https://doi.org/10.1016/j.autcon.2021.103816

# Exploring Smart Construction Objects as Blockchain Oracles in Construction Supply Chain Management

Weisheng Lu, Xiao Li, Fan Xue, Rui Zhao, Liupengfei Wu, and Anthony G.O. Yeh

## Abstract

Blockchain technology has attracted the interest of the global construction industry for its potential to enhance the transparency, traceability, and immutability of construction data and enables collaboration and trust throughout the supply chain. However, such potential cannot be achieved without blockchain "oracles" needed to bridge the on-chain (i.e., blockchain system) and off-chain (i.e., real-life physical project) worlds. This study presents an innovative solution that exploits smart construction objects (SCOs). It develops a SCOs-enabled blockchain oracles (SCOs-BOs) framework. To instantiate this framework, the system architecture of a blockchain-enabled construction supply chain management (BCSCM) system is developed and validated using a case study, whereby four primary smart contracts are examined in the context of off-site logistics and on-site assembly services. The validation results show that accurate data is retrieved against malicious data in each request, and the corresponding reputation scores are successfully recorded. The innovativeness of the research lies in two aspects. In addition to mobilizing SCOs as blockchain oracles to bridge the on-chain and off-chain worlds, it develops a decentralized SCO network to avoid the single point of failure (SPoF) problem widely existing in blockchain systems. This study contributes to existing research and practice to harness the power of blockchain in construction.

## Keywords

Blockchain; Oracles; Smart Contract; Supply Chain Management; Smart Construction Objects; Prefabricated Construction

## 1. Introduction

The construction industry is one of the most fragmented sectors globally due to a high degree of specialization amongst its professions, businesses and processes, and most construction projects are one-offs involving adversarial relationships (Turk and Klinc, 2017). As a result, construction projects worldwide have a scattered and complex supply chain (Dainty et al., 2001). For example, the construction of a two-tower student hostel at the University of Hong Kong involved more than 20 suppliers throughout China, the transportation of prefabricated modules over 1500 miles from Jiangsu and, at its peak, over 200 workers on the supply chain (Zhang et al., 2021). Collaboration is needed to manage such a fragmented supply chain, and accountability is needed to ensure the sharing of trustworthy data on progress, quality, safety, costs, payments, and resources (Tezel et al., 2020). A lack of accountability gives rise to disputes, deflection of blame, and corner-cutting, which in turn lead to depressed productivity, cost overrun, and accidents (Zhong et al., 2020). Blockchain technology can offer this missing accountability in construction and make supply chains traceable, transparent, and immutable for all project participants (Wan et al., 2020).

Blockchain is a decentralized trust infrastructure that combines distributed ledgers, cryptography, and consensus protocols. It can track and store the past and present status of tangible assets or intangible events across a decentralized peer-to-peer network. Data is saved in a set of blocks linked in a consecutive chain. Cryptographic protocols prevent any change in the data stored in a block without the collusion or collaboration of the majority of participants (Kosba et al., 2016). Smart contracts can run on a blockchain, implement consensus protocols, and allow participants to reach a consensus-based on predefined rules without a trusted third party (Li et al., 2019a) (Yang et al., 2020). Several applications for blockchain-enabled smart contracts have been explored in different domains, including document control (Hasan and

Salah, 2018a), delivery assurance (Hasan and Salah, 2018b), Internet of Things (IoT) access control and authentication (Salah et al. 2019; Almadhoun et al. 2018). In the construction industry, applications have been explored in payment security (Chong and Diamantopoulos, 2020), quality management (Sheng et al., 2020), traceability of prefabricated components (Wang et al., 2020), Building Information Modeling (BIM) data audit (Zheng et al., 2019; Xue and Lu, 2020), and integrated project delivery (IPD) transactions (Elghaish et al., 2020). However, the execution of smart contracts in the construction supply chain often requires an exchange of real-world data, which cannot be accomplished by blockchain (Wang et al., 2017).

Oracles are middleware agents that can capture and validate real-world information and feed it to a blockchain for the use of smart contracts (Al-Breiki et al., 2020). They may be software or hardware, inbound or outbound, or consensus-based. While humans can serve as oracles to trigger communication between the on-chain and off-chain worlds, this is interruptive, time-consuming, and error-prone. The use of software oracles, such as BIM manipulated by human operators, is possible but ensuring the authenticity of external data sources is challenging. Software oracles also bring back the blockchain centralization problem since relying on centralized sources increases the risk of feeding erroneous data to the blockchain system. There is a need for oracles with autonomous and decentralized computational power scattered in key construction processes or nodes to verify construction data correctness and accuracy before it is fed into a smart contract, and to ensure data privacy.

Smart embedded technologies have excellent potential as blockchain oracles. Specifically, according to a concept developed by Niu et al. (2016), construction resources (e.g., machinery, tools, devices, materials, components, and structures) can be turned into smart construction objects (SCOs) able to convey their designated properties and with new properties of awareness, communicativeness, and autonomy to enable various smart applications. Thus, opportunities exist to mobilize SCOs into hardware oracles to bridge the communication between blockchain and real-life construction processes. However, this research area is uncharted territory.

This study, therefore, aims to investigate the extended use of SCOs as trustworthy hardware oracles for blockchain applications in the construction industry. It has three specific objectives: (1) to establish a deployment framework for exploring SCOs as decentralized blockchain oracles; (2) to instantiate the framework by proposing a system architecture of SCOs-as-oracle blockchain-enabled construction supply chain management (BCSCM); and (3) to validate the framework and system architecture in a case study. The study makes three main contributions to the body of knowledge. First, it is one of the first investigations on decentralized blockchain oracles in the construction industry to improve the single point of failure (SPoF). Second, to achieve data selection and validation through on- and off-chain interactions, the study presents the oracles smart contract (OSC) with an unbiased random sortation mechanism and the aggregator smart contract (ASC) with cross-reference mechanisms. Third, to realize the cross-chain activities between the main (service blockchain) and side chain, it develops the reputation smart contract (RSC) with the reputation system and service smart contract (SSC). The rest of the paper is organized as follows. After this introductory section is Section 2, which elaborates on some basics of blockchain and its oracles. Section 3 delineates the SCOs-enabled blockchain oracles (SCOs-BOs) framework, and Section 4 describes the system architecture of the BCSCM system. Section 5 is a case study that uses logistics and on-site assembly traceability services to validate the smart contracts used in the BCSCM system. Our findings are discussed in Section 6, and conclusions are drawn in Section 7.

## 2. Background

### *2.1 Construction Supply Chain Management*

Construction supply chain management (CSCM) aims to ensure the smooth flow of goods and services to the construction site through cooperation between supply chain participants (Dainty et al., 2001). CSCM is massively challenging in practice due to long-standing issues, including lack of trust, fragmentation, and discontinuity (Luo et al., 2020). Product provenance issues and disputed inspection of products contribute to the lack of trust (Bankvall et al., 2010), while fragmentation issues result from a geographical distribution of stakeholders and multiple CSCM stages (Hsu et al., 2019). Discontinuity arises because the current CSCM system lacks good-quality data for coordinated functional modules such as compliance check, process control, and quality assurance, and can also be ascribed to low levels of information visibility and traceability. For example, product data is still mainly conveyed from the prefabrication factory to the construction site on paper (Zhai et al., 2019). Such manual processes are time-consuming and may lead to input errors, file loss, and data tampering. Integrating IoT and BIM to connect physical construction objects with virtual BIM objects has been proved to alleviate the fragmentation and discontinuity of CSCM (Li et al., 2019b). Researchers and practitioners have proposed many technologies to realize IoT, such as radio-frequency identification (RFID), near-field communication (NFC) for short-range wireless (Xue et al., 2018), 5G for medium-range wireless (Li et al., 2018b), and low-power wide-area networking (e.g., LoraWan, NB-IoT) for long-range wireless (Mekki et al., 2018). In construction, Niu et al. (2016) proposed a robust IoT model in the form of smart construction objects (SCOs). The integration of SCOs with BIM has been recognized as a compelling paradigm for digital twin applications to enhance construction efficiencies. These applications have been widely explored in construction resource and progress monitoring (Li et al., 2018a; Zhong et al., 2017), occupational health and safety management (Niu et al., 2019), and construction logistics and supply chain management (Niu et al., 2017). However, merely integrating BIM and SCOs is not enough to achieve data privacy, security and trust among stakeholders. For example, the shared cloud BIM model and its data can be tampered with, leaving data changes untraceable, and SCO sensors (e.g., RFID, GPS) may suddenly run out of power or report noise that reduces data quality.

### *2.2 Blockchain and Smart Contracts*

A potential solution to the above issues is blockchain, a distributed ledger of pertinent data and transactions mutually agreed upon and shared among all participants in a peer-to-peer network (Nakamoto, 2008). Three components support the functioning of blockchain: cryptography, a distributed database, and a consensus mechanism (Zheng et al., 2017). Cryptography, in the form of hashing algorithms, is used to encrypt transactional data based on the agreed protocol, making the data difficult to tamper with (Beck et al., 2016). A widespread network of computers supports distributed ledgers, recording all data in each participant's ledger. According to the consensus, data transactions are kept synchronized across the network (Nguyen and Kim, 2018). Current blockchain platforms include permissionless blockchain and permissioned blockchain (Helliar et al., 2020). A permissionless blockchain, such as Bitcoin or Ethereum, is entirely decentralized and allows any participant to access the data in blocks (Buterin, 2014). In permissioned blockchain, such as Hyperledger Fabric, only identified users can validate transactions and access block data (Cachin, 2016). Permissionless blockchain highlights openness and decentralization, while permissioned blockchain can provide higher throughputs by designing deterministic consensus protocols (Gupta et al., 2020). Therefore, permissioned blockchains are more applicable for time-sensitive CSCM applications in terms of transparency, traceability, immutability, decentralization, privacy, and smartness (Qian and Papadonikolaki, 2020).

The first application of blockchain, or “Blockchain 1.0”, was in cryptocurrencies, while the technology has expanded into other sectors through smart contracts, the primary advancement of “Blockchain 2.0” (Buterin, 2014). Smart contracts are self-executing contracts that run on an “if-then” basis (Hasan and Salah, 2018a). They can track on-chain or cross-chain data changes and off-chain data sources in real-time and automatically respond under preset trigger conditions (Uriarte et al., 2020). Smart contracts can be either deterministic or non-deterministic (Kosba et al., 2016). The former, such as tokenization of assets, can be independently executed in the blockchain without interaction with the external world. Non-deterministic smart contracts, as in the case of construction industry applications, require off-chain data to trigger execution. For example, the location data of a prefabricated product can be captured from its mounted GPS sensors. When the product arrives at the construction site, the smart contract can extract the off-chain location data as proof of location to activate the blockchain’s product status change. To facilitate this data exchange between the on-chain and off-chain worlds, blockchain oracles can serve as intermediaries. Without blockchain oracles, smart contracts have to trust only data already within the chain, and the functions of blockchain would be seriously constrained.

### *2.3 Blockchain Oracles*

In ancient Greece, an oracle was a messenger passing advice or prophecies from gods to mortals. In modern society, any accurate source of information can be considered an oracle (Al-Breiki et al., 2020). In blockchains, an oracle is a middleware agent that queries, verifies, and authenticates external data sources and then delivers them to the blockchain for subsequent use by smart contracts (Kochovski et al., 2019). The data transmitted by oracles in CSCM processes include workers’ health and safety information, operation and energy information from machinery, location and quality data from material and components, cost and progress status, and building information contained in BIM models. Oracles can also be classified according to the source of data (software, hardware, human), information flow direction (inbound, outbound), design pattern (request-response, publish-subscribe, immediate–read), and trust model (centralized, decentralized) (Beniiche, 2020).

Oracles are not a built-in functionality of blockchain and do not have consensus mechanisms. 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. Previous studies have explored the use of oracles to improve data quality and authenticity in CSCM. Shrestha and Behzadan (2018) developed an evolutionary algorithm to refine sensor data noise and enhance data quality for better construction planning simulation, while Bangaru et al. (2020) found that multiple sensors or combined sensors can achieve higher accuracy in activity classification than individual sensors. Addressing data authenticity, Chong and Diamantopoulos (2020) used smart sensors as hardware oracles and integrated them with smart contracts to improve payment security. Zheng et al. (2019) considered BIM as software oracles for blockchain to store historical processes of file modification. Involving human oracles, Wang et al. (2020) used smart contracts in blockchain to update precast components status and operation information, and Sheng et al. (2020) explored smart contracts for handling construction quality information such as inspection forms. However, the quality and authenticity of off-chain data before input to the blockchain have not yet been investigated, and some CSCM process data are noisy or miscellaneous in nature.

Several studies have investigated blockchain oracles in other industries to achieve proof of location. For example, Vivekanandan et al. (2021) proposed an IoT device-to-device authentication protocol for smart city applications, facilitating the registration of IoT location data in the blockchain. Victor and Zickau (2018) stored location encoding systems in the smart contract to represent a geofence used to evaluate the location data provided by the oracle. Boeira et al. (2019) developed a scheme using cryptographic primitives and mobility awareness to improve the trustworthiness of shared vehicle location information in high-speed scenarios. Zafar et al. (2020) also summarized a state-of-the-art location proof system and highlighted current challenges such as collusion resistance (malicious location and noise) and storage (redundancy in blockchain but untrustworthy in distributed devices). Some off-the-shelf decentralized blockchain oracle solutions for commercial applications have been summarized in Al-Breiki et al. (2020), such as Witnet (De Pedro et al., 2017), Augur (Peterson et al., 2015), Chainlink (Ellis et al., 2017), ASTRAEA (Adler et al., 2018), and Aeternity (Hess et al., 2018) taking the reputation system, voting game, or consensus mechanism into account.

The research gaps identified can be summarized as follows: (1) there is a lack of a framework to guide the establishment of a decentralized hardware oracle sidechain for specific CSCM functions and to help card the logic of cross-chain (off-chain, sidechain, main chain) interactions; (2) the automatic consensus mechanism (e.g., cross reference), reputation system, and unbiased random sortation mechanism have not been investigated for hardware oracles in CSCM to avoid SPoF and help get trustworthy data in an empirical study.

## 3. A Framework for Using Smart Construction Objects as Blockchain Oracles

### *3.1 Transferring SCOs as Oracles: Definition & Properties*

Smart construction objects (SCOs), proposed by Niu et al. (2016), represent a robust IoT model with sensing, processing, and communicating capacities to facilitate information exchange among various construction resources. Here, construction resources could be men, machines, or materials. The core properties of SCOs are awareness, communicativeness, and autonomy

(Niu et al., 2017). Awareness shows the ability of SCOs to sense and record their real-time situation and that of the vicinity. Communicativeness indicates the power of SCOs to exchange information they have obtained through their awareness. Autonomy refers to the ability of SCOs to alert people of the need for actions or take actions autonomously based on predefined rules. These properties of SCOs are well-matched with the design patterns of blockchain oracles (see Table 1). For example, the activity aware with passive autonomy and pull communicativeness property is similar to the request-response pattern, where oracles can monitor, retrieve, and record data when the specific activity or event requests are triggered. The policy-aware with active autonomy and push communicativeness property is identical to the publish-subscribe pattern, where oracles can broadcast real-time conditions when the changes comply with rules and regulations. The process-aware with mixed autonomy and mixed communicativeness property works the same as the immediate-read pattern, where oracles can store data available for any immediate need in any construction process.

**Table 1.** The core and sub-properties of SCOs

| **Properties** | **Description** | **Oracles Design Patterns** |
|---|---|---|
| **Awareness** | | |
| **Activity aware** | To be aware of and respond when an activity or event is triggered | request-response |
| **Policy aware** | To be aware of situations compliant with published rules and regulations | publish-subscribe |
| **Process aware** | To be immediately aware of activities in workflows and processes | immediate-read |
| **Communicativeness** | | |
| **Pull** | To offer information on request | request-response |
| **Push** | To proactively issue updated information or alerts at regular intervals | publish-subscribe |
| **Mixed** | To immediately offer and issue information | immediate-read |
| **Autonomy** | | |
| **Passive** | To assist in making decisions and taking action upon request | request-response |
| **Active** | To proactively take action based on changes at regular intervals | publish-subscribe |
| **Mixed** | To execute autonomy in both passive and active manners | immediate-read |

SCOs can work in a similar way to oracles in construction blockchain. They can act as data feed providers to sense and capture data from various construction resources and scenarios and serve as oracle node operators to process and transfer accurate, reliable, and verified data to blockchain systems. In addition, IoT sensors installed on SCOs can serve other relevant purposes. For example, inertial measurement units (IMU) and air pressure units can supplement GPS locations with accurate motion and height data, while passive RFID and QR codes can be attached to construction objects for lifelong facility management. For different construction tasks, a combination of different SCOs design profiles can offer the optimal performance-price ratio. Figure 1 shows a detailed SCO plan for construction processes with two types of models:

*Model 1:* Low-energy, single GPS sensor for location-based service in off-site logistics

*Model 2:* High-frequency multiple motions and environmental sensors for off-site production and on-site assembly

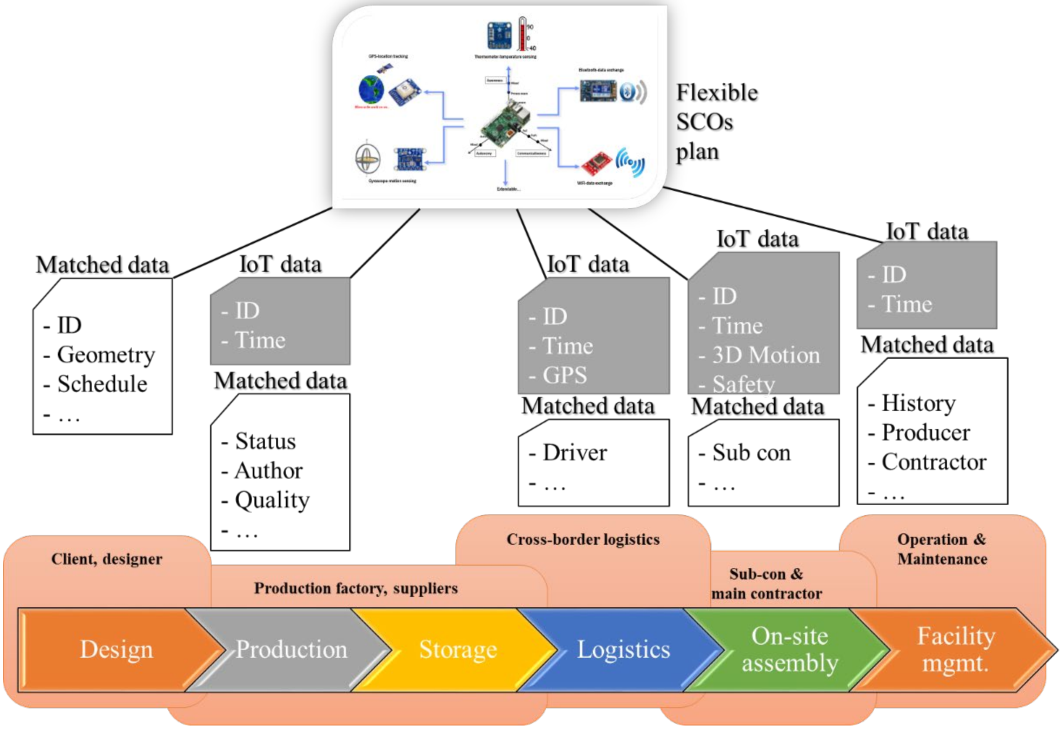


**Figure 1.** An SCO plan for CSCM processes: Off-site production, logistics, and on-site assembly services

### *3.2 The SCOs-BOs Framework*

This section proposes an SCOs as blockchain oracles (SCOs-BOs) framework providing a decentralized oracle solution with multiple smart contracts to manage interactions and data access. It can help randomly select and monitor the oracles for specific services in CSCM processes and offer reputation scores to each oracle by cross-reference. The framework tries to improve two bottlenecks in the data exchange between the physical CSCM processes and the cyber blockchain worlds: (1) single point of failure (SPoF), which means only relying on one source of information from the centralized BIM platform; and (2) the need for trustworthy, good-quality data.

The framework is shown in Figure 2 with the components described below.

- *Oracles pool*: Stakeholders can provide and register new SCOs in the oracles pool. The SCOs for specific services in the oracles pool are randomly selected and registered in the oracle smart contract (OSC) to form a decentralized oracle network.
- *Oracles sidechain*: Since oracles are not as reliable as blockchain, the sidechain is a substitution that also packages the oracles network as a side blockchain that communicates to the main blockchain (Singh et al., 2020) (Uriarte et al., 2020) (Li et al., 2021a). The sidechain includes the OSC and the aggregator smart contract (ASC). The former provides a frequently used interface to select SCOs from the oracles pool by using an unbiased sortation algorithm for specific construction management services (Zhou et al., 2019), and these selected SCOs can be registered in the OSC. This registration process is verified by all stakeholders. Their data is also hashed and returned to the ASC. The ASC receives all data hashes from the OSC, cross-references their hash values, and broadcasts reputation scores for each involved SCO to the reputation smart contract (RSC).

- *Service blockchain*: This includes a service smart contract (SSC) and an RSC. The SSC receives ASC reputation scores to compute and record the average reputation scores for all SCOs in the oracles sidechain and then selects the winning SCO. Also, the updated reputation scores are returned to the SCOs in the OSC. The SSC is triggered and can call upon the ASC's selection interface when the specific need arises (e.g., the logistics service is used to monitor prefabricated products' location status).
- *Stakeholder consensus*: Stakeholders in the CSCM processes can request data from the SSCs. All SSCs and registered SCOs in the OSC should reach a consensus from all stakeholders before execution in the services blockchain and oracles sidechain.

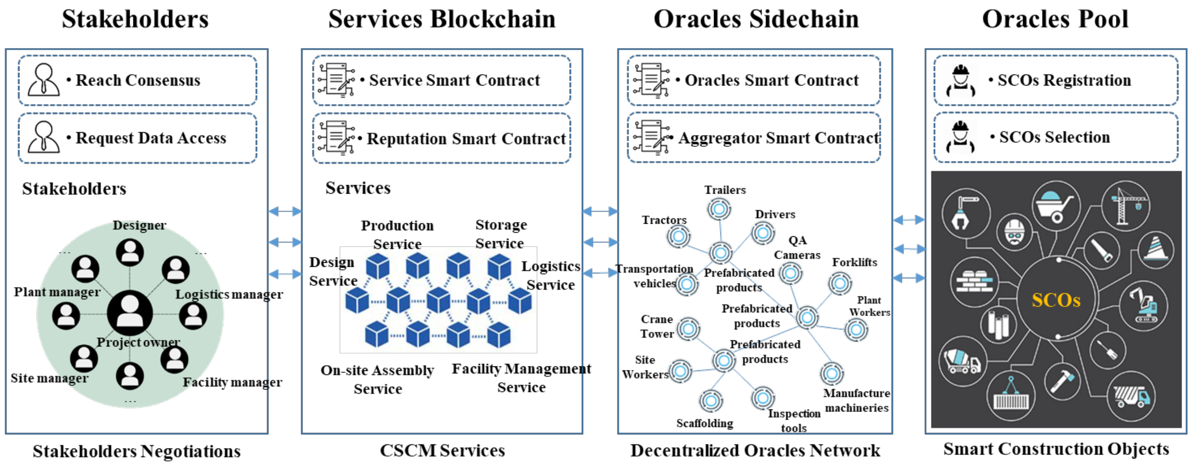


**Figure 2.** The SCOs-Bos framework

Interactions between components in the framework are summarized in the sequence diagram in Figure 3. These interactions can occur on-chain, cross-chain, or off-chain. Stakeholders are responsible for deploying SCOs and reach a consensus when the SSC is ready and selected SCOs are registered. For example, all the stakeholders can make a service consensus on the SSC in the logistics stage to monitor prefabricated products' real-time position status. The logistics manager should arrange for embedding GPS sensors (e.g., Model 1) into the prefabricated products, tractors, trailers, and drivers in the transportation process. Furthermore, these SCOs can be registered into the OSC, where they will be agreed upon by all stakeholders.

Stakeholders can send data requests to the SSC, and the SSC conducts permission verification. If the stakeholder has valid access permission, the request proceeds as follows:

- SSC forwards stakeholder request to ASC
- ASC invokes the interface of OSC, and OSC randomly selects from the off-chain oracles pool
- All selected SCOs can be registered into the OSC to form a decentralized sidechain, and each registered SCO hashes its data and sends it back to ASC
- ASC introduces a cross-reference method on all received data hashes from SCOs, gets the most similarity on returned hashes, and reports each SCO's reputation score to RSC
- RSC updates reputation scores for all SCOs in the oracles sidechain and selects a winner SCO based on the highest reputation score for SSC
- SSC delivers the data in the winner SCO to the stakeholders

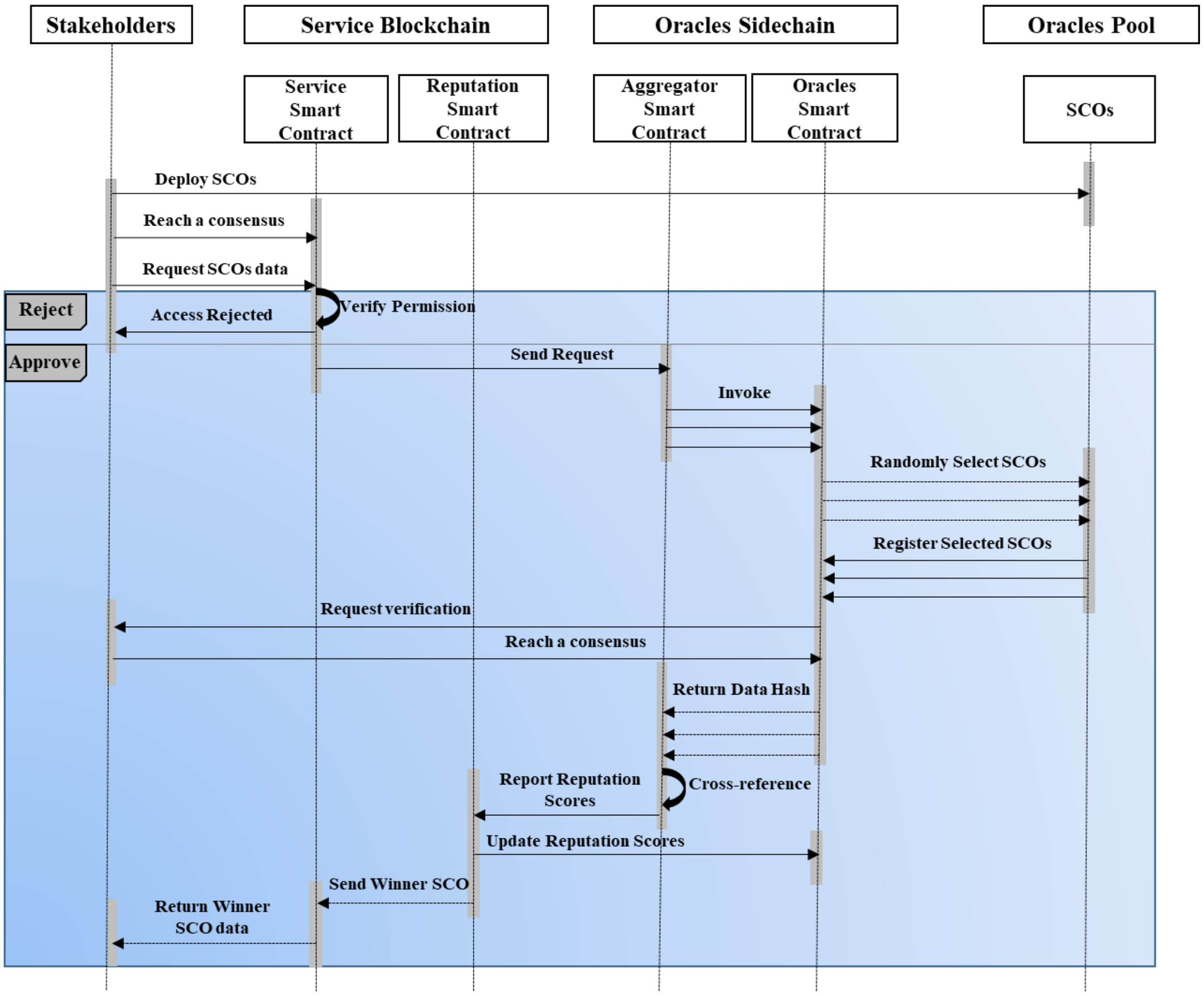


**Figure 3.** Sequence diagram for the proposed framework

## 4. System Architecture

This section instantiates the SCOs-BOs framework by developing and enriching a blockchain-enabled construction supply chain management (BCSCM) system. The supporting stakeholders and main services are explained below.

### *4.1 Overview*

Figure 4 shows the architecture of the BCSCM system, which comprises four main layers: (1) smart construction objects (SCOs), (2) oracles sidechain network, (3) service blockchain network, and (4) services.

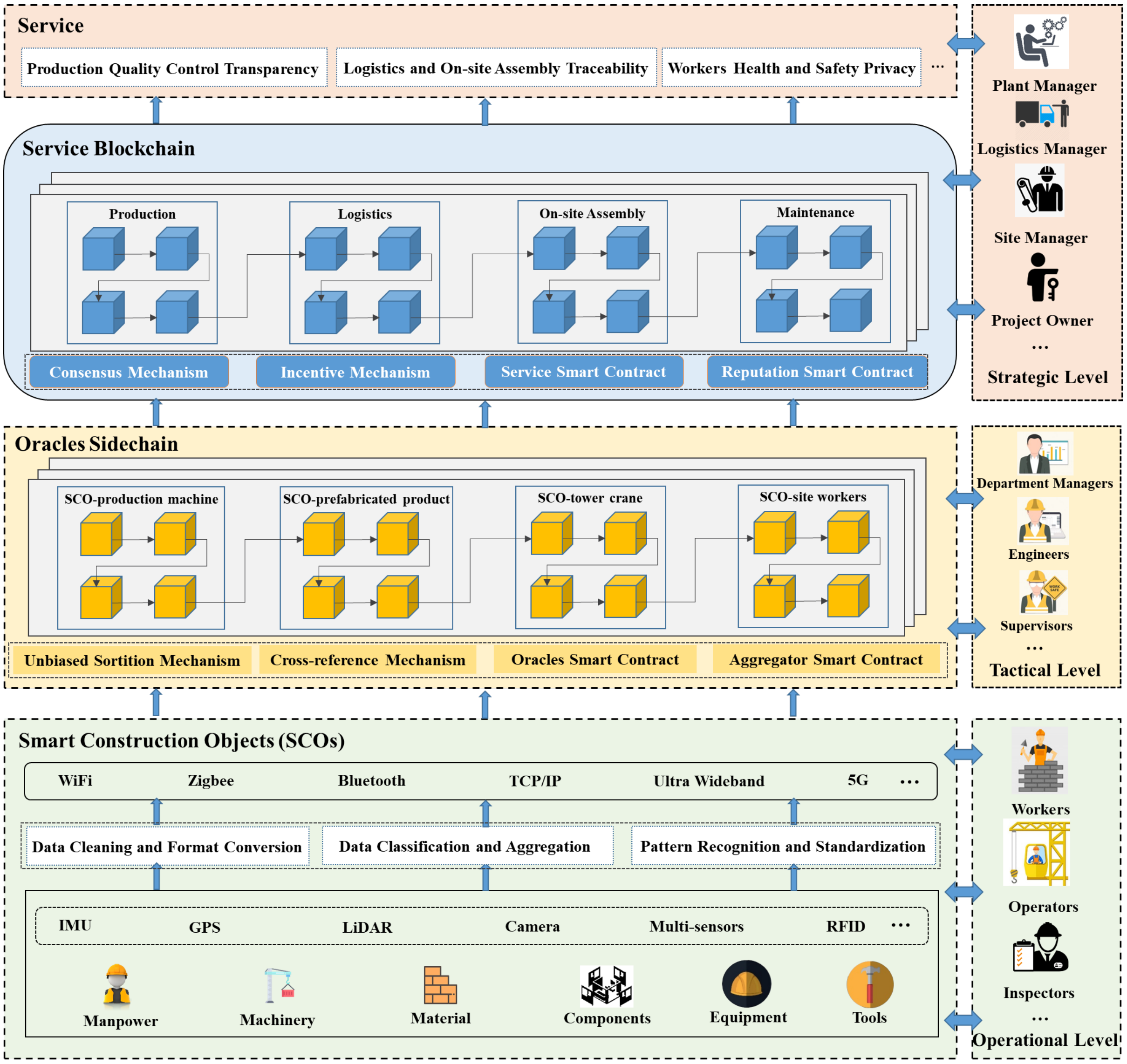


**Figure 4.** BCSCM system architecture

SCOs serve as the foundation of this architecture. SCOs are the construction resources in the BCSCM system and can be equipped with smart IoT devices (Zhong et al., 2020). For example, site workers' data, including heartbeat, heat stress, location, and motion, can be monitored and tracked using wearable devices such as smart wristbands, vests, and helmets. These collected construction data can be retrieved and broadcast through the SCO layer communication channels, such as ZigBee, Bluetooth, WiFi, Ultra-wideband, 5G, and Transmission Control Protocol/Internet Protocol (TCP/IP). The procedures of data cleaning and transformation, aggregation and classification, standardization, and pattern recognition are also processed in each SCO. This layer mainly has capacities in awareness, autonomy, and communicativeness of multimode data from different IoT sensors.

In the oracles sidechain network layer, an unbiased sortation algorithm is used to randomly select the SCOs for a specific service (e.g., production quality assurance), and these selected SCOs can be registered and used to form a set of blocks. Each block in the oracles sidechain comprises a header and the selected SCOs' data. A cross-reference method is applied in this layer to find out the authentic data and report their reputation scores (e.g., 0 or 100) based on similarity to the authentic data.

In the service blockchain network layer, a reputation system is adopted to elect the SCO with the highest reputation scores as the winner and updates the reputation scores of other SCOs in the oracles sidechain network layer. The data from the winner SCO are used to establish a set of blocks. Each block in this network comprises a header and winner SCOs' data, such as the location of prefabricated products, physiological signals of workers, and operation status of the tower crane. In the service blockchain network establishment process, a consensus mechanism named Practical Byzantine Fault Tolerance (PBFT) is used to ensure collective decision-making and reduce the faulty nodes' influence. An incentive mechanism is also devised to reward stakeholders who deploy SCOs in the construction processes.

Numerous service-oriented applications for the BCSCM system are deployed in the service layer, such as production quality control transparency, logistics and on-site assembly traceability, and workers' health and safety privacy. These services are supported by the service main blockchain and oracles sidechain using the data from SCOs.

### *4.2 Supporting Stakeholders*

In this BCSCM system, strategic-level, tactical-level, and operational-level stakeholders can support and facilitate implementation of related layers.

Frontline workers and operators are representative stakeholders at the operational level. All the SCOs' fresh data for the BCSCM system comes from operational tasks. For example, supported by deep cameras and laser scanners in the production plant, prefabricated products' dimensions and smoothness can be detected and recorded. These point cloud data can be further used for assembling the prefabricated products onsite. Operational-level stakeholders should verify authenticity of data in the selected and registered SCOs. For example, truck drivers should verify whether location signals of on-transport prefabricated products are consistent.

Department managers and construction engineers are representative stakeholders at the tactical level, aiming to ensure data-exchange reliability between oracles and blockchain by designing specific smart contracts with various algorithms and models. For example, the reputation mechanism and cross-reference method are designed by tactical stakeholders. These data can be further used for service blockchain when all tactical and operational stakeholders agree and verify the data from SCOs-BOs.

Project owners and senior managers such as plant managers, logistics managers, and site managers are the strategic-level representative stakeholders. They are accountable for forming the service blockchain, including the provision of SCOs, and the design of incentive and consensus mechanisms. They are also the decision-makers of the construction supply chain for specific strategies using the information from the service blockchain network.

### *4.3 Critical Services*

To instantiate the SCOs-BOs framework, three construction management services may be achieved through the proposed system architecture, as follows.

- *Production quality control transparency* means that quality inspection information for each prefabricated product production process is readily available to stakeholders. This service can facilitate remote stakeholders (e.g., contractors and project owners in Hong Kong) access detailed quality information from an off-site manufacturing plant (e.g., Jiangsu province). Deep cameras and laser scanners can work as SCOs-BOs to compute and retrieve quality inspection information for formworks (e.g., smoothness, cleanliness, and dimensions), steel reinforcing bars (e.g., size, pattern, fixing and layout, spacers, and concrete covers), concrete (e.g., placing and compaction), and finished products (e.g., surface, size, and dimensions, anchor bar). These data recorded in blockchain can improve production process inspection accountability.
- *Logistics and on-site assembly traceability* enable stakeholders to track construction resources and events' latest and historical status. For example, this service can assist site managers in monitoring the real-time location status and assembly progress of prefabricated products (Li et al., 2020). The low-energy, single GPS sensor for a location-based logistics service can be mounted in each prefabricated product. These location data can be cross-referenced as proof of location and the one with the highest reputation scores recorded in the blockchain for further decision-making.
- *Workers' health and safety privacy* relate to data including images or sensor signals of fatigue, unsafe motions, heartbeat, heat stress, and locations, which can be captured and processed by the SCOs-BOs. Furthermore, Federated learning is used for decentralized SCOs-BOs (Yang et al., 2019) (Li et al., 2021b), where a model can be trained by using the local health and safety data samples in each SCO-BO without extracting them.

Instead, only the high-level insights from the data are retrieved and stored in the blockchain.

## 5. Case Study

A case study using cross-border logistics and supply chain management, focusing on the services of off-site logistics and on-site assembly traceability, is conducted to verify the SCOs-BOs framework and the system architecture of the BCSCM system. According to Wan et al. (2020), blockchain can offer clear accountability and make the construction supply chain more traceable, transparent, and immutable among all participants involved in a project. In this case, low-energy, single GPS sensors mounted in prefabricated beams as SCOs were transported from mainland China to Hong Kong for a prefabricated construction project comprising five high-rise public housing residential towers surrounding one commercial center. To validate the proposed SCOs-BOs framework and system architecture, we implement four primary smart contracts (SSC, RSC, ASC, OSC) using the GPS data of prefabricated beams for the commercial center. Although this project is complete (nine highlighted beams have been erected as shown in Figure 5), the full record of GPS data (shown in the green data list of Figure 6) for the nine beams (C1023–C1031) in the same batch can be put into the oracles pool for validation.

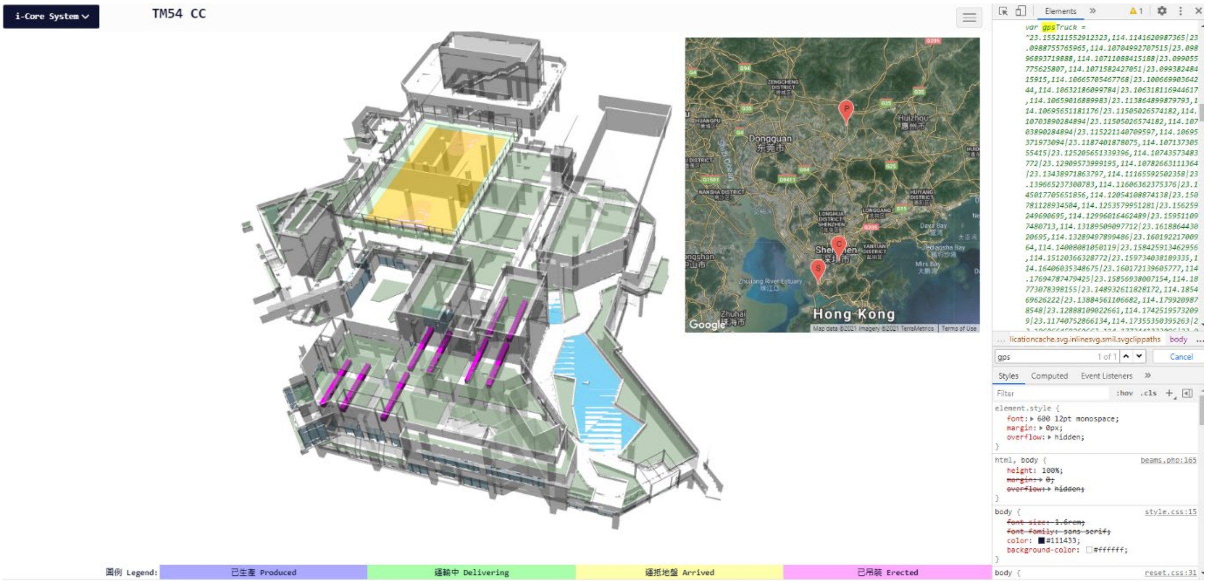
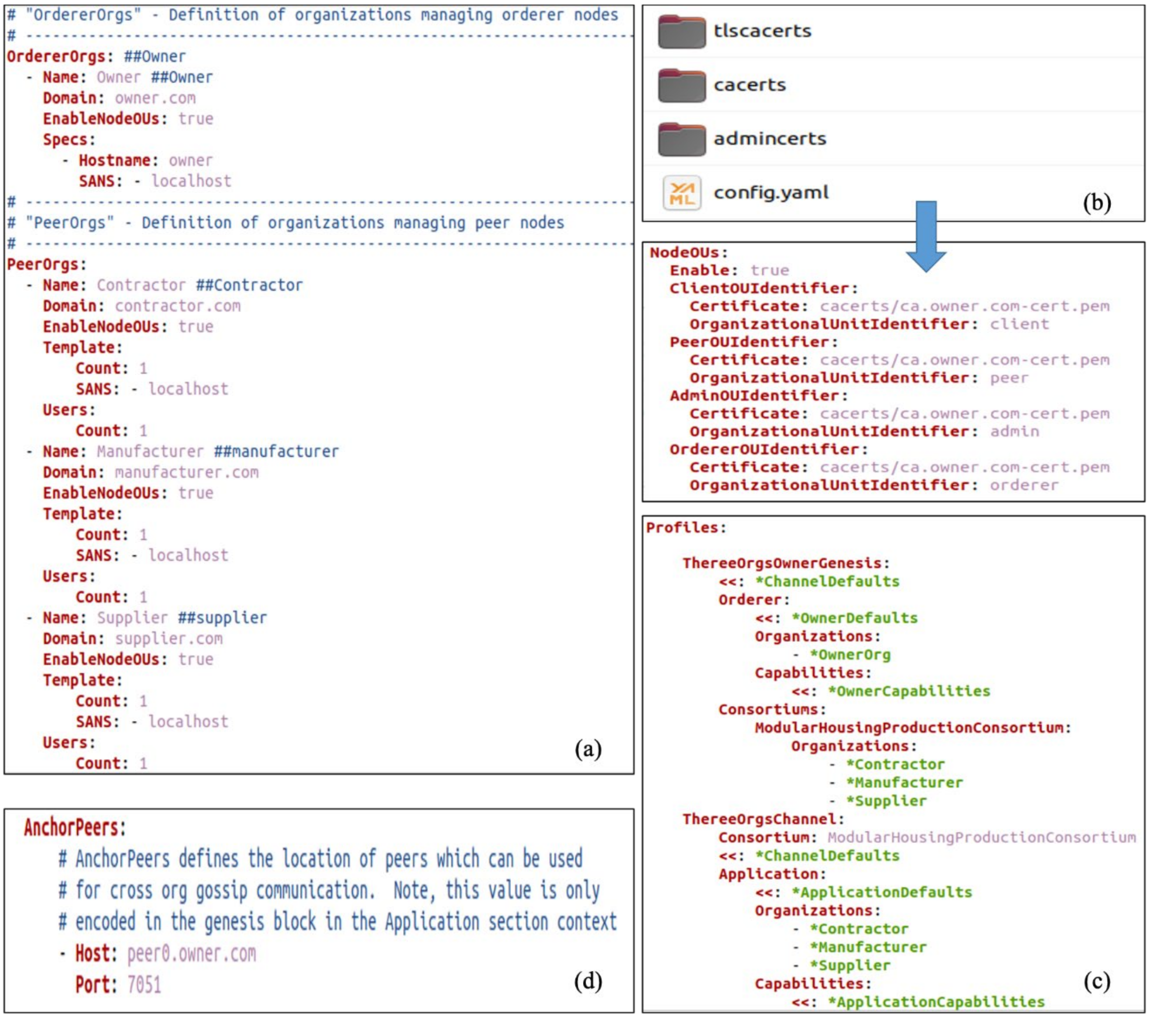


**Figure 5.** Service for off-site logistics and on-site assembly

### *5.1 Implementation*

The proposed SCOs-Bos framework was implemented on *Hyperledger Fabric* (version 2.2), and Javascript was used to write the smart contracts in the chaincode. The development environment was in Ubuntu 18.04, and docker with isolated containers is used to facilitate system prototype development, which uses fewer resources than virtual machines. In the prototype, four stakeholders are involved in the service blockchain: (1) the owner, who serves as the orderer in the ordering service; (2) the contractor; (3) the manufacturer; and (4) the supplier. Figure 6 (a) presents the configuration information for these stakeholders, and cryptogen in *Hyperledger Fabric* is used to facilitate the registration process by issuing the certificates, such as admincert (for each stakeholder's administrator), cacert (for the owner), and tlscacert (for establishing connections), as shown in Figure 6(b).

Each stakeholder in Figure 6(a) has an administrator registered in both the service blockchain and sidechain. The stakeholders can receive certificates and public-private keys from the Fabric CA module of the service blockchain. The administrator can also send requests to the Fabric CA of the sidechain for offering certificates and the publicprivate key to operators in the affiliated organization, which is responsible for registering SCOs in the sidechain. The main blockchain's genesis block is configured, including information of ordering service, consortium, and each stakeholder [see Figure 6(c)]. An anchor peer is devised in each stakeholder for cross-stakeholder communication in the service blockchain and cross-chain interactions between the service blockchain and sidechain [see Figure 6(d)].



```
# "OrdererOrgs" - Definition of organizations managing orderer nodes
# ---------------------------------------------------------------------------
OrdererOrgs: ##Owner
  - Name: Owner ##Owner
    Domain: owner.com
    EnableNodeOUs: true
    Specs:
      - Hostname: owner
        SANS: - localhost
# ---------------------------------------------------------------------------
# "PeerOrgs" - Definition of organizations managing peer nodes
# ---------------------------------------------------------------------------
PeerOrgs:
  - Name: Contractor ##Contractor
    Domain: contractor.com
    EnableNodeOUs: true
    Template:
        Count: 1
        SANS: - localhost
    Users:
        Count: 1
  - Name: Manufacturer ##manufacturer
    Domain: manufacturer.com
    EnableNodeOUs: true
    Template:
        Count: 1
        SANS: - localhost
    Users:
        Count: 1
  - Name: Supplier ##supplier
    Domain: supplier.com
    EnableNodeOUs: true
    Template:
        Count: 1
        SANS: - localhost
    Users:
        Count: 1
```

```
NodeOUs:
  Enable: true
  ClientOUIdentifier:
    Certificate: cacerts/ca.owner.com-cert.pem
    OrganizationalUnitIdentifier: client
  PeerOUIdentifier:
    Certificate: cacerts/ca.owner.com-cert.pem
    OrganizationalUnitIdentifier: peer
  AdminOUIdentifier:
    Certificate: cacerts/ca.owner.com-cert.pem
    OrganizationalUnitIdentifier: admin
  OrdererOUIdentifier:
    Certificate: cacerts/ca.owner.com-cert.pem
    OrganizationalUnitIdentifier: orderer
```

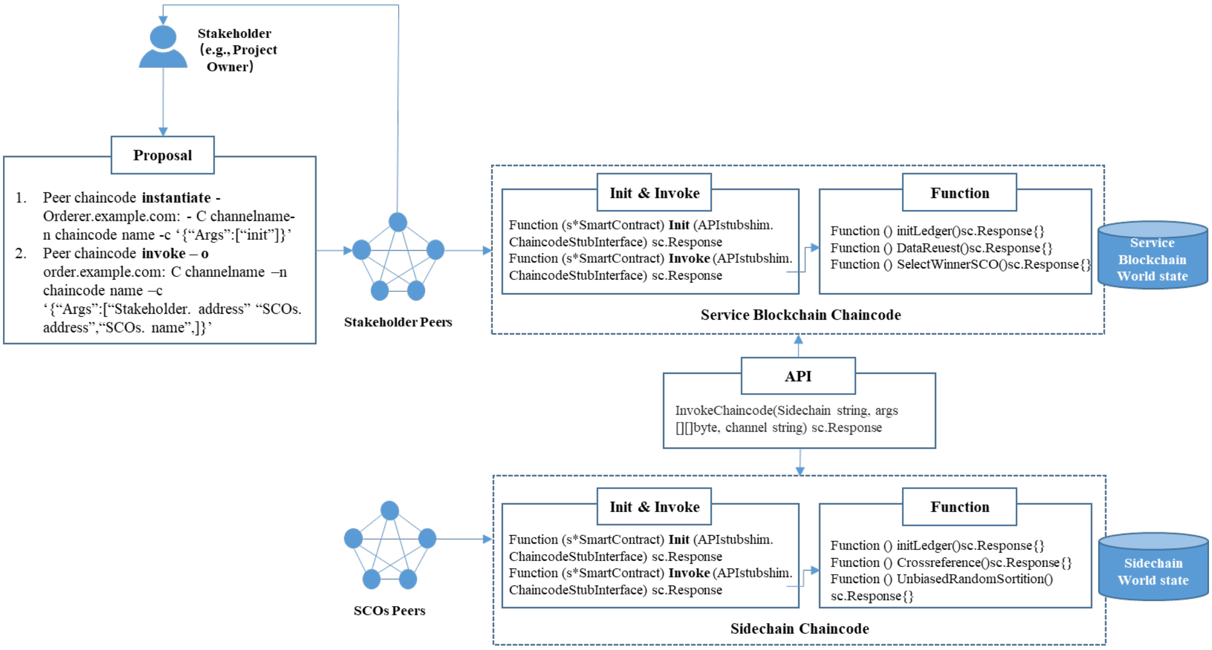
```
Profiles:

    ThereeOrgsOwnerGenesis:
        <<: *ChannelDefaults
        Orderer:
            <<: *OwnerDefaults
            Organizations:
                - *OwnerOrg
            Capabilities:
                <<: *OwnerCapabilities
        Consortiums:
            ModularHousingProductionConsortium:
                Organizations:
                    - *Contractor
                    - *Manufacturer
                    - *Supplier
    ThereeOrgsChannel:
        Consortium: ModularHousingProductionConsortium
        <<: *ChannelDefaults
        Application:
            <<: *ApplicationDefaults
            Organizations:
                - *Contractor
                - *Manufacturer
                - *Supplier
            Capabilities:
                <<: *ApplicationCapabilities
```

```
AnchorPeers:
    # AnchorPeers defines the location of peers which can be used
    # for cross org gossip communication.  Note, this value is only
    # encoded in the genesis block in the Application section context
    - Host: peer0.owner.com
      Port: 7051
```

Figure 6. System configuration for: (a) participant; (b) certificate; (c) genesis block; and (d) anchor peer

The blockchain's execution logic is to invoke smart contracts (chaincode), deployed in both service blockchain and sidechain to enable their cross-interactions and interactions with the shared ledger (world state). The smart contracts SSC, RSC, ASC, and OSC in SCOs-BOs are implemented and provide rich features for testing and debugging before deployment. Figure 7 presents the interaction patterns of chaincodes. SCO information, e.g., address and GPS data, is stored in the ledger. The stakeholder inputs the address and arguments to initialize the transaction and the peers access the ledger via chaincode based on APIs. Two operations are mainly involved in chaincode: the “init” and “invoke” functions, the former when initializing or upgrading chaincode and the latter in response to transaction proposals to query or update the ledger. The “invoke” function comprises functions in four smart contracts: data request, select winner SCO, cross reference, and unbiased random sortition. Cross-chain interactions are implemented with the help of InvokeChaincode() API. The initledger function creates the initial inputs of the ledger. The main functions and interactions of the four smart contracts in this case study are detailed in Figure 7.



```
1. Peer chaincode instantiate - Orderer.example.com: - C channelname-n chaincode name -c '{"Args":["init"]}'
2. Peer chaincode invoke – o order.example.com: C channelname –n chaincode name –c '{"Args":["Stakeholder. address" "SCOs. address","SCOs. name",]}'
```

```
Function (s*SmartContract) Init (APIstubshim. ChaincodeStubInterface) sc.Response
Function (s*SmartContract) Invoke (APIstubshim. ChaincodeStubInterface) sc.Response
```

```
Function () initLedger()sc.Response{}
Function () DataReuest()sc.Response{}
Function () SelectWinnerSCO()sc.Response{}
```

```
InvokeChaincode(Sidechain string, args [][]byte, channel string) sc.Response
```

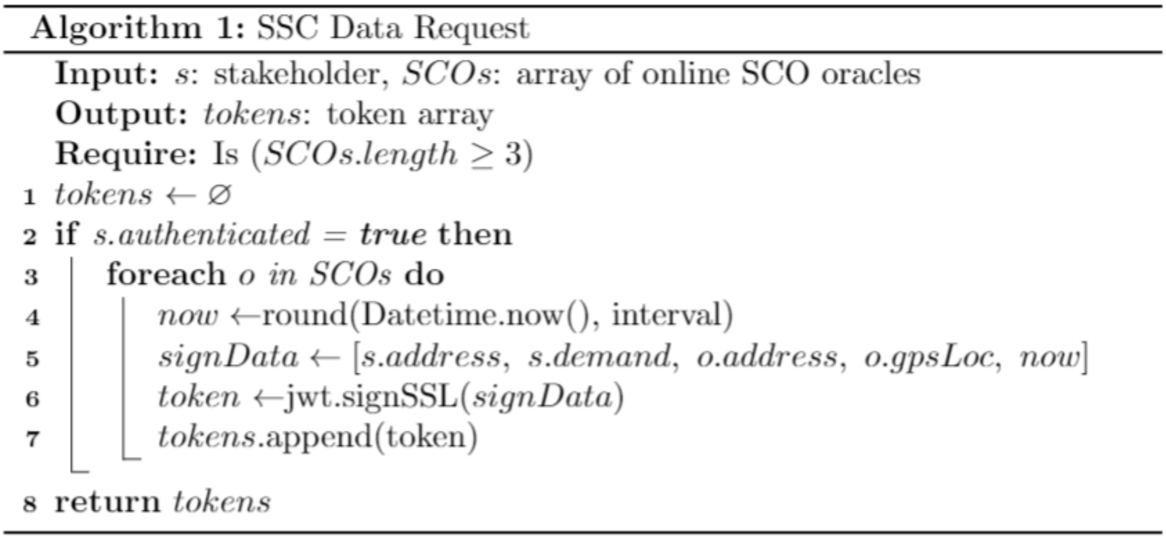
```
Function (s*SmartContract) Init (APIstubshim. ChaincodeStubInterface) sc.Response
Function (s*SmartContract) Invoke (APIstubshim. ChaincodeStubInterface) sc.Response
```

```
Function () initLedger()sc.Response{}
Function () Crossreference()sc.Response{}
Function () UnbiasedRandomSortition() sc.Response{}
```

Figure 7. Interaction patterns for chaincodes of four smart contracts

(1) Service Smart Contract (SSC)

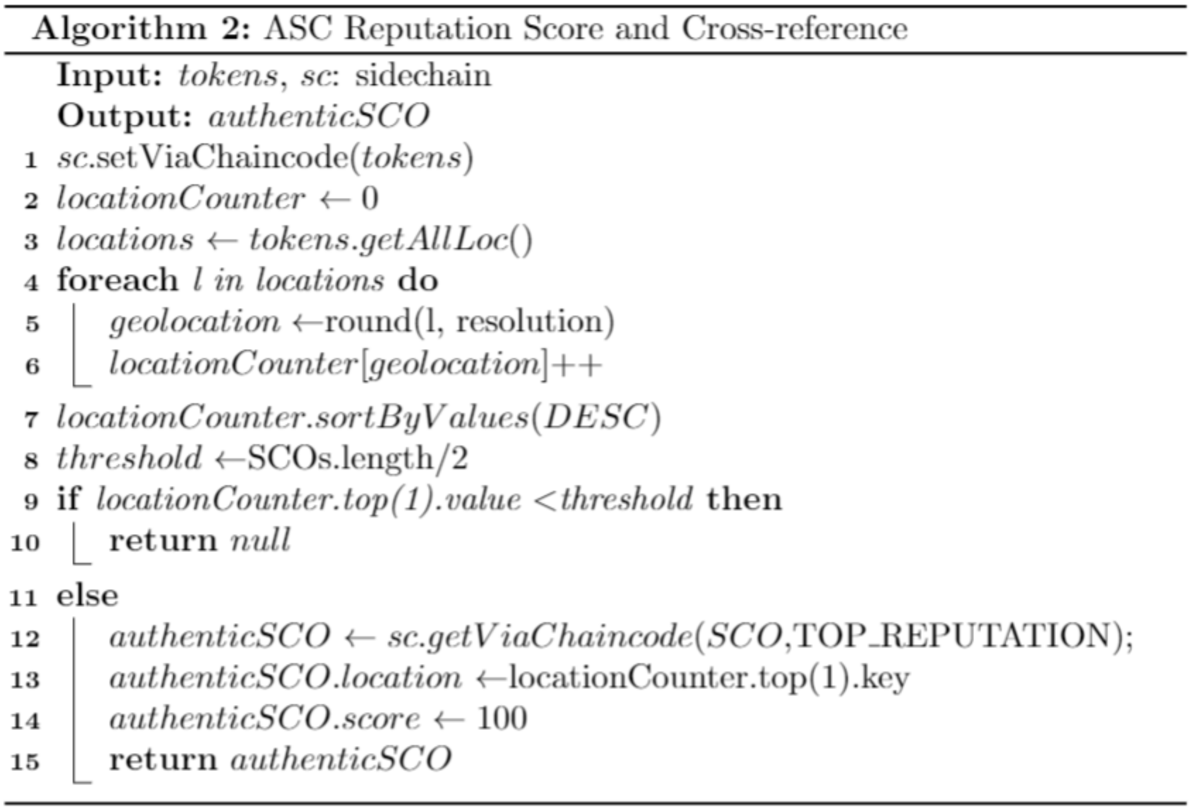
SSC manages the on-chain interactions that can help check if the site manager, project owner, or other stakeholders can access the location data from SCOs and then send data requests. Each service matches one smart contract, and each stakeholder in the service blockchain can request the location data by providing the address of stakeholders and SCOs, and the demanded number of SCOs. To reach a more than 51% consensus on accurate location data, the demanded number of SCOs should be no less than three. Algorithm 1 presents the algorithm of the main function for SSC. Once the stakeholder sends a request, the SSC will check if the stakeholder can access data and then check if the SCOs are online. Once the stakeholder's request is approved, the SSC will generate a token which comprises: (1) request for data, created from the address of stakeholder, address of online SCO oracles, and demand number of SCOs, (2) address of stakeholder, (3) address of SCOs, (4) demand number of SCOs, and (5) GPS location data of SCOs. The tokens are generated and then delivered to the ASC. After implementing the ASC, the final data is received by OSC, RSC, and SSC through cross-chain interaction.



```
Algorithm 1: SSC Data Request
  Input: $s$: stakeholder, $SCOs$: array of online SCO oracles
  Output: $tokens$: token array
  Require: Is ($SCOs.length \geq 3$)
tokens ← ∅
if $s.authenticated = true$ then
    foreach $o$ in $SCOs$ do
        $now$ ←round(Datetime.now(), interval)
        $signData \leftarrow [s.address,\ s.demand,\ o.address,\ o.gpsLoc,\ now]$
        $token$ ←jwt.signSSL($signData$)
        $tokens$.append(token)
return $tokens$
```

(2) Aggregator Smart Contract (ASC)

The ASC coordinates the cross-chain interactions between OSC, SSC, and RSC. SSC forwards the stakeholder's location data requests to the ASC, which invokes the oracles to satisfy them by retrieving and validating SCOs' responses from the OSC. The ASC finally reports the reputation scores of selected SCOs to the RSC. Algorithm 2 presents the algorithm of the main function for the ASC. A minimum of three SCOs should be sent to the ASC to conduct SCO cross-referencing for achieving proof of location, in which at least 51% consensus on the same location data should be reached. Otherwise, the ASC cannot judge and provide the accurate location data sent from the OSC. The ASC cross-references the SCO data based on 51% agreement and reports a binary reputation score to the RSC. If more than 51% of SCOs send back the same result, these SCOs will receive a 100 reputation score, and other SCOs that return different results will get a zero score. If less than 51% of SCOs return the same result, a new aggregation round can be conducted.

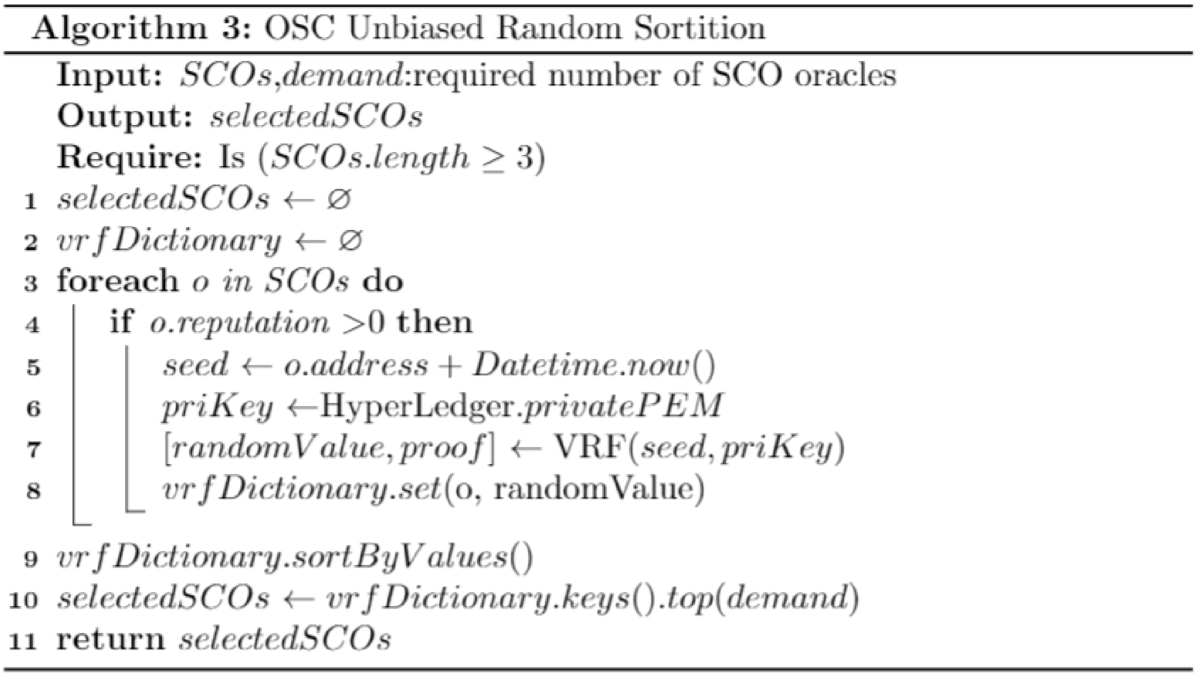


**Algorithm 2:** ASC Reputation Score and Cross-reference

```
Input: $tokens$, $sc$: sidechain
Output: $authenticSCO$
$sc$.setViaChaincode($tokens$)
$locationCounter \leftarrow 0$
$locations \leftarrow tokens.getAllLoc()$
foreach $l$ in $locations$ do
    $geolocation \leftarrow$round(l, resolution)
    $locationCounter[geolocation]$++
$locationCounter.sortByValues(DESC)$
$threshold \leftarrow$SCOs.length/2
if $locationCounter.top(1).value < threshold$ then
    return $null$
else
    $authenticSCO \leftarrow sc.getViaChaincode(SCO,$TOP_REPUTATION);
    $authenticSCO.location \leftarrow$locationCounter.top(1).key
    $authenticSCO.score \leftarrow 100$
    return $authenticSCO$
```

### (3) Oracles Smart Contract (OSC)

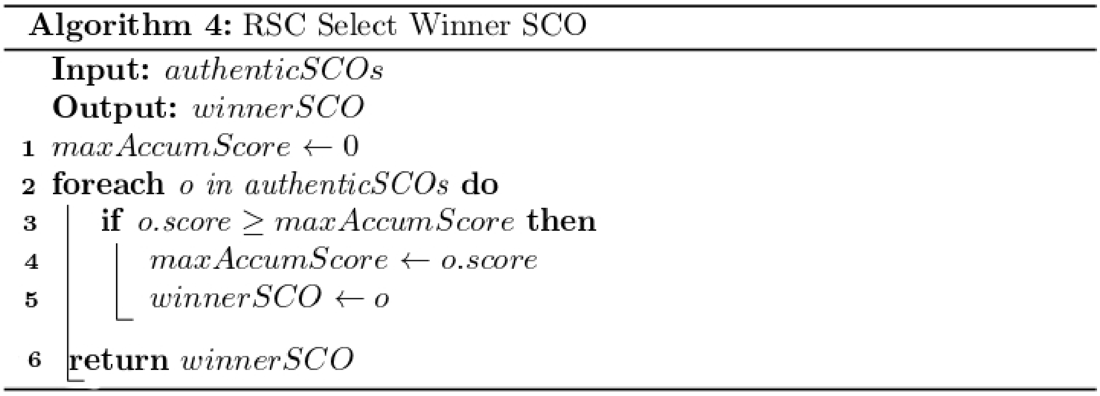
The OSC manages the off-chain interactions to select SCOs from the oracles pool to retrieve the location data. An unbiased random sortition algorithm (RSA) was developed to ensure that the selection processes are independent and random (see Algorithm 3). It is also vital to ensure that stakeholders reach a consensus in the selection processes, which means selected SCOs cannot represent any counterpart's benefits. RSA in OSC uses a verifiable random function (VRF) of Algorand (Gilad et al., 2017) for the oracles pool (A set of SCOs), and they can generate a proof and a random value for each SCO. SCOs aligned with the demanded number are selected and recognized in the sidechain's OSC.



```
Algorithm 3: OSC Unbiased Random Sortition
  Input: SCOs,demand:required number of SCO oracles
  Output: selectedSCOs
  Require: Is ($SCOs.length \geq 3$)
selectedSCOs ← ∅
vrfDictionary ← ∅
foreach o in SCOs do
    if o.reputation >0 then
        seed ← o.address + Datetime.now()
        priKey ←HyperLedger.privatePEM
        [randomValue, proof] ← VRF(seed, priKey)
        vrfDictionary.set(o, randomValue)
vrfDictionary.sortByValues()
selectedSCOs ← vrfDictionary.keys().top(demand)
return selectedSCOs
```

(4) Reputation Smart Contract (RSC)

The RSC aims to compute each SCO's accumulative reputation score and return the winner SCO with the highest accumulative reputation score to the SSC. The RSC receives an input array of authentic SCO addresses, and it sends back an output array address of the winner SCO to the SSC. Algorithm 4 presents the algorithm of the main function for the RSC.

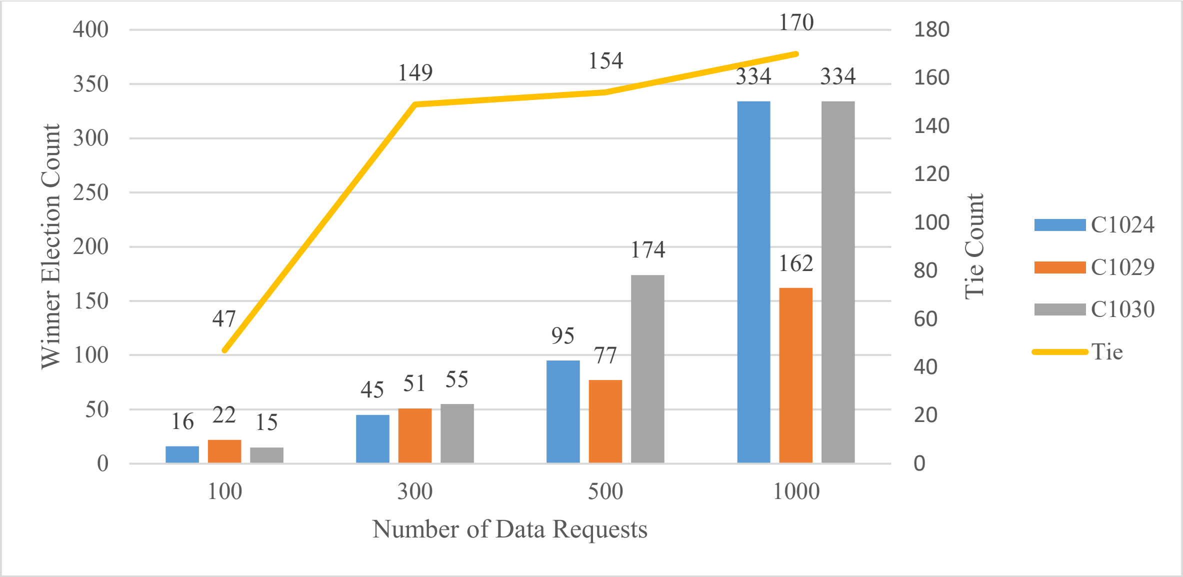


```
Algorithm 4: RSC Select Winner SCO
  Input: authenticSCOs
  Output: winnerSCO
maxAccumScore ← 0
foreach o in authenticSCOs do
    if $o.score \geq maxAccumScore$ then
        maxAccumScore ← o.score
        winnerSCO ← o
return winnerSCO
```

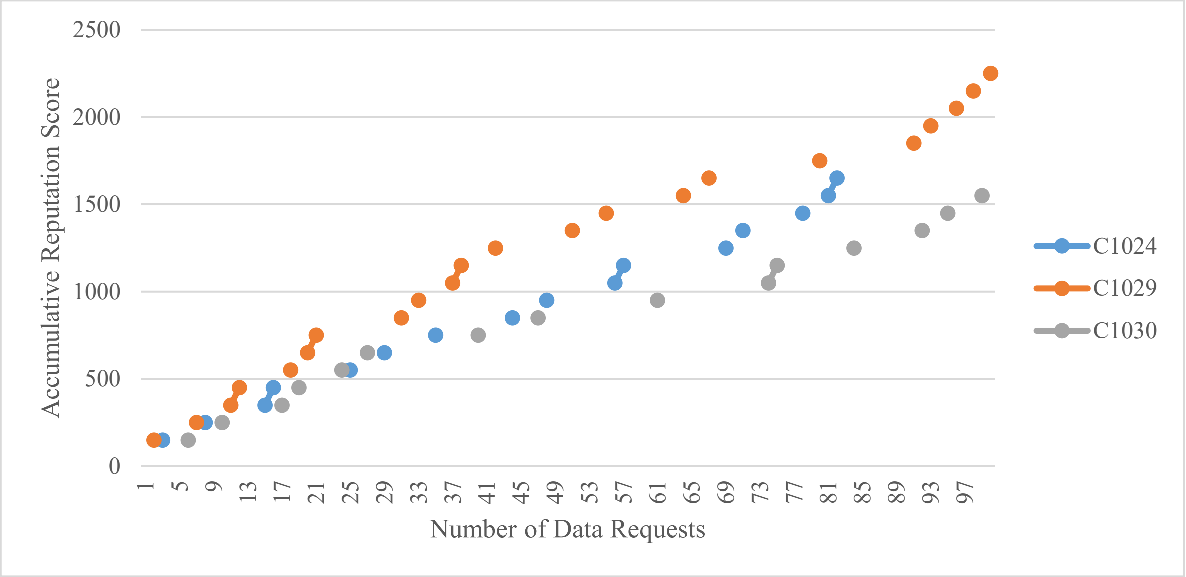
### *5.2 Evaluation*

The most critical design philosophies in SCOs-BOs are to avoid SPoF and offer authentic construction data to the service blockchain. To this end, an evaluation is conducted to prove SCOs-BOs usability in screening malicious data. We assume the total number of SCOs is nine, three of which are authentic (C1024, C1029 & C1030), two malicious (C1025 & C1026), and four offline (C1023, C1027, C1028 &C1031), and these SCOs were transported in the same batch, which means they should be shipped in the same vehicle with identical location data given ignoring GPS accuracy error. We suppose that the total number of data requests is 100, 300, 500, and 1000, the required number of SCOs in each request is 3, and the default reputation score is set to 50 for each SCO. The accumulative reputation scores and winner election count are used as the index to validate SCOs-BOs usability.

As shown in Figure 8 (a), all three authentic SCOs are elected under a different number of data requests, and the tie count is also significant, which indicates that each selection is independent and random. The results prove that SCOs-BOs can avoid SPoF and reject malicious data when authentic data occupy the majority. However, when malicious SCOs hold the majority and form collusion, this would limit the automatic capacity to retrieve authentic data, even though the oracles sidechain can record the data history of each selected SCO. Figure 8 (b) shows the records of all three authentic SCOs' accumulative reputation scores under 100 rounds of SCOs requests. As 47-round requests reach a tie (all selected location data are different), C1024, C1029, C1030 only receive their reputation scores 16, 22, 15 times, respectively. It is also interesting to find that C1029 has the most winner election counts under 100 rounds of SCOs requests but falls to the least under 1000 requests.



(a) Winner election count and tie count



(b) Accumulative reputation score at 100 rounds of requests

**Figure 8**. Evaluation of SCOs-BOs usability

According to Hasan et al. (2018) and Almadhoun et al. (2018), security analysis of the proposed system with smart contracts can be discussed through the aspects of confidentiality, integrity, non-repudiation, authentication and authorization.

- **Confidentiality:** this study enables confidentiality through its cross-chain architecture (e.g., channels in Hyperledger Fabric) and private data. The former supports SCOs in establishing a sidechain, where only those peers (e.g., stakeholders, SCOs) who participate in a sidechain can access the smart contract and data transacted. The latter means that SCO data is stored in the private state database of authorized peers (e.g., project owner) and encrypted with a hash. The proposed system thus preserves both privacy and confidentiality.

- **Integrity and non-repudiation:** For data to have integrity, it cannot be modified during its transmission. All exchanged SCO data in both sidechain and service blockchain are tamper-proof with timestamps. Furthermore, to prevent “man in the middle” attacks and otherwise secure communications, Transport Layer Security (TSL) in Hyperledger Fabric facilitates a data integrity check between a stakeholder and a winner SCO. The unbiased random sortition process will be recorded in tamper-proof logs in the interactions between on-chain and off-chain (e.g., SCOs selection processes).
- **Authentication:** Authentication mechanisms rely on digital signatures requiring each peer to hold two cryptographically corresponding keys: a public key is made widely available and acts as an authentication anchor, and a private key is used to produce digital signatures on data. This study also develops a cross-referencing mechanism to assure the authentic SCO is selected.
- **Authorization:** This study uses membership service provider (MSP) in Hyperledger Fabric to prove authorized peers’ identity. Only authorized peers can trigger the functions of smart contracts. For example, a stakeholder uses its private key to make a consensus (e.g., using a digital signature) on the selected SCO data. The MSP on the ordering service contains this stakeholder’s public key, which is then used to verify that this transaction’s signature is valid.

## 6. Discussion

Three novel aspects of the proposed SCOs-BOs framework are summarized as follows.

- Blockchain oracles are yet to be fully investigated in the construction context because they are reliant on big data from humans, hardware, and software for real-time project management. SCOs offer an innovative alternative able to satisfy the design patterns of oracles and facilitate data exchange between blockchain and real-world CSCM processes.
- Construction data for existing blockchain systems mainly relies on human inputs or a centralized BIM platform. The innovative establishment of a decentralized SCO network as a sidechain has been proven in our case study to avoid the SPoF, and registration of SCOs can make them more accountable when compared with unregistered SCOs. The unbiased random sortation mechanism is also deployed to ensure fairness in selecting and registering SCOs for the sidechain.
- The proposed cross-reference mechanism together with the reputation system, effectively screen out malicious construction data in the evaluation section. This innovation supports the obtaining of trustworthy SCOs and sustains their on-chain reputation.

Despite these innovations, our study has several limitations.

- Firstly, the proposed framework is conceptual and does not claim to be able to solve all blockchain oracle issues. Instead, the framework can work as a guideline and provide insights to help other scholars or practitioners design the four smart contracts, conceive the structure of main (service blockchain) and sidechains, and develop the interaction logics of the four smart contracts. Future studies will provide other detailed solutions under this framework, such as proof-of-inspection and proof-of-assembly by involving blockchain network operations such as ordering service and network configuration.

- Secondly, the only request-response pattern is designed in the smart contracts for SCOs-BOs. As limited by the case study scenarios, the publish-subscribe and immediate-read patterns have not been explored in this study.
- Thirdly, the cross-reference mechanism in this study sets a 51% consensus on the data. However, there is a risk that malicious data could also reach a 51% agreement when the quantity of data or SCOs is small enough. Thus, a more flexible consensus rate range (e.g., from 51% to 67%) in the cross-reference mechanism should be devised and matched according to the required number of SCOs.
- Lastly, we only test the usability of the SCOs-BOs by using the index of average reputation scores and winner election count in a case study. Enabling multiple SCOs reporting the same data streams may increase the cost of the overall system. Thus, other performance metrics, such as the cost of SCOs, throughput, latency, and scalability, will be considered in the future.

## 7. Conclusions

With its characteristics of decentralization, immutability, and consensus, blockchain can improve construction process coordination and collaboration in an isolated deterministic network. Meanwhile, smart construction objects (SCOs) can offer data for blockchain by capturing, processing, verifying, and taking action with the external construction environment in real-time. Harnessing SCOs as blockchain oracles has the potential to enable massive value-added services in construction but also presents Gordian knots in the form of SPoF and malicious data. Blockchain's power may be limited when the offered data heavily rely on a single centralized source or low-quality sources.

This study presents a SCOs-enabled blockchain oracles (SCOs-BOs) framework to offer a decentralized SCO network and related data authenticity mechanism. The SCOs-BOs framework has four parts: stakeholder, service blockchain, sidechain, and oracles pool, which can interact with each other under the request-response pattern in an on-chain, cross-chain, or off-chain manner. Accordingly, a blockchain-enabled construction supply chain management (BCSCM) system is developed to instantiate SCOs-BOs. The services, such as production quality control transparency, logistics and on-site assembly traceability, workers' health and safety privacy, are illustrated. A case study for logistics and on-site assembly traceability service with four main smart contracts is implemented to evaluate its usability. The oracles smart contract (OSC) helps form the decentralized SCO network and select the SCOs from the oracles pool randomly and independently. The aggregator smart contract (ASC) cross-references the data and reports the reputation scores. Then, the reputation smart contract (RSC) manages the authentic SCOs and selects the winner. The service smart contract (SSC) monitors the data requests and responses in an overall process. The evaluation results show that accurate data are retrieved in each request, and the corresponding reputation scores are successfully recorded.

Future research works are recommended to enrich the SCOs-BOs framework. For example, logics in the four smart contracts can be developed and extended for publish–subscribe and immediate–read patterns. A cooperative game theory-based reputation system can be used to improve the performance of rating scores for SCOs. The data semantics enrichment can be enhanced to ensure cross-chain, off-chain, and on-chain communication. More tests are needed for different services, such as production quality control transparency and workers health and safety privacy.

## Acknowledgments

The work presented in this paper was financially supported by the Hong Kong Innovation and Technology Commission (ITC) with the Innovation and Technology Fund (ITF) (No. ITP/029/20LP). This funding source had no role in the design and conduction of this study. The authors wish to express their appreciation to the anonymous reviewers for their constructive comments.

## References

Adler, J., Berryhill, R., Veneris, A., Poulos, Z., Veira, N., & Kastania, A. (2018, July). Astraea: A decentralized blockchain oracle. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (1145-1152). IEEE. DOI: 10.1109/Cybermatics_2018.2018.00207

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. DOI: 10.1109/ACCESS.2020.2992698

Almadhoun, R., Kadadha, M., Alhemeiri, M., Alshehhi, M., & Salah, K. (2018, October). A user authentication scheme of IoT devices using blockchain-enabled fog nodes. In *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)* (pp. 1–8). IEEE. DOI: 10.1109/AICCSA.2018.8612856

Bangaru, S. S., Wang, C., & Aghazadeh, F. (2020). Data quality and reliability assessment of wearable EMG and IMU sensor for construction activity recognition. *Sensors*, *20*(18), 5264. DOI: 10.3390/s20185264

Bankvall, L., Bygballe, L., Dubois, A., & Jahre, M. (2010). Interdependence in supply chains and projects in construction. *Supply Chain Management*, *15*(5), 385–393. DOI: 10.1108/13598541011068314

Beck, R., Stenum Czepluch, J., Lollike, N., & Malone, S. (2016). Blockchain–the gateway to trust-free cryptographic transactions. DOI: aisel.aisnet.org/ecis2016_rp/153

Beniiche, A. (2020). A study of blockchain oracles. *arXiv preprint arXiv:2004.07140*. Retrieved from: https://arxiv.org/abs/2004.07140 on 8th June 2021.

Boeira, F., Asplund, M., & Barcellos, M. (2019). Decentralized proof of location in vehicular Ad Hoc networks. *Computer Communications*, *147*, 98–110. DOI: 10.1016/j.comcom.2019.07.024

Buterin, V. (2014). A next-generation smart contract and decentralized application platform. *white paper*, *3*(37). Retrieved from: https://cryptorating.eu/whitepapers/Ethereum/Ethereum_white_paper.pdf on 8th June 2021.

Cachin, C. (2016, July). Architecture of the hyperledger blockchain fabric. In *Workshop on distributed cryptocurrencies and consensus ledgers* (Vol. 310, No. 4). Retrieved from: https://www.zurich.ibm.com/dccl/papers/cachin_dccl.pdf on 8th June 2021.

Chong, H. Y., & Diamantopoulos, A. (2020). Integrating advanced technologies to uphold security of payment: Data flow diagram. *Automation in Construction*, *114*, 103158. DOI: 10.1016/j.autcon.2020.103158

Dainty, A. R., Millett, S. J., & Briscoe, G. H. (2001). New perspectives on construction supply chain integration. *Supply Chain Management: An International Journal*. DOI: 10.1108/13598540110402700

De Pedro, A. S., Levi, D., & Cuende, L. I. (2017). Witnet: A decentralized oracle network protocol. *arXiv preprint arXiv:1711.09756*. Retrieved from: https://arxiv.org/abs/1711.09756 on 8th June 2021.

Elghaish, F., Abrishami, S., & Hosseini, M. R. (2020). Integrated project delivery with blockchain: An automated financial system. *Automation in Construction*, *114*, 103182. DOI: 10.1016/j.autcon.2020.103182

Ellis, S., Juels, A., & Nazarov, S. (2017). Chainlink: A decentralized oracle network. *Retrieved March*, *11*, 2018. Retrieved from: https://link.smartcontract.com/whitepaper on 8th June 2021.

Gilad, Y., Hemo, R., Micali, S., Vlachos, G., & Zeldovich, N. (2017, October). Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles* (pp. 5168). DOI: 10.1145/3132747.3132757

Gupta, S., Hellings, J., Rahnama, S., & Sadoghi, M. (2020). Building high throughput permissioned blockchain fabrics: Challenges and opportunities. *Proceedings of the VLDB Endowment*, *13*(12), 3441–3444. DOI: 10.14778/3415478.3415565

Hasan, H. R., & Salah, K. (2018a). Proof of delivery of digital assets using blockchain and smart contracts. *IEEE Access*, *6*, 65439–65448. DOI: 10.1109/ACCESS.2018.2876971

Hasan, H. R., & Salah, K. (2018b). Blockchain-based solution for proof of delivery of physical assets. In *International Conference on Blockchain* (pp. 139–152). Springer, Cham. DOI: 10.1007/978-3-319-94478-4_10

Helliar, C. V., Crawford, L., Rocca, L., Teodori, C., & Veneziani, M. (2020). Permissionless and permissioned blockchain diffusion. *International Journal of Information Management*, *54*, 102136. DOI: 10.1016/j.ijinfomgt.2020.102136

Hess, Z., Malahov, Y., & Pettersson, J. (2017). Aeternity blockchain: The trustless, decentralized and purely functional oracle machine. *Accessed: Jan*, *23*, 2018. Retrieved from: https://blockchainlab.com/pdf/%91ternity-blockchain-whitepaper.pdf on 8th June 2021.

Hsu, P. Y., Aurisicchio, M., & Angeloudis, P. (2019). Risk-averse supply chain for modular construction projects. *Automation in Construction*, *106*, 102898. DOI: 10.1016/j.autcon.2019.102898

Kochovski, P., Gec, S., Stankovski, V., Bajec, M., & Drobintsev, P. D. (2019). Trust management in a blockchain-based fog computing platform with trustless smart oracles. *Future Generation Computer Systems*, *101*, 747–759. DOI: 10.1016/j.future.2019.07.030

Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016, May). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *2016 IEEE Symposium on Security and Privacy (SP)* (pp. 839–858). IEEE. DOI: 10.1109/SP.2016.55

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. DOI: 10.1016/j.autcon.2019.02.005

Li, C. Z., Xue, F., Li, X., Hong, J., & Shen, G. Q. (2018a). An Internet of Things-enabled BIM platform for on-site assembly services in prefabricated construction. *Automation in Construction*, *89*, 146–161. DOI: 10.1016/j.autcon.2018.01.001

Li, S., Da Xu, L., & Zhao, S. (2018b). 5G Internet of Things: A survey. *Journal of Industrial Information Integration*, *10*, 1–9. DOI: 10.1016/j.jii.2018.01.005

Li, X., Wu, C., Wu, P., Xiang, L., Shen, G. Q., Vick, S., & Li, C. Z. (2019b). SWP-enabled constraints modeling for on-site assembly process of prefabrication housing production. *Journal of Cleaner Production*, *239*, 117991. DOI: 10.1016/j.jclepro.2019.117991

Li, X., Chi, H. L., Wu, P., & Shen, G. Q. (2020). Smart work packaging-enabled constraint-free path re-planning for tower crane in prefabricated products assembly process. *Advanced Engineering Informatics*, *43*, 101008. DOI: 10.1016/j.aei.2019.101008

Li, X., Chi, H. L., Lu, W., Xue, F., Zeng, J., & Li, C. Z. (2021a). Federated transfer learning enabled smart work packaging for preserving personal image information of construction worker. *Automation in Construction*, *128*, 103738. DOI: 10.1016/j.autcon.2021.103738

Li, X., Wu, L., Zhao, R., Lu, W., & Xue, F. (2021b). Two-layer adaptive blockchain-based supervision model for off-site modular housing production. *Computers in Industry*, *128*, 103437. DOI: 10.1016/j.compind.2021.103437

Luo, L., Jin, X., Shen, G. Q., Wang, Y., Liang, X., Li, X., & Li, C. Z. (2020). Supply chain management for prefabricated building projects in Hong Kong. *Journal of Management in Engineering*, *36*(2), 05020001. DOI: 10.1061/(ASCE)ME.1943-5479.0000739

Mekki, K., Bajic, E., Chaxel, F., & Meyer, F. (2018, March). Overview of cellular LPWAN technologies for IoT deployment: Sigfox, LoRaWAN, and NB-IoT. *In 2018 IEEE International Conference on Pervasive Computing and Communications Workshops* (percom workshops) (pp. 197-202). IEEE. DOI: 10.1109/PERCOMW.2018.8480255

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. (2008). Retrieved from: https://cryptoexpert.net/wp-content/uploads/2018/11/bitcoinwhitepaper.pdf on 8th June 2021.

Nguyen, G. T., & Kim, K. (2018). A survey about consensus algorithms used in blockchain. *Journal of Information Processing Systems*, *14*(1). DOI: 10.3745/JIPS.01.0024

Niu, Y., Lu, W., Chen, K., Huang, G. G., & Anumba, C. (2016). Smart construction objects. *Journal of Computing in Civil Engineering*, *30*(4), 04015070. DOI: 10.1061/(ASCE)CP.1943-5487.0000550

Niu, Y., Lu, W., Liu, D., Chen, K., Anumba, C., & Huang, G. G. (2017). An SCO-enabled logistics and supply chain–management system in construction. *Journal of Construction Engineering and Management*, *143*(3), 04016103. DOI: 10.1061/(ASCE)CO.1943-7862.0001232

Niu, Y., Lu, W., Xue, F., Liu, D., Chen, K., Fang, D., & Anumba, C. (2019). Towards the “third wave”: An SCO-enabled occupational health and safety management system for construction. *Safety Science*, *111*, 213–223. DOI: 10.1016/j.ssci.2018.07.013

Peterson, J., Krug, J., Zoltu, M., Williams, A. K., & Alexander, S. (2015). Augur: A decentralized oracle and prediction market platform. *arXiv preprint arXiv:1501.01042*. DOI: 10.13140/2.1.1431.4563

Qian, X. A., & Papadonikolaki, E. (2020). Shifting trust in construction supply chains through blockchain technology. *Engineering, Construction and Architectural Management*. DOI: 10.1108/ECAM-12-2019-0676

Salah, K., Nizamuddin, N., Jayaraman, R., & Omar, M. (2019). Blockchain-based soybean traceability in agricultural supply chain. *IEEE Access*, *7*, 73295–73305. DOI: 10.1109/ACCESS.2019.2918000

Sheng, D., Ding, L., Zhong, B., Love, P. E., Luo, H., & Chen, J. (2020). Construction quality information management with blockchains. *Automation in Construction*, *120*, 103373. DOI: 10.1016/j.autcon.2020.103373

Shrestha, P., & Behzadan, A. H. (2018). Chaos theory-inspired evolutionary method to refine imperfect sensor data for data-driven construction simulation. *Journal of Construction Engineering and Management*, *144*(3), 04018001. DOI: 10.1061/(ASCE)CO.1943-7862.0001441

Singh, A., Click, K., Parizi, R. M., Zhang, Q., Dehghantanha, A., & Choo, K. K. R. (2020). Sidechain technologies in blockchain networks: An examination and state-of-the-art review. *Journal of Network and Computer Applications*, *149*, 102471. DOI: 10.1016/j.jnca.2019.102471

Tezel, A., Papadonikolaki, E., Yitmen, I., & Hilletofth, P. (2020). Preparing construction supply chains for blockchain technology: An investigation of its potential and future directions. *Frontiers of Engineering Management*, 1–17. DOI: 10.1007/s42524-020-0110-8

Turk, Ž. & Klinc, R. (2017). Potentials of blockchain technology for construction management. *Procedia Engineering* 196 (2017): 638–645. DOI: 10.1016/j.proeng.2017.08.052

Uriarte, R. B., Zhou, H., Kritikos, K., Shi, Z., Zhao, Z., & De Nicola, R. (2020). Distributed service‐level agreement management with smart contracts and blockchain. *Concurrency and Computation: Practice and Experience*, e5800. DOI: 10.1002/cpe.5800

Victor, F., & Zickau, S. (2018, November). Geofences on the blockchain: Enabling decentralized location-based services. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)* (pp. 97–104). IEEE. DOI: 10.1109/ICDMW.2018.00021

Vivekanandan, M., & Sastry, V. N. (2021). BIDAPSCA5G: Blockchain-based Internet of Things (IoT) device-to-device authentication protocol for smart city applications using 5G technology. *Peer-to-Peer Networking and Applications*, *14*(1), 403-419. DOI: 10.1007/s12083-020-00963-w

Wan, P. K., Huang, L., & Holtskog, H. (2020). Blockchain-enabled information sharing within a supply chain: A systematic literature review. *IEEE Access*, *8*, 49645–49656. DOI: 10.1109/ACCESS.2020.2980142

Wang, J., Wu, P., Wang, X., & Shou, W. (2017). The outlook of blockchain technology for construction engineering management. *Frontiers of Engineering Management*, 67–75. DOI: 10.15302/J-FEM-2017006

Wang, Z., Wang, T., Hu, H., Gong, J., Ren, X., & Xiao, Q. (2020). Blockchain-based framework for improving supply chain traceability and information sharing in precast construction. *Automation in Construction*, *111*, 103063. DOI: 10.1016/j.autcon.2019.103063

Xue, F., Chen, K., Lu, W., Niu, Y., & Huang, G. Q. (2018). Linking radio-frequency identification to Building Information Modeling: Status quo, development trajectory and guidelines for practitioners. *Automation in Construction*, *93*, 241–251. DOI: 10.1016/j.autcon.2018.05.023

Xue, F. & Lu, W. (2020). A semantic differential transaction approach to minimizing information redundancy for BIM and blockchain integration. *Automation in Construction*, *118*, 103270. DOI: 10.1016/j.autcon.2020.103270

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *10*(2), 1-19. DOI: 10.1145/3298981

Yang, R., Wakefield, R., Lyu, S., Jayasuriya, S., Han, F., Yi, X., ... & Chen, S. (2020). Public and private blockchain in construction business process and information integration. *Automation in Construction*, *118*, 103276. DOI: 10.1016/j.autcon.2020.103276

Zafar, F., Khan, A., Anjum, A., Maple, C., & Shah, M. A. (2020). Location proof systems for smart internet of things: Requirements, taxonomy, and comparative analysis. *Electronics*, *9*(11), 1776. DOI: 10.3390/electronics9111776

Zhai, Y., Chen, K., Zhou, J. X., Cao, J., Lyu, Z., Jin, X., ... & Huang, G. Q. (2019). An Internet of Things-enabled BIM platform for modular integrated construction: A case study in Hong Kong. *Advanced Engineering Informatics*, *42*, 100997. DOI: 10.1016/j.aei.2019.100997

Zhang, S., Rong, X., Bakhtawar, B., Tariq, S., & Zayed, T. (2021). Assessment of feasibility, challenges, and critical success factors of MiC projects in Hong Kong. *Journal of Architectural Engineering*, *27*(1), 04020047. DOI: 10.1061/(ASCE)AE.1943-5568.0000452

Zheng, R., Jiang, J., Hao, X., Ren, W., Xiong, F., & Ren, Y. (2019). bcBIM: A blockchain-based big data model for BIM modification audit and provenance in mobile cloud. *Mathematical Problems in Engineering*, *2019*. DOI: 10.1155/2019/5349538

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017, June). An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 IEEE International Congress on Big Data (BigData Congress)* (pp. 557–564). IEEE. DOI: 10.1109/BigDataCongress.2017.85

Zhong, R. Y., Peng, Y., Xue, F., Fang, J., Zou, W., Luo, H., ... & Huang, G. Q. (2017). Prefabricated construction enabled by the Internet-of-Things. *Automation in Construction*, *76*, 59–70. DOI: 10.1016/j.autcon.2017.01.006

Zhong, B., Wu, H., Ding, L., Luo, H., Luo, Y., & Pan, X. (2020). Hyperledger fabric-based consortium blockchain for construction quality information management. *Frontiers of Engineering Management*, 1–16. DOI: 10.1007/s42524-020-0128-y

Zhou, H., Ouyang, X., Ren, Z., Su, J., de Laat, C., & Zhao, Z. (2019). A blockchain-based witness model for trustworthy cloud service level agreement enforcement. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications* (1567–1575). IEEE. DOI: 10.1109/INFOCOM.2019.8737580