Commercial Property Tokenizing With Smart Contracts

Alex Norta
Large-Scale-Systems Group
Tallinn University of Technology,
19086, Tallinn, Estonia
Email: alex.norta.phd@ieee.org

Chad Fernandez
Blockgemini, Dubai UAE
Email: chad@blockgemini.com

Stefan Hickmott
Evarei and Evareium, Downtown Dubai, UAE
Email: info@evarei.com

***Abstract*—Trading commercial property currently involves several middlemen who are extra time-consuming and cost-consuming elements in a transaction. With recent blockchain-based smart contract innovations, it is possible to use distributed applications (Dapps) for disintermediating third parties and enabling direct peer-to-peer (P2P) transactions. The blockchain serves as an immutable, event-recording ledger that facilitates trust-less P2P transactions. The Evareium system is a distributed application for enabling blockchain-based commercial-property trades that make middlemen superfluous and consequently, investors benefit from cost reductions, faster transaction times, greater transparency and reduced regulatory burdens. As a consequence of the Evareium system, investors and other stakeholders are able to avail a direct interface with the creation and dissemination of optimal financial gains from such investments without interdiction. The corresponding Evareium token is structured for governing platform-related incentives. The Evareium token is also intended for Internet-of-Things (IoT) based hotel-management services such as controlling hotel-room blindfolds, A/C, smart keys, checkin, spa, restaurant, room service, and so on.**

***Index Terms*—Commercial property, tokenize, real-estate investment fund, smart contracts, blockchains, institution retail investors**

I. Introduction

Commercial property trading is plagued by being an inherently illiquid asset class [13] involving many hidden costs [18], regulations [8], middlemen and a lack of transparency [9]. In a recent study [3], blockchain-based smart contract solutions are mentioned as a remedy for enabling dis-intermediated peer-to-peer (P2P) transactions that allow investors to save costs and time while being less burdened with regulations.

With the advent of blockchain-based smart contracts [22], a trend emerges for establishing dis-intermediated collaboration structures [6] to engage in the formation of P2P transactions. Briefly, a smart contract is a computerized form of transaction protocol [21] that carries out terms of contracts. Thus, blockchain technology [15] is suitable to achieve immutable event-tracking of commercial property transactions. A blockchain is comparable to a distributed database that independently verifies artifact-ownership chains [12] that cryptographic digests create in hash values. To additionally support electronic P2P-transaction platforms, the emergence of service-oriented cloud computing (SOCC) [1] promises an, ad-hoc integration and coordination of information- and business-process flows [10], [11] to orchestrate and choreograph commercial property trades.

The state of the art shows that technologically it is possible to create P2P commercial transaction platforms that eliminate cost-creating and time-consuming intermediaries and obsolete stage processes. This paper fills the gap by answering the question of how to provide a smart-contract/blockchain based commercial property trading and value-chain platform that avoids dis-intermediated middlemen for enabling P2P trades with low costs and optimized time consumption. To establish a separation of concerns, we deduce the following subquestions; What are the system requirements for P2P commercial property trading- and holdings? What is the architecture of this trading- and holding platform? What is the dynamic stakeholder-engagement behavior within the platform?

Based on a pre-existing ICO[1] whitepaper [17], the remainder of this paper is structured as follows; Section II presents a running case about the status quo and the larger context of Dubai commercial property trading. Next, Section III gives functional- and quality requirements for the trading platform together with a positioning of stakeholders in that spectrum of requirements. Section IV describes the trading-platform architecture we derive from the requirement sets and Section V partially describes the dynamic engagement behavior of stakeholders that result in immutable event recordings in the trade-supporting blockchain. Section VI gives a deployment-feasibility evaluation and finally, Section VII concludes the paper and presents future work.

II. Preliminaries

We introduce in Section II-A a real-life running case about the commercial property market status quo in Dubai. Next, Section II-B presents additional background preliminaries.

A. Running Case

Too many institutions and companies act as intermediaries between investors who want to purchase real estate, while in the advent of blockchain, many of these middlemen add no fundamental value in the process. As Figure 1 shows,

[1] https://www.evareium.io/home

978-1-5090-6014-6/18/$31.00 ©2018 IEEE

banks, pension funds, and other real estate intermediaries take a percentage of the overall funds under their management; although owned ultimately by investors and the general public. Moreover, these institutions decide unilaterally how to lend and invest, keeping in many instances a large proportion of all the profits and gains for themselves. In addition, property developers pre-selling real estate units are often able to mask the profit the investor could really generate in such a project, and instead keep profits for themselves. Note that property developers use the investors capital to fund the entire project and are in doing so taking marginal-, or no meaningful financial risk in the development process accordingly.

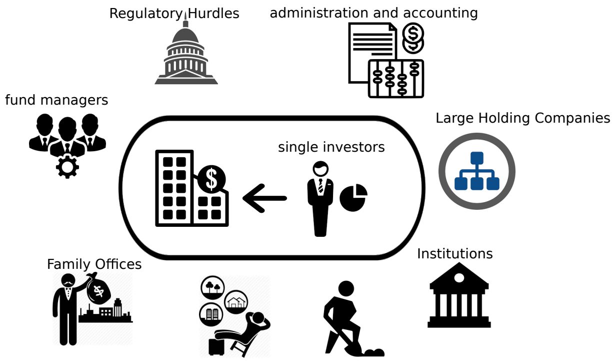


Fig. 1. Flow in architecture-types development.

B. Background Preliminaries

Goal models as part of the agent-oriented modeling (AOM) method [20] are used to specify the requirements of the Evareium system. Thus, thereby we acknowledge the Evareium system is highly decentralized, distributed and disintermediated. AOM goal models comprise the simple notation in Figure 2 (a) to specify the functional goals of a system, ‘quality goals’ or non-functional requirements, and agents with specified roles that may be human or artificial.

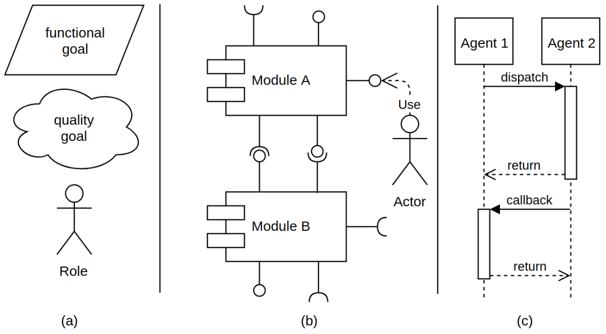


Fig. 2. Icons of AOM goal-model notation in (a), UML component diagrams in (b) and UML sequence diagrams in (c).

The root of a hierarchically decomposed AOM goal model is a ‘value proposition’ depicted as a functional goal that denotes the overall systems goal. Attached to functional goals are quality goals and roles. Note that attached quality goals and roles also hold for lower-level, refining functional goals.

From the goal model we deduce a unified modeling language (UML) component diagrams [4] to specify static structure of the Evareium system. Figure 2 (b) depicts UML notation elements with components being labeled rectangles that have attached provided- and required interfaces depicted as a line with a circle and required interfaces are lines with a cup at the end respectively.

To express the dynamic behavior of the Evareium system, we use UML sequence diagrams [19]. Figure 2 (c) shows two entities with the labels *Agent 1* and *Agent 2* with dashed timelines emanating vertically downwards. Directed straight arrows show messages between the depicted entities and the bars are so-called activations that denote processing threads.

III. Requirements for Automated Commercial Property Trading

While all refinement levels are explained in [17], the center of Figure 3 is the value proposition of the Evareium system with the label of business-to-business (B2B) crowdfunding platform investing in commercial real estate at the base level, automating transparency and structuring the operating entities to benefit retail investors. Hierarchically, below is the first refinement level of functional goals, namely, crowdsourcing pool of funds, Evarei fund management of fund distribution and resource allocation, property profile management, decentralized leasing platform for property management and stakeholder integrations, a dynamic synchronization of auction-pool- and property exit. The refining goals we explain in the subsections below in further detail.

Connected to the value proposition in Figure 3 are a set of quality goals that hold for the entire Evareium crowdfunding platform. *Strategic* pertains to an incentivized model that all stakeholders in the hierarchy inherit, enabling investor long-term investment decisions to be optimally met.

Secure [2] must hold for the entire system and is further decomposed into confidentiality, integrity, and availability. *Seamless* as a quality goal for the Evareium system stands for the precise allocation of integrable information exchange between external and internal platform infrastructures. *Direct* denotes P2P engagement between stakeholders, digitizing interactions that usually require in-person, or third-party authentication with a considerably increase of turn over time and human resources. *Liquid* pertains to the platform providing at all times sufficient funding for commercial property investment. *Verifiable* pertains to critical transactions on the Evareium platform being recorded on the blockchain(s) as immutable evidence. Finally, *mutable* means system settings of the crowdfunding platform are changeable based on commercial property trading needs.

The first refinement level under the value proposition also has quality goals attached. Decentralized and transparent do not hold for all lower-level functional goals and are therefore selectively assigned. *Decentralized* stands for collaboration

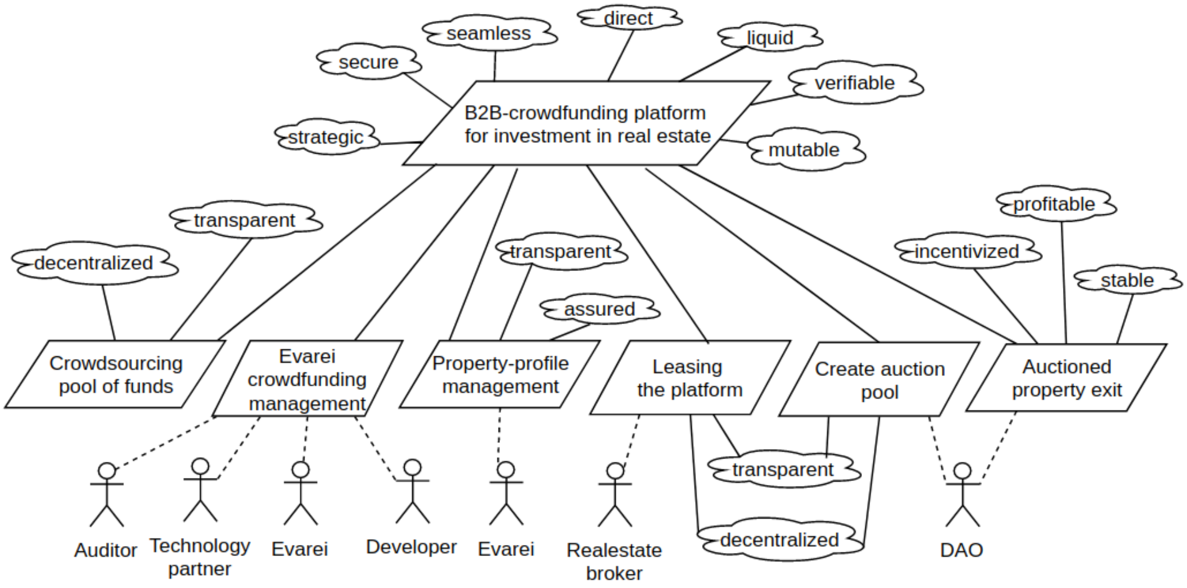


Fig. 3. The value proposition and first refinement level of the Evareium goal model.

taking place in a P2P way without any component of orchestration concentration taking control of communication channels. *Transparent* pertains to transactions and communications between parties being traceable for other system stakeholders. This is achieved by logging key events in the blockchain.

The functional goal of property profile management the quality goal *assured* linked to indicate it is certain provided stakeholder and commercial property data is based on evidence and truthful. Also the functional goal of auctioned property exit links to a separate set of quality goals. *Incentivized* means that stakeholders are encouraged by the Evareium system to engage and transact. Thus, closely related is the goal of *profitability*, meaning that commercial property auction exits result in a financial gain. Finally, *stable* pertains to the exit transaction being constant and as such, stored on the blockchain in an immutable way.

Figure 3 also shows roles attached to specific functional goals who are respective stakeholders in the Evareium system. The *technology partner* is associated to the functional goal for crowdfunding management and indicates that third parties are integrated in the platform for their complementary technological service provisions. *Evarei* as an organization also actively involves itself in crowdfunding management that additionally requires a *developer* for its establishment and a *committee* too. *Evarei* is also involved in profile management. A *real-estate broker* takes care of leasing the commercial property trading platform. Finally, a decentralized autonomous organization (DAO) [16] is orchestrated via business rules such as obligations and rights that are encoded as machine-readable smart contracts. Figure 3 associates a DAO role with the functional goals for creating an auction pool and the auctioned property exit.

IV. Architecture of the Trading Platform

The crowdsourcing pool component at the top left in Figure 4 has an embedded component for investment management. We omit refinement beyond first refinements levels in the architecture due to space limitation. Note the dashed gray box denotes EVT use correspondingly with the refining goal models in [17]. These components have several provided- and required interfaces.The embedded investment management component has a providing interface that delivers the Evarei actor with investment information. The providing interface receives investment information from the investor actor. The top-level component for the crowdsourcing funds pool also has a set of interfaces with a providing one delivering profile information of investors to the Evarei actor. A corresponding required interface exists for the investor to deliver profile information. Finally, two providing interfaces deliver investment- and profile information to the next component for Evarei crowdfunding management. Simultaneously, the crowdsourcing funds pool also has receiving interfaces that connect to the component for auction exit management, which we address in the sequel. Note the data-exchange labels in Figure 4 are conceptual placeholders for exchanges that are significantly more complex than we are able to express in the depicted top-level architecture.

The Evarei crowdfunding-management component at the top right of Figure 4 comprises a contained EVM token-management component that is gray-shaded to express EVT are applied on the first refinement level, while EVT are not applicable on the top-level per se. The only required interface of the EVM token-management component allows investors to authorize corresponding fund management. The remaining interfaces are connected to the Evarei crowdfunding-management component with only one being required so that

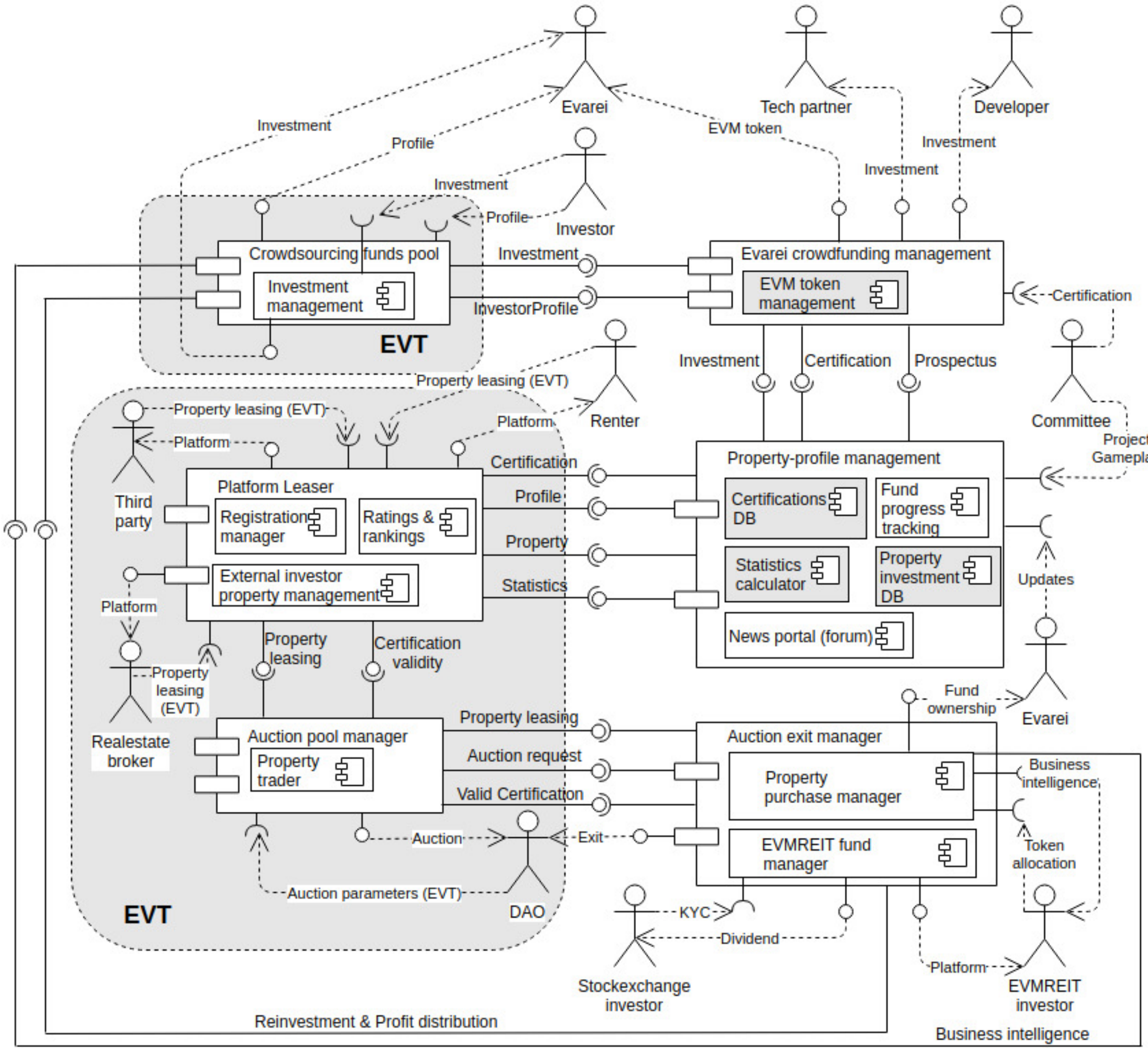


Fig. 4. The Evarei component diagram derived from earlier goal models.

a committee is able to submit certifications about commercial property. Out of the other providing interfaces, three allow for stakeholders to engage, i.e., the Evarei actor receives EVM tokens, the third technology partner and property developer receive investment information. The remaining two providing-interfaces deliver investment- and certification information to the component for property-profile management. Finally, a committee delivers valid certificates for commercial property via a receiving interface.

Note that EVT are employed in the components for the certification database (DB), the statistics calculator and the property investment DB. The components for fund-progress tracking and the news portal do not require EVT. Two interfaces receive investment and certification information while the third receiving interface has the Evarei actor deliver updates about the property management. The remaining providing interfaces of the property-profile management deliver to the platform-leaser component certification, profile, property and statistics information.

The platform-leaser component in the center left of Figure 4 has first-level refinement components termed registration manager, ratings and rankings and external investor property manager. A third-party actor receives platform information while he delivers EVT for leasing property. The same holds for the realestate broker actor and renter in Figure 4. The renter receives platform information for leasing and submits in return EVT. The remaining providing interfaces of the platform-leaser component deliver information to the auction-pool manager component about property leasing and validating certificates.

The auction-pool manager at the bottom left of Figure 4, employs EVT together with the platform leaser. The contained property-trader component is the first-level architecture refinement and besides two required interfaces for receiving information about property leasing and certification validity, a third providing interface allows DAOs to deliver auction parameters and EVT. The DAO receives via a providing interface auction information while the remaining two providing interfaces deliver information for property leasing and auction requests to the auction-exit manager component.

The auction exit manager component at the bottom right of Figure 4 is the commercial property lifecycle sink. Note that the functional goals in Figure 4 for democratic voting and determine voting period, we merge into a first-refinement-level component termed voting manager. Also the functional goals for generating EVMREIT funds and for reinvesting into EVM Evareium we merge into a component for EVMREIT fund manager. The final refining component is a property purchase manager that has attached a providing interface so that the Evarei actor can withdraw fund-ownership information. The second providing interface of the property purchase manager component allows EVMREIT investors to deliver information about token allocations and business intelligence. The refining EVMREIT fund manager component has providing interfaces for allowing the EVMREIT investor to withdraw platform information and also the stockexchange investor is able to withdraw dividend information. For the latter, the stockexchange investor must deliver know-your-customer (KYC) information. Finally, on the top level of the auction exit manager component, besides receiving information about property leasing and auction requests via required interfaces, the DAO receives concrete exit information via a providing interface. Finally, generated business intelligence together with reinvestment and profit distribution is channeled back to the top left component termed crowdsourcing funds pool to support the next commercial property deal.

V. Dynamic Stakeholder Engagement

We describe and specify the dynamic behavior of the Evareium system with UML sequence diagrams [19], following the notation of Figure 2(c). The sequence diagrams show data-exchange protocols and data services between roles with the components described in Figure 4.

The remainder is structured as follows. Section V-A lists the blockchain operations for achieving event traceability for commercial property trade. Section V-B explains the leasing-platform protocol as one example and we refer the reader to [17] for the complete set of sequence diagrams.

A. Blockchain Operations

Storing operations of events on blockchains is costly with respect to transaction fees[2] and consumed energy[3]. Note that PoW to solve cryptographic riddles for transaction validation is a performance and scalability bottleneck for highly distributed systems such as Evareium. Thus, as outlined in Table I, it is important to know the minimal set of transactions for the required traceability in the Evareium property trading system.

Table I lists the operations with the *event* column giving identification numbers of respective blockchain operations. Additionally, we also list the *stakeholder* set per operation and a corresponding *explanation* per operation. The first column categorizes the operations into subsets that are assigned to respective components from Figure 4. Note the event IDs we use below in the sequence diagrams are to show at which moments in the protocols the blockchain is used.

B. Platform Leaser

The sequence diagram in Figure 5 starts with the platform property manager (PPM) dispatching the valid certificate *cert*, profile *prof* of the commercial property *prop* and related statistics *stats*. Next, the platform leaser registers the broker, profile and tenant with the registration manager who then subsequently also receives registration information from the third party, realestate broker, renter. The component for external investor property management *EIPM* receives the accumulated registration information.

The depicted loop in Figure 5 by a rounded rectangle denotes that the leasing process is repetitive and involves the renter, realestate broker and the third party. In order to make their respective leasing, these stakeholders use EVT. Finally, a dashed rounded rectangle denotes that the last part of the protocol in Figure 5 is optional. Thus, the EIPM puts forward a service offer to a third party, e.g., for delivering commercial property statistics to the third party. Assuming the latter confirms the service offer, the EIPM requests relevant statistics for a specific ongoing leasing context to the component for generating ratings and rankings. The received information the EIPM next dispatches to the third party, realestate broker and renter for educated decision making.5

Several blockchain transactions are depicted as red circles in Figure 5 numbered from 9 to 15 in correspondence with Table I. In Transaction *10*, the platform leasing details are stored in the blockchain. Also the registration of third party, realestate broker and renter are stored in the blockchain with Transactions *11* to *13* respectively. For the EVNBnB smart leasing in the loop of Figure 5, the concrete leasing events Transaction *9* stores in the blockchain. The optional final part of the leasing protocol stores with Transaction *15* the fact that a specific service offer is confirmed by a third party. Also the distribution of the statistics by the EIPM are stored as to then have evidence for service provision compensation.

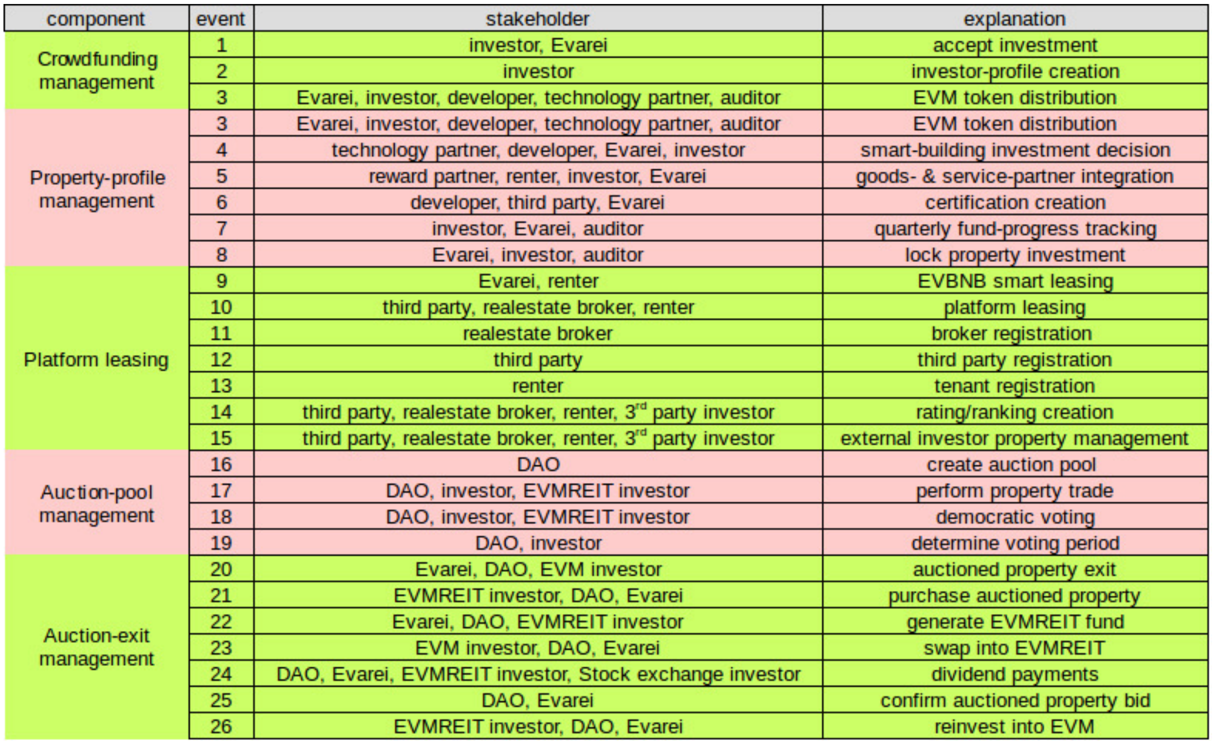
VI. Feasibility Evaluation

For blockchain-based mass data storage of the Evareium system, the InterPlanetary File System[4] (IPFS) [5] is suitable as a content-addressable, peer-to-peer (P2P) and distributed hypermedia protocol that yields an open-source file system. IPFS uses a high performance, decentralized block-storage model being an alternative for the regular HTTP protocol. IPFS allows for hyperlinks to address data sets in a highly performing block-storage model with data distribution across

[2] https://www.coindesk.com/ethereums-double-edged-sword-will-rising-price-hurt-users/

[3] https://www.technologyreview.com/s/609480/bitcoin-uses-massive-amounts-of-energybut-theres-a-plan-to-fix-it/

[4] https://ipfs.io/



component	event	stakeholder	explanation
Crowdfunding management	1	investor, Evarei	accept investment
	2	investor	investor-profile creation
	3	Evarei, investor, developer, technology partner, auditor	EVM token distribution
Property-profile management	3	Evarei, investor, developer, technology partner, auditor	EVM token distribution
	4	technology partner, developer, Evarei, investor	smart-building investment decision
	5	reward partner, renter, investor, Evarei	goods- & service-partner integration
	6	developer, third party, Evarei	certification creation
	7	investor, Evarei, auditor	quarterly fund-progress tracking
	8	Evarei, investor, auditor	lock property investment
Platform leasing	9	Evarei, renter	EVBNB smart leasing
	10	third party, realestate broker, renter	platform leasing
	11	realestate broker	broker registration
	12	third party	third party registration
	13	renter	tenant registration
	14	third party, realestate broker, renter, 3rd party investor	rating/ranking creation
	15	third party, realestate broker, renter, 3rd party investor	external investor property management
Auction-pool management	16	DAO	create auction pool
	17	DAO, investor, EVMREIT investor	perform property trade
	18	DAO, investor, EVMREIT investor	democratic voting
	19	DAO, investor	determine voting period
Auction-exit management	20	Evarei, DAO, EVM investor	auctioned property exit
	21	EVMREIT investor, DAO, Evarei	purchase auctioned property
	22	Evarei, DAO, EVMREIT investor	generate EVMREIT fund
	23	EVM investor, DAO, Evarei	swap into EVMREIT
	24	DAO, Evarei, EVMREIT investor, Stock exchange investor	dividend payments
	25	DAO, Evarei	confirm auctioned property bid
	26	EVMREIT investor, DAO, Evarei	reinvest into EVM

TABLE I
BLOCKCHAIN TRANSACTIONS FOR THE EVAREIUM SYSTEM.

several computers. A single computer participates with storing data subsets using content addressing, hash-linked lists and allows to develop distributed blockchain applications on top by placing immutable, permanent IPFS links into a blockchain transaction.

For complex operations on large datasets, BigchainDB [14] is a suitable system also for constructing profiles. BigchainDB is a blockchain database with immutable decentralized control and the ability of creating and moving digital assets. Essential for BigchainDB as a candidate for Evareium system consideration is the ability of connecting to other decentralized systems such as IPFS, Ethereum, Qtum and so on.

As a non-permissioned smart-contract platform for the Evareium system, the earlier discussed Ethereum is a potential candidate. Still, one of the profound Ethereum disadvantages is that PoW creates performance and scalability challenges that may not allow a large-scale use in the Evareium system. It is our observation based on Table I that the Evareium system requires a considerable amount of blockchain operations to immutably store critical collaboration events. The equally non-permissioned Qtum smart-contract solution uses SPV and UTXP for supporting lite wallets that may run on mobile devices. While Ethereum has announced the development of a PoS version, it is still not clear when such a version will be available. On the other hand, PoS is already fully functioning in a third-party Ethereum version[5] and also in the Qtum system. Both Ethereum and Qtum use Solidity as a smart-contract development language. A permissioned alternative to Ethereum and Qtum is Hyperledger [23] that aims to create an open-source distributed ledger for enterprise-scaling applications with a code base. Hyperledger Fabric [7] is a modular implementation specifically for running smart contracts that also provides pluggable implementations of various functions.

Since the Evareium system assumes smart commercial property management that employs IoT devices, other classes of distributed ledger technology exist that are specifically suitable for such cyber-physical systems (CPS). For example, IOTA[6] uses directed acyclic graphs (DAG) as an alternative to a blockchain that yields free and highly scaling simultaneous transactions with fast confirmation times irrespectively of their size. More recently, Hashgraph[7] is an alternative DAG system comparable to IOTA that is equally suitable for CPS applications.

Developing sound smart-contracts without security flaws is a challenge and the architecture of Figure 4 leads to the conclusion an Evareium system implementation requires a large set of cross-dependent smart contracts that must not yield concurrency conflicts or dependability issues [2]. Several systems exist for evaluating contracts. The Securify[8] online service formally verifies Ethereum smart contracts and checks

[5] https://www.cryptoninjas.net/2018/01/19/ethereum-proof-stake-implementation-launched-applicature/

[6] https://iota.org/

[7] https://hashgraph.com/

[8] https://securify.ch/

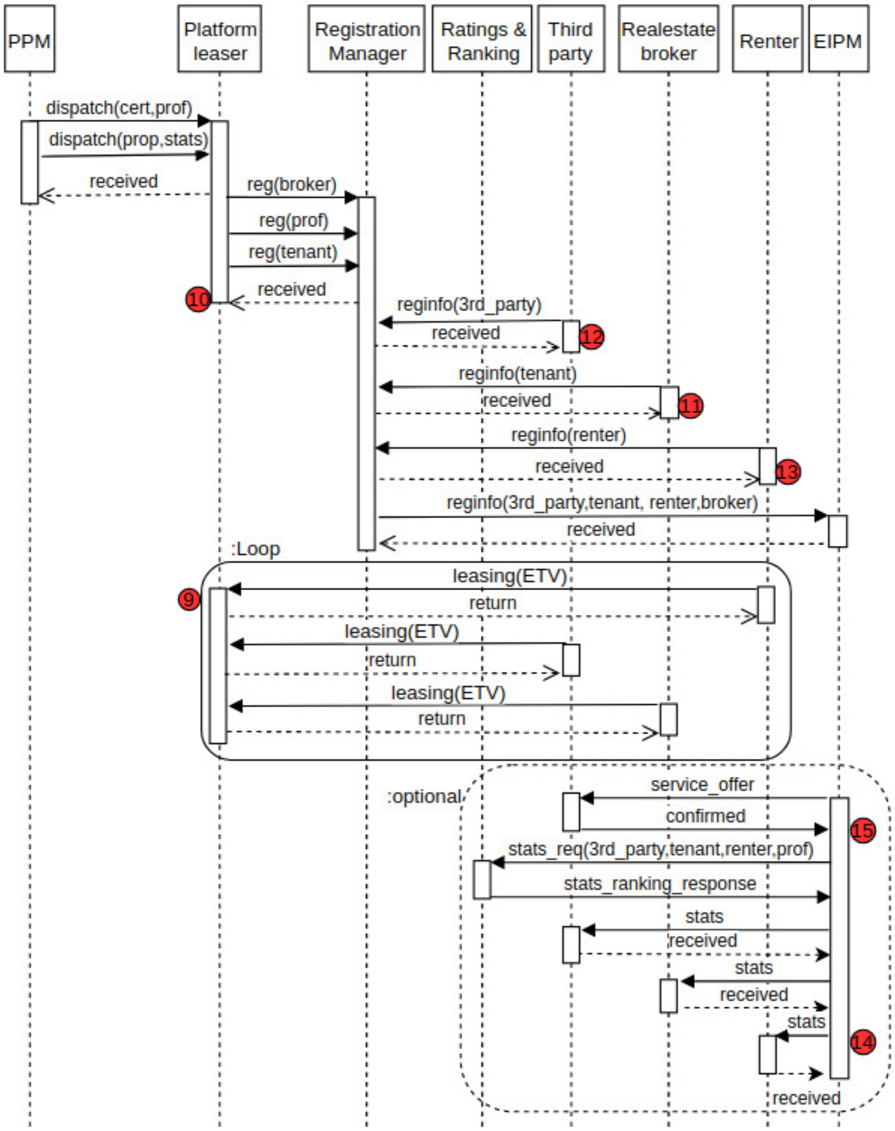


Fig. 5. The Evarei leasing-platform sequence diagram.

insecure coding with critical security issues. Currently, only a beta-version of Securify exists and lacking documentation poses a challenge to estimate which formal properties are checked in what specific way. An online Securify example checks for transaction recordings, recursive calls, insecure coding patterns, unexpected Ether flows and the use of untrusted inputs in security operations. Also the Embark-framework[9] and Populus[10] for smart-contract development and deployment are currently not mature enough for satisfactory formal verification and evaluation.

VII. Conclusion

This paper is based on an ICO whitepaper and presents the novel blockchain technology-based Evareium system for a smart-contracts driven trade of commercial property. We first define the requirement that must be satisfied by presenting goal models comprising tree-organized sets of refining functional goals. Assigned to the tree of functional goals are quality goals and roles of stakeholders in the Evareium system. Derived from the goal model, we next give a UML component-diagram architecture that gives the static structure of the Evareium system and is organized in a lifecycle starting with the crowdsourcing funds pool. Subsequently, the dynamic behavior of the components are expressed in UML sequence diagrams that show blockchain transactions for events that must be immutably stored. The sequence diagram shows the most amount of predictable blockchain transactions are part of the protocol for platform leasing. The feasibility study shows with what pre-existing industry solutions the Evareium system can be quickly deployed. At the same time, we also show

[9] https://github.com/iurimatias/embark-framework
[10] http://populus.readthedocs.io/en/latest/

that the aspect of creating verifiable smart contracts for the Evareium system is still a topic of ongoing research.

To answer the relevant questions for the development of the Evareium system, the value proposition of the goal model is the development of a B2B-crowdfunding platform for investment in real estate. This value proposition is refined into six branches, namely for crowdsourcing pool funds, Evarei crowdfunding management, property profile management, auction platform leasing, the creation of an auction pool and the auction-property exit. The majority of quality goals are associated with the value proposition being strategic, secure, seamless, direct, liquid, verifiable and mutable. Assigning these quality goals with the value proposition denotes the former hold for all functions of the system. Important is also which branches of the goal model must be realized in a decentralized and transparent way, namely, the crowdsourcing of pool funds, platform leasing and the creation of auction pools. Additionally, the goal model assigns the Evareium stakeholders being a renter, reward partner, investor, Evarei, committee, technology partner, developer of commercial property, third party, realestate broker, DAO, EVM investor, EVM-REIT investor and stock exchange investor.

The component architecture we derive from the goal model is organized in a lifecycle starting with the crowdsourcing funds pool. The component diagram shows conceptually labeled interface exchanges in one direction towards the lifecycle sink being the component for auction-exit management. Furthermore, we also assign actors in the component diagram corresponding to the roles of actors in the goal diagram. Gray-shaded areas and components depict in which parts of the architecture EVT tokens are used for monetization means. The sink of the commercial-property lifecycle termed auction-exit manager, connects to the lifecycle start of crowdsourcing funding by transferring via interfaces reinvestment with profit distribution and also accumulated business intelligence that both are instrumental for the next commercial property trade.

The dynamic behavior of the Evareium system that we depict with sequence diagrams, show that the auction-exit management protocol is the most elaborate one, while the most amount of predictable blockchain transactions are part of the protocol for platform leasing. The latter protocol is less elaborate than the auction-exit management that requires almost the same amount of blockchain transactions. Property profile management is ranked third in terms of required blockchain transactions and involves the most amount of entities. Still, the sequence protocol is not as elaborate.

Finally, the limitation of the Evareium system that requires scientific attention is the challenge of high distribution. This informal representation of the Evareium system does not address in detail the predictable dependability and concurrency conflicts that will occur in an application implementation. Understanding these dependabilities and concurrency conflicts is important to prevent deadlocks and unnecessary loops between Evareium system components. Furthermore, the sound development of smart contracts for the Evareium system is still a challenge.

Bio

Alex is an associate professor[11] at TTU.ee and has written the whitepapers for the ICOs of Qtum.org, Agrello.org, Everex.io, CEDEX.com, Datawallet.io, Evareium.io, more to come.

References

[1] M. Aiello, E.B. Johnsen, S. Dustdar, and I. Georgievski. *Service-oriented and Cloud Computing*. Springer, 2016.

[2] A. Avizienis, J.C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions on dependable and secure computing*, 1(1):11–33, 2004.

[3] A. Baum. Proptech 3.0: the future of real estate. 2017.

[4] D. Bell. Uml basics: The component diagram. *IBM Global Services*, 2004.

[5] J. Benet. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*, 2014.

[6] V. Butterin. A next-generation smart contract and decentralized application platform, 2014.

[7] C. Cachin. Architecture of the hyperledger blockchain fabric. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, 2016.

[8] E. Cerutti, J. Dagher, and G. Dell'Ariccia. Housing finance and real-estate booms: a cross-country perspective. *Journal of Housing Economics*, 38:1–13, 2017.

[9] B. Danielsen, D. Harrison, R. Van Ness, and R. Warr. Liquidity, accounting transparency, and teh cost of capital: Evidence from real estate invetment trusts. *Journal of Real Estate Research*, 36(2):221–251, 2014.

[10] R. Eshuis, A. Norta, O. Kopp, and E. Pitkanen. Service outsourcing with process views. *IEEE Transactions on Services Computing*, 99(PrePrints):1, 2013.

[11] R. Eshuis, A. Norta, and R. Roulaux. Evolving process views. *Information and Software Technology*, 80:20–35, 2016.

[12] E. Gaetani, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone. Blockchain-based database to ensure data integrity in cloud computing environments. In *ITASEC*, pages 146–155, 2017.

[13] D.C. Ling, A. Naranjo, and B. Scheick. Credit availability and asset pricing dynamics in illiquid markets: Evidence from commercial real estate markets. *Journal of Money, Credit and Banking*, 48(7):1321–1362, 2016.

[14] 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.

[15] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Consulted*, 1(2012):28, 2008.

[16] A. Norta. Creation of smart-contracting collaborations for decentralized autonomous organizations. In *International Conference on Business Informatics Research*, pages 3–17. Springer, 2015.

[17] A. Norta, C. Fernandez, and S. Hickmott. Tokenizing commercial property with smartcontracts. *URL: https://www.evareium.io/assets/img/pdf/Evareium-technical-whitepaper.pdf*, 2018.

[18] D. Pogue and N. Miller. Sustainable real estate and corporate responsibility. In *Routledge Handbook of Sustainable Real Estate*, pages 19–36. Routledge, 2018.

[19] J. Rumbaugh, I. Jacobson, and G. Booch. *Unified Modeling Language Reference Manual, The (2Nd Edition)*. Pearson Higher Education, 2004.

[20] L. Sterling and K. Taveter. *The art of agent-oriented modeling*. MIT Press, 2009.

[21] M. Swan. Blockchain thinking: The brain as a dac (decentralized autonomous organization). In *Texas Bitcoin Conference*, pages 27–29, 2015.

[22] N. Szabo. Formalizing and securing relationships on public networks. *First Monday*, 2(9), 1997.

[23] M. Vukolić. Rethinking permissioned blockchains. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, BCC '17, pages 3–7, New York, NY, USA, 2017. ACM.

[11] https://tinyurl.com/Alex-Norta