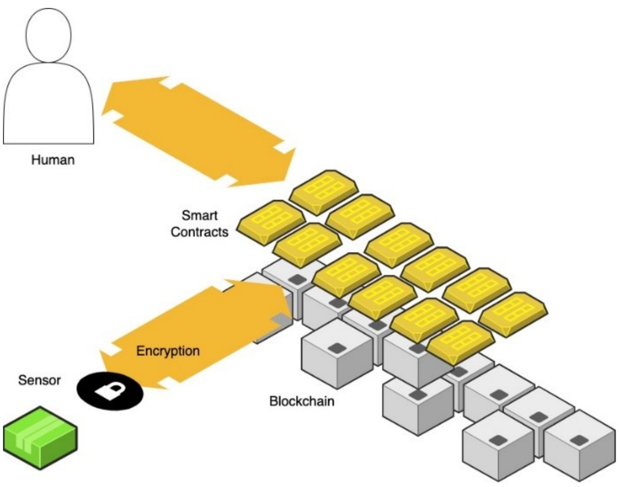


*Figure 1: Potential Oracles communicating with smart contracts that operate on blockchain*

Although blockchain technologies are considered trustless, meaning there is no need to measure trust in a deterministic system, these systems are prone to inaccuracies and falsehood when bridging between the software and the bi-directional data flow. Applying blockchain technologies to the Architecture, Engineering, and Construction (AEC) industry is gaining increased traction. Nevertheless, most of the literature emerging often has little to no testing and discussion on the Oracle system used. Considering that the Oracle can become the single point of failure (SPoF) for the whole system, the authors use this as motivation to propose design principles for an Oracle for AEC by reviewing existing taxonomies and relevant work.

The research design and subsequent structure of this paper begins with a review of existing literature on blockchain broadly that provides useful taxonomies of Oracle systems. Following this, a review is conducted of current literature discussing Oracles used in the AEC industry. Next, a synthesis of both sections leads to proposed design dimensions for implementing an Oracle in AEC. Lastly, is a discussion on the limitations of these dimensions and high-level applications.

The authors of this paper aim to show a large lapse in the literature regarding the design of Oracles and emphasize the need for authors in this space to investigate and assess the design approach. As found in a review of Oracle literature, more efforts are needed for a widely accepted taxonomy and better collaboration between researchers and practitioners (Caldarelli, 2022). This call for action provides grounds from which researchers may assess and compare Oracle design patterns for AEC.

## Oracles Definitions and Classifications

A blockchain is both a state machine able to execute smart contracts, but also a continuous, immutable chain, built out of discrete blocks of information that each contains a cryptographic hash of the previous discrete block. Each block includes a series of transactions that contain information, for example in the discipline of architectural design these can be a series of design synthetic actions, or in construction practice, information about a series of material actions, executed in a bottom-up fashion, and encoded into a block. Blockchains operate on a set of

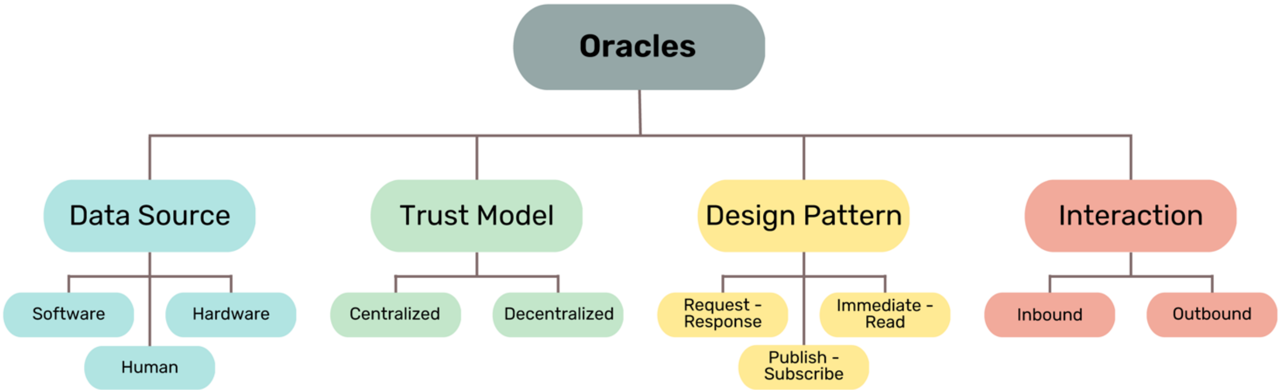


*Figure 2: Oracle Taxonomy – redrawn by the authors (Al-Breiki et al., 2020)*

distributed nodes that all hold the same ledger. The blockchains necessarily possess a mechanism to synchronize the chains contained in each node, i.e., achieve consensus on the block to add next to the chain.

Smart contracts are software that operates on top of the blockchain and are the equivalent of Turing complete software classes that can run autonomously, i.e., without human intervention once a set of conditions are met. Ontologically and practically a blockchain is a secure, self-contained computation environment, that needs data bridges to communicate with the outside world. These bridges are called Oracles, stemming from the cryptography literature where an Oracle is the mathematical provider of true random numbers (Schneier, 1996)

### Existing Oracle Taxonomies

Literature generally converges on a similar high-level Oracle Taxonomy based on the Data Source, Trust Model, Design Pattern, and Interaction type. These can be used not only to classify but to interrogate design choices for Oracles, in the AEC industry (Al-Breiki et al., 2020; Pasdar et al., 2021; Sadawi et al., 2022) (Figure 2).

### Data Source

A data source can be a piece of software, a human, or a hardware input. Software Oracles are one of the most common types of blockchain Oracles as the connection to the internet allows for information to transmit to and from in real-time (Al-Breiki et al., 2020; Beniiche, 2020). Sometimes also referred to as deterministic Oracles, the information can come from websites, online databases, servers, and other internet sources.

Hardware Oracles are those that collect and translate data from the physical world onto the smart contract by means of various sensors and hardware technologies (Al-Breiki et al., 2020; Beniiche, 2020). This information could be collected from sensors such as RFID tags, IoT, barcodes, and QR code scanners. Potential issues arise in connectivity and data encryption. Both hardware and software Oracles can be considered automated Oracles.

Human Oracles, at a high level, are individuals that provide external data to a smart contract technologies (Al-Breiki et al., 2020; Beniiche, 2020). Often, they possess specialized knowledge and can have their identity verified through cryptography. An example of a Human Oracle implementation can be through voting on an event. Human Oracles produce truth through consensus; thus, they are also called multi-source Oracles. Human Oracles can also be considered traditional manual-input Oracles, but are often disregarded because of intrinsic disadvantages due to relatively slow cognition and single points of failure (Beniiche, 2020).

### Trust model

The trust model can be either centralized or decentralized. Decentralization is important so that blockchain systems cannot be controlled by single or malicious actors, ultimately leading to the expected affordances of blockchain systems like immutability of transactions. Multiple technical parts need to be designed with decentralization in mind to achieve immutability. Next to the blockchain network itself (e.g., access to nodes, number of nodes, and other important design choices), the design of the Oracles is key. Even when a blockchain is decentralized, if the real-world data input is needed for smart contracts and this data input comes from a centralized source, the final workflow could be centralized and controllable by a single entity. Centralization and single entities are not necessarily undesired, but it is important that these design choices are deliberate and made transparent to everyone using the system. Even if a single entity designed to use the centralized Oracle is trustworthy, the risk of manipulation and hacking increases from attackers. According to Lu, et al. (2021), “If Oracles are compromised, smart contracts will also be compromised. Thus, centralized Oracles with a single data source may suffer single point of failure (SPoF) problems.”

Centralized trust models for an Oracle mean there is only a single provider of information. This produces a SPoF for the smart contract, and it becomes difficult to protect against malicious interference. Decentralized trust systems indicate multiple Oracles are queried for the accuracy of the information, thus also called consensus Oracles. Because of this mechanism, decentralized Oracles often are less efficient while being more robust to vulnerabilities. Decentralized Oracles should also be permissionless so that users can join, leave, and have an equal privilege. Although blockchain networks and smart contracts are both trustless systems, the chosen Oracles system will have to be a trusted system.

### Design pattern

This criterion classifies the behavioural sequence of the Oracle and has a consequence on the structure of the connector and the smart contract architecture. The types recognized are request-response, immediate-read, and publish-subscribe. These design patterns also have implications on the data saturation and cost of transactions, hence the overall cost structure for the Oracle selection. Beniiche (2020) defines each of the three design patterns. Immediate-read designs provide the information needed for immediate action. Examples of this pattern for AEC could involve querying for designer certifications or immediate room temperature. The publish-subscribe design is set up to broadcast data that is continuously updated and must be polled or listened to for updates. Applications for AEC could include tracking the price of certain building materials or power consumption data. Lastly, is the request-response design, the most complicated, as it resembles more towards a client-server architecture. In this design approach, a query is received, payment and access are verified, relevant data is retrieved off-chain, and the transaction is signed with the data and broadcast to the network. The request and the response can happen to or from all three data source types (human, hardware, or software). This design pattern is used if data sources are large, diverse, or dynamic.

### Interaction type

This classification explores whether an Oracle is inbound, outbound, or both in relation to the blockchain entity. Oracles can be uni- or bi-directional for data interaction and the blockchain (Beniiche, 2020). Outbound Oracles deliver data to the external world when a smart contract event is triggered. An example for AEC may be the release of payment or project information with satisfied smart contract criterion. Conversely, an inbound Oracle writes data into the blockchain from the external world, such as when payment thresholds have been reached or temperature targets are satisfied.

## Examples of Oracles in AEC/FM Research

### Why are Oracles important in Blockchain in AEC

Blockchain and smart contracts are digital technologies acting only on digital data. It is not a coincidence that the first application of blockchain (Bitcoin) was designed to be a monetary system. Money is a concept that can be completely represented digitally without the need for Oracles, but the AEC industry builds physical products beyond the digital world. Using the affordances of blockchain and smart contracts, this industry always needs to be connected to events happening in the physical world. For example, automatic payouts based on construction progress or delivery of logistic items on site needs data input of when this event happened.

In the following, the authors provide examples from current literature in the AEC industry for implemented approaches to Oracles. These do not intend to be comprehensive, nor a finished review, but rather a carefully selected sample so that emerging design dimensions can be formulated. For now, we examine AEC Oracles in three categories, human, automated (both software and hardware) and hybrid Oracles. While software might be considered as one sub-category, there are no clear examples in literature where only software is used as the source of truth. BIM for example, uses software, but with an additional step of human action or validation. This is due to often more detailed information about other aspects of the introduced framework by Al-Breiki, et al. are hard to obtain or classify from the descriptions in the literature or not described at all (2020). The lack of detail in the nature and set up of the Oracles is an additional motivation for writing the present paper.

### Human-Based Oracles

For example, Celik, et al. (2023) describe a blockchain supported BIM data provenance for construction projects, where the blockchain and the smart contracts act as the connecting tissue for a complete BIM project. In this case the Oracles are humans and BIM software feeding data in and out of the blockchain.

Dounas, et al. (2020) describe a form of decentralized BIM where only the difference in data is passed unto a decentralized CDE, using again a theoretical abstraction of an Oracle, in software and human form. In this instance Oracles are assumed to also activate payments towards the agents participating in the system. Dounas, et al. (2021) have also discussed a similar form of Oracles where design agents, humans or machines, coordinate the solution of a design problem by using blockchain as the substrate of data coordination and financial rewards.

Tezel, et al. (2021) describe three blockchain prototypes for project bank accounts, reverse-auction based tendering and asset tokenization, where the Oracles are humans.

Ahmadisheykhsarmast & Sonmez (2020) describe a smart contract for security of payment of construction contracts, where human validation is needed for a smart contract to be activated for payments. Gunasekara, et al. (2021) developed a framework for Facilities management that uses a blockchain, but where, again, human Oracles are central to validating information to be passed to a smart contract.

Li, et al. (2021) describe a blockchain based supervision model for off-site modular production, where coordination takes place through the blockchain, but humas are again relied upon to validate critical information. Van Groesen and Pauwels (2022) tested the use of a smart contract for construction supply chain tracking via QR code, BIM, smart contract, and a web app. The implemented process involves the semi-automation of processing physical asset data, comparing the as-planned and as-built states, and tokenized payments between stakeholders. The authors used the Provable Oracle Service connected to Google Firebase via an API.

**Automated Oracles**

Examples of automated Oracles used in AEC blockchain literature include the supply chain tracking of façade panels in an Australian construction project with smart sensors (Chong and Diamantopoulos, 2020). The smart sensors are Bluetooth low energy embedded devices that capture location and status information across the supply chain. Data is then fed into a smart contract to trigger automatic payments.

Hamledari & Fischer (2021) used reality capture technology, i.e. cameras and laser scanning devices, installed on drones and robotic ground vehicles to monitor construction site progress. They then stored this data in a distributed file system and connected it to payouts and transfer of tokenized lien rights for contractors through a smart contract.

Also, Lee, at al. (2021) connected automatic smart contract payouts to robotic construction. They connect the digital twin of a robot fed by data of the robotic sensors to identify when a defined work task, in this case the positioning of a brick, is completed to trigger the payments. Similarly, Hunhevicz, et al. (2022b) connect smart building sensors to a performance based smart contract via the digital building twin of the house. Finally, the no1s1 prototype for a self-owning house connects the installed sensors to the blockchain over a mini-board computer by automatically triggering transactions to the smart contracts (Hunhevicz et al., 2021).

**Hybrid Oracles**

This category combines data sources, humans and automated Oracles which introduces further need for exploration on governance and (de)centralization of the Oracle. Within this hybrid encapsulation, a range of hardware-software-human mix can be conceived, in the sense that projects might need a range of dimensions to be combined, so that they achieve the best governance of data required for the task. Governance seems to be the main driver for selecting this type of Oracle, as one might allow humans to make decentralised decisions, while incorporating data or hardware input to assist human decisions. Lu, et al. (2021) explore blockchain Oracles through smart construction objects for supply chain management. The work is one of the first explorations on decentralized Oracles for AEC and proposes an Oracle smart contract for data selection and validation on- and off chain via stakeholder peers and smart construction object peers (Lu et al., 2021). Also, Dounas et al (2022) present an abstract high level connection between Oracles in BIM and token pools on smart contracts, where the performance of a building model in terms of architectonic functionality, carbon and waste reduction results in increased payments to the design team that is developing the models. Their high-level abstraction discusses the development of a data cycle that feeds into an evaluating smart contract multiple times when the performance of the model is improved (Dounas et al 2022). In this example, the Oracle triggers the performance of a contract through combining design governance of an abstract building model. The governance decisions are taken by human and software design agents in a feedback loop with the smart contract that regulates the incentives for the design collaboration.

## Emerging Oracle Design Dimensions

The previous sections showed that Oracles can be designed differently based on e.g., the data source, trust model, design pattern, and interaction type (Beniiche, 2020) of a use case in the AEC industry. However, current blockchain research in the AEC industry rarely gives justification for the implemented Oracle type or discusses possible alternative implementations for an investigative use case. A well-designed Oracle is the foundation for trusted data in blockchain applications. Further work is needed on how to implement Oracles in construction to reach appropriate trust levels for data in each use case.

Since we realized that it is hard to directly map the somewhat abstract categories proposed by Beniiche on the examined use cases in the AEC industry, we instead propose three emerging design dimensions for the design of Oracles in AEC use cases. By focusing on those, the paper simplifies and structures thinking about Oracle implementation for researchers and practitioners alike.

**Dimension 1: Human vs. Machine**

The first design dimension is the human – machine spectrum (Figure 3). It was the most broadly applicable category that we could classify for current Oracle implementations in the AEC industry. Therefore, we propose to keep it as an initial design decision that a designer of a blockchain application for construction needs to take, i.e., whether the Oracle system is purely based on human input, machine input, or a combination of both, which we refer to as hybrid.

This decision depends on a variety of criteria. Most obvious, the kind of data needed for the blockchain processes and the levels of automation desired. For example, information about the current weather could be obtained by either a human observing the weather or a weather station. Both would work, but one might be the

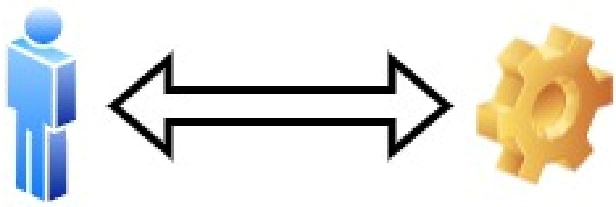

*Figure 3: Human-machine Oracle design dimension.*

more desirable option regarding automation with implications to cost or speed. In other cases, only a machine or a human Oracle would work. If both types are an option, a more subtle criterion is the expected level of trust when using either a human or machine Oracle. A hybrid approach is likely best-suited, e.g., the machine delivering the data and a human checking upon the data.

The technology stack and data security also play a big role in the trustworthiness of the Oracle system but are not described here in detail. Human input requires a well-designed and secure front end so the user can connect the wallet to input information and sign the transaction. Machine input requires a secure data pipeline from the sensor to the machines running the middleware aggregating data and signing a transaction. Secure can mean encrypted on a data level, protected from physical access to the hardware, or protected from cyber threats when connected to the internet.

Overall, AEC applications need to consider for each use case whether human or machine inputs are more applicable given constraints like cost, speed, trust, and technical implementation.

**Dimension 2: Single vs Multiple**

After thinking about whether to use human or machine Oracles, a second important dimension is how many humans or machines should be involved in the Oracle solution (Figure 4). This is a scale ranging from one to hundreds of humans or machines.

Trust may increase with a proportional increase of the number of data points or verifiers in an Oracle system. Cross-validating data among various sources is possible and the solution is therefore less dependent on a single data source. An increased number of data points from different sources also contributes to decentralizing the Oracle and overall blockchain system. Many blockchain-based systems take expensive measures to ensure decentralization (independency from single actors) for the trustworthiness of transactions. If a use case intends to use the decentralized blockchain environment but the Oracle data input is not trustworthy or centralized, it compromises the trustworthiness of the overall use case implementation.

As an example, trust in the outcome of the Oracle system increases if a data batch input has been validated by a high percentage of experts. The same is even more true for machine input where we can model and predict the trust vector of the machine output. For example, if out of 100 temperature sensors 98 show a particular temperature for a room, then this increases the trust factor. Manipulation of one sensor is easier than simultaneously manipulating 100 sensors.

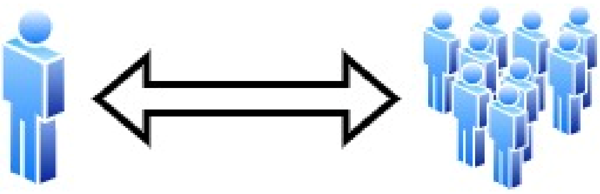

*Figure 4: One-Multiple Oracle design dimension.*

When implementing an AEC use case, one should think about the appropriate amount of data inputs required to generate a trustworthy Oracle system. At this point, we intentionally do not claim that more data sources are always better. There might be good reasons why one or a few selected experts or machines are more trustworthy than many, but it should be justified why this is the case.

**Dimension 3: Ungoverned vs. Governed**

The last dimension is described as ungoverned vs. governed (Figure 5), describing the processes in place to make sure trustworthy data from one or many humans or machines enters the blockchain system. A governance process could define which combination of machines and humans can be used, along with a decision on what kind of data types, their frequency, but also the sequence of data validation before passing a particular data batch to a smart contract on the blockchain for processing. Note that we are more concerned with validating the data for blockchain input rather than the transmission from the blockchain to other cyber-physical systems because data input has increased chance for data misrepresentation, errors, and attacks.

The main decision on Oracles is whether a governance process is needed. Most of the reviewed AEC blockchain papers implemented ungoverned Oracle systems. If the first two dimensions are already designed to ensure trustworthy data input, a governance process may not be needed. However, in many cases a governance process is desirable and the implementation mechanisms will depend heavily on the use case. A near unlimited number of governance mechanisms exist, both off-chain before entering the blockchain and on-chain through smart contracts and cryptoeconomic mechanisms after data is in the blockchain system.

Examples of off-chain governance mechanisms include monitored decision-making by involved stakeholders on the frequency of transaction execution, deciding which humans or machines are able to control addresses with permissions to write to the smart contract, or regulations for the type and quality of data produced by humans and machines. The advantage of off-chain governance significantly reduces the data input to blockchain systems and therefore transaction costs.

For example, Hunhevicz et al. (2022b) determined the frequency and random selection of building sensors with a middleware to reduce the number of transactions to the smart contract. Also, Chainlink implements additional data validation mechanisms by humans through API-

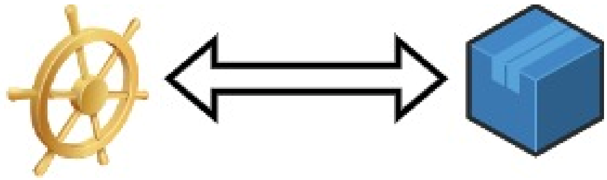

*Figure 5: Ungoverned-Governed Oracle design dimension.*

based input before entering data onto the blockchain. One potential problem with off-chain governance mechanisms is that it could still lead to wrong or unauthorized data inputs to the blockchain. The governance process is not transparent and could still be attacked or bypassed without notification.

Alternatively, on-chain governance can implement mechanisms so that they are visible and transparent on the blockchain to everyone using smart contracts. For example, a data validation pattern could be implemented via a multi-signature smart contract that demands a certain number of valid signatures before executing an action. Many potential mechanisms of using smart contracts to implement governance processes exist and are usually referred to as cryptoeconomic mechanisms. Hunhevicz, et al. (2022a) identifies potential cryptoeconomic mechanisms for the governance of project delivery. Some identified mechanisms that could also be useful for the governance of data input are:

- Smart contract voting mechanisms using either address-based or token-based weighting. Authorized users controlling an address, or a token can vote on certain governance decisions, the validity of certain Oracles or other matters.
- Verification mechanisms incentivizing validators through automated e.g., monetary rewards or reputation rewards to participate and behave honestly.
- For more uncertain data, prediction markets such as Augur could be an interesting implementation (Peterson et al., 2018).

Listing and describing a complete list of possible cryptoeconomic mechanisms goes beyond the scope of this paper and should be subject of further research.

An on-chain governance system is likely useful if the ecosystem involves many decentralized data sources passing data to the blockchain that are hard to coordinate off-chain in a trusted way. Moreover, they could be useful when operating Oracles needs to be incentivized so that trust, security, or other desirable traits are accomplished. In essence, a well-designed on-chain governance mechanism can reduce potential attack vectors and increase the trust dimension of the Oracle system without the need to know the location or entity doing the data input.

However, on-chain governance systems are not trivial to design (Voshmgir and Zargham, 2020). The purpose and scope of the system need to be investigated thoroughly by the designer, along with a map of when interactions are occurring, orchestrated according to the time when data exchange takes place, its purpose, and its stakeholders.

Realistically, a combination of off-chain and on-chain governance is desirable dependent on the use case. On-chain governance mechanisms should mainly be introduced when decentralization or a multitude of incentives are present and one needs a collective incentive alignment. For most other cases, no on-chain or off-chain governance is likely enough to accomplish the purpose of the Oracle.

## Conclusions

While blockchain has many promising applications in the AEC industry that could increase the industry's transparency and trustworthiness, most research on blockchain in AEC has not focused on how Oracles connect trusted applications to real data. Without well-designed Oracles, trust in the whole blockchain solution could be compromised. Trust is especially applicable to the AEC industry which builds physical products that use digital applications with real world data input. Therefore, it is important that more research investigates how to design trusted Oracles for AEC use cases.

To kickstart more discussion and research around Oracles in AEC blockchain applications, this paper provides an overview and guidelines on which factors to focus on. While the reviewed taxonomies mainly created discretized categories post-hoc for existing Oracle implementations, these categories provide little help to build new Oracle implementations. For example, how to decide on the level of centralization or decentralization of an Oracle when thinking about a use case. This paper proposes three design dimensions that intend to simplify thinking about important aspects when implementing Oracles: Human vs. Machine, One vs. Many, Ungoverned vs. Governed (Figure 6). The authors believe this is a more practical way to consider important design dimensions to satisfy requirements of the technical solution. When having specified these, the subsequent technical implementation will naturally consider the specified Oracle classification categories from previous taxonomies.

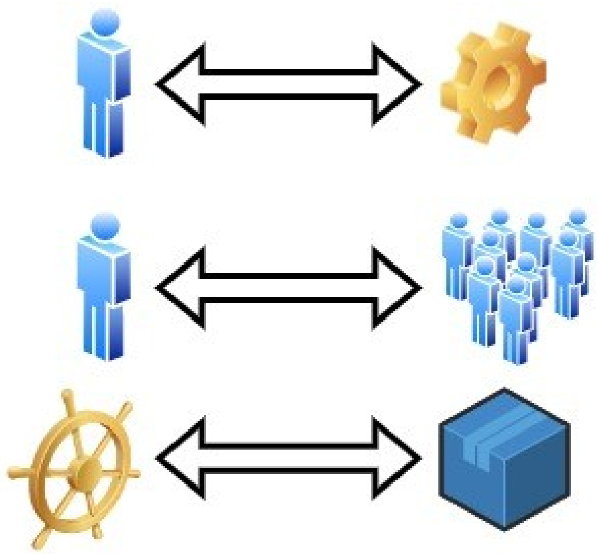

*Figure 6: Summary of proposed emerging Oracle design dimensions: Human vs. Machine, One vs. Multiple, Governed vs. Ungoverned*

Nevertheless, the provided emerging design dimensions could be further detailed and should be considered more as a starting point for an elaborated decision framework. As a next step research could try to map existing technical solutions with the respective decisions in each design dimension. For implementation many Oracle platforms are available to be used for AEC use cases instead of implementing an Oracle system from scratch, e.g. Chainlink ("Blockchain Oracles for Hybrid Smart Contracts | Chainlink," March 2023), API3 ("API3 | The Web3 API Economy," March 2023.), or Band Protocol ("Band Protocol - Cross-Chain Data Oracle," April 2023).

In addition to improving this work, further research may explore validation and governance systems for Oracle data input in the AEC industry. Validation and governance are important factors often overlooked in literature that will likely play an ever-increasing role in blockchain and smart contract prototypes for the AEC industry. Next to the described off-chain approaches, on-chain governance through cryptoeconomic mechanisms seems promising to make Oracles more trustworthy in AEC and should be a subject of further research efforts, more so when considering the physical dimension of the AEC industry.

Overall, this paper emphasizes the importance of well-designed Oracles in the AEC industry and the challenges in doing so. The provided overview and design dimensions then inspire more researchers to investigate and discuss Oracle applications in AEC.

## References


Ahmadisheykhsarmast, S., Sonmez, R., 2020. A smart contract system for security of payment of construction contracts. Autom. Constr. 120, 103401. https://doi.org/10.1016/j.autcon.2020.103401

Al-Breiki, H., Rehman, M.H.U., Salah, K., Svetinovic, D., 2020. Trustworthy Blockchain Oracles: Review, Comparison, and Open Research Challenges. IEEE Access 8, 85675–85685. https://doi.org/10.1109/ACCESS.2020.2992698

API3 | The Web3 API Economy [WWW Document], n.d. URL https://api3.org/ (accessed 1.31.23).

Band Protocol - Cross-Chain Data Oracle [WWW Document], n.d. URL https://bandprotocol.com/ (accessed 1.31.23).

Beniiche, A., 2020. A Study of Blockchain Oracles. https://doi.org/10.48550/arXiv.2004.07140

Blockchain Oracles for Hybrid Smart Contracts | Chainlink [WWW Document], n.d. URL https://chain.link/ (accessed 1.31.23).

Caldarelli, G., 2022. Overview of Blockchain Oracle Research. Future Internet 14, 175. https://doi.org/10.3390/fi14060175

Celik, Y., Petri, I., Barati, M., 2023. Blockchain supported BIM data provenance for construction projects. Comput. Ind. 144, 103768. https://doi.org/10.1016/j.compind.2022.103768

Chong, H.-Y.Y., Diamantopoulos, A., 2020. Integrating advanced technologies to uphold security of payment: Data flow diagram. Autom. Constr. 114, 103158. https://doi.org/10.1016/j.autcon.2020.103158

Dounas, T., Jabi, W., Lombardi, D., 2021. TOPOLOGY GENERATED NON-FUNGIBLE TOKENS - Blockchain as infrastructure for a circular economy in architectural design. Presented at the 26th CAADRIA Conference, Hong Kong, pp. 151–160.

Dounas, T., Lombardi, D., Jabi, W., 2020. Framework for decentralised architectural design BIM and Blockchain integration. Int. J. Archit. Comput. https://doi.org/10.1177/1478077120963376

Gunasekara, H.G., Sridarran, P., Rajaratnam, D., 2021. Effective use of blockchain technology for facilities management procurement process. J. Facil. Manag. https://doi.org/10.1108/JFM-10-2020-0077

Hamledari, H., Fischer, M., 2021. The application of blockchain-based crypto assets for integrating the physical and financial supply chains in the construction & engineering industry. Autom. Constr. 127, 103711. https://doi.org/10.1016/j.autcon.2021.103711

Hunhevicz, J.J., Brasey, P.-A., Bonanomi, M.M.M., Hall, D.M., Fischer, M., 2022a. Applications of Blockchain for the Governance of Integrated Project Delivery: A Crypto Commons Approach. https://doi.org/10.48550/arXiv.2207.07002

Hunhevicz, J.J., Motie, M., Hall, D.M., 2022b. Digital building twins and blockchain for performance-based (smart) contracts. Autom. Constr. 133, 103981. https://doi.org/10.1016/j.autcon.2021.103981

Hunhevicz, J.J., Wang, H., Hess, L., Hall, D.M., 2021. no1s1 - a blockchain-based DAO prototype for autonomous space, in: Proceedings of the 2021 European Conference on Computing in Construction. University College Dublin, pp. 27–33. https://doi.org/10.35490/ec3.2021.185

Lee, D., Lee, S.H., Masoud, N., Krishnan, M.S., Li, V.C., 2021. Integrated digital twin and blockchain framework to support accountable information sharing in construction projects. Autom. Constr. 127, 103688. https://doi.org/10.1016/j.autcon.2021.103688

Li, X., Wu, L., Zhao, R., Lu, W., Xue, F., 2021. Two-layer Adaptive Blockchain-based Supervision model for off-site modular housing production. Comput. Ind. 128, 103437. https://doi.org/10.1016/j.compind.2021.103437

Lu, W., Li, X., Xue, F., Zhao, R., Wu, L., Yeh, A.G.O., 2021. Exploring smart construction objects as blockchain oracles in construction supply chain management. Autom. Constr. 129, 103816. https://doi.org/10.1016/j.autcon.2021.103816

Pasdar, A., Dong, Z., Lee, Y.C., 2021. Blockchain Oracle Design Patterns. https://doi.org/10.48550/arXiv.2106.09349

Peterson, J., Krug, J., Zoltu, M., Williams, A.K., Alexander, S., 2018. Augur: a decentralized oracle and prediction market platform. https://doi.org/10.13140/2.1.1431.4563

Sadawi, A.A., Hassan, M.S., Ndiaye, M., 2022. On the Integration of Blockchain With IoT and the Role of Oracle in the Combined System: The Full Picture. IEEE Access 10, 92532–92558. https://doi.org/10.1109/ACCESS.2022.3199007

Schneier, B., 1996. Applied Cryptography: Protocols, Algorithms, and Source Code in C. Second Edition, John Wiley and Sons, New York.

Tezel, A., Febrero, P., Papadonikolaki, E., Yitmen, I., 2021. Insights into Blockchain Implementation in Construction: Models for Supply Chain Management. J. Manag. Eng. 37, (ASCE)ME.1943-5479.0000939. https://doi.org/10.1061/(ASCE)ME.1943-5479.0000939

van Groesen, W., Pauwels, P., 2022. Tracking prefabricated assets and compliance using quick response (QR) codes, blockchain and smart contract technology. Autom. Constr. 141, 104420. https://doi.org/10.1016/j.autcon.2022.104420

Voshmgir, S., Zargham, M., 2020. Foundations of Cryptoeconomic Systems. Found. Cryptoeconomic Syst., Working Paper Series / Institute for Cryptoeconomics / Interdisciplinary Research.