Designing a Collaborative Construction-Project Platform on Blockchain Technology for Transparancy, Traceablity and Information Symmetry

Alex Norta[1], Christoph Wenna[2], and Chibuzor Udokwu[1]

[1] Blockchain Technology Group, Department of Software Science,
Tallinn University of Technology, Estonia
alex.norta.phd@ieee.org,cjobuzor@gmail.com
[2] Wenna Consult GmbH, Linz, Austria
cw@wenna.com

Abstract. The construction industry is a $6 trillion industry world wide with a prediction to grow towards $10,3 trillion by 2023 and constitutes an essential part of the global economy. Nevertheless, the management of the construction effort is still very manual. The construction process from design, to sourcing of material, contract managements, and so on, is a convoluted and in-transparent process filled with risks the collaborating parties are exposed to. A need exists for management platforms that streamline and automate collaborative construction processes, establish transparency, traceability, and information symmetry between business parties. This whitepaper presents the Construction Project Managament (CoPM) platform that is based on blockchain- and smart-contract technologies for enabling peer-to-peer collaboration between construction parties that enhances the flow of information for reducing cost- and time expenditures while improving the quality of service. The CoPM system is based on diligent up-front requirement studies from which we derive a coherent system architecture and set of cooperation protocols. Thereby, the CoPM system overcomes the currently existing fractured value propositions for construction-management systems. Furthermore, we consider a deduced state-of-the-art technology stack that comprises a significant blockchain- and smart-contract subset, together with an artificial agent layer for a scalable and elastic support of highly effective and efficient cooperation support.

Keywords: Construction, project management, blockchain, smart contract, collaboration, business processes, inter-organisational, data logistics

1 Introduction

There is a strong demand for restructuring construction processes while in reality, the management of construction projects still lacks the required attention with respect to meaningful automation and thus, this industry is one of the least digitised industries[3] in its current existence, i.e., slightly above agriculture and hunting[4]. More specifically, the construction industry suffers from a lack of investment in innovation that results in limited collaboration sophistication with structural fragmentation [23]. The lack of a collaboration-automation context results in hardly any of the available data in the construction industry to be utilised[5], thus, knowledge management is currently dysfunctional. The research- and development investment is currently less than 1% of net sales[6], causing a labour-productivity decline over the last decades [49].

Smart contracts [12] are computer protocols that automate the enforcement of machine-readable contracts to allow for trackable and irreversible transactions that render trusted third parties unnecessary. Blockchain technology [48] achieves these trackable and irreversible transactions with the first usecase being the crypto-currency Bitcoin [37]. Still, smart contracts yield the option to establish agile collaboration configurations between decentralised autonomous organisations (DAO) [39] that comprise loosely coupled process-aware and inter-organisational exchange interfaces that protect internal process privacy. The digital-assets that drive the internal processes of such autonomous organizations are referred to as tokens. Recently, a trend emerges to consider such distributed ledger technology (DLT) also for ensuring in the construction industry [8] that data is secure, decentralised and immutably traceable. Thus, DLT and smart contracts are driving the development of novel business models that rely heavily on financial transactions and the exchange of information. As a key feature of blockchain technology, construction business on a blockchain system [54] allows to collaborate in a trustless context where information exchange flows through project lifecycles securely across distributed systems. A detected key application area of DLT is for so-called building information modelling (BIM) [28] that is a process for generating and managing digital representations, i.e., files, of physical and functional characteristics of places to support decision-making for building, or other built asset.

Digital assets in the construction industry are building objects and models used for representing and communicating the requirements of a building project [36]. In building projects, the tender specification shows detailed requirements of a building project and these are represented as building models digital assets [33]. The standards for specifying the tenders varies across different countries. Some of the common standards include the European standard for building specifications [7], building standards produced by American Society for Testing and

[3] http://www.bimplus.co.uk/people/blockchain-and-construction-how-why-and-when/
[4] https://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/imagining-constructions-digital-future
[5] https://geniebelt.com/blog/go-mobile-go-cloud-but-keep-the-data
[6] https://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/the-construction-productivity-imperative
[7] EU building specification standards https://ec.europa.eu/oib/pdf/mit-standard-building-specs$_en.pdf$

Materials (ASTM) etc. These standards are not machine-executable. The Building Information Model (BIM) provides an optimized code for creating machine readable building requirements. The BIM standard is described by the BIM objects and BIM models used in creating digital building representations [4]. The Original Equipment Manufacturers (OEM) provides specifications for describing building objects (BO) they produce, while designers such as engineers and architects use the building objects as inputs for creating building models (BM).

The state of the art shows a recognised need for construction-collaboration automation to inter-organisationally integrate processes along which the right type of information can flow from the right sources to the right sinks at the right time. However, we find a gap with respect to the available automation systems being not well studied with respect to the complete requirement sets that should be satisfied for construction-collaboration systems. Furthermore, the technology stack such construction-collaboration systems are based on, are outdated and inappropriately assembled, given the actual requirement sets. This whitepaper fills the gap by posing the question of how to design automation systems that address the actual requirements for achieving high quality construction-project collaborations without fractures and frictions causing unnecessary time delays and cost increases? To establish a separation of concerns, we deduce the following sub-questions: What are the sets of requirements that include and assign also the diverse stakeholders who collaborate? What is the architecture topology of a requirements satisfying construction-collaboration system? What are the dynamic exchange protocols between stakeholders and system components into which embedded on-chain transactions assure immutable traceability? Consequently, we introduce the conceptual model of the CoPM system that is a blockchain-technology employing distributed applications (DApp).

The remainder of this whitepaper is structured as follows. Section 2 discusses further preliminaries and also gives a running case that captures the AS-IS state of the construction industry. Section 3 arranges the requirement sets for a construction-collaboration system into a logical space that allows for a targeted association of key stakeholders. Next, Section 4 deduces a distributed architecture from the logical space of system requirements. Section 5 then detects key onchain transactions that are system relevant for potential litigation situations and positions these onchain transaction into dynamic exchange protocols that occur between stakeholders and the architecture of the construction-collaboration system. Section 6 specifies a blockchain-technology stack of pre-existing applications that can be assembled for a rapid deployment of the CoPM construction-collaboration system and also discusses relevant related work. Finally, Section 7 concludes this whitepaper and also discusses limitations, open issues and future work.

2 Preliminaries

Section 2.1 gives a running case that captures the AS-IS state of deficiencies that exist in the construction industry. Next, Section 2.2 gives further presuppositions that expands on the running case.

2.1 Running Case

This article uses as a running case the construction disaster of the new Berlin Brandenburg international airport as described by [55] for which we show in Figure 1 a depiction. Accordingly, the numerous problems that have arisen with the construction of the new Berlin Brandenburg international airport are representative examples within construction projects that can be overcome using the CoPM platform.

Fig. 1: Running case about the Berlin airport construction that reflects the AS-IS situation of the construction industry.

From a technical and management perspective, the problems of the new Berlin Brandenburg international airport are mainly caused by a lack of collaboration-automation, distrust between collaborators and inefficient knowledge management among the project participants. Examples of these problems are the approx. 150,000 construction defects, the exploding costs of the project from roughly 2 thousand million euros to begin with, to over 5 thousand million euros at the last count, the fact that the opening date of the airport was delayed by almost 20 years and ensuing (legal) conflicts among the project participants.

The long planning period and the large number of subcontractors employed results in a high level of complexity. The fact that individual areas of tasks that would better have been carried out by either a single contractor, or several well-coordinated contractors were spread out over several contractors that did neither efficiently coordinate nor efficiently exchange knowledge among each other, has added to the complexity of the project. The complexity further increases by numerous and significant changes in the size of and standards set for the airport while the construction project is already under way. (See also, e.g., [22, 30]). The resulting administrative and co-ordination work is extremely difficult to manage, considering the low degree of digitization of the construction industry that results in chaos at the construction side.

2.2 Related Work

There is a reluctance in the construction industry to adopt BIM and share information for several reasons[8] for fear of legal consequences in case of low performance. Building performance becomes comparable with BIM and assignable to architects, engineers, builders and material supplies. For example, equipped with sensors such as IoT devices, a smart building component can send information about its performance to a distributed database, rather than directly to the manufacturer who is not interested in disclosing actual performance. The study [19] already describes a standard for integrating IoT and BIM objects.

The AS-IS state of the construction industry is plagued by several structural problems [42] that we list below. First, construction projects are complex [3] because of interrelated processes, sub-processes and involved stakeholders such as architects, contractors, subcontractors, customers, suppliers, etc. Second, a lack exists of complete specification for processes and sub-processes and uniformity of materials, work and teams at construction sites that result in uncertainty [21] as each construction project is time-limited and site-based unique. Third, the structure of the construction industry is tightly coupled in individual projects and loosely coupled in permanent networks that foster short-term thinking to the detriment of long-term innovation and learning. Fourth, industry specific is also the high fragmentation [31] in supply chains comprising small and medium size enterprises (SME) with undifferentiated products and services and limited capabilities for investments in new technologies. Finally, the construction industry fosters a rigid culture [3] with a strong resistance to change.

Existing construction collaboration systems are fractured in that when construction-team members collaborate with each other on the same BIM model, many issues arise, due to the usage of different software and tools [1]. Thus, due

[8] https://www.geospatialworld.net/blogs/bim-adoption-around-the-world/

to a lacking industry governance consensus, no inter-organisational coordination occurs for organising the internal data structure, policies, procedures, and control processes for accessing company-internal data sets. With such fragmentation, the development and deployment of integrative and collaborative technologies in the construction industry lags behind [46] that in other sectors. Furthermore, the fractured setup of collaborations in construction industry does not support an efficient implementation of the concept-modular construction involving several parties. Modular construction involves a process of constructing several modular parts of a building off-site using standardized building materials under controlled condition [32]. As a result, smaller organizations cannot effectively take part in construction collaboration due to a lacking proper governance system for such inter-organizational processes. The e-Sourcing Reference Architecture (eSRA) is a conceptual framework that provides technical governance for managing complex inter-organizational collaborations between businesses [41]. Thereby, the current difficulty in executing cross-team collaborations in construction due to different software systems, can easily be managed by eSRA. In addition, eSRA also provides a method for managing processes in inter-organizational collaborations and we refer the reader to [40] for further information.

With blockchain-enabled smart contracts, it is possible to monetize digital and non-digital assets created during a construction project. These types of legitimized assets commonly referred to as tokens, can easily be exchanged between parties involved in a construction project when certain conditions are reached without involving a trusted third party [15]. These conditions are outlined in a smart contract and thereby represent the rules that govern the inter-organizational collaborations between the parties in a construction project. Utility- and security tokens are commonly used in decentralized applications [10] where the former provides the users of decentralized applications the resource to access and consume services that are provided on such platforms. The security tokens represent the stakeholders' investments and ownership in a decentralized platform. In describing the requirements and architecture of the proposed CoPM platform in this whitepaper, only the utility tokens are considered.

The importance of collaboration to ensure project success, requires overcoming the complexities and limitations caused by the fragmentations [43] in the construction industry. Collaboration comprises core entities, i.e., structure, process, agents and artefacts that are conditioned by their respective context. Overcoming fractions requires aligning processes [35] inter-organisationally. A failure persists to adjust inter-organisational processes and working arrangements with different sources of data [7, 45]. Data for collaboration in construction projects are formed [25] across multiple organisations [27] in so-called virtual teams.

3 Stakeholders Associated to CoPM-Requirements

We present the stakeholders and their associated requirements in the decentralized CoPM system. An AOM goal model [47] is used in outlining the functional requirements and the associated non-functional requirements and assigned roles depicted as stick men. Furthermore, the functional requirements are referred to as the goals and depicted as parallelograms, while the non-functional requirements are referred as the quality goals and depicted as clouds.

The stakeholders in the CoPM platform are system participants that perform different functions. Based on expert knowledge, we identify the following classes of users for CoPM. The *designers*, *contractors*, *auditors*, *project owners* (project managers), original equipment manufacturers *(OEM)*, *admin* and *software agents*. The designers are the architects, planners, engineers, and specifiers that perform goals that result in the creation of digital assets on CoPM. The contractors are the engineers responsible for the day-to-day construction activity of a building project. Auditors are certified individuals that perform checks to verify that a construction work meets the required-, or stated standards. The project owners and managers represent the individuals responsible for coordinating the building projects. The OEM produces physical building objects used by contractors. They also produce digital assets such as BIM objects used by designers in creating corresponding building models. The admins are the platform administrators responsible for onboarding new users. The software agents represent the autonomous and semi-autonomous agents. They include *IoT agent*, *KCY agent* and *auditor agent*. The IoT agent records and transmits the status of building projects. The KYC agents perform a verification of identity and other documents submitted by new users. The auditor agent along with human auditor verifies the status of building projects in a semi-autonomous way. The rest of this section is presented as follows. Section 3.1 of this whitepaper presents the main value proposition of the system that is the highest-level root functional goal as well as the first-level goals (FLG) derived from the main value propositions. Section 3.2 describes the user-asset refinements and Section 4 describes the project refinements. Finally, Section 5 describes the payment refinements.

3.1 System Value Proposition

The main value proposition of CoPM in Figure 2 is to provide a decentralized platform for supply chain and project management in the construction sector. Due to the use of blockchain technology, the proposed system inherits the trustworthiness associated with blockchain systems. Furthermore, blockchain ensures that the activities recorded on the system are immutable and traceable. The platform is also designed to be affordable resulting in a low entry barrier for even small construction organizations. The supply-chain management capability of CoPM also encourages modular construction and with smart contracts, this system ensures participants deliver projects that meet the requirements that are immutably stored on a blockchain.

To design a system that satisfies the main value proposition in Figure 2, we derive the following requirements that are referred to as the FLGs. These are as follows, *onboard new user*, *manage digital asset*, *onboard new project*, *manage project*, *manage payment* and *manage wallet*. The *onboard new user* goal enables the execution of know-your-customer (KYC) related goals and assignment of roles to the users. The ***manage digital asset*** goal enables the users to create digital assets such as BIM objects and models and then manage the ownership and usage of such assets. The digital assets are design models used in deriving physical properties of a construction-, or building project. Therefore, properly managing these assets and rewards associated is a crucial property of the proposed platform. The ***onboard new project*** is a goal that enables users to initiate construction projects, manage technical- and business requirements associated to the project, manage the tender process and assign roles to relevant stakeholders. The ***manage project*** goal addresses the possibility to manage an ongoing-, or already completed project. For an ongoing project, the project managers and project owners can update, or modify requirements associated to a project. In the case of an already completed project, the user can manage tasks associated to building repairs and -maintenance. The ***manage payment*** goal provides the possibility of tracking milestones associated to a project and processing payments under clearly stated conditions. The ***manage wallet*** goal caters for allowing all involved stakeholder to manage their different token types and also for transferals between users of the platform.

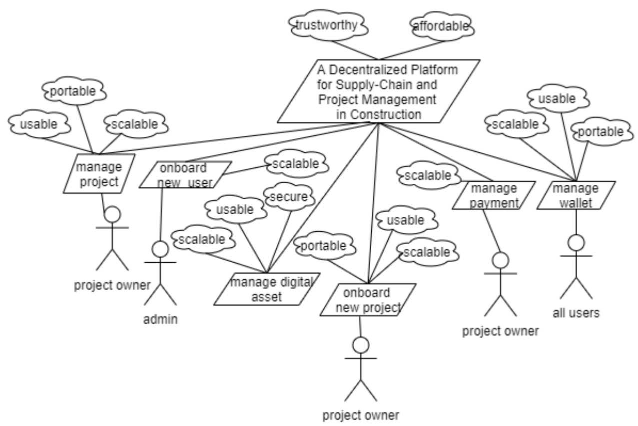

Fig. 2: CoPM-system value proposition and the FLG model.

The quality goals represent the non-functional system properties associated to respective goals of the proposed platform. We now identify and define all the relevant quality goals that are necessary in designing the CoPM platform. The quality goal descriptions provided in this whitepaper are based on standard literature definitions and are further adapted in the CoPM context [41]. They quality goals include; ***scalable***, ***secure***, ***transparent***, ***trustworthy***, ***performant***, ***usable***, ***a ordable***, ***portable*** and ***automated(highly/semi)***. Scalable implies that an associated goal (or asset) can grow rapidly. Security in this case implies unauthorized access, or manipulation to a system goal and associated assets. Transparency refers to the observabilty of a system goal and the ability of the participant to verify that the end-result derived from realising a goal is based on clearly outlined rules of CoPM. Trustworthy is the property of the system, or goal that the executed behavior follows expectations. Performant is the measure of the amount of useful work performed by executing a system goal. Usability is the easiness of the involved stakeholders to understand the system goal in its executable realisation. Affordability implies that executing the goals associated to CoPM are with feasible costs. Portable implies that the associated system goal can be associated in multiple heterogeneous technology stacks. Finally, automated signifies that the associated goal is self-executed by a software-, or hardware agent with embedded software.

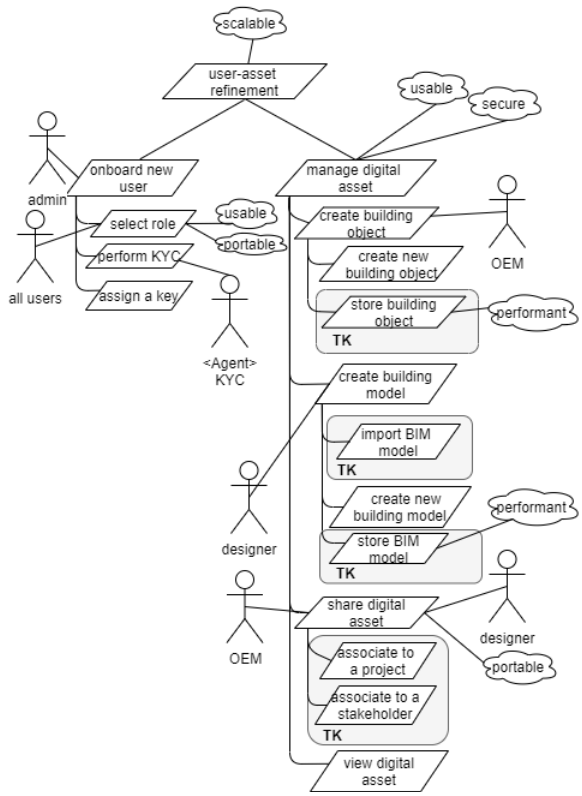

Fig. 3: CoPM-system user-asset refinement goal model.

Based on the above definitions of the quality goals, the non-functional requirements are assigned to the FLGs. The quality goals apply hierarchical inheritance implying that a quality goal associated to a goal is the same to all other sub-goals derived from it. Trustworthy and affordable are associated to the main value proposition of the platform signifying that all goals executed on the platform are trusted and affordable. As shown in Figure 2, all the functions performed in onboarding new user and manage payment are scalable. The manage project goal should be usable, portable and scalable as well. Manage digital asset is usable, secure and scalable. Onbaord new project is portable, usable and scalable. Finally, manage wallet is scalable, usable and portable.

The stakeholders in Figure 2 are assigned to their relevant goals as follows and similarly to quality goals, there exists an assignment inheritance down the hierarchy of the goal branches. Thus, the admin software agent is associated to the goal *onboard new user*. The goals for onboard new project and manage project are associated to the project owner and project manager. The project owner is also associated to manage payment. Finally, manage wallet goal can be performed by all non-software agents on the platform.

3.2 User-Asset Refinements:

The user-asset refinements in Figure 3 contain the goals FLG *onboard new user* and *manage digital asset*. Both goals are scalable implying that an unlimited amount of project stakeholders must be on-boarded on the platform and an inexhaustible amount of digital assets can be created and stored on the platform.

Onboard New User: The FLG *onboard new user* shown at the left of Figure 3 comprises the following goal refinements; *select role*, *perform KYC* and *assign a key*. The select role goal enables users to select the role they wish to perform on CoPM. The roles include; designer, contractor, auditor, project owner/manager, and OEM. A new user provides evidence about the capacity to function in a particular role. Such evidence can include skill certifications, educational qualifications etc. The perform KYC goal enables a KYC agent to verify data submitted by the user. If the KYC is successfully completed, the user is assigned a cryptographic key for the purpose of identification on the platform. The goal select role is expected to be usable by all the participants and portable. As a result, the users can easily perform tasks associated to onboarding a new user on the mobile devices.

Manage Asset: The FLG *manage digital asset* at the right of Figure 3, contains the following sub-goals; *create building object*, *create building model* and *share digital asset*. The digital assets that are managed in the construction collaboration platform are all defined based on the standardization defined in the BIM. The building objects are a clearly defined digital representation of a building object produced by OEM. The objects serve as an input for designing in developing building, or construction models used by engineers in translating to a physical building construction project. Therefore, manage digital asset goal provides the possibility for an OEM and building designers to create and manage these types of assets. BIM objects can be created on external channels and then stored on blockchain for further use. This is enabled by the goal to create new BIM objects and store BIM object. Storing building objects as digital assets is processed using the *utility token*. This ensures that the system is not spammed with unnecessary building objects that provide no real value to the stakeholders such the engineers and designers.

A designer can create a new BIM model by importing an already existing model (on the blockchain) and creating a new model from several BIM objects. These provisions are included with the goals import BIM model and create new BIM model. The *utility token* is used in processing model importation and thereby rewarding appropriately the original model creator. The BIM models created by the designer are stored on the blockchain for further use. This is enabled by the goal *store BIM model* and the goal is processed using the *utility token*.

The goal *share digital asset* provides the possibility for OEMs and designers to provide access to their created digital assets for building contractors and project owners. This goal ensures that the designers and OEM can efficiently and transparently track the use of their intellectual property (IP) and receive appropriate credits and payments for its use. The sub-goals share digital assets shows the created assets can be used by others in the platform by directly associating the asset a project, or a project owner. Finally, both the OEMs and designers can view the building objects and models stored on the CoPM. This is enabled by the goal *view digital asset*.

The goals *store BIM object* and *store BIM model* are expected to be performant as they result in a state change in the blockchain network. The goal *share digital asset* is expected to be portable to ensure that BIM models and objects stored on a blockchain can be easily shared and viewed using mobile devices.

3.3 Project Refinements

The project refinement in Figure 4 contains the FLG *onboard new project* and *manage project*. Both goals are scalable and usable. This implies that both goals allow for a sudden increase in number of projects deployed on the platform and associated stakeholders can easily execute the associated goals correctly.

Onboard New Project: The FLG *onboard new project* shown at the right of Figure 4 provides the project owner the possibility of creating technical and business requirements before a new project is initiated on the platform. In addition, the project owner and project manager can also manage the tender process. The FLG onboard new project contains the sub-goals; set requirements and manage tender. Using the set requirement goal, a project owner can outline technical and business requirements associated to a new project. To set technical requirements, the project owner import BIM models containing building objects as well as their clearly defined properties. The BIM model is then associated to the project and published on the blockchain. A *utility token* is used in importing BIM models and publishing requirements on the blockchain. The business requirements contain a set of milestones that are fulfilled before payment is issued to the contractor. Tokens are set aside, locked in a smart contract and released to the contractor after the condition in the milestone is satisfied. A *utility token* is used in issuing payment to the contractor on reaching a project milestone.

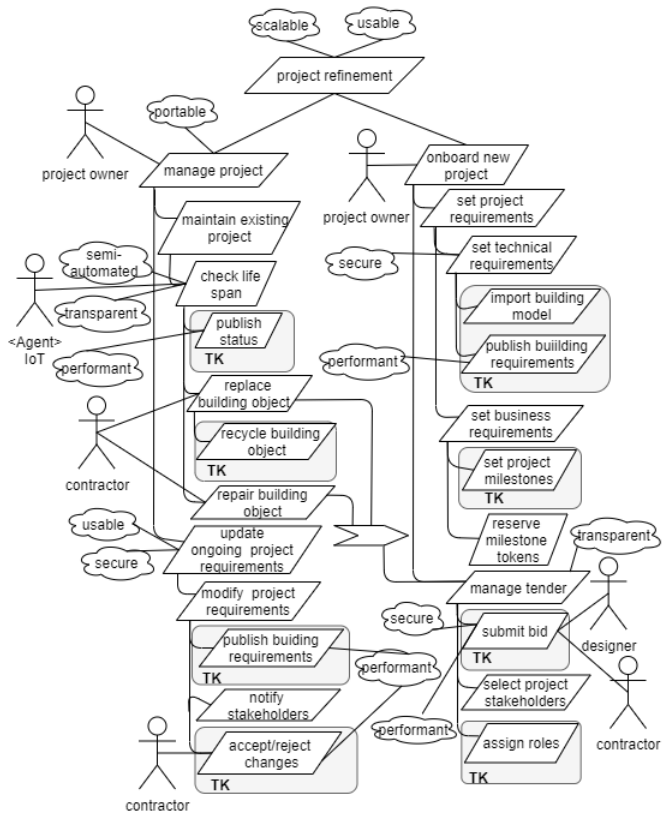

Fig. 4: CoPM-system project refinement goal model.

The ***manage tender goal*** enables the project owner to initiate a bid for a project and manage roles associated to a project. The following sub-goals are contained in manage tender; submit bid, select project stakeholders and assign roles. Project designers and contractors can submit a bid for a new project design (model), or for developing an already designed project. The submit bid goal is processed using a ***utility token*** to prevent users from spamming the platform with invalid bids. The goal ***select stakeholders*** provides the possibility for software bid agents to identity the winning bid. The goal ***assign roles*** allows the project owner to manually assign roles to the stakeholders involved in the project and this is processed with a ***utility token***.

The goal that enables project owners to set requirements for a new project is expected to be secure ensuring the project requirements, cannot be manipulated by any malicious user. Furthermore, the goal that enables bidding by the contractors and designers is expected to be secure as well. Publishing project requirements to the blockchain as well as the bid submission goal are expected to be performant.

The rules that guide the bidding process enabled by manage tender goal is expected to be transparent.

Manage Project: The FLG *manage project* shown to the left of Figure 4 ensures that the project owner and project manager are capable of updating requirements for an ongoing project and also maintain existing projects. Maintaining existing projects using the collaborative construction platform, involves checking the lifespan of a building object to determine if repair, or replacement is needed. The status of a building object that needs repair, or replacement is published using the *utility token*. The goals replace *building object* and *repair building object* to trigger a new bidding process and allow contractors to submit bids for the maintenance work. The goal *recycle building object* is a sub-goal of the replace building object, allowing a project owner to earn a *token* for using an environmentally friendly building maintenance method.

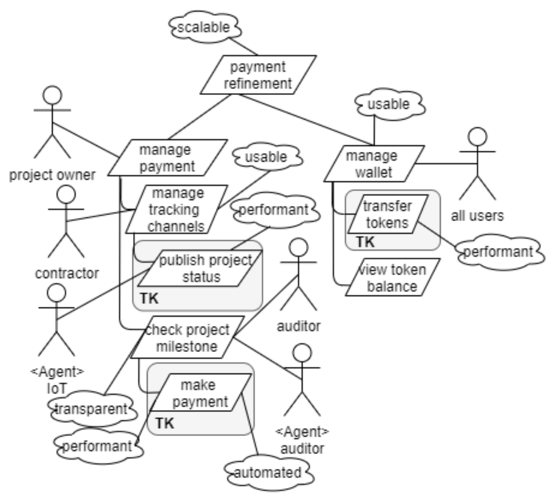

Fig. 5: CoPM-system payment-refinement goal model.

The goal *update ongoing project requirement* allows the project owner to modify the requirements of a project that is already being executed by a contractor. The new requirements are then published to the blockchain using the goal *publish BIM requirements* and this is processed using the *utility token*. The goal *notify stakeholders* messages all parties involved in the project about the new requirements. An offline discussion can take place and the additional payment may be requested by project contractors. Additional payments can be performed by initiating a *utility token* transfer from the wallet of the project owner to the smart-contract enabled escrow account of the project. The escrow project payment is further described in Section 3.4. The project contractors can either accept, or reject the projects. The goal *accept/reject changes* is published on the blockchain and processed using the *utility token*.

The relevant quality goals of the sub-goals of the *manage project* FLG are presented as follows. Checking the lifespan of a building project is semi-automated involving records from IoT-recording devices and manual checks done by project owners. The rule that defines how the checks are conducted is transparent and can be verified by all the stakeholders of CoPM. Publishing building status, BIM requirements as well as responses from contractors regarding changes to the project requirement are expected to be performant. The listed goals result in state changes on the blockchain. The *update ongoing project requirement* is expected to be secure and usable by the project owner.

3.4 Payment Refinements:

The goal *payment re nement* in Figure 5 contains the FLG *manage payment* and *manage wallet* for which all the associated goals are scalable. A large amount of project payments can be handled by the manage payment goal and an unlimited number of token transfers can be executed from the wallet.

Manage Payment: The FLG *manage payment* shown at the left of Figure 5 provides the possibility to track an ongoing project and unlock an escrow type of payment when a specific project milestone has been reached. The FLG *manage payment* has the following sub-goals; *manage tracking channels* and *check project milestone*. The ¡tracking channels¿ are the artificial agents that a contractor uses to record the project status and publish them on a blockchain. These devices may include RFID readers, barcode scanners for scanning bar-codes of physical building objects delivered at a construction site and IoT-video cameras for recording the graphical/pictorial state of a project. The *manage tracking channels* goal enables the construction engineer to select tracking devices and publish the project status on a blockchain. The goal *publish project status* is processed using the *utility token*.

The *check project milestone* goal provides the possibility for auditors on the platform to verify that the project meets the technical- and business requirements described during the project onboarding. The auditing is a two-staged task executed by both a human auditor and software agents - auditor. The auditor agent verifies that the BIM-objects delivered at the construction project meets the requirements of the BIM-model requirements published in the project onboarding. The bar-codes of the physical building objects contain information that outlines the BIM-objects therein. The human auditors perform checks to verify the visible properties of the completed project and confirm that its in line with the original building requirements published by the project owner. A payment is issued to the contractor after checks are successful. The *make payment* goal is a process using the *utility token*.

The *manage project tracking channels* must be usable by the contractors. The rules for checking project milestone must be transparent to all participants in the CoPM. *Publishing of the project status* and *making payments* must be performant.

Manage wallet: The FLG *manage wallet* shown at the right of Figure 5 to provide all the participants the possibility to manage their tokens and transfer tokens between users in the platform. The FLG *manage wallet* has the sub-goals *transfer tokens* and *view tokens*. The goal *transfer tokens* allows all stakeholders to share and exchange tokens among each other. To view the token history and check the details of token exchanges, the goal *view token balance* is used to achieve that. The goal *transfer tokens* is expected to be performant as it results in a state change on the blockchain network.

4 CoPM Architecture

The static components and information-exchange interfaces are derived directly from the goal models above. First, we show in Section 4.1 the methodology that is used in transforming the system requirements represented by the goal models into architectural components of the CoPM-platform. In Section 4.2, we describe the components, actors and information-exchange interfaces that comprise the CoPM-system architecture. Lastly, Section 4.3 maps the static architecture components to eSRA to enable a proper management of inter-organizational collaborations in the construction processes.

4.1 Mapping Goal Models to Component

We show the main components, sub-components, interfaces, and ports of a system and also outline relationships that exist between them [53]. In deriving the component diagram of the CoPM platform from the goal models, the following heuristics are used. The FLGs shown in Figure 2 are used in deriving the main components of the system that contain themselves sub-components equally derived from the refinements of the first-level goals. Due to limited space, only the first- and second goal refinements are used in deriving the components and sub-components of the CoPM-architecture.

Actors who are mapped from the roles associated with the goal-model refinements, are linked to the corresponding components and sub-components. The directional component interfaces are used in representing the data exchange between components. The directional broken-line arrows denote the architecture's dependency represented by the action performed by various actors associated with a component as well as the direction of data flow in the architecture. The thick lines show the direct exchange of data between components without involving any actors.

The quality goals representing the non-functional requirements of the proposed CoPM-platform are not used in the heuristics mapping for deriving the static architecture. Instead, the quality goals are realized in the system-development stage using standard architectural patterns [26, 51] for developing non-functional requirements in computer systems.

4.2 Component Diagram

We show in Figure 6 the component diagram of the CoPM-architecture obtained by applying the heuristic mapping rule described in Section 4.1. The architecture shows six main components and the sub-components of the main components. Each component contains interfaces for reading and writing operations and actor exchanges with components are shown with broken lines. The actors' interactions are also described by the information transfer involved per activity. The inter-components' read/write operations are outlined by the thick lines and grey components show those gaining, or spending tokens depending on the actors' activity.

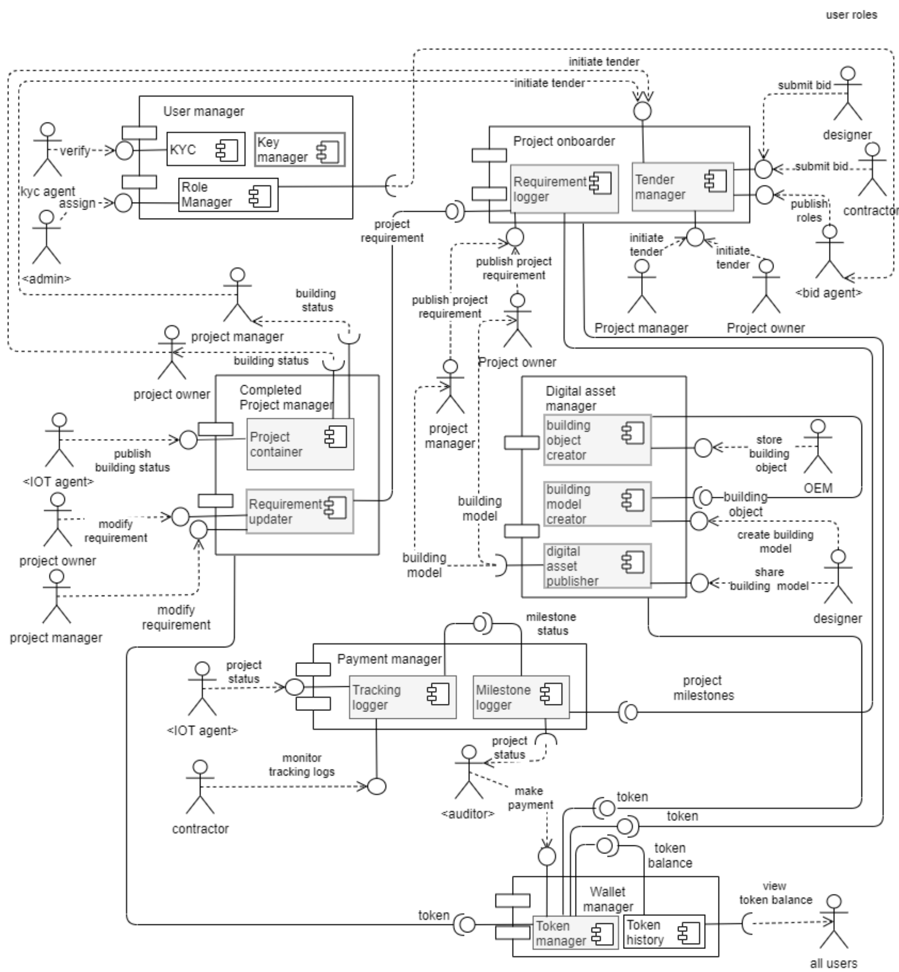

Fig. 6: CoPM-platform architecture topology.

User manager: The user manager component has the following sub-components *KYC, Key manager and Role manager*. The KYC component has an interface for a KYC-agent to verify the identity of the users of CoPM-platform. The role manager has two interfaces, one for assignment of roles by the admin actor and the other interface allows specified users to read the role's data that resides in the component. The following data types are available on the user-manager component, identity data, user roles and user keys.

Digital asset manager: The component enables the creation and exchange of digital assets such as building objects and models on the platform. The component has three sub-components and they include *building object*, *building model creators* and *digital asset publisher*. The building-object components provide the interface for the OEM to create and store BIM-objects on the platform that serve as input for creating building models. The dedicated creator component is accessible to designers such as architects and engineers. The component provides an interface for creating building models using the building objects from the OEM as inputs. The building models created represent the technical requirements of construction-, or building projects. The sub-component digital asset publisher enables the designers to share building models created with other users on the platform. The digital asset publisher also has a read interface that provides the possibility for specified users to access the building models as building requirements. The following data types are available on the digital asset manager component, building-object- and building- model data.

Project onboarder: The component enables the project owner to initiate a new project by outlining the technical requirements and business requirements of the project. Also, the component manages user-biding activities that result from the project tender. The project onboarder has the following sub-components including *requirement logger* and *tender manager*. The requirement logger provides an interface for the project owner to write and store technical- and business requirements of a new project. The technical requirements are accessed from the building models shared from the digital-asset publisher. The business requirements contain milestones and deadlines for completing the outlined technical requirements of the project initiated. The tender-manager sub-component enables the project manager and -owner to initiate bidding for a new project. The tender component provides another interface for designers and contractors to submit bids for a project. There exists also another interface for assigning project roles based on the result of the bidding. The user-roles data used by the biding agent are accessed from the role-manager component. The following data types are available on the project onboarder, project-technical requirement, project-business requirements and user-bid data.

Completed project manager: The component enables the project owner to maintain existing building projects and update the requirements of an ongoing building project when necessary. The project manager has the sub-components *maintain project container* and *requirement updater*. The project container has logs about the status of a completed building project to identify when repairs are necessary. The sub-component provides interfaces for external IOT-agents to monitor the lifespan of building objects in a completed building project and to publish the status of the building objects. The project manager and project owner can as well read the status of building objects to initiate tenders for the repair, or replacement of a building object. The requirement updater provides an interface for the project owner to modify and update the requirements of an ongoing building project. The new requirement is updated on the requirement logger component. The data types available on the project manager are building-status data and updated building-requirement data.

Payment manager: The component enables the platform to automatically monitor ongoing projects and sends payments to the contractors and designers when both technical- and business criteria of the project are satisfied. The payment has two sub-components, *tracking logger* and *milestone logger*. The former provides an interface for IoT-agents to publish the status of a project. The contractor/designer handling the project ensures that the project status is correctly published by IoT-agents. The milestone logger contains information about the milestones that have been achieved during the execution of the project. The component provides an interface for an auditor agent to process payments once specific milestones have been reached. The general information about a milestone data relevant for the project is read from the requirement logger component while the information about completed milestones is read from the tracking logger as milestone status. The payment manager component contains the project status and milestone-status data types.

The setup of the sub-components of the project onboarder and payment manager enable the implementation of a modular construction concept. This is because a single project can be spitted into several parts and bids are submitted separately for each part. The software-agent auditor monitors the status of each of the sub-projects and issues payments accordingly.

Wallet manager: The component provides interfaces for other components to access tokens and exchange tokens with other users. The wallet manager also allows users to access and view their token balances. The wallet manager has the sub-components, *token manager* and *token history component*. In the token manager component, an interface is provided for the auditor agent to access tokens and perform payments to project executors. There occurs an exchange of the token balance between the *token manager* and *completed project manager* sub-components for updating requirements and initiating tenders. Furthermore, the token balance is exchanged with the project-onboarder component for publishing requirements and initiating tenders for new projects. The token-history sub-component has a read interface for all users in the platform to view their token balances. The token-history component accesses the users' token-balance data from the token manager. The token-balance data is the only type available in the wallet manager.

4.3 Construction-Management Business Collaborations on eSRA

The static architecture of the construction management platform in Figure 6 contains inter-organizational collaborations between stakeholders such as project owner, building designer and OEM in creating building requirements. These collaborations are observed in the sub-component of the project onboarder, the *requirement logger* and sub-components of *digital asset manager*, the *building object creator* and *building model creator* components. The eSRA provides a framework for managing these types of collaborations [41] that is achieved by introducing an eSourcing middleware and translational engine to interface between the legacy systems used in executing the collaboration workflows in the construction management. Thereby, an eSRA compliant CoPM-system enables a seamless exchange of information between collaborating parties. Figure 7 shows the highest abstraction level of two orthogonal collaboration directions among the stakeholders in construction management. The first collaboration direction is between the building designer and OEM, the second is between the building designer and project owner. The former enables the exchange of building objects for creating building models and the latter enables the exchange of building models for creating building requirements.

To create building models on the CoPM-platform, the building-designer requests for building objects (AT) digital objects from the OEM through the *construction management broker*. The *construction management exchange* provides an interface for establishing a binding exchange of information (AB - BA) through the construction management broker between the building-model creator component and building-object creator component. On receiving the data, the *eSourcing Setup Support* and the *translator* component transform the data such that it can be readable on the legacy-management components for creating and managing building models. The same applies to the project owner in creating building requirements. The project owner requests for a building model (CX) from the building designer through the construction-management broker. A binding exchange of data is established between the construction-management exchange interfaces (AC - CA). On receiving the BM-data, the process-translational engines transform the data such that it is readable on the legacy management systems for creating and managing building tenders.

The main benefit of the eSRA in application to the CoPM-platform is to ensure that the digital objects such as building objects, building models and building requirements created with legacy managements systems in collaborating organizations are readily available to collaborating partners. In addition, the setup of eSRA-collaboration in construction ensures that information exchanged between parties are machine readable and provided in a transparent manner. The transparency of the construction-management broker in eSRA provides the possibility of managing complex inter-organizational processes in construction collaborations, thereby enabling a full implementation of modular construction concepts involving several parties in a single construction project.

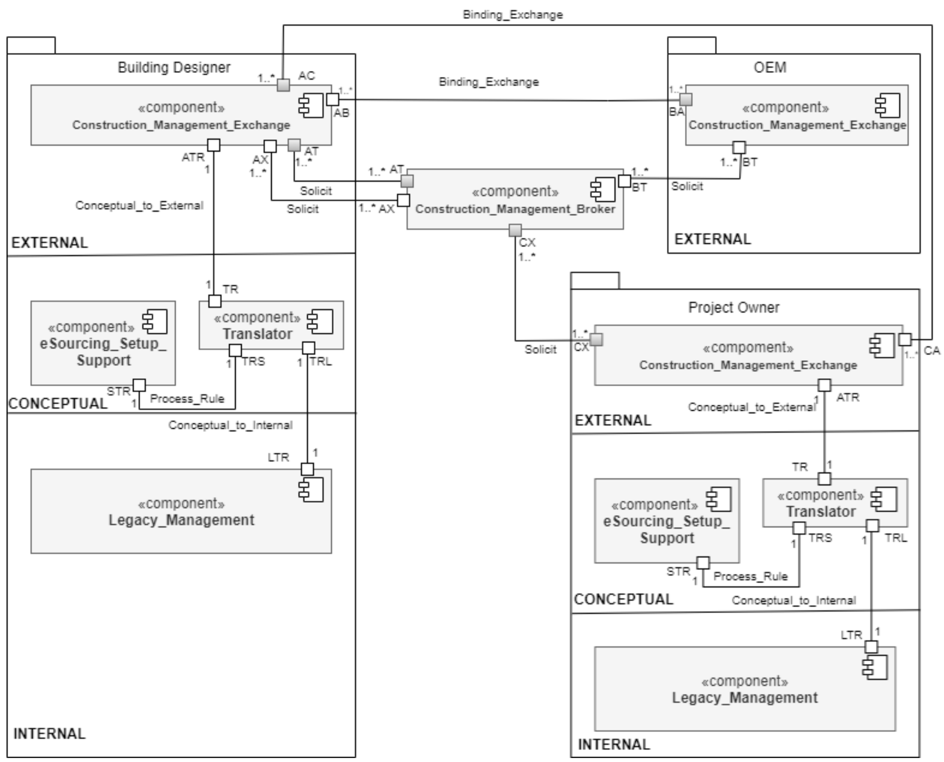

Fig. 7: CoPM-architecture eSRA-components.

4.4 Construction-Management Mapping of eSRA to the Static Architecture

We show in Table 1 the mapping of the components in eSRA of Figure 7 to the static architecture in Figure 6 of the CoPM-platform. The first column shows the eSRA-components being the construction-management exchange, legacy management and construction management broker. The eSRA conceptual components are excluded in the mapping since they do provide any collaboration interface for the involved parties. The main components and sub-components of the static architecture are represented in the row items.

For the legacy management, the components *Requirement logger, Building object creator* and *Building model creator* are mapped to. The requirement logger component enables the project owner to create building technical requirements using existing legacy systems. The requirements are based on the data (building models) received from the building designers. The building-object creator enables

Table 1: CoPM eSRA-Static Architecture Components Mapping.

eSRA Components	Static Architecture Components													
	UM			PO		CPM		DSM			PM		WM	
	KYC	KM	RM	RL	TM	PC	RU	BOC	BMC	DAP	TL	ML	TM	TH
Legacy management				x				x	x					
Construction management exchange					x					x				
Construction management broker												x		

Accronyms
UM: User Manager, PO: Project Manager, CPM: Completed Project Manager
DSM: Digital Asset Manager, PM: Payment Manager, WM: Wallet Manager
KYC: Kyc Component, KM: Key Manager, RM: Role Manager, RL: Requirement Logger
TM: Tender Manager, PC: Project Container, RU: Requirement Updater
BOC: Building Object Creator, BMC: Building Model Creator, TH: Token History
DAP: Digital Asset Publisher, TL: Tracking Logger, ML: Milestone Logger, TM: Token Manager

the OEMs to create digital representations of their building objects using existing legacy systems for computer aided designs (CAD). The building-model creator enables building designers such as architects to create building models using legacy CAD-systems.

For the construction management exchange, the components *tender manager* and *digital asset publisher* are mapped to. The tender manager provides an interface for the project owner to share and exchange building requirements and other project details with other stakeholders. The digital-asset publisher enables both the OEMs and building designers to share and exchange building objects and -models with other stakeholders in the building project.

The eSRA-construction management broker provides a trusted and transparent exchange of information between collaborating parties that is mapped to the *milestone logger*. The latter enables the auditor (software agent) to provide a transparent exchange of information among the collaborating parties regarding the status of an ongoing construction project.

5 Dynamic Blockchain Protocol

This section provides the exchange protocols of dynamic behavior on the CoPM-platform showing the transactions that are executed on blockchain referred to as the on-chain transactions. The use-cases are also presented showing how the stakeholders interact with the platform while executing the on-chain transactions. The on-chain transactions are presented in a table showing the components, transaction event, description and associated stakeholders. The use-cases are presented using UML-sequence diagrams showing the order of activities that occurs as the system stakeholders execute the on-chain transactions. Section 5.1 shows the on-chain transactions and Section 5.2 presents the platform use-cases.

5.1 Construction Platform On-Chain Transactions

Table 2 shows the transactions executed on-chain outlining eight events that are stored on-chain on the specified CoPM-platform. These events involve the creation of digital assets, initiating project tenders, issuing project payments and monitoring, and maintaining completed building projects.

The first two events that are publishing building object and publishing building model, occur at the digital-asset manager component. The building object and building model are the digital assets that are created and shared on the platform and therefore are stored on-chain. The OEM is the stakeholder involved in creating and sharing building objects, while the building designers are responsible for creating and sharing building models.

Table 2: CoPM Onchain Transactions.

Component	Event	Description	Stakeholders
Digital Asset manager	1	Publishing building object	OEM
	2	Publishing building model	Designer
Project onboarder	3	Publishing building requirement	Project owner
	4	Submitting bid	Contractor
	5	Publishing project stakeholders	Bid agent
Payment manager	6	Logging ongoing project status	IoT agent, Contractor
	7	Making project payment	Auditor
Completed project manager	8	Logging completed project status	Project owner, IoT agent

The on-chain events such as publishing building requirement, submitting bid and publishing project stakeholders, occur in the project onboarder. These events are necessary for initiating project tenders and selecting the correct stakeholders for executing the project. The project owner is responsible for publishing building requirements to initiate the project tender. The contracts submit bids to potentially execute the project and a software-agent bid agent selects the right stakeholders for the project.

The on-chain events such as logging ongoing project status and performing project payment, occur in the payment-manager component. IoT-agents as well as contractors provide the status of the project and log the information on the blockchain. An auditor agent verifies the project status and issues payment when requirements/milestone have been reached.

The last on-chain event logging completed project status occurs at the completed project manager and it occurs in the completed project manager. The project status is necessary for determining if a repair-, or a replacement activity is executed on the building objects. The project owner, as well as the IoT-agents are responsible for executing this event.

5.2 Construction Platform Use-Cases

The use-cases for the CoPM-platform show the set of message exchanges that are necessary for executing the on-chain events. The use-cases are presented as UML-sequence diagrams that depict interactions between actors and components of a system in a sequential order [53]. Note that the red-circled numbers in Figures 8-11 correspond to the list of on-chain events in Table 2. The message exchanges between components and agents of the CoPM-platform are classified according to the following operations: digital asset creation operations, tender initiation operations, project payment operations and building maintenance operations.

Digital asset-creation operations: This operation shows the set of message exchanges necessary for creating digital assets such as building objects (BO) and models. First, an OEM-agent imports BO from external sources and then publish the same on the platform. A building designer accesses the BO, using it as input for creating building models (BM). The latter is then published by the designer and becomes accessible to other users for creating technical requirements of building projects. These activities are shown in Figure 8.

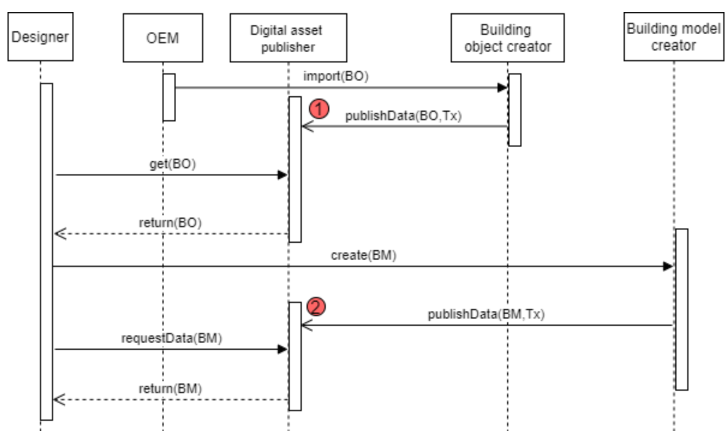

Fig. 8: Digital-assets creation operations sequence diagram.

Tender-initiation operations: To set up a project tender and identify project stakeholders, the project first accesses existing building requirements from the digital asset manager component. The component returns to the user building models containing the necessary technical requirements to start the tender process. The project owner outlines the necessary deadlines for the project milestones and then publishes the building requirements. The contractors and engineers can then submit their bids containing payment details and possible deadlines for executing the project. The bid agent identifies the winning bids and then allocates stakeholders to the project by returning the identification-details public key (PK) and project roles of the selected contractors to the project owner. These activities are shown in Figure 9.

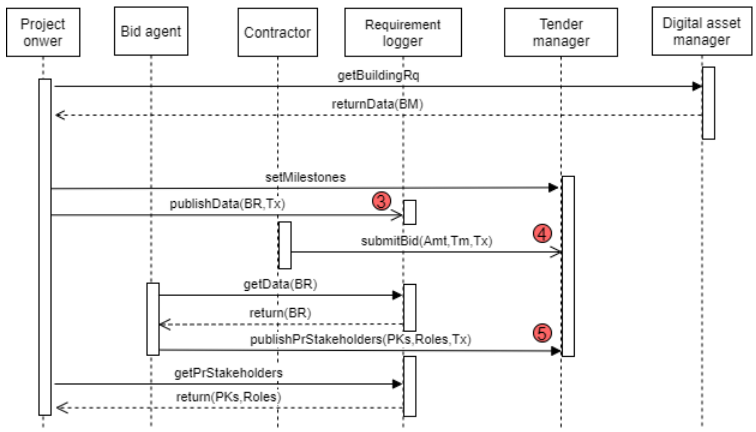

Fig. 9: Tender-initiation operations sequence diagram.

Project-payment operations: On completing projects, or reaching particular project milestones, payments are performed to contractors. The activities necessary for executing payments are outlined as follows. First, the contractor uses an IoT-agent to publish the status of the project. The data associated with project status are logged on the tracking-logger component. Based on the information contained on the project-status data, the project-milestone status is then logged on the milestone-logger component. The auditor agent reads the milestone status and issues the necessary payment to the contractor. The payment data contain the contractor identification (PK), milestone completed (Mlst) and payment amount (amt). These activities are shown in Figure 10.

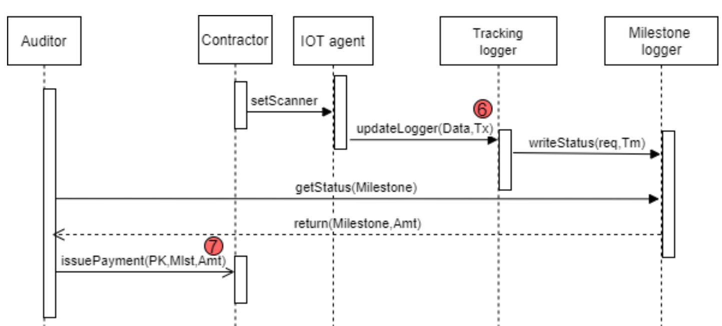

Fig. 10: Project-payment operations sequence diagram.

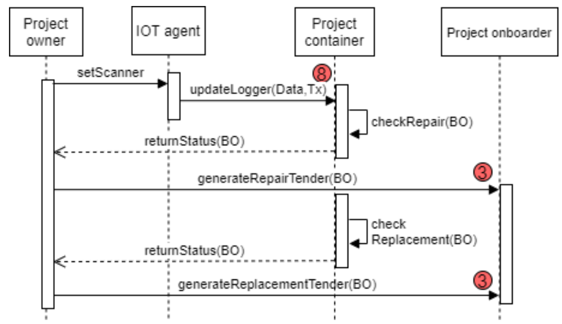

Fig. 11: Building-maintenance operations sequence diagram.

Building-maintenance operation: This operation contains activities necessary for monitoring completed projects and initiating a tender process for building-object repairs, or -replacements. First, the project owner sets an IoT-agent to read a building status to then log the status data to the project-container component. A check is performed to confirm if a repair is needed for the building objects, or if it is necessary to replace the object. In the first case, a repair-tender process is initiated and executed on the project-onboarder component for the damaged building object. In the second case, a replacement-tender process is initiated on the project-onboarder component. These activities are shown in Figure 11.

6 Feasibility Evaluation and Discussion

The TO-BE state of the modernized construction ecosystem we present based on the proposed CoPM-platform where the new running case in Figure 12 shows how the CoPM-platform transforms the low-level digitization in the construction industry into a highly automated collaboration-enabled community. Next, a paper-based evaluation of existing technologies show the feasibility of a rapid development and deployment of the CoPM-architecture described in Section 4. The technologies outlined comprise of blockchain and non-blockchain projects that provide the possibility of deploying in accordance with the specific requirements of the proposed CoPM-platform. The blockchain-related projects examined show smart-contract platforms for realising software agents such as the bid agent and auditor where the former manages the tender processes while the latter manages the payment processes for completed projects. The examined blockchain-technology stack also includes projects that provide decentralized storage of digital assets such as building objects and -models generated on the proposed platform. The non-blockchain technologies describe projects that provide standards for storing digital assets related to building construction. Furthermore, analyzed are available hardware/IoT-scanners that provide the possibility of generating machine-readable data for monitoring the status of ongoing- and completed building projects.

The rest of this section is structured as follows. Section 6.1 shows the digitized construction management of the running case in the resolved TO-BE state. Section 6.2 provides an evaluation of smart-contract platforms and Section 6.3 discusses decentralized storage platforms for blockchain systems. Next, Section 6.4 describes available standards for coding digital assets in the construction industry. Lastly, Section 6.5 shows available radio-frequency identifiers (RFID) and IoT-technologies for implementing scanners for building objects.

6.1 CoPM-Modernized Construction Running Case

In Figure 12, we show the TO-BE state of the construction ecosystem after the deployment of the CoPM-platform. Unlike the current case where the complexity of projects results in project delays, rising cost and defects, the CoPM construction ecosystem enables properly managed collaboration for all the stakeholders in the construction project. First, as shown in Part A of the figure, the platform provides a collaboration system that supports the decoupling of complex project ideas into digitized building models that contain technical requirements of a construction project. This type of cooperation involving stakeholders such as project owners, building designers and OEMs in creating building requirements, eliminates the delays and complexities in communicating project requirement across stakeholders. As a result, the CoPM supports a highly digitized collaboration among stakeholders during the initial onboarding of a building project.

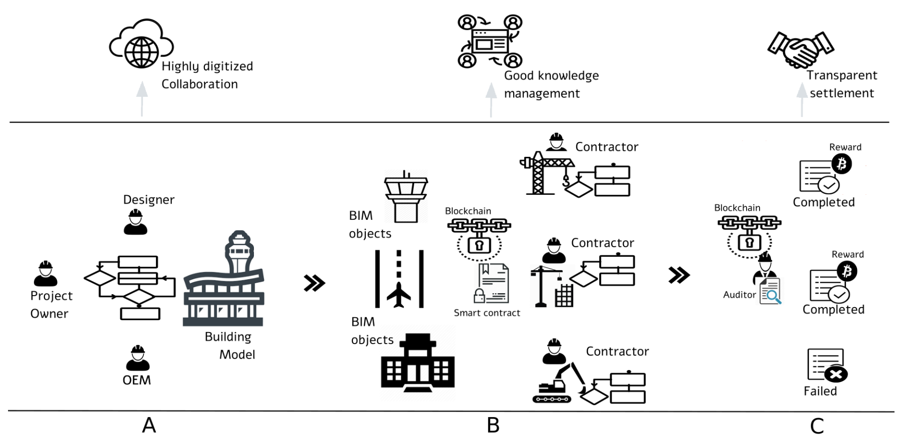

Fig. 12: Proposed TO-BE state CoPM-platform ecosystem.

The running case of the proposed CoPM-platform ecosystem shows in Part B that digitized building models are further decomposed into subsequent building objects containing different parts of a building project. These objects are machine-executable and equipped with smart-contracts where the requirements contained are stored on a blockchain. The CoPM-platform, as depicted in part B of the Figure 12, thereby enables good knowledge management of building requirements in construction. This system provides an environment for impartial and automated bidding of a construction project that is made possible by blockchain-enabled smart-contracts.

The final phase of the CoPM-ecosystem, outlined in Part C of Figure 12, shows a semi-autonomous auditor accessing building requirements from blockchains to examine whether the completed project milestones satisfy the outlined requirements. The auditor uses sensors to obtain the current state of a building project and uses a smart contract to retrieve the requirements of a building project. Payments are only issued to completed projects that meet the building requirements. This thereby results in transparency during project settlements. Unlike the current case where distrust, defects and rising costs are common during the construction of complex projects, the CoPM-platform addresses these issues by providing a system for managing many contractors, digitizing construction processes and rewarding only projects that meet requirements using fully autonomous and semi-autonomous approaches. These effects are enabled by introducing machine-executable building requirements, blockchain-enabled smart-contracts, sensors and semi-automated building auditors into the construction ecosystem.

6.2 Smart-Contract Platforms

The largest blockchain network for smart contracts, Ethereum network [11], is a potential network for deploying smart-contracts for the CoPM-system. The Ethereum network still uses a proof-of-work (PoW) consensus method for the verification of transactions on the network. The PoW-consensus method requires considerable computational power for confirming transactions in the network and it is also a scalability bottleneck. This limits the number of on-chain transactions that can be executed on the CoPM-platform. The proposed Ethereum 2.0 [13] combines PoW with the consensus method termed proof-of-stake (PoS) to address scalability issues and also maintain the security of the network. If Ethereum 2.0 is finally deployed, the network can provide a stable system for executing smart-contracts for the CoPM-platform. Although there exist several blockchain systems that supports smart-contracts [56], we considered only platforms that have shown a high degree of stability and enterprise-wide adoption. Therefore, we identify Qtum [18], Cardano [5] and Hyperledger Fabric [14] as potential blockchains for executing smart contracts on CoPM.

The Qtum network [18] is a highly scalable blockchain network for executing smart-contracts and uses an incentivized PoS consensus method for verifying transactions on the network. To address storage scalability and provide the possibility of executing smart-contracts on mobile devices, the Qtum network combines the Ethereum virtual machine (EVM) with the unspent transaction output protocol (UTXO) in bitcoin and a simple payment verification (SPV) method. This also enables backward compatibility with smart-contracts on the

Ethereum network. As a result, the Qtum network is a good candidate for executing smart contracts for the proposed CoPM-platform.

The Cardano blockchain network is another promising network for executing smart contracts for the CoPM-platform. Just as Qtum, the network uses PoS for verifying transactions, thereby making the network highly scalable when compared with the Ethereum network. The PoS method used in the Cardano network is referred to as the Ouroboros Proof of Stake Algorithm Protocol. The Ouroboros consensus protocol provides an additional layer of security on the network by using secure digital signatures and verifiable random functions to achieve unpredictability during key generation [29].

The Hyperledger Fabric is an example for a permissioned blockchain network for executing smart contracts. The network uses flexible consensus methods that are based on Byzantine fault tolerance (BFT) systems. The BFT consensus family are highly scalable and secure [2]. Still, the main limitation of these consensus types is that the scalability level decreases as the number of users increases. Therefore, there is a limited number of participants that can be involved in executing smart contracts deployed on Hyperledger [6]. To deploy a fully decentralized platform for executing construction projects with an unlimited number of stakeholders such as engineers, contractors, project managers, designers etc, Hyperledger is not a suitable choice.

6.3 Decentralized Storage

The proposed CoPM will generate a considerable amount of data objects such as digital assets that contain building requirements and data from scanners that contain building status. Due to corresponding blockchain limitations [52], decentralized storage provides extended addressable repositories for blockchain systems [17] and therefore, can be used in preserving digital assets in the proposed platform. The Interplanetary File System (IPFS) and BigchainDB are common decentralized storages used in blockchain applications [24], [16] and [50].

The IPFS is a peer-to-peer (p2p) system that provides the possibility for different computing devices to connect to the same file system for enabling access to large data sets with high efficiency over a non-trusting network [9]. As a result, the IPFS-system is transparent and resilient as well because there is no single point of failure. The system uses hyperlinks to address data sets and provide a structure for building a file-versioning system on the blockchain.

BigchainDB provides a scalable and distributed database for storing large data sets generated from blockchain applications [34]. The main strength of BigchainDB is that it also extends the storage- and file-management capabilities of smart-contract applications. The BigchainDB can be easily integrated into popular smart-contract platforms such as Ethereum, Qtum, etc. In addition, the BigchainDB is compatible with file versioning systems such as IPFS.

6.4 Digital Assets Standard

A universally acceptable and machine-readable standard is necessary for specifying digital assets created on the proposed decentralized CoPM-platform. The BIM-model provides an optimized system for designing building models ensuring the integration of software systems in building construction [38]. BIM is useful in construction tendering by providing a common environment for defining building data, specify technical requirements of a building project as BIM-objects and models, specify business requirements of a project as BIM-metadata. These are useful for storing digital assets created in CoPM. Furthermore, by outlining building requirements as BIM-objects and -models, it is possible to identify residual risks associated with physical building objects. This is applicable to the CoPM-platform ensuring that completed projects are properly monitored for possible defects and the building status scanned by IoT-agents are stored using the BIM-standard.

In the traditional construction-tendering applications such as procore-submittals[9], hundreds and thousands of PDF files are submitted by project stakeholders. The data in files are manually sorted by the project manager to identify necessary project stakeholders. However, with the BIM-standard, the building specifications are received as machine-readable codes that can result in the execution of smart-contracts in the decentralized CoPM-platform. Although the traditional construction-tendering applications are not suitable for gathering building requirements for the proposed CoPM-platform, form builders that are based on the BIM-standard can be implemented on the user interfaces to capture machine-readable data executable by smart contracts.

6.5 Technologies for Building Scanners and -Sensors

The architecture of the proposed CoPM-platform shows significant involvement of IoT-devices in the broadcasting status of building objects when executing construction project and when monitoring already completed project. The project IOTA [44] provides a decentralized platform and real-time economy for processing data from IoT-devices and performing payments for services rendered. The project IOTA has the potential for providing a trusted environment for IoT agents involved in the CoPM project. This implies that the building-status data from RFID sensors with IoT-capabilities can be transparently transmitted to the data-logging component of the proposed CoPM-platform. Still, the IOTA-project is not compatible with other major blockchain platforms for executing smart-contracts and cannot be effectively integrated into the proposed CoPM-platform.

The new IBM microscopic chip[10] is a grain-sized computer with an independent processor, memory, storage and communication module that provides the possibility for transparent data transmission without the involvement of third-parties. For instance, the chip can be embedded into a physical building

[9] Procure Submittals https://www.procore.com/

[10] IBM Chip Computer https://cointelegraph.com/news/ibm-reveals-blockchain-computer-smaller-than-grain-of-salt-to-track-objects-devices

object and the status of the object is transmitted in real-time to the receiving components of the proposed CoPM-platform. The chip is considered an anti-fraud device to prevent a possible malicious behaviour of building contractors within the ecosystem of the proposed construction platform.

The integration of autonomous IoT-agents on the CoPM-platform and the interaction between these agents and human users outlines the social-technical aspects of the platform. Therefore, a proper digital identity-management system is necessary to enable secure exchange of information produced by the stakeholders and agents in the system. The decentralized digital identity governance described in the machine-to-everything (M2X) economy [20], provides a formalized technique for identifying all the entities on the CoPM-platform. The M2X identity management describes a secure protocol for validating and authenticating entities in decentralized network.

7 Conclusion

This whitepaper presents the CoPM-platform, a decentralized platform for managing and executing construction projects. It describes a trusted and affordable system for outlining technical- and business requirements of construction projects and payments are performed to the building contractors when specific requirements have been reached in the specified project milestone. The platform gives building-project owners and -managers the possibility of deploying a modular construction project and with smart contracts, ensures that the projects are executed according to the specified requirements.

The main value proposition of the platform is to provide a decentralized system for supply-chain and project management in the construction industry. The main value proposition is further refined into the functional requirements of the system. The first-level functional requirement shows that the platform should be able to onboard new users, onboard new projects, manage digital assets, manage projects, manage payments and manage wallets. The stakeholders that participate on the platform include human users and software agents. The human users are project managers, project owners, OEM, designers, engineers, contractors and human auditors. The software agents include the admin, bid agent, and auditor. The auditing role represents a combination of human- and software agents. The project owners and -managers are responsible for managing and onboarding new projects. The OEM and designers are responsible for creating digital assets such as building objects and models on the platform. The contractors and engineers are responsible for executing projects according to the outlined requirements. The admin agent is responsible for onboarding users and assigning user roles. The bid agent manages the tender function during onboarding of a new project. Finally, the auditor performs payments when all the building conditions and -requirements are satisfied.

The static architecture of the proposed CoPM-platform is provided to further describe the system and a UML-component diagram, is used in showing the system architecture embedded in the eSRA architecture that shows the management of inter-organizational collaborations in building construction. The main components of the static architecture are derived from the heuristics mapping of the first level requirement of the platform and they include an user manager, project onboarder, completed building manager, digital asset manager, payment manager and wallet manager. Each component contains sub-components that further shows the hierarchically nested structure of the architecture. The stakeholders are also associated with components showing the actions performed as well as the data objects exchanged between components. The KYC- and admin agents are associated with the user-manager component and the data object in this component is the user data. The project manager, project owner, contractor, designer and bid agents are associated with the project onboarder component. The data objects that are stored in project onboarder are the project's technical requirements, business requirements and the user-bid data. The OEM and designers are associated with the component digital asset manager and the data object stored are building objects and building models. The IoT-agent, project owner and -manager are the stakeholders associated with the component completed project manager and the data object stored is building status data and updated building requirement data. The IoT-agent, auditor and contractor are the stakeholders associated with the component payment manager. The data objects stored are project status and milestone status. Lastly, all users associated with the manage wallet component and the data object are stored in the users' token balance. The eSRA architecture depicts the collaborations that exist between the project owner, building designers and OEMs in creating and transmitting building requirements across the stakeholders in a construction project. Furthermore, eSRA outlines a framework for an efficient management of legacy systems involved in creating and managing building requirements.

We use UML-sequence diagrams to describe the dynamic behaviour of the CoPM-platform. The UML-diagrams show the use-cases that outline the necessary activities that are required in executing on-chain events on the platform. Four use-cases are described and they include digital-asset creation operations, tender-initiation operations, project-payment operations and building-maintenance operations. The first use-case shows the activities that are required for creating digital assets on the platform and contains two on-chain events that are publishing building object and publishing building model. The second use-case shows activities that are necessary for setting up a project tender and identifying project stakeholders. The use-case description has three on-chain events that are publishing building requirement, submitting a bid, and publishing project stakeholders. The third use-case describes a set of activities necessary for executing project payments containing two on-chain events that are logging the ongoing project status and performing project payment. The fourth use-case shows the activities necessary for monitoring completed projects and initiating a tender process for repairs, or replacements of building objects. The use-case has only one on-chain event that is logging completed project status.

Finally, a paper-based evaluation is carried out to identify existing blockchain- and non-blockchain technologies as candidates for a rapid development and deployment of the CoPM platform. The Ethereum, Qtum and Cardano networks are identified as the potential blockchain networks for executing smart contracts from the CoPM-platform. The IPFS and BigchainDB are identified as the extended data repository for storing digital assets created on the platform. The BIM-standard is identified as the ontology for representing digital assets such as building models and objects on the platform. The new IBM grain-sized chip computers in combination with RFIDs are identified as potential IoT-scanners for -monitoring and transmitting in realtime the status of a building object. As future work, we aim for a prototype implementation of the CoPM-platform.

References

1. E. Alreshidi, M. Mourshed, and Y. Rezgui. Requirements for cloud-based bim governance solutions to facilitate team collaboration in construction projects. *Requirements engineering*, 23(1):1–31, 2018.
2. E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, page 30. ACM, 2018.
3. Y. Arayici, P. Coates, et al. *A system engineering perspective to knowledge transfer: a case study approach of BIM adoption*. InTech, 2012.
4. Y. Arayici, P. Coates, L. Koskela, M. Kagioglou, C. Usher, and K. O'Reilly. Bim adoption and implementation for architectural practices. *Structural survey*, 29(1):7–25, 2011.
5. C. Badertscher, P. Gaži, A. Kiayias, A. Russell, and V. Zikas. Ouroboros genesis: Composable proof-of-stake blockchains with dynamic availability. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 913–930, 2018.
6. A. Baliga. Understanding blockchain consensus models. In *Persistent*. 2017.
7. M. Bassanino, T. Fernando, and K. Wu. Can virtual workspaces enhance team communication and collaboration in design review meetings? *Architectural Engineering and Design Management*, 10(3-4):200–217, 2014.
8. I. Belle. The architecture, engineering and construction industry and blockchain technology. *Digital Culture*, pages 279–284, 2017.
9. J. Benet. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*, 2014.
10. M. Benoliel. Understanding the difference between coins, utility tokens and tokenized securities. *URL https://medium. com/startup-grind/a6522655fb91*, 2017.
11. V. Buterin et al. Ethereum white paper. *GitHub repository*, pages 22–23, 2013.
12. V. Buterin et al. A next-generation smart contract and decentralized application platform,(2014). *URl: https://github. com/ethereum/wiki/wiki/White-Paper (visited on 10/09/2016)*, 2014.
13. V. E. Buterin. 2.0 spec–casper and sharding, 2018. *Available [online].[Accessed: 30-10-2018]*.
14. C. Cachin et al. Architecture of the hyperledger blockchain fabric. In *Workshop on distributed cryptocurrencies and consensus ledgers*, volume 310, page 4, 2016.
15. Y. Chen. Blockchain tokens and the potential democratization of entrepreneurship and innovation. *Business Horizons*, 61(4):567–575, 2018.

16. J. L. B. Cisneros, F. M. Aarestrup, and O. Lund. Public health surveillance using decentralized technologies. *Blockchain in Healthcare Today*, 2018.
17. K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, et al. On scaling decentralized blockchains. In *International Conference on Financial Cryptography and Data Security*, pages 106–125. Springer, 2016.
18. P. Dai, N. Mahi, J. Earls, and A. Norta. Smart-contract value-transfer protocols on a distributed mobile application platform. *URL: https://qtum. org/uploads/-les/cf6d69348ca50dd985b60425ccf282f3. pdf*, page 10, 2017.
19. B. Dave, A. Buda, A. Nurminen, and K. Främling. A framework for integrating bim and iot through open standards. *Automation in Construction*, 95:35–45, 2018.
20. A. S. Devices. The m2x economy.
21. A. Dubois and L. Gadde. The construction industry as a loosely coupled system: implications for productivity and innovation. *Construction Management & Economics*, 20(7):621–631, 2002.
22. R. During. *Lachnummer BER: Das Debakel um den Hauptstadt ughafen, Eine Chronik*. Rotbuch Verlag, 2013.
23. M. Farmer. The farmer review of the uk construction labour model. *Construction Leadership Council*, 2016.
24. E. García-Barriocanal, S. Sánchez-Alonso, and M.-A. Sicilia. Deploying metadata on blockchain technologies. In *Research Conference on Metadata and Semantics Research*, pages 38–49. Springer, 2017.
25. A. Grilo and R. Jardim-Goncalves. Value proposition on interoperability of bim and collaborative working environments. *Automation in construction*, 19(5):522–530, 2010.
26. D. Gross and E. Yu. From non-functional requirements to design through patterns. *Requirements Engineering*, 6(1):18–36, 2001.
27. M. Hosseini and N. Chileshe. Global virtual engineering teams (gvets): A fertile ground for research in australian construction projects context. *International Journal of Project Management*, 31(8):1101–1117, 2013.
28. M. Kassem, J. Lia, and D. Greenwood. Blockchain in the built environment: analysing current applications and developing an emergent framework. 2018.
29. A. Kiayias, A. Russell, B. David, and R. Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*, pages 357–388. Springer, 2017.
30. G. Kostka and J. Fiedler. *Large Infrastructure Projects in Germany: Between Ambition and Realities*. Springer, 2016.
31. J. Kraatz and A. Sanchez. R&d investment and impact in the global construction industry, 2014.
32. N. Lu and T. Korman. Implementation of building information modeling (bim) in modular construction: Benefits and challenges. In *Construction Research Congress 2010: Innovation for Reshaping Construction Practice*, pages 1136–1145, 2010.
33. Z. Ma, Z. Wei, and X. Zhang. Semi-automatic and specification-compliant cost estimation for tendering of building projects based on ifc data of design model. *Automation in Construction*, 30:126–135, 2013.
34. T. McConaghy, R. Marques, A. Müller, D. De Jonghe, T. McConaghy, G. McMullen, R. Henderson, S. Bellemare, and A. Granzotto. Bigchaindb: a scalable blockchain database. *white paper, BigChainDB*, 2016.
35. G. Mignone, N. Hosseini, M.R.and Chileshe, and M. Arashpour. Enhancing collaboration in bim-based construction networks through organisational discontinuity

theory: a case study of the new royal adelaide hospital. *Architectural Engineering and Design Management*, 12(5):333–352, 2016.
36. S. Mihindu and Y. Arayici. Digital construction through bim systems will drive the re-engineering of construction business practices. In *2008 international conference visualisation*, pages 29–34. IEEE, 2008.
37. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.
38. N. O. Nawari. Bim standard in off-site construction. *Journal of Architectural Engineering*, 18(2):107–113, 2012.
39. A. Norta. Creation of smart-contracting collaborations for decentralized autonomous organizations. In *International Conference on Business Informatics Research*, pages 3–17. Springer, 2015.
40. A. Norta. Designing a smart-contract application layer for transacting decentralized autonomous organizations. In *International Conference on Advances in Computing and Data Sciences*, pages 595–604. Springer, 2016.
41. A. Norta, P. Grefen, and N. C. Narendra. A reference architecture for managing dynamic inter-organizational business processes. *Data & Knowledge Engineering*, 91:52–89, 2014.
42. T. Oesterreich and F. Teuteberg. Understanding the implications of digitisation and automation in the context of industry 4.0: A triangulation approach and elements of a research agenda for the construction industry. *Computers in Industry*, 83:121–139, 2016.
43. E. Poirier, D. Forgues, and S. Staub-French. Collaboration through innovation: implications for expertise in the aec sector. *Construction management and economics*, 34(11):769–789, 2016.
44. S. Popov. The tangle, iota whitepaper, 2018.
45. R. Sebastian. Changing roles of the clients, architects and contractors through bim. *Engineering, Construction and Architectural Management*, 18(2):176–187, 2011.
46. W. Shen, Q. Hao, H. Mak, J. Neelamkavil, H. Xie, J. Dickinson, R. Thomas, A. Pardasani, and H. Xue. Systems integration and collaboration in architecture, engineering, construction, and facilities management: A review. *Advanced engineering informatics*, 24(2):196–207, 2010.
47. L. Sterling and K. Taveter. *The Art of Agent-Oriented Modeling*. The MIT Press, 2009.
48. D. Tapscott and A. Tapscott. *Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world*. Penguin, 2016.
49. P. Teicholz. Labor-productivity declines in the construction industry: causes and remedies (a second look). *AECbytes Viewpoint*, 2013.
50. C. D. Thimmaiah, S. Disha, D. Nayak, B. Diya, and H. Gururaj. Decentralized electronic medical records. 2019.
51. S. Thompson. *Haskell: the craft of functional programming*, volume 2. Addison-Wesley, 2011.
52. C. Udokwu, A. Kormiltsyn, K. Thangalimodzi, and A. Norta. The state of the art for blockchain-enabled smart-contract applications in the organization. In *2018 Ivannikov Ispras Open Conference (ISPRAS)*, pages 137–144. IEEE, 2018.
53. O. UML. Omg (2017) unified modeling language®(omg uml®) version 2.5. 1 https://www. omg. org/spec. UML, 2017.
54. J. Wang, P. Wu, X. Wang, and W. Shou. The outlook of blockchain technology for construction engineering management. *Frontiers of engineering management*, 4(1):67–75, 2017.

55. A. Wulf. Der rechtsformenmangel der öffentlichen hand. eine ökonomische fallstudie zu den rechtlich-institutionellen hindernissen in öffentlichen unternehmen am beispiel des flughafens berlin brandenburg (ber). In *Bedeutung des Wirtschaftsrechts für die volkswirtschaftliche Entwicklung*, pages 127–174. Springer, 2018.
56. Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. Chen, J. Weng, and M. Imran. An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems*, 105:475–491, 2020.