We need a fancy title
Whitepaper - Version 0.1
May 1, 2018

Benjamin Leiding[1], Will Vorobev[1], Peter Zverkov[1], and Lena Cherry[1]

Chorus Technology

Abstract. I am an abstract - pet me.

DON'T FORGET THE PLAGIARISM CHECK

Keywords: keywords

1 Introduction

Despite steadily growing public transport networks and systems, especially in most first world countries, cars and similar vehicles are still the default standard for urban transportation. In the US, "about 86 percent of all workers commuted to work by private vehicle, either driving alone or carpooling" [23] even though in recent years the numbers remained relatively stable after decades of consistent increase - similar applies to other industrial countries [24][36] even though the overall percentage of vehicle commuters in Europe is lower than in the US [9]. While it was normal for the last few decades to own a vehicle and commute on a day-by-day basis, the future will be radically different due to the progressing evolution of self-driving cars and autonomous vehicles. The car-sharing economy that developed in recent years in combination with autonomous cars results in a so called *passenger economy* [16]. Users no longer own cars, instead just hop on an autonomous cars, pick a destination and get delivered without any human interaction. An Intel report estimates the size of this economy to be around US$ 7 Trillion in 2050 [16].

Despite some recent setback, e.g. Uber and Tesla accidents [?][?][?], academic researchers as well as companies from the private sector make fast progresses in the research area of self-driving cars [?][?]. It took less than 15 years from the first DARPA Grand Challenge (a prize competition for autonomous vehicles) to self-driving cars operating on public streets on a regular base (Tesla, Waymo, Uber, etc.) [?][?][?]. Besides the cars, several projects are also already working on system solutions for trucks, rovers, drones, ships and even airplanes [?][?][?][8]. But progressing automation and driverless transport that enables the passenger economy is only a small aspect of the potential of these new technologies. During a talk[1] at the 2013 Turing Festival in Edinburgh, Mike Hearn did not only

---

[1] https://www.youtube.com/watch?v=MVyv4tOOKe4

described a vision where most users don't own cars anymore and instead use services provided by autonomous vehicles that own itself, but also the potentials of a vehicle-to-vehicle (V2V) as well as a vehicle-to-infrastructure (V2I) economy. Autonomous vehicles (AVs) may own themselves, offer services and goods to earn money, and pay money to acquire services that they cannot provide on their own, e.g., car renting a parking lot, paying for a charged battery, using toll roads, or simple service check ups. The idea of V2V and V2I or in general V2X (vehicle-to-everything) will fuel various new business fields.

Certainly, traditional payment systems such as paper money or fiat currencies in general are not suited to be part of this new economy. There are slow, depend on third parties (e.g., banks) and suffer from bureaucratic overhead. Blockchain technology and cryptocurrencies offer a promising alternative payment solution that comes with several additional advantages that we will discuss later on. The blockchain technology, also referred to as distributed ledger system, is most noticeably known for providing the foundation of the peer-to-peer (P2P) cryptocurrency and payment system Bitcoin [25], but nowadays there a various different platforms out there, e.g., [15][30][38]. Several companies already started to prototype applications that combine vehicles and blockchains. Porsche is researching different payment-related applications for vehicles [31] whereas Ford focuses on traffic marshaling [20]. As expected in the early days of a new technology, companies focus on selective solutions for a selection of very specific problems or use cases and the resulting solutions are only compatible with their own products. What is currently missing in the new business field of V2X economy is an industry standard that can easily be integrated with self-driving and (semi)-autonomous cars or even nowadays cars.

This whitepaper addresses the detected gap by introducing the Chorus Technology solution, thereby answering the question of how to implement a library for (semi)-autonomous vehicles that enables a V2X trading platform for goods and services? In order to answer this question with a separation of concerns, we pose the following sub-questions: What is the long term vision of Chorus Technology? What are the critical requirements and the corresponding architecture of the Chorus platform? What are the system-engagement processes for the stakeholders?

The remainder of this paper is structured as follows: Section 2 introduces supplementary literature and related work. Section 3 then outlines the vision of Chorus Technology as well as different use-cases. Afterwards, Section 4 analyses the requirements of the our system and outlines the resulting system architecture that we derive from the requirements. Afterwards, Section 5 expands on the system-engagement processes for the stakeholders, followed by Section 6 that introduces the Chorus prototype as well as feasibility evaluation. Section 7 provides an discussion and an analysis of related projects. Finally, Section 8 concludes this work and provides an outlook on future work.

2 Technical Background and Related Works

The following section provides background information and describes related works regarding previous ideas and concepts that focus on a blockchain-based VANET platforms. First, Section 2.1 introduces the general concepts of blockchain technology, terms and frameworks. Afterwards, Section 2.2 and Section 2.3 focus on the fundamentals of autonomous vehicles as well as vehicular ad-hoc networks. Finally, Section 2.4 focuses on related work.

2.1 Blockchain Technology

As the name suggests, a blockchain consists of a chronologically ordered chain of blocks. Every block consists of a certain number of validated transactions and each of those block links to its predecessor by a hash reference. As a result, changing the content of one block also changes all succeeding blocks and hence breaks the chain. All blocks are stored on and verified by all participating nodes. While the initial Bitcoin blockchain only supported a very limited set of scripting instructions, the next generation of blockchain platforms, e.g., Ethereum [38], Qtum [10], or Tezos [15], provide Turing-complete programming languages on the protocol-layer level in order to enable smart contract capabilities. Smart contracts are, "orchestration- and choreography protocols that facilitate, verify and enact with computing means a negotiated agreement between consenting parties" [10]. Hence, the parties participating in the enactment of a smart contract establish binding agreements and deploy applications using such smart contracts in order to provide blockchain-based applications. Those application are as versatile as smart contracts itself and enable services including the finance sector [26][33], academic and business authentication and identity solutions [3][7][18][22][34], reputation systems [5] as well platforms for Internet-of-Things (IoT) applications [6][28].

The blockchain concept is particularly interesting for the V2X economy for three reasons. First, it removes the need for trusted third parties and instead enables trust-less transaction enactment. Second, transactions that were agreed up on cannot be changed any more since the underlying blockchain is tamperproof. Third, no human interaction is required for any kind of transaction between vehicles or machines in general.

2.2 Autonomous Vehicles

Work-In-Progress

### 2.3 Vehicular Ad-Hoc Networks - VANETs

Communication between vehicles, road infrastructure and Internet-based services is a key enabler of upcoming generation of vehicles. So called vehicular ad-hoc networks provide an abstract concept that models the different components that are required for V2V, V2I and V2X communication. As illustrated in

Figure 1, the basic components of VANETs are vehicles, on-board-units (OBUs), application-units (AUs) and road-side-units (RSUs).

RSUs are placed along the road side or in dedicated locations such as at cross-roads. Usually, RSUs provide short range communication based on IEEE 802.11p radio technology but can also be equipped with other network devices in order to provide communication within the infrastructural network [1]. An OBU is typically mounted onto a vehicle and used for exchanging information with RSUs or other OBUs. Short range wireless communication or other radio technologies are usually used to exchange information [2].

Closely linked to the OBU is the AU. They might even reside in the same physical unit or is mobile and might be regularly removed from the vehicle (e.g smartphones). The AU provides an execution environment for applications that utilize the OBU's communication capabilities [1][2].

Communication in VANETs occurs either inside a vehicle between AUs and
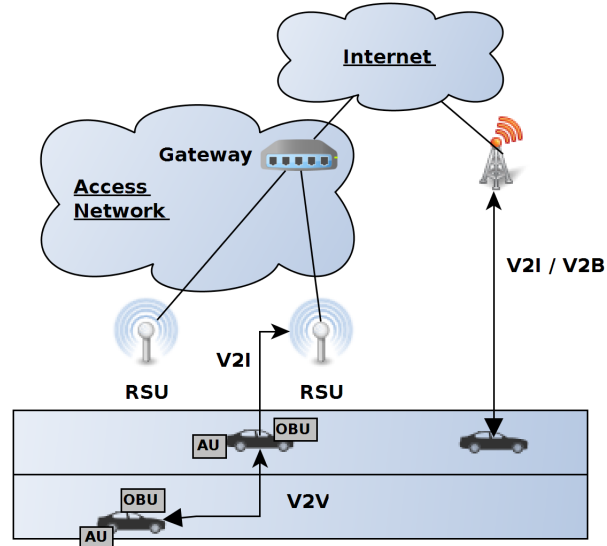
Fig. 1. General VANET architecture (Based on: [2] and [19])

OBU, wirelessly between different vehicles (V2V), vehicles and infrastructure (V2I) or vehicles and the infrastructure via broadband (V2B) [13]. For authentication purposes, each network participant is equipped with a unique public/private key pair which usually resides in a tamper-proof-device (TPD). In blockchain terms, the TPD is similar to an external hardware wallet.

2.4   Related Work

Work-In-Progress

3 The Chorus Vision

Intro

3.1 Use Cases

Work-In-Progress

3.2 Human to Human

3.3 Human to Vehicle

3.4 Vehicle to Vehicle

3.5 Vehicle to Infrastructure

4 System Design and Architecture

The vision of Chorus outlined in Section 3 is now analyzed from technical perspective as part of the following section. In order to identify, structure and formalize the critical requirements and stakeholders on an abstract level, we use one part of an Agent-Oriented Modeling (AOM) method [37], i.e., goal models. Section 4.1 introduces AOM goal models and the Chorus specific goal model. The produced goal model is used in subsequent Section 4.2 to derive the Chrous system architecture . The resulting system architecture and specifications serve as implementation guidelines. Finally, Section 4.3 provides some details on the smart contract lifecycle management within our solution, whereas Section 4.4 outlines a general outlook on the APIs and how to integrate the Chorus library into external applications.

4.1 Functional Goals, Quality Goals, Stakeholders and Requirements

In system development and software engieerng, good requirements follow certain characteristics. According to [11][14] requirements address one issue only and are completely specified without missing information. Moreover, they have to be consistent and do not contradict itself, or in correlation with other requirements. Finally, a requirement must also be atomic and without conjunctions [27].
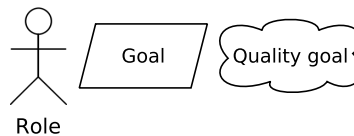
Fig. 2. Selection of AOM notation elements.

The AOM methodology is a socio-technical requirements-engineering approach used to model complex systems that consist of humans, devices, and software agents. An AOM goal model enables both, technical- and non-technical stakeholders, to capture and understand the functional- and non-functional requirements of a complex system. Figure 2 depicts the three main elements that an AOM goal model comprises in order to capture the system requirements and goals. Roles of involved entities are represented in form of sticky men, whereas functional requirements are depicted as parallelograms. Functional requirements are also referred to as goals. Non-functional requirements are depicted as clouds and refer to quality goals of the modeled software system. The AOM goal model follows a tree-like hierarchy with the root value proposition of the modeled system at the top. Subsequently, this main goal is decomposed into sub-goals where each sub-goal represents an aspect for achieving its parent goal [21]. The goals are further decomposed into multi-layered sub-goals until the lowest atomic level is reached. Additionally, roles and quality goal may be assigned to goals and are inherited to lower-level goals.

4.1.1 Top-Level AOM Goal Model

work-in-progress

Maybe reference to [17] as reference architecture and that the AOM goal model looks pretty similar since it is an instance of what is described in that academic paper.

Note that a sticky man does not exclusively represent human entities but rather all kinds of entities, e.g., humans, vehicles, infrastructure.
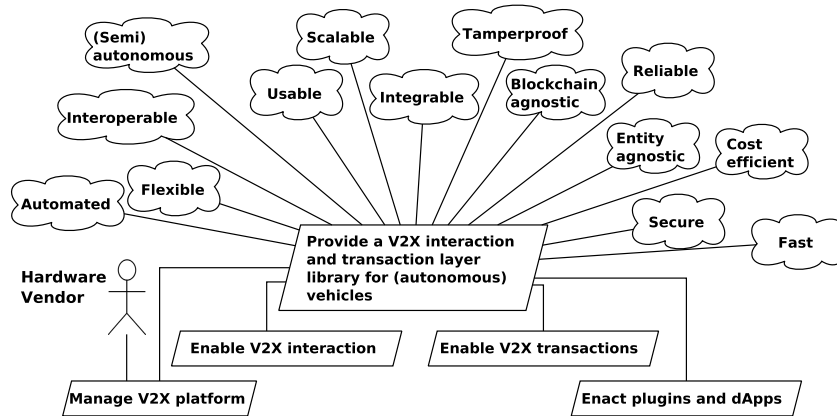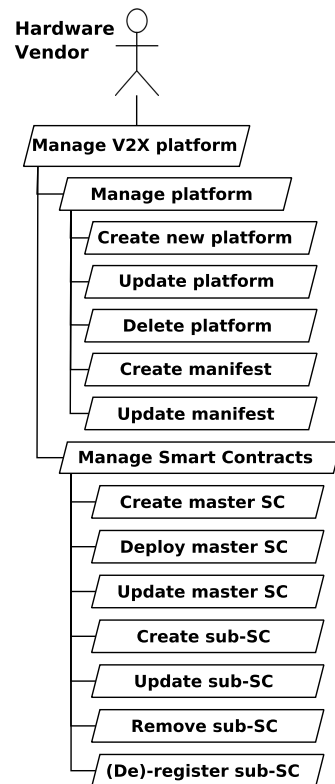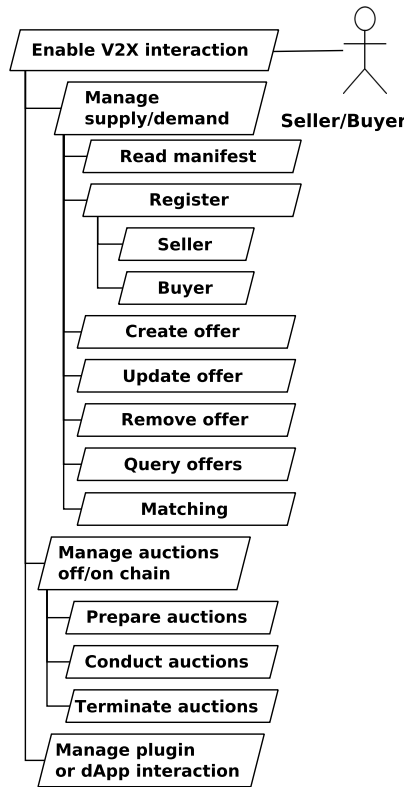

Fig. 3. Chorus - Top-level goal model representation (Source: Based on [17]).

4.1.2 Refined AOM Goal Model

**Fig. 4.** Chorus - Goal model refinement *Manage V2X platform.*

Enable V2X interaction

Seller/Buyer

Manage supply/demand

Read manifest

Register

Seller

Buyer

Create offer

Update offer

Remove offer

Query offers

Matching

Manage auctions off/on chain

Prepare auctions

Conduct auctions

Terminate auctions

Manage plugin or dApp interaction

Fig. 5. Chorus - Goal model refinement *Enable V2X interaction.*
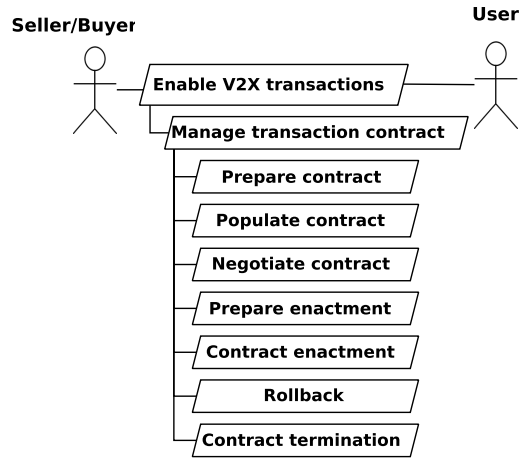
**Fig. 6.** Chorus - Goal model refinement *Enable V2X transaction.*
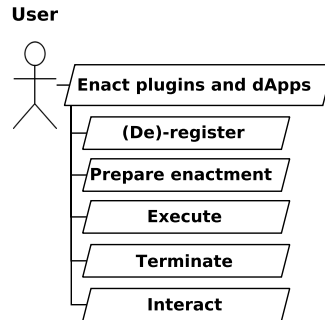
Fig. 7. Chorus - Goal model refinement *Plugin and dApp enactment.*

4.2 Component Diagrams

The abstract system- and business architecture is derived from the functional- and non-function requirements of the AOM goal model presented earlier. The services are powered by a service-oriented architecture (SOA) that is comprised of different designated components. Each of these components is self-contained, well-defined components and provides a specific set of services [12][29]. Dedicated services and components may also consist of other underlying sub-services [32].

In the following, a technology-agnostic UML-component-diagram represen- tation is used to illustrate the system architecture [4][35]. The UML notation elements used to model the architecture are presented in Figure 8. In UML, components are represented as rectangular boxes and labeled either with the

keyword *component*, or with the component icon in the right-hand upper corner. A component may consists of further sub-components and is implemented by one, or more classes, or objects. Moreover, components are reusable and communicate via two types of interfaces as illustrated in Figure 8. Small squares depict ports that are attached to the border of components and expose required and provided interfaces. Ports may also specify inputs and outputs as they operate uni-, or bi-directionally [4][35]. Once more, sticky men are used to depict entities and their interactions with the system.
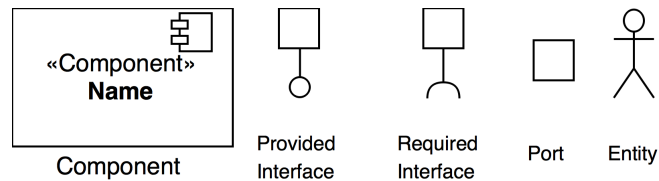
Fig. 8. UML-component diagram notation elements.

The remainder of this section first introduces an abstract high-level overview of the system architecture and components. Afterwards, further illustration present selected sub-components of our architecture.

4.2.1   High-Level Architecture

4.3   Smart Contract Lifecycle Management

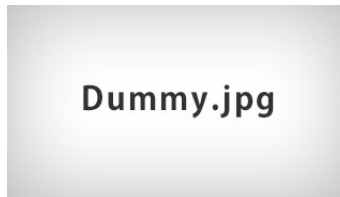**Work-In-Progress**

4.4   Library / API

Work-In-Progress

5   System Engagement Processes

Intro

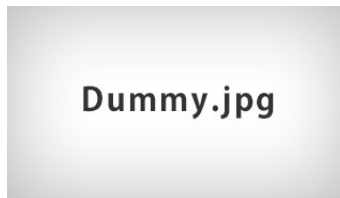### 5.1   Sequence Diagrams — or BPMN representation of Processes

Work-In-Progress

Fig. 9. Caption.

5.2 Auction Algorithm

Work-In-Progress

Fig. 10. Caption.

5.3 Token Economics and Value Proposition

Work-In-Progress

6 Prototype and Feasibility Study

Intro

6.1 Prototype

Work-In-Progress

6.2 Evaluation

Work-In-Progress

7 Discussion

Intro

7.1 Critical Analysis

Work-In-Progress

7.2 Related Work

Work-In-Progress

8 Conclusion and Future Work

We conclude ...

Work-In-Progress

References

1. Al-Sultan, Saif and Al-Doori, Moath M and Al-Bayatti, Ali H and Zedan, Hussien: A Comprehensive Survey on Vehicular Ad Hoc Network. Journal of Network and Computer Applications 37, 380–392 (2014)
2. Baldessari, R., Bödekker, B., Deegener, M., Festag, A., Franz, W., Kellum, C.C., Kosch, T., Kovacs, A., Lenardi, M., Menig, C., et al.: Car-2-Car Communication Consortium - Manifesto (2007)
3. Bochem, A., Leiding, B., Hogrefe, D.: Unchained Identities: Putting a Price on Sybil Nodes in Mobile Ad hoc Networks. In: Security and Privacy in Communication Networks (SecureComm 2018). Singapore (August 2018)
4. Booch, G., Jacobson, I., Rumbaugh, J., et al.: The Unified Modeling Language. Unix Review 14(13), 5 (1996)
5. Calcaterra, C., Kaal, W.A., Vlad, A.: Semada Technical Whitepaper - Blockchain Infrastructure for Measuring Domain Specific Reputation in Autonomous Decentralized and Anonymous Systems. URL: `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3125822` (2018), (Accessed April 18, 2018)
6. Christidis, K., Devetsikiotis, M.: Blockchains and Smart Contracts for the Internet of Things. IEEE Access 4, 2292–2303 (2016)
7. Civic Technologies, Inc.: CIVIC - Whitepaper. URL: `https://tokensale.civic.com/CivicTokenSaleWhitePaper.pdf` (2017), (Accessed May 01, 2018)
8. Copel, N., Ater, T.: DAV - Decentralized Autonomous Vehicles (Whitepaper). URL: `https://dav.network/whitepaper.pdf` (2017), (Accessed April 27, 2018)
9. Cortright, J.: Sprawl Tax: How the US Stacks Up Internationally . URL: `http://cityobservatory.org/sprawl-tax-how-the-us-stacks-up-internationally/` (2016), (Accessed April 26, 2018)
10. Dai, P., Mahi, N., Earls, J., Norta, A.: Smart-Contract Value-Transfer Protocols on a Distributed Mobile Application Platform. URL: `https://qtum.org/uploads/files/a2772efe4dc8ed1100319c6480195fb1.pdf` (2017), (Accessed May 01, 2018)
11. Davis, A.M.: Software Requirements: Objects, Functions, and States. Prentice-Hall, Inc. (1993)
12. Erl, T.: Service-Oriented Architecture: Concepts, Technology, and Design. Pearson Education India (2005)

13. Faezipour, M., Nourani, M., Saeed, A., Addepalli, S.: Progress and Challenges in Intelligent Vehicle Area Networks. Communications of the ACM 55(2), 90–100 (2012)
14. IEEE Computer Society. Software Engineering Technology Committee and Institute of Electrical and Electronics Engineers: IEEE Recommended Practice for Software Requirements Specifications. IEEE Std, Institute of Electrical and Electronics Engineers (1994)
15. L. M. Goodman: Tezos - A Self-Amending Crypto-Ledger (White paper). URL: `https://www.tezos.com/static/papers/white_paper.pdf` (2014), (Accessed April 27, 2018)
16. Lanctot, R.: Accelerating the Future: The Economic Impact of the Emerging Passenger Economy. URL: `https://newsroom.intel.com/newsroom/wp-content/uploads/sites/11/2017/05/passenger-economy.pdf` (2017), (Accessed April 27, 2018)
17. Leiding, B.: The Machine-to-Machine Economy Revolution - A blockchain-based trading platform for the Internet of Things (2018), unpublished
18. Leiding, B., Cap, C.H., Mundt, T., Rashidibajgan, S.: Authcoin: Validation and Authentication in Decentralized Networks. In: The 10th Mediterranean Conference on Information Systems - MCIS 2016. Cyprus, CY (September 2016)
19. Leiding, B., Memarmoshrefi, P., Hogrefe, D.: Self-Managed and Blockchain-Based Vehicular Ad-Hoc Networks. In: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct. pp. 137–140. ACM (2016)
20. MacNeille, P.R., Wisniewski, J., DeCia, N.: Vehicle-to-Vehicle Cooperation to Marshal Traffic (Mar 27 2018), uS Patent 9,928,746
21. Marshall, J.: Agent-Based Modelling of Emotional Goals in Digital Media Design Projects. International Journal of People-Oriented Programming (IJPOP) 3(1), 44–59 (2014)
22. McCorry, P., Shahandashti, S.F., Clarke, D., Hao, F.: Authenticated key exchange over bitcoin. In: International Conference on Research in Security Standardisation. pp. 3–20. Springer (2015)
23. McKenzie, B.: Who Drives to Work? Commuting by Automobile in the United States: 2013. American Community Survey Reports (2015)
24. Mieke Berends-Ballast et al.: Transport and Mobility 2016. URL: `https://www.cbs.nl/en-gb/publication/2016/25/transport-and-mobility-2016` (2016), (Accessed April 26, 2018)
25. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System. URL: `https://bitcoin.org/bitcoin.pdf` (2008), (Accessed May 01, 2018)
26. Nguyen, Q.K.: Blockchain - A Financial Technology for Future Sustainable Development. In: Green Technology and Sustainable Development (GTSD), International Conference on. pp. 51–54. IEEE (2016)
27. Norta, A., Grefen, P., Narendra, N.C.: A Reference Architecture for Managing Dynamic Inter-Organizational Business Processes. Data & Knowledge Engineering 91, 52–89 (2014)
28. Ouaddah, A., Elkalam, A.A., Ouahman, A.A.: Towards a Novel Privacy-Preserving Access Control Model Based on Blockchain Technology in IoT. In: Europe and MENA Cooperation Advances in Information and Communication Technologies, pp. 523–533. Springer (2017)
29. Perrey, R., Lycett, M.: Service-Oriented Architecture. In: Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium on. pp. 116–119. IEEE (2003)

30. Popov, S.: The Tangle - Version 1.4.2. URL: `https://iota.org/IOTA_Whitepaper.pdf` (2018), (Accessed April 22, 2018)
31. Porsche AG: Porsche Introduces Blockchain to Cars. URL: `https://newsroom.porsche.com/en/themes/porsche-digital/porsche-blockchain-panamera-xain-technology-app-bitcoin-ethereum-data-smart-contracts-porsche-innovation-contest-14906.html` (2018), (Accessed April 27, 2018)
32. Rosen, M., Lublinsky, B., Smith, K.T., Balcer, M.J.: Applied SOA: Service-Oriented Architecture and Design Strategies. John Wiley & Sons (2012)
33. SALT Technology, Ltd.: SALT - Blockchain-Backed Loans - Whitepaper. URL: `https://membership.saltlending.com/files/abstract.pdf` (2017), (Accessed April 25, 2018)
34. SelfKey Foundation: SelfKey - Whitepaper. URL: `https://selfkey.org/whitepaper/` (2017), (Accessed April 27, 2018)
35. Specification, O.A.: OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2. Object Management Group (2007)
36. Statistics New Zealand: Car, bus, bike or train: What were the main means of travel to work (2006)
37. Sterling, L., Taveter, K.: The Art of Agent-Oriented Modeling. MIT Press (2009)
38. Wood, G.: Ethereum: A Secure Decentralized Generalised Transaction Ledger. URL: `http://gavwood.com/paper.pdf` (2014), (Accessed May 01, 2018)