The M2X Economy

–

Concepts for Business Interactions, Transactions and Collaborations Among Autonomous Smart Devices

Dissertation
for the award of the degree
“Doctor rerum naturalium” (Dr.rer.nat.)
of the Georg-August-Universität Göttingen

within the Doctoral program Ph.D. Programme in Computer Science (PCS)
of the Georg-August University School of Science (GAUSS)

Submitted by
Benjamin Leiding

from Rostock (place of birth)
Göttingen 2019

Thesis committee

Prof. Dr. Dieter Hogrefe,
Georg-August-Universität Göttingen, Faculty of Mathematics and Computer Science, Institute of Computer Science

Prof. Dr. Clemens H. Cap,
University of Rostock, Faculty of Computer Science and Electrical Engineering, Department of Computer Science

Assoc-Prof. Dr. Alexander Norta,
Tallinn University of Technology, Faculty of Information Technology, Department of Software Science

Members of the examination board

First Reviewer: **Prof. Dr. Dieter Hogrefe**,
Georg-August-Universität Göttingen, Faculty of Mathematics and Computer Science, Institute of Computer Science

Second Reviewer: **Prof. Dr. Clemens H. Cap**,
University of Rostock, Faculty of Computer Science and Electrical Engineering, Department of Computer Science

Other members of the examination board:

Assoc-Prof. Dr. Alexander Norta,
Tallinn University of Technology, Faculty of Information Technology, Department of Software Science

Prof. Dr. Xiaoming Fu,
Georg-August-Universität Göttingen, Faculty of Mathematics and Computer Science, Institute of Computer Science

Prof. Dr.-Ing. Delphine Reinhardt,
Georg-August-Universität Göttingen, Faculty of Mathematics and Computer Science, Institute of Computer Science

Prof. Dr. Carsten Damm,
Georg-August-Universität Göttingen, Faculty of Mathematics and Computer Science, Institute of Computer Science

Date of the oral examination: December 11th, 2019

Declaration of Authorship

I, Benjamin Leiding, declare that this thesis titled "The M2X Economy – Concepts for Business Interactions, Transactions and Collaborations Among Autonomous Smart Devices" and the work presented in it are my own. I confirm that I authored this thesis independently, that I have not used any sources other than the declared sources, and that I have explicitly marked all material which has been quoted either literally or by content from external sources.

Date: ____

Signed: ______________________________________

Abstract

Nowadays, business transactions almost exclusively focus on human-to-human transactions. The persistent growth and expansion of the Internet of Things, the ubiquitousness of so called smart devices, as well as progressing digitalization of our daily life, enables business transactions without human intervention among autonomously acting machine agents; a concept referred to as the Machine-to-Machine (M2M) economy. Besides M2M interactions, machines interact with humans (Machine-to-Human – M2H), or infrastructure components (Machine-to-Infrastructure – M2I). The term Machine-to-Everything (M2X) economy represents a more general view on use cases that involve autonomous smart devices and also encompasses M2M, M2H and M2I scenarios. While the technical concepts of IoT, Smart Homes, Smart Cities and Industry 4.0 that enable the M2X economy have been around for a while now, a widespread adoption as well as applications that use their full potential are still missing. Many isolated applications exist that aim to solve very specific and simplified use cases that fall within the spectrum of the M2X economy. However, an interoperable, integrated, scalable model that facilitates the M2X economy is non-existing. Likewise, concepts for a M2X value transfer and collaborations among machines to achieve shared objectives within this ecosystem are missing as well.

This work focuses on the emerging M2X ecosystem in the context of Information System research and makes three contributions: First, it suggests architectural concepts that encompass a blockchain-based interaction-, transaction- and collaboration model for M2X use cases, a business collaboration lifecycle and governance structure as well as a set of modalities for these use cases derived through an exploratory research approach. Second, it presents a decentralized self-sovereign identity solution in combination with a validation and authentication mechanism that is suitable for the M2X ecosystem. Sybil attacks are a common issue of decentralized networks. Thus we present a mechanism to price the costs of a sybil node attack, thereby providing an easy to use metric for the sybil resistance of a decentralized M2X system. As a step towards a formal validation of these novel infrastructural concepts, a Colored Petri Net model is provided covering the protocol-driven data exchange of the M2X identity solution. The developed identity protocols are validated using CPN models and proof-of-concept implementations, while specific aspects of the presented M2X identity solution are evaluated using historical data to asses its suitability. Finally, the feasibility of the M2X interactions-, transactions- and collaboration model as well as the identity solution is demonstrated.

Acknowledgements

Here, I would like to appreciate and thank all of the people that contributed to the successful completion of this thesis. Most importantly, I want to thank my wife Nami who supported me along the way, not only during the creation process of this thesis, but also over the period of many years. I would like to express my deepest gratitude to Nami for her love, support, encouragement in times of doubt and for keeping me sane over the past few months.

I would like to express my gratitude to all my supervisors who patiently guided me through this thesis. I would like to thank Prof. Dieter Hogrefe for his contributions, patience, guidance and feedback as well as the funding which enabled this dissertation. I am also very grateful to him for providing me with the opportunity to pursue this rather unusual research direction.

Next, I would like to thank Prof. Clemens H. Cap who inspired my scientific career, initiated and fostered many ideas and never hesitated to help or provide critical feedback. It was a true pleasure to create this thesis under his guidance.

I am also deeply indebted to Prof. Alexander Norta – not only for supervising this work, but also for being a great mentor who provided countless research ideas and never hesitated to help or provide feedback no matter whether it was day, or night.

I also thank Prof. Xiaoming Fu, Prof. Delphine Reinhardt, and Prof. Carsten Damm, for serving on the examination board of this dissertation.

I am also very grateful to all the members of the Telematics group and all my colleagues at the University of Göttingen. The same applies to all my previous colleagues at the University of Rostock who fostered my interest in science. Moreover, I would like to thank all my co-authors as well as Bachelor and Master students.

Special thanks goes to Mayutan Arumaithurai, Simon Schuler and the whole Chaindrium team for bearing with me.

Many thanks also to William V. Vorobev and the Chorus Mobility team who helped to test some of the ideas presented in this work in a prototype environment as well as endless discussions on the economy of autonomous vehicles.

Last but not least, I thank my parents and my family for their support, patience and help during all these previous years.

Contents

List of Abbreviations

AI	Artificial Intelligence
AOM	Agent-Oriented Modeling
AU	Application Unit
BNMA	Business-Network Model Agent
BPMN	Business Process Model Notation
CA	Certificate Authority
CPN	Colored Petri Net
CPS	Cyber-Physical System
CPPS	Cyber-Physical Production System
CR	Challenge Record
DAG	Directed Acyclic Graph
DAO	Decentralized Autonomous Organization
dApp	Decentralized Application
DGI	Distributed Governance Infrastructure
DID	Decentralized Identifier
DLT	Distributed Ledger Technology
DoS	Denial of Service
DPKI	Decentralized PKI
DSR	Design Science Research
ECU	Electronic Control Unit

EIR	Entity Identity Record
ERP	Electronic Road Pricing
EVM	Ethereum Virtual Machine
GPIO	General Purpose Input/Output
HMAC	Hash Message Authentication Code
IoT	Internet of Things
IS	Information Systems
ISSRM	Information Systems Security Risk Management
I2V	Infrastructure-to-Vehicle
MANET	Mobile Ad Hoc Network
M2H	Machine-to-Human
M2I	Machine-to-Infrastructure
M2M	Machine-to-Machine
M2X	Machine-to-Everything
OBU	On-Board-Unit
OBDII	On-Board-Diagnostics 2
OEM	Original Equipment Manufacturer
OS	Operating System
OTA	Over-the-Air
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
PoA	Proof-of-Authority
PoS	Proof-of-Stake
PoW	Proof-of-Work
P2P	Peer-to-Peer
ROS	Robotic Operating System

RPA	Robotic Process Automation
RR	Response Record
RSU	Road-Side-Unit
SCC	Strongly Connected Component
SOA	Service-Oriented Architecture
SR	Signature Record
SRP	Security Risk-Oriented Pattern
TaaS	Transportation-as-a-Service
TPD	Tamper-Proof-Device
TR	Traffic Regulation
TX	Transaction
UML	Unified Modeling Language
V&A	Validation and Authentication
VAE	V&A Entry
VANET	Vehicular Ad Hoc Network
VAR	Validation and Authentication Request
VIN	Vehicle Identification Number
V2H	Vehicle-to-Human
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
WAVE	Wireless Access for Vehicular Environments
WoT	Web of Trust
WSN	Wireless Sensor Network

List of Figures

List of Tables

Chapter 1

Introduction

The ubiquitousness of smart devices, the persistent growth and expansion of the Internet of Things (IoT) [1][2] as well as the progressing digitalization of our daily life, e.g., [3] and [4], foster the development of new technical and economical business models. While nowadays, business transactions almost exclusively focus on human-to-human transactions, the IoT enables business transactions without human intervention via autonomously acting machines; a concept that we refer to as the Machine-to-Machine (M2M) economy. Besides M2M interactions, machines interact with humans (Machine-to-Human – M2H) or infrastructure components (Machine-to-Infrastructure – M2I). The Machine-to-Everything (M2X) economy is the result of business interactions, transactions and collaborations among participants of the corresponding ecosystem and represents a more general view on use cases that involve autonomous smart devices while also encompassing M2M, M2H and M2I scenarios. In the M2X economy, smart sensors may offer collected sensor data such as temperature or air contamination to interested buyers that rely on the aforementioned data for their own computations. In the context of autonomous and self-driving vehicles, scenarios such as automated tollbooth payments, autonomous battery charging services as well as general Transportation-as-a-Service (TaaS) applications [5] and business models are among the most discussed use cases. More complex scenarios focus on Smart Homes and Smart Cities [6] as well as Industry 4.0 [7]. Even potential successors of Industry 4.0 with fully automated and autonomous smart factories which independently handle supply and demand management as well as corresponding logistics – including supply-chain management – are part of the M2X economy.

Besides the technical perspective and the corresponding technical challenges, the upcoming M2X economy also poses sociotechnical problems and challenges. In an M2X scenario, we are not only enabling interactions, transactions and collaborations among machines, or between machines and infrastructure components, but also among machines and humans. A main requirement is to enable an integration of humans and smart devices into a well-functioning sociotechnical system that puts the M2X concept in a human-centered context. When considering collaborations, interactions and transactions of autonomous smart devices, even M2M and M2I can be seen in a sociotechnical context that is similar to humans interacting with each other, or humans interacting with machines of the M2X ecosystem. In order to provide non-trivial services or products smart devices are not only required to interact with their potential clients; they may also have to collaborate, interact and transact on-demand with other entities to be able to achieve a shared goal. While providing services or products, they might even migrate to different geographical locations based on supply and demand. The interleaved on-demand collaborations, interactions and transactions among autonomous, heterogeneous and highly dynamic entities (humans, machines, software agents, etc.) lead to a decentralized, distributed and heterogeneous sociotechnical system consisting of a large number of micro-services of different vendors and solution as well as infrastructure providers.

1.1 The M2X Economy

The upcoming M2X economy and the corresponding ecosystem will influence our daily lives in many ways. Even though minor applications and use cases already exist, more complex and impactful applications that provide more than marginal value to society are still missing. Despite technological limitations that are still being researched and developed, further restrictions arise from missing concepts on how smart devices will interact and transact in complex collaboration scenarios among each other, with the infrastructure or with humans. We previously defined the M2X economy as the result of business interactions, transactions and collaborations among participants of the M2X ecosystem, while the M2X ecosystem encompasses all M2X entities and their interlinked relations required to provision its goods and services.

Throughout the subsequent sections we first discuss overlaps and differences between the concepts of M2X, the IoT, cyber-physical systems (CPS) and more. Afterwards, example applications are introduced as well as running cases that illustrate concepts of subsequent chapters.

1.1.1 Overview and State of the Art

Before going into detail with regards to the M2X scenarios, we first clarify overlaps and differences to closely related concepts and applications such as wireless sensor network (WSN), machine-to-machine, cyber-physical system, cyber-physical production systems (CPPS), the Internet of Things, cybernetics and robotic process automation (RPA). For each of these terms and concepts, applications and use cases exist that overlap with those of the M2X economy. Hence, it is necessary to explain the differences, overlaps and correlations among them.

In 1948, Wiener defined the concept of cybernetics as “the scientific study of control and communication in the animal and the machine” [8] which is concerned with providing mathematical means for studying adaptive and autonomous systems while mimicking information communicated in machines with that of the brain and nervous system [9]. According to [10], an alternative definition of cybernetics is provided by the mathematician Kolmogorov who defines it as the “science concerned with the study of systems of any nature which are capable of receiving, storing and processing information so as to use it for control”. Usually, such systems incorporate closed signaling loops where an action by a system causes a change in the systems environment once detected via sensors which – in a circular manner – causes the system to change as well. Actions of a system are performed to advance from the current state to the desired goal state. Due to the broad definitions of cybernetics, many of the following concepts somehow overlap.

WSNs “consist of spatially distributed autonomous sensors to monitor physical or environmental conditions, and to cooperatively pass their data through a variety of networks to a main location. WSNs emphasizing the information perception through all kinds of sensor nodes are the very basic scenario of IoT” [11].

The concept of M2M most commonly refers to the communication capabilities of wireless and wired systems such as computers, embedded processors, sensors, actuators as well as mobile devices to facilitate information exchange [11][12]. “The rationale behind M2M communications is based on two observations: 1) a networked machine is more valuable than an isolated one; and 2) when multiple machines are effectively interconnected, more autonomous and intelligent applications can be generated” [11][13].

The term cyber-physical systems refers to systems with integrated computational, physical and networking capabilities [14][15][16]. Such “CPS are engineered systems that are built from, and depend upon, the seamless integration of computation and physical components. CPS tightly integrate computing devices, actuation and control, networking infrastructure, and sensing of the physical world” [17].

The concept of cyber-physical production systems can also be interpreted as the next evolution of M2M systems. [11] and [18] argue: “Through interfacing with WSNs, M2M systems can collect a wide range of information by all kinds of sensors. Thus, in addition to M2M communications, machines also can make action through the collected information with the integration with WSNs. From a long-term point of view, M2M systems with the capabilities of decision-making and autonomous control can be upgraded to cyber-physical systems”.

Example applications from the field of CPS research may include: Smart electric grids, autonomous automobile systems, medical monitoring, process control systems, robotics systems, automatic pilot avionics, precision agriculture and advanced manufacturing [14][19].

Monostori et al. [20] define CPPS as a collection of “autonomous and cooperative elements and sub-systems that are connected based on the context within and across all levels of production, from processes through machines up to production and logistics networks. Three main characteristics of CPPS are to be underlined here: 1.) Intelligence (smartness), i.e. the elements are able to acquiring information from their surroundings and act autonomously. 2.) Connectedness, i.e. the ability to set up and use connections to the other elements of the system – including human beings – for cooperation and collaboration, and to the knowledge and services available on the Internet. 3.) Responsiveness towards internal and external changes”.

Gubbi et al. [21] defines the Internet of Things as an “interconnection of sensing and actuating devices providing the ability to share information across platforms through a unified framework, developing a common operating picture for enabling innovative applications. This is achieved by seamless large scale sensing, data analytics and information representation using cutting edge ubiquitous sensing and cloud computing”. From an architectural perspective, the IoT consists of four main components: *i.*) Sensing, *ii.*) heterogeneous access, *iii.*) information processing and *iv.*) applications/services.

“Although M2M, WSNs and CPS are quite similar in many networking aspects, there are still some major differences from architecture and design philosophy. Generally, M2M is for supporting communications without or with limited human intervention. WSNs are particularly designed for delivering sensor-related data. CPS typically involves multiple dimensions of sensing data, crosses multiple sensor networks and the Internet, emphasizes control functions, and aims at constructing intelligence across multiple domains. Thus, we propose CPS is an evolution of M2M by the introduction of more intelligent and interactive operations, under the architecture of internet of things” [11].

Classification	Correlations
WSNs, M2M and CPS	All belong to IoT from the architecture perspective.
WSNs	WSNs are the very basic scenario of IoT and the foundation of CPS, and are regarded as the supplement of M2M.
M2M	Is the main pattern of IoT at the present stage.
CPS	Is an evolution of M2M in intelligent information processing, and will be an important technical form of IoT in the future.

TABLE 1.1: Correlation among M2M, WSNs and CPS – Source: [11] and [13]

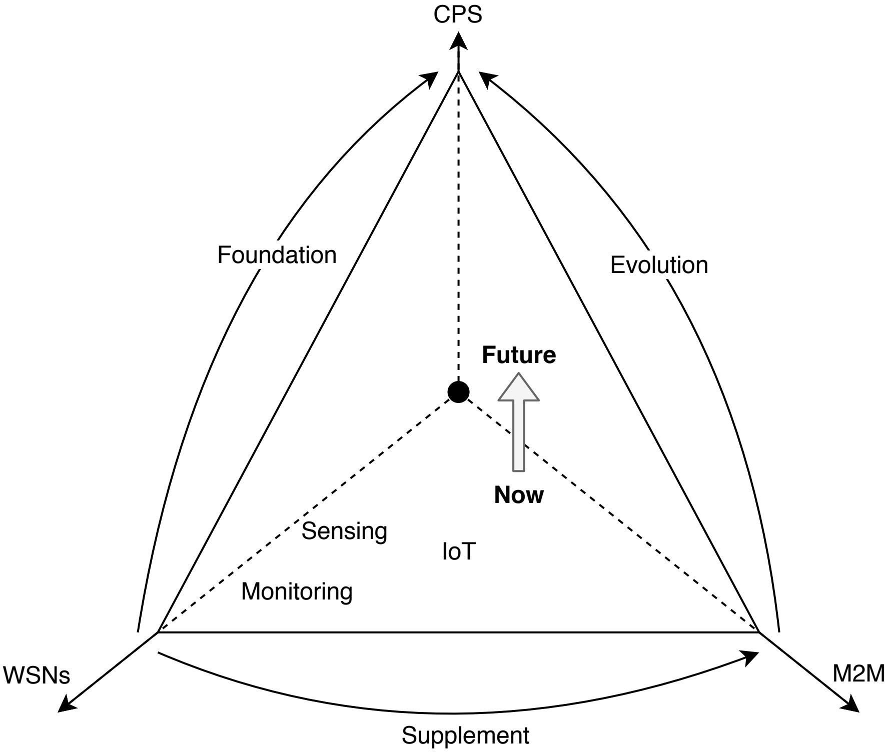


FIGURE 1.1: Correlation among M2M, WSNs, CPS and IoT – Based on [13]

Since WSNs, M2M, CPS (and CPPS as part of CPS) all must have the same components (on a conceptual level – differences in proportion and design may exist) as listed above, they belong to the concept of the IoT [11][22]. Table 1.1 briefly summarizes the described findings above. In addition, Figure 1.1 presents a visual illustration of the correlations between WSNs, M2M, CPS and the IoT where the the space formed by the three axes (dimensions) represents the IoT universe. A further time dimension (*now* ⇒ *future*) describes the progressing developments of WSNs and M2M that promote the dimension of CPS applications [13].

Ivančić et al. [23] define robotic process automation (RPA) "as the application of specific technology and methodologies which is based on software and algorithms aiming to automate repetitive human tasks [24][25][26][27]. It is mostly driven by simple rules and business logic while interacts with multiple information systems through existing graphic user interfaces [28]". Other definitions go even further and specifically include more advanced use cases that utilize artificial intelligence (AI), cognitive computing, process mining, and data analytic in order to perform more complex tasks [23][29][28][30].

Finally, smart devices are equipped with software that governs and controls how the (autonomously acting) machine achieves its objectives. In the context of the concepts described above this software is executed on the machine itself. However, software agents might also reside in a cloud environment, or other locations, e.g., a users' mobile phone. Franklin and Graesser define an autonomous agent as "a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future" [31]. Therefore, the notion of an agent – no matter whether it is situated in a machine or not – is an important part within the notion of the M2X ecosystem.

After clarifying the terms and concepts above, the question remains: Where does the M2X economy fit in? Several publications list and survey CPS and IoT applications (e.g., [19][32][33][34][35]) as well as the economic value and impact of WSN, M2M, CPS, CPPS, IoT and RPA (e.g., [20][35][23]). However, the potential economy resulting from interactions, transactions and collaborations among humans, smart devices, software agents and physical systems within is rarely considered [36][5][37]. **This work focuses on the emerging M2X economy covering interactions, transactions, collaborations and business models for machine-to-human (M2H), machine-to-machine (M2M) and machine-to-infrastructure (M2I) applications.**

1.1.2 Running Case – Vehicle-to-Everything (V2X) Applications

We introduce three example use cases of the M2X economy in order to provide the reader with a better understanding and intuition on the variety as well as the scope of M2X applications. The selected examples are illustrated in Figure 1.2 and belong to a sub-set of the vehicle-focused M2X applications, i.e., V2X (Vehicle-to-Everything), and serve as running cases. These running cases are used throughout this thesis for further illustration purposes. Examples for V2H (Vehicle-to-Human), V2V (Vehicle-to-Vehicle) as well as for V2I (Vehicle-to-Infrastructure) are presented, thereby encompassing a sample of each main subcategory of the M2X ecosystem.

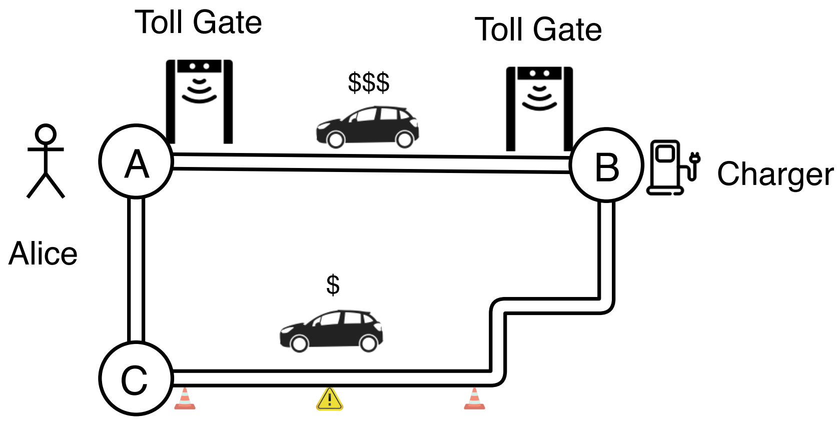


Figure 1.2: V2X Application Running Cases

1.1.2.1 V2H – Transportation-as-a-Service

In the future, people might not possess their own vehicles any more. Instead, vehicles may own themselves, or they are owned by the government or private corporations [38][39][40]. The self-owning vehicles are produced by a manufacturer and pay off their production cost by offering transportation services – Transportation-as-a-Service (TaaS). As illustrated in Figure 1.2, Alice is issuing a transportation request from location A to location B via an application interface. She transmits her location, the travel destination, departure time as well as further constraints such as vehicle size, or different comfort features. Next, Alice negotiates a corresponding transportation contract with available transportation providers (i.e., the autonomous vehicles) and triggers the service enactment after an agreement is reached. In case multiple competing vehicles are available for the same service, common market mechanisms such as auctions determine which vehicle services the transportation contract [5]. In the illustrated example, two vehicles of different price ranges are available. A vehicle that, besides offering more comfort features, is offering to take the direct route from location A to location B via the toll road is available. Alternatively, another cheaper vehicle that offers a route via location C is available too. However, this vehicle is less comfortable and will take longer due to traffic congestion between location C and location B.

Such a TaaS concept provides several advantages: First, a vehicle-sharing economy reduces the number of vehicles required to manage the general transportation of entities. Second, vehicles that own themselves can charge lower prices compared to profit-oriented companies since they only have to cover their own expenses and not make a profit. In case a vehicle cannot find a new transportation task, it may search for an empty parking spot and idle for some time until new job offers are available. Finally, the TaaS concept is not limited to human transportation and applies to the transportation of goods via drones, ships, planes and trains as well.

1.1.2.2 V2V – Road Space Negotiation and Mitigating Traffic Congestion

Traffic congestion is the result of an over-utilization of a scarce resource, i.e. road space. V2V communication and collaboration offer new opportunities to mitigate or reduce traffic congestion. The interconnection of vehicles allows for predictive- and intelligent traffic flow management by using alternative routes, or by optimizing utilization of available road capacities [41][42][43][44].

In case even perfect traffic management cannot provide sufficient throughput, road space may become a tradable resource. Network participants that need to reach a destination urgently have the option to pay other road users to yield in a traffic jam in order to arrive faster at their destination [36][45]. Finally, paid priority lanes [46], or congestion tax systems may be additional options to mitigate traffic congestion [47][48]. Based on these approaches, the cheaper vehicle from Figure 1.2 may reach its destination faster using micro-payments for road space, or to utilize priority lanes that circumvent the traffic congestion. Alternatively, assuming that Alice is not in a hurry, the vehicle could earn some extra money for giving the rights of way to other vehicles that are willing to pay. The additionally earned money can be used to reduce Alice's travel fee.

1.1.2.3 V2I – Automated Payment Services

Toll road payments, paid parking, or related fees are well-established means to fund road- and infrastructure maintenance. Likewise, minor maintenance fees for the technological infrastructure of Vehicular Ad Hoc Networks (VANETs) may apply in the future. Apart from funding road- and infrastructure maintenance, fees also act as incentivization mechanisms for ecosystem participants, e.g., increased parking costs in the city center incentivize the use of public transport as a means of transportation. Despite the progressing digitization, toll road payments still require tedious human interaction. Similar thoughts apply to traffic mitigation incentive systems such as Singapore's Electronic Road Pricing (ERP) system [49] that charges vehicles usage of the road according to the congestion they are causing. In the context of V2I, payment automation uses a predefined model (traveled distance, time, data consumption, etc.), thereby illustrating one of the most common examples of V2I transaction and interaction services. Besides such mandatory fees, Leiding et al. [36] suggest additional infrastructure-enabled optional applications and services, e.g., traffic jam notifications or platooning services.

In the context of our running cases presented in Figure 1.2, automated payment services are used by the more expensive vehicle while traveling on the toll road, and by both vehicles after arriving at the final destination (location B) to recharge their batteries before serving the next customer.

1.2 Research Questions

While the technical concepts of IoT, Smart Homes, Smart Cities, Industry 4.0 and so on that enable the M2X economy have been around for a while now, a widespread adoption as well as applications that use the full potential of these concepts are still missing. Many isolated applications exist that aim to solve very specific and simplified use cases which fall within the spectrum of the M2X economy. However, an interoperable, integrated, scalable model that allows to establish a new economy is non-existing. Moreover, for such an economy a transfer of values resulting from interactions and transactions among smart devices is necessary – the same applies for a model that enables collaborations among machines to achieve a common objective within this ecosystem. Furthermore, proper conflict resolution strategies in case of contract violations have to be put in place – preferably mechanisms that do not require human supervision. In order to unlock the full potential of a M2X ecosystem, interoperability among all involved entities – which are often structured in a decentralized manner – is required. Finally, in the context of autonomously acting smart devices, a variety of requirements such as tamperproof data processing, transparency, accountability and non-repudiation pose challenges.

This thesis focuses on the M2X economy in the context of Information System (IS) research and explores key concepts that are essential to the upcoming M2X economy. As a result of this process, the thesis answers the research question of **how to enable the Digital Transformation of Information Systems in the Context of the Machine-to-Everything (M2X) Economy?** In order to answer this question with a separation of concerns, the main research question is divided into two sub-questions:

1. **RQ-1:** How to enable interactions, transactions and collaboration as well as reliable value transfer among entities of the M2X ecosystem?
2. **RQ-2:** How to identify, authenticate and validate entities in a decentralized M2X ecosystem?

1.3 Research Contributions

This work makes three contributions: First, it suggests architectural concepts that encompass an interaction-, transaction- and collaboration model for M2X use cases and scenarios, a business collaboration lifecycle and governance structure as well as a set of modalities for these use cases derived through an exploratory research approach.

Second, it presents a decentralized self-sovereign identity solution as well as a validation- and authentication mechanism that is suitable for the M2X ecosystem. Sybil attacks are a common issue of large-scale peer-to-peer (P2P) networks, where hostile or faulty computing elements threaten the security of the whole network. Single faulty entities may be able to present multiple identities, thereby controlling a substantial fraction of the system, consequently undermining its functionality and security [50]. Therefore, we present a mechanism to price the costs of a sybil node attack, thereby providing an easy to use metric for the sybil resistance of a decentralized M2X system. As step towards a formal validation of these novel infrastructural concepts, a Colored Petri Net model is provided covering the protocol-driven data exchange of the M2X identity solution.

Finally, the feasibility of the M2X interactions-, transactions- and collaboration model as well as the identity solution is demonstrated. The developed identity protocols are validated using CPN models and proof-of-concept implementations, while specific aspects of the presented M2X identity solution are evaluated using historical data to asses its suitability.

A majority of the contributions of this work are based on peer-reviewed academic publications that are listed in Appendix B.

1.4 Research Methodology

This work focuses on the emerging M2X economy in the context of IS research and contributes a set of architectural concepts for business interactions, transactions and collaborations within the M2X context as described in the previous Section 1.3. Thus, a research methodology that supports the development and evaluation of such conceptual artifacts is required. The design-science research (DSR) paradigm “seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts” [51]. According to Hevner et al. [51], artifacts are defined as constructs, models, methods and utilities. A construct may be a vocabulary, or symbols. Models refer to abstractions and representations. Algorithms and practices constitute the methods. Finally, the utility may be an implementation, or a prototype system. “Broadly speaking, DSR aims to add to knowledge of how things can and should be constructed or arranged (i.e., designed), usually by human agency, to achieve a desired set of goals” [52]. As a result, this work follows design-science research in the domain of information systems as a research methodology.

Figure 1.3 represents the DSR instantiation of the framework as outlined by Hevner et al. within the context of this thesis. As shown on the left of Figure 1.3, artifacts have to address a relevant need of the defined environment. The environment consists of people, organizations, processes and technologies. The existing knowledge base on the

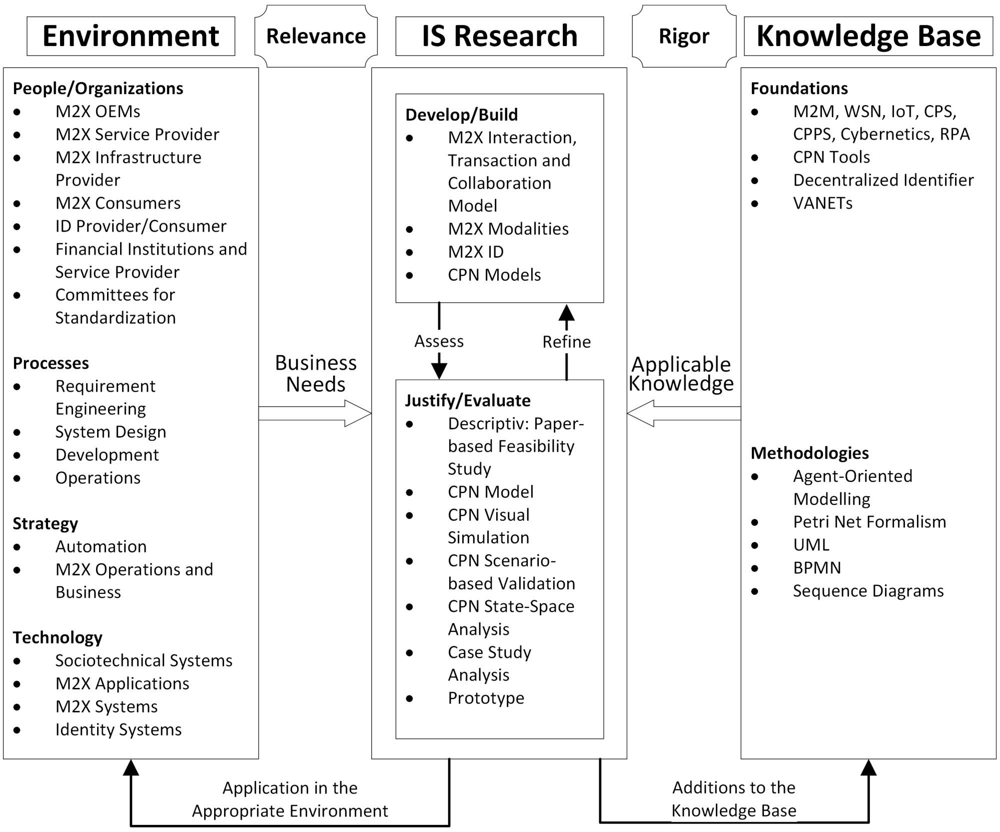


FIGURE 1.3: Design Science Research Framework Instantiation – Based on [51]

right side provides foundations (e.g., existing research and theories) and methodologies (e.g., evaluation guidelines or formalisms). The produced artifact contributes to the knowledge base and is applied to the environment thereby addressing the initial need. In addition to the DSR framework depicted in Figure 1.3, Hevner et al. further propose guidelines for the DSR process. The guidelines consist of seven steps and are illustrated in Table 1.2. The subsequent sections detail the application of the guidelines within this work and in relation to the presented DSR framework instantiation.

1.4.1 Design as an Artifact

The artifacts created in this work aim to close the existing gap in the knowledge base as described previously. The goal is to:

1. Enable interactions, transactions and collaboration as well as reliable value transfer among entities of the M2X ecosystem.
2. Identify, authenticate and validate entities in a decentralized M2X ecosystem.

The produced artifacts are the M2X model for interactions, transaction and collaborations and the corresponding M2X modalities as presented in Chapter 3. An identity

Guideline	**Description**
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contribution	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Table 1.2: Design Science Research – Guidelines (Source: [51])

solution for the M2X ecosystem is described in Chapter 4. Moreover, a set of CPN models covering the protocol-driven data exchange of the M2X identity solution is developed.

1.4.2 Problem Relevance

The importance and need for solutions that foster the upcoming M2X economy is motivated and explained in the introduction of this chapter. The growing number of connected (autonomous) smart devices enables a new economy among machines themselves and also among machines and humans, i.e., sociotechnical systems [53]. As illustrated in the environment pillar of Figure 1.3, concepts and strategies to structure this economy as well as the corresponding identity solutions are crucial components in this regards.

The results of this research work are relevant for OEMs (Original Equipment Manufacturers), service providers, infrastructure provider and consumers related to the M2X ecosystem. The same applies to financial institutions and financial service providers (e.g., for payment processing), committees for standardization as well as academic-, research-

and industrial organizations. The produced artifacts support these entities throughout the requirement elicitation process, the system design process, the implementation and development work and also while operating M2X services and systems.

1.4.3 Design Evaluation

As illustrated in Figure 1.3, the knowledge base contains methodologies that provide guidance for the evaluation and justification of new artifacts. Based on Hevner et al., Table 1.3 summarizes the available evaluation methods. Evaluation methods must rigorously demonstrate the utility, quality, and efficacy of design artifacts [54]. The evaluation of the created artifacts of this work is performed using analytical and descriptive evaluation methods. The resulting system designs and architectures are evaluated based on an architecture analysis to study the fit of the artifacts into technical IS architecture. Especially for innovative artifacts – for which other forms of evaluation may not be feasible – Hevner et al. propose descriptive methods of evaluation.

Evaluation Method	**Description**
1. Observational	Case Study: Study artifacts in depth in business environment
	Field Study: Monitor use of artifact in multiple projects
2. Analytical	Static Analysis: Examine structure of artifact for static qualities (e.g., complexity)
	Architecture Analysis: Study fit of artifact into IS architecture
	Optimization: Demonstrate inherent optimal properties of artifact or provide optimality bounds on artifact behavior
	Dynamic Analysis: Study artifact in use for dynamic qualities (e.g., performance)
3. Experimental	Controlled Experiment: Study artifact in controlled environment for qualities (e.g., usability)
	Simulation: Execute artifact with artificial data
4. Testing	Functional (Black Box) Testing: Execute artifact interfaces to discover failures and identify defects
	Structural (White Box) Testing: Perform coverage testing of some metric (e.g., execution paths) in the artifact implementation
5. Descriptive	Informed Argument: Use information from the knowledge base (e.g., relevant research) to build a convincing argument for the artifact's utility
	Scenarios: Construct detailed scenarios around the artifact to demonstrate its utility

Table 1.3: Design Science Research – Evaluation Methods (Source: [51])

Chapter 3 relies on descriptive methods due to the fact that a real-world proof-of-concept, or even a prototype implementation are out of scope of this work while other evaluation methods such as a simulation, or quantitative evaluation methods are not applicable. Several artifacts resulting from this research can neither be analyzed using observational, experimental, testing, nor even analytical evaluation methods since a real-world implementation is not feasible (yet). Thus, for the M2X platform a paper-based feasibility evaluation is performed, which considers existing and planned solutions that allow for a simplified and minimal proof-of-concept implementation of the M2X platform based on existing technologies and on-going research. The proposed technology stack represents a tentative proposal based on available solutions.

In contrast to the previous chapter, Chapter 4 relies on the Petri Net formalism [55] – more specifically Colored Petri Nets (CPNs) [56] and implementations of proof-of-concepts for evaluation purposes. CPN is a graphical language for the design, specification, simulation as well as the verification of systems and describes the states of a modeled system and the events (transitions) that cause the system to change states. Moreover, Chapter 4 analyses simulation results by CPN Tools that include visualization, scenario-based validation and state-space analysis results. Finally, specific aspects of the presented M2X identity solution are evaluated using historical data to assess the suitability of the proposed system based on past events.

1.4.4 Research Rigor

In the context of DSR, “rigor is derived from the effective use of the knowledge base” (theoretical foundations and research methodologies) and “must be assessed with respect to the applicability and generalizability of the artifact” [51]. To do so, we select appropriate techniques from the knowledge base to construct and evaluate artifacts that answer the posed RQs. The produced artifacts extend existing concepts such as M2M, WSN, IoT, CPS, Cybernetics and RPA to the notion of a M2X ecosystem (see Section 1.1). The concept of VANETs is required for this works’ running case. The rigor of the CPN models is ensured by CPN Tools[1] [57] and the CPN modeling language [56]. Furthermore, the concepts of Agent-Oriented Modeling [58] (AOM) is utilized during the process of requirement elicitation. Business Process Model and Notation (BPMN) [59], UML and sequence diagram representations [60][61] are used throughout the design phase and to illustrate key functionalities and architectures.

[1] http://cpntools.org/

1.4.5 Design as a Search Process

DSR is an inherently iterative process with the goal to establish the best, or an optimal solution. However, due to the complexity of IS research problems, researchers often simplify problems “by explicitly representing only a subset of the relevant means, ends, and laws, or by decomposing a problem into simpler subproblems” [51]. In this work, the goal of enabling the M2X economy for information systems is decomposed into subproblems and also simplified. Since this work is describing a solution for an ecosystem that does not exist yet due to limitations in technical advances, simplifications and assumptions are made, e.g., assuming certain functionalities of autonomous self-driving cars that are not yet ready but essential for the widespread adoption of this technology and hence, the subject of on-going research. In addition, as a result of the complexity and size of the overall problem, we focus on the two selected challenges of the M2X economy that we consider most relevant: First, the transaction-, interaction- and collaboration model. Second, an identity, validation and authentication solution for decentralized networks of M2X entities.

1.4.6 Communication of Research

Most of the underlying research of this thesis is already published in an academic environment, either presented at conferences, or as articles in scientific journals (see Appendix B). The results of this thesis itself will be made available to the public as well.

1.4.7 Demarcations

As mentioned in previous sections, this work focuses on two main challenges that have to be solved in order to enable the M2X economy. First, presenting an interaction, transaction and collaboration model for the M2X economy. Second, a decentralized identity solution for the same ecosystem. Topics such as privacy and security – which are crucial for information systems – are either excluded from the scope of this work, or only briefly discussed. Furthermore, instead of tackling technical details and presenting specific implementations for sub-functionalities, we introduce an overall concept for such an ecosystem – running cases are utilized solely for illustration and evaluation purposes. While we discuss some business models and payment solutions, a broader economical analysis and discussion of the M2X economy is not part of this work. The same applies to sociological and legal implications of this new economy.

1.5 Thesis Structure

This thesis is structured as follows: Chapter 2 introduces fundamental concepts and technologies that are used throughout this work. Next, Chapter 3 deals with M2X use cases and scenarios to deduce requirements and modalities for an interaction-, transaction- and collaboration platform. Afterwards, the focus shifts to the interactions and transactions among entities, followed by governance mechanisms for the ecosystem such as conflict resolution management. Identities are an essential part of each economy that rely on collaboration and value exchange – hence, Chapter 4 deals with the issues and challenges of existing validation- and authentication mechanisms in decentralized networks. Afterwards, we present a protocol workflow for validation and authentication that suits the M2X ecosystem and discuss the security guarantees and implications of binding an identity to a distributed ledger system. Subsequently, Chapter 5 covers the evaluation of this work as well as a critical discussion of the findings and comparison with related work. Finally, Chapter 6 concludes the thesis and provides an outlook on future work.

Chapter 2

Technological Foundations

The following chapter introduces the technological foundations that are used throughout this work. The specific motivation to select any of the concepts and technologies are described later in Chapter 3 and Chapter 4 when they are applied in a specific context. First, Section 2.1 provides a detailed overview on distributed ledger technology (DLT). Afterwards, Section 2.2 covers the basics of vehicular ad hoc networks (VANETs). Finally, Section 2.3 presents an overview on decentralized identifiers (DIDs) which provide a self-sovereign identity mechanism that is controlled by the owning entity instead of an external party, or a central authority.

2.1 Distributed Ledger Technology – DLT

Over the last decade, distributed ledgers majored and spread in popularity – most noticeably by providing the foundation of the cryptocurrency Bitcoin [62]. Inspired by the Bitcoin system, several further DLT platforms emerged, e.g., Ethereum[1], Hyperledger[2], Corda[3], or Tezos[4]. Moreover, a variety of applications for blockchains have been proposed, e.g., as a platform for IoT applications [63][64], applications in the automotive sector [36][5], in the finance sector [65][66], as part of supply chains [67], or in security- and authentication protocols [64][68][69].

Even though DLT and blockchain are often used synonymously, a technical difference exists. Both, DLT and blockchain, store data in a distributed and highly replicated manner across several nodes for that each maintains a copy of the complete dataset

[1] https://ethereum.org/

[2] https://www.hyperledger.org/

[3] http://www.corda.net/

[4] https://tezos.com/

while updating information through a P2P network without requiring a central authority. However, the specific nature of blockchains is that data – more specifically transactions – are grouped in forms of blocks, while each new block is added to the existing chain of blocks in an append-only structure that is cryptographically secured using hashes and signatures.

2.1.1 Blocks and Transactions

Figure 2.1 illustrates the classical structure of a blockchain block as used by, e.g., the Bitcoin [62], or Ethereum blockchain [70]. As the name suggests, a blockchain consists of a sequentially ordered number of blocks that records transaction events (denoted as TX), e.g., transfer of a cryptocurrency from person A to person B. In addition, each block contains the hash of the previous ancestor block, thereby chaining all blocks together. Changing a transaction of a block, results in a hash mismatch of the succeeding block. As a result, tampering with one block requires the recalculation of all succeeding blocks. Section 2.1.2 provides more details regarding this issue. The exact number of transactions in a block depends on the maximum block size for the specific blockchain and varies from platform to platform [62][71][72][73].

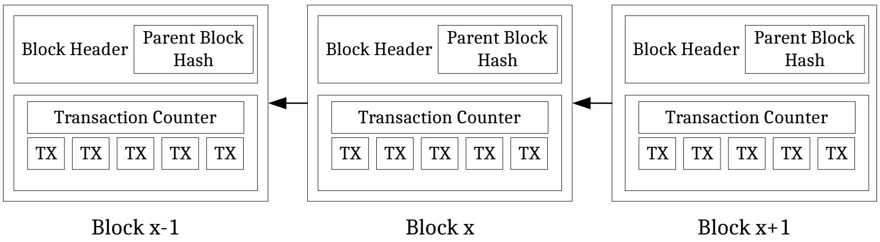


FIGURE 2.1: General Blockchain Structure (Based on [62] and [74])

Each user owns a key pair consisting of a public- and a private key. Owning an asset (e.g., Bitcoin, or any other tokenized asset) is equivalent to owning the key pair that corresponds to the signature of the tokenized asset. The subsequent Figure 2.2 further details the structure of a typical transaction. In the context of Bitcoin, a blockchain asset is transferred "by digitally signing a hash of the previous transaction and the public key of the next owner" [62].

A new transaction is broadcast to the participating nodes of the network to be included in the next block. Still, in order to reach a global consensus on which transaction to include in the next block as well as the order of the transactions a variety of so-called consensus algorithms emerged. The following Section 2.1.2 focuses on the fundamentals of such consensus algorithms.

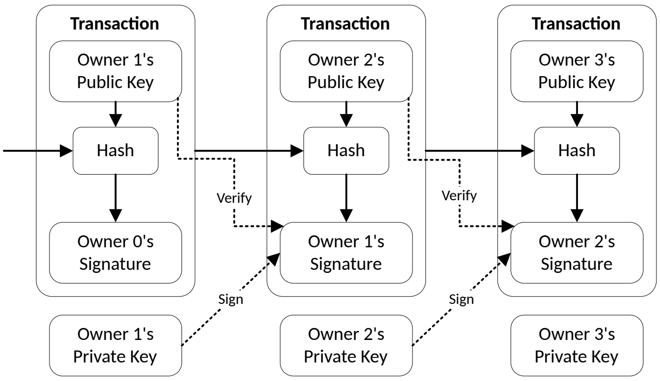


FIGURE 2.2: Blockchain Example Transaction – Based on [62]

2.1.2 Consensus

Participation in a DLT system might be open to all interested entities (public ledger), or restricted to a specific subset of entities (permissioned ledger). Most cryptocurrencies are an example for the first option, whereas a supply-chain use case with a supplier, a transport provider and a receiving party may represent the later option. In public as well as permissioned blockchains, a consensus among all participants is required to form the next block and append it to the existing chain. Reaching consensus among distributed nodes that do not necessarily trust each other (or might even act maliciously) without a central authority represents an instantiation of the Byzantine Generals Problem as introduced by Lamport et al. [75]. For example, a malicious node may attempt to reverse, or redirect a valid cryptocurrency payment to its own benefit by manipulating a past transaction, or the receiving address of a pending transaction.

Proof-of-Work (PoW) [62], Proof-of-Stake (PoS) [76] and Proof-of-Authority (PoA) [77][78] are among the most commonly used consensus algorithms for DLT systems. For each of them, different flavors exists as well as a large variety of alternative protocols. Having said that, so far all consensus algorithms require trade-offs, or suffer from disadvantage, e.g., scalability issues, security issues, efficiency issues, etc. [79][80]. Hence, consensus algorithms within the context of DLT systems remain a topic of on-going research. The following section briefly introduces the PoW, PoS and PoA consensus algorithms.

2.1.2.1 Proof-of-Work

The Proof-of-Work (PoW) consensus algorithm is most commonly used among blockchain platforms, e.g., by Bitcoin [62] and Ethereum [70]. The concept of PoW was initially proposed by Hashcash [81] as a measurement to prevent spam emails by requiring each email to have a small piece of data attached – a *proof of work*. On one hand, the proof should be costly and time-consuming to produce, but on the other hand easy to verify.

Bitcoin uses the Hashcash PoW approach to reach consensus on the next valid block [62][81]. After broadcasting a new transaction to the network, validator nodes – also referred to as miners – pick up these transactions and group them into a new block. A PoW is attached to the block. The PoW resembles a hash-based puzzle. As described in Figure 2.1, each block contains (simplified) a set of transactions and the hash of the previous block. For the PoW, a further random seed value is added. Afterwards, the block is hashed using a pre-defined hash algorithm, e.g., SHA256, SHA-3, or Scrypt. The resulting hash is required to match a specified structure to be accepted as a valid block. In the context of Bitcoin, the resulting hash has to start with a number of zero bits. "The average work required is exponential in the number of zero bits required and can be verified by executing a single hash" [62]. The varying number of zero bits is used to adjust the difficulty of finding a valid block depending on the computing power provided by users searching for a valid hash. By iterating the random seed value, miners eventually find a valid block that is broadcast to the network. Changing a transaction of an already existing block changes the hash of the block and thereby breaks the chain of hashes. In order to successfully tamper with the blockchain, a potential attacker has to recalculate all successor blocks and the corresponding proofs-of-work at a higher speed than the remaining network. By design, creating a PoW requires a certain amount of computation power and due to the popularity of the Bitcoin cryptocurrency, the energy- and hardware resources consumed by Bitcoin-related mining activities became a problem [82][83][84]. Hence, alternative consensus algorithms emerged.

2.1.2.2 Proof-of-Stake

Proof-of-Stake (PoS) [76][85] is an alternative consensus algorithm to the commonly used PoW algorithm. In contrast to PoW, the creator of the next block in PoS is chosen using a selection algorithm that requires a stake to be deposited. The algorithm selects the validator of the next block with a probability proportional to the stake in comparison to the overall stake of all participants. The larger the stake, the higher the probability to become the next block creator. Including conflicting transactions into a new block, or other attacks (which are detected by the network participants) are punished by losing the stake [86]. The absence of a computationally expensive PoW calculation increases the overall transaction rate of the blockchain and lowering the required energy- and hardware resources significantly. The popular blockchain platform Ethereum is planning to switch from its PoW consensus algorithm Ethash [70] to a PoS-type of algorithm called Casper [87]. However, proper staking, selection and punishment mechanisms require additional complexity.

2.1.2.3 Proof-of-Authority

While PoW and PoS are predominantly used and well suited for public ledgers (permissionless), other consensus algorithms focus on permissioned networks where all consensus participants are known and reputable (not necessarily trusted), e.g., in enterprise business applications. Essentially, Proof-of-Authority (PoA) [77][78] is based on a modified PoS algorithm. Instead of providing a monetary stake, validators put their identity and reputation at stake. In permissioned systems the approved authorities might be business partners involved in a shared collaboration process. Alternatively, the POA Network[5] requires validators of the core network to obtain an active U.S. public notary license [88]. For a new block to become valid and attached to the chain, a pre-defined number (often a majority) of elected authorities have to cryptographically sign the new block. Besides the POA Network, the Ethereum test networks Rinkeby, Kovan [89] and Parity [78] as well as enterprise chains such as Hyperledger [90] offer PoA-like consensus algorithms.

2.1.3 Smart Contracts

The concept of smart contracts (SCs) is generally perceived to be closely connected with the emerging popularity of DLTs. However, the idea of SCs was already introduced in 1994 by Nick Szabo who defined a smart contract as “a computerized transaction protocol that executes the terms of a contract” [91] in a self-enforcing manner, thereby minimizing the need for trusted intermediaries among transacting entities [92].

While Bitcoin is only capable of executing a very limited set of scripting commands [62], most modern blockchain platforms with smart contract capabilities support Turing-complete programming languages for smart contract development, e.g., Ethereum [70], or Tezos [93]. In the context of blockchain technology, smart contracts are deterministic code segments, or scripts that reside on the blockchain. Each smart contract has a unique address as an identifier. Via this address, the smart contract can be triggered using an incoming transaction that results in the execution of the smart contract. Since the network reaches consensus on the transaction via the consensus algorithm (as described earlier in Section 2.1.2), the smart contract is executed independently and automatically by all nodes of the network. Hence, smart contracts enable a large variety of general-purpose computation, even though some practical limitations exist, e.g., storing data on a blockchain is often expensive, hence computations that deal with large amounts of data are unusual [94].

Nonetheless, the concept of smart contracts is very well suited for enabling interactions, transactions and collaborations among network entities that do not trust each other. A

[5] https://poa.network/

smart contract that is: *i.*) agreed upon by all involved parties (and cannot be changed by a single party), *ii.*) is transparently available on the blockchain, and *iii.*) provides accountable verifiable execution of processes based on digital signatures. “The possibility of a dispute is eliminated (when all possible outcomes are accounted for) since the participants cannot disagree over the final outcome of this verifiable process they engaged in” [63].

2.1.4 Blockchain Interoperability

At the time of writing this thesis, more than 2000 cryptocurrencies [95] and a large variety and quantity of blockchain platforms [96][97] had emerged since the first introduction of the Bitcoin whitepaper in 2008. Even though many cryptocurrencies share the same underlying platform, e.g., Ethereum, the platforms themselves are rarely interoperable among each other and do not allow for cross-platform transactions, or yet alone value transfer. As a result, not only similar blockchain applications and cryptocurrencies compete among each other, but also all platforms aim for market domination thereby fragmenting the ecosystem in so many isolate parts that none of them reaches significant market adoption.

Therefore, the blockchain ecosystem is in a similar position as the Internet in the 1980s. Hardjono et al. [98] propose to adopt the Internet architecture and its fundamental goals as the blueprint for interoperable blockchains. They define the concept of an *interoperable blockchain* as follows: “An interoperable blockchain architecture is a composition of distinguishable blockchain systems, each representing a distributed data ledger, where transaction execution may span multiple blockchain systems, and where data recorded in one blockchain is reachable and verifiable by another possibly foreign transaction in a semantically compatible manner” [98]. Moreover, Hardjono et al. propose a set of design principles for interoperable blockchains and demonstrate how the MIT project Tradecoin [99] is designed using this interoperability model.

An alternative approach with the same goal is suggested in [100] where the author proposes the concept of relay-chains that act as hubs between different chains. The relay-chain itself is a distinct blockchain that relays messages between chains, but may also tracks the state of connected chains, thereby essentially acting as an interoperability layer. However, a downside of the relay-chain concept is that in the future we might end up in a similarly fragmented ecosystem with many relay-chain hubs that all try to become the most popular platform while still being not interoperable among each other.

The research on blockchain interoperability is still in its infancy. Besides the approaches introduced above, further proposals exist, e.g., [101][102] and [103], but a blockchain interoperability standard comparable to the standards of the Internet is still missing.

2.2 Vehicular Ad Hoc Networks – VANETs

The progressing digitization of current and forthcoming generations of vehicles results in a demand to manage the communication between vehicles, road infrastructure and Internet-based services. Vehicular ad hoc networks (VANETs) are an abstract concept that models the different components that are required for this kind of communication [104][105][106].

The subsequent Section 2.2.1 introduces the network structures and components of VANETs. Afterwards, Section 2.2.2 focuses on the different types of communication among VANET entities. Finally, Section 2.2.3 categorizes VANET application and lists corresponding examples.

2.2.1 Network Structure and Components

Figure 2.3 illustrates the key building blocks of VANETs: Vehicles – equipped with on-board-units (OBUs) and application-units (AUs) –, road-side-units (RSUs) as well as tamper-proof-devices (TPDs). The term *vehicle* is usually used synonymously with *cars* but may also refer to drones, or ships and other devices with transportation capabilities [107][108].

OBUs are typically mounted onto a vehicle and enable data exchange with other OBUs or RSUs, usually via short-range wireless or radio communication, depending on the use case [104][105]. "The main functions of the OBU are wireless radio access, ad hoc and geographical routing, network congestion control, reliable message transfer, data security and IP mobility" [104].

AUs are typically closely linked to the OBU and might even reside in the same physical device unit. Alternatively, AUs might reside in a separate mobile device that is regularly removed from the vehicle (e.g smartphones). The vehicles AU offers an execution environment for applications that utilize and rely on the OBU's communication capabilities [104][105]. The different types of applications are detailed later in Section 2.2.3.

RSUs are placed "along the road side or in dedicated locations such as at junctions or near parking spaces. The RSU is equipped with one network device for a dedicated

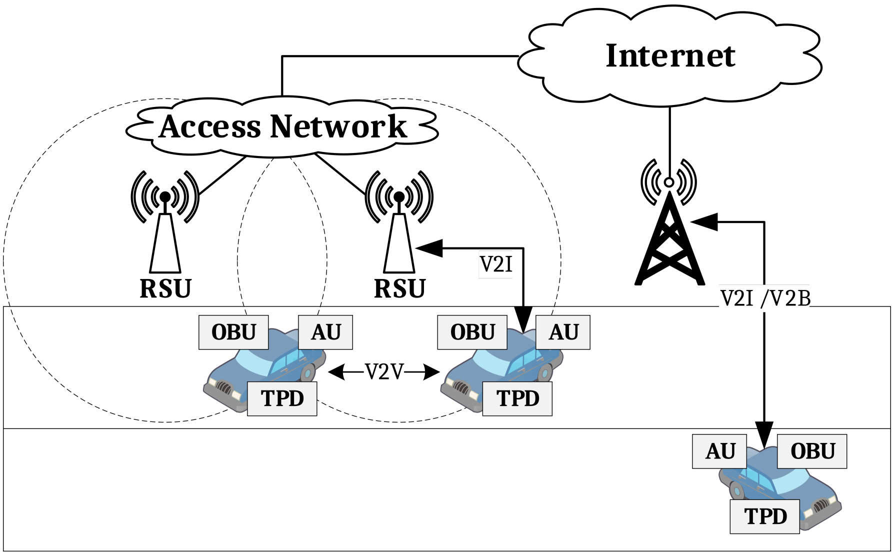


FIGURE 2.3: VANET Model Overview – Based on [36] and [105]

short-range communication based on IEEE 802.11p radio technology, and can also be equipped with other network devices so as to be used for the purpose of communication within the infrastructural network" [104]. On one hand, RSUs are used to forward OBU information to other RSUs or OBUs. On the other hand, they further provide Internet access to OBUs and may also host safety applications, e.g., relaying traffic jam and accident warnings via infrastructure-to-vehicle communication (I2V) [105]. The optimal distribution and deployment of RSUs is crucial for VANETs and discussed for example in [109].

Vehicles are often assumed to be equipped with a TPD that protects sensitive information such as secret key pairs for message signing or the vehicles' identity. As later detailed in Chapter 3, the TPD may also be in charge of ensuring a secure execution and boot environment for firmware components to avoid software manipulations. As the name suggests, TPDs are constructed in such a way that they detect hardware manipulation or unauthorized access which triggers a routine to erase all stored information [110]. Access to the TPD "should be restricted to authorized people. For example, cryptographic keys can be renewed at the periodic technical checkup of the vehicle" [110].

2.2.2 Communication

As illustrated in Figure 2.3, communication within VANETs can be divided into three main categories: *i*) intra-vehicle communication, *ii*) vehicle-to-vehicle communication (V2V) and *iii*) vehicle-to-infrastructure communication (V2I) [104][111]. Intra-vehicle communication concern the data exchange between OBU and AUs. In order to communicate among different vehicles (V2V), an ad hoc network is established for decentralized and distributed communication purposes without relying on a fixed infrastructure [112][113]. V2I communication allows vehicles to communicate with static roadside infrastructure, i.e., RSUs. Finally, vehicles may utilize a hybrid approach of V2V and V2I "in order to increase the range of communication by sending, receiving and forwarding data from one node to another or to benefit from the ability of the RSU to process special applications forming vehicle to infrastructure communication (V2I)" [104].

2.2.3 Applications

VANETs enable a large number of applications with varying utilities and goals. Most commonly, VANET applications are categorized as either *safety*-applications or *comfort/entertainment*-applications [104][113][114]. Safety applications aim to improve road safety, avoiding accidents and ensuring a clean environment – e.g, intersection collision warning, pedestrian crossing information, traffic jam notifications, enforcement of traffic regulations, and so on. On the other hand, comfort- and entertainment applications aim to improve the driver's and passenger's comfort levels as well as enhancing traffic efficiency – e.g., road congestion management, location of available parking, or route navigation [36][104][113][114].

2.3 Decentralized Identifiers – DIDs

The concept of decentralized identifiers (DIDs) has been proposed and is also currently under development by the W3C [115]. DIDs provide an identity that is controlled by the owning entity while at the same time being "independent from any centralized registry, identity provider, or certificate authority" [115]. The design goals of this new type of digital identifier are outlined in Table 2.1 and comprise decentralization, self-sovereignty, privacy, security, a proof-based authentication and authorization mechanism, discoverability, interoperability, portability, simplicity and extensibility.

A DID (*did:example:123456789abcdefghi*) consists of three parts. First, the so-called scheme (*did*), second the method (*example*) and last the method-specific identifier (*123456789abcdefghi*). The scheme part simply explains that we are handling a DID. A DID method specification defines how to create, read, update and delete a DID and its DID document. The last section of the example details the actual unique identifier. The W3C DID specifications propose a distributed network such as blockchains for

Goal	Description
Decentralization	DID architecture should eliminate the requirement for centralized authorities, or single points of failure in identifier management, including the registration of globally unique identifiers, public verification keys, service endpoints, and other metadata.
Self-Sovereignty	DID architecture should give entities, both human and non-human, the power to directly own and control their digital identifiers without the need to rely on external authorities.
Privacy	DID architecture should enable entities to control the privacy of their information, including minimal, selective, and progressive disclosure of attributes or other data.
Security	DID architecture should enable sufficient security for relying parties to depend on DID Documents for their required level of assurance.
Proof-based	DID architecture should enable the DID subject to provide cryptographic proof of authentication and proof of authorization rights.
Discoverability	DID architecture should make it possible for entities to discover DIDs for other entities to learn more about or interact with those entities.
Interoperability	DID architecture should use interoperable standards so DID infrastructure can make use of existing tools and software libraries designed for interoperability.
Portability	DID architecture should be system and network-independent and enable entities to use their digital identifiers with any system that supports DIDs and DID Methods.
Simplicity	To meet these design goals, DID architecture should be (to paraphrase Albert Einstein) ”as simple as possible but no simpler”.
Extensibility	When possible, DID architecture should enable extensibility provided it does not greatly hinder interoperability, portability, or simplicity.

Table 2.1: Design Goals and Principles of the DID Architecture – Source: [115]

this purpose. Table 2.2 provides an overview on a selection of DID methods and their underlying blockchains (and/or identity solution based on that blockchain).

2.3.1 DID Documents

A DID itself is essentially an URL that corresponds to an entity and resolves to a so-called DID Document. DID documents are represented by JSON-LD documents and describe how to use the corresponding DID. As illustrated in Listing 2.1, DIDs consist of a reference that links them to the corresponding DID, public keys that can be used

Method Name	**Blockchain**
did:btcr:	Bitcoin
did:sov:	Sovrin
did:uport:	Ethereum (uPort)
did:selfkey:	Ethereum (SelfKey)

Table 2.2: Example DID Methods – Source: [116]

for verification purposes, authentication methods to authenticate a DID or the owning entity and service endpoints [115]. "Service endpoints enable trusted interactions with the DID controller" [115].

```
{
   "@context": "https://w3id.org/did/v1",
   "id": "did:example:123456789abcdefghi",
   "publicKey": [{
      "id": "did:example:123456789abcdefghi#keys-1",
      "type": "RsaVerificationKey2018",
      "controller": "did:example:123456789abcdefghi",
      "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n"
   }],
   "authentication": [{
      // this key can be used to authenticate as did:...fghi
      "id": "did:example:123456789abcdefghi#keys-1",
      "type": "RsaVerificationKey2018",
      "controller": "did:example:123456789abcdefghi",
      "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n"
   }],
   "service": [{
      "id": "did:example:123456789abcdefghi#service123",
      "type": "ExampleService",
      "serviceEndpoint": "https://example.com/endpoint/8377464"
   }]
}
```

Listing 2.1: DID Document Example – Source [115]

As illustrated in Listing 2.1, DID documents may contain an *authentication* property, a mechanism "by which a DID subject can cryptographically prove that they are associated with a DID" [115]. The *authentication* property provides a list of various verification methods, e.g., public keys. Proving control over a DID document is exerted by resolving the DID to a DID document according to its DID method specification. Proving control over the public key specified in a DID document is achieved via a signature-based challenge-response mechanism using the private key corresponding to the public key.

2.3.2 Verifiable Claims and Credentials

DIDs and DID documents itself are not sufficient to establish a decentralized identity – a concept to describe any arbitrary entity is missing. Verifiable claims overcome this issue and enable an entity to collect selective claims pertaining itself which are then linked to the DID and the DID document, thereby constituting an identity. Claims are represented in JSON format, issued by an issuer and can be cryptographically verified [115][117]. In the context of humans, such a claim might be: Alice is older than 21. Hence, she is allowed to enter any bar and consume alcohol. For the specific purpose of acquiring or consuming alcohol the exact age is not relevant as long as it is above a certain threshold, e.g., 21 years in the USA. The bartender simply verifies the claim presented by Alice and that it is linked to the DID controlled by Alice. Note that the issuer might need to be a party trusted by the bartender, e.g., a governmental institution and not Alice herself.

A subset of verifiable claims are verifiable credentials as specified in [117]. Verifiable credentials are used to structure claims around common credentials from our daily lives such as academic degrees or licenses. Such credentials provide “a mechanism to express these sorts of credentials on the Web in a way that is cryptographically secure, privacy respecting, and machine-verifiable” [117].

In the context of the M2X ecosystem, a DID-based identity solution in combination with verifiable claims can be used in many ways. First and foremost, to establish identifiers and corresponding identities for all entities of the M2X ecosystem, e.g., a DID/claim combination for a car’s vehicle identification number (VIN), or an identifier for a human using or offering services. DIDs also enable us to create digital twins (digital representations) of analogue objects or entities for digital business collaborations, interactions and transactions. Moreover, the DID concept also allows for a secure and privacy-preserving data sharing concept since each entity is not limited to one identity (DID) and instead can hold thousands of DIDs, DID documents and claims to be useful in different scenarios and to share with different entities.

Chapter 3

Collaboration, Value Exchange and Governance of the M2X Ecosystem

Nowadays, business collaborations and transactions occur almost exclusively among humans. However, the M2X economy requires business transactions and collaboration beyond this specifically human-focused approach. Papazoglou and Kratz [118] define a business transaction "as a trading interaction between possibly multiple parties that strives to accomplish an explicitly shared business objective, which extends over a possibly long period of time and which is terminated successfully only upon recognition of the agreed conclusions between the interacting parties. A business transaction is driven by well-defined business tasks and events that directly or indirectly contribute to generating economic value". Traditionally, business transactions and further collaboration platforms are governed by contracts either in the form of an oral or written agreement that is enforceable by law in which all involved parties voluntarily engage. Such contracts – usually represented in the form of documents [119] – uniquely identify the participating parties, specify and define provisioned services or goods, monetary compensations, eventual penalties, as well as further constraints and requirements that vary depending on the context. Norta [120] argues that "subsequent transactions are trust-based and contracting parties usually consider contracts as a symbol for an existing business deal", while "the enforcement of traditional contracts [121] proves to be either too complicated, time-consuming, or impossible, certainly in international circumstances".

An alternative approach to the traditional oral, or paper written contracts for transactions and collaborations are electronic smart contracts that allow to govern business transactions in a M2X-compatible manner using a computerized transaction protocol such as a blockchain. Moreover, smart contracts allow for the automated, globally-available orchestration and choreography of heterogeneous sociotechnical systems with a loosely coupled, P2P-like network structure. Additionally, a blockchain-based smart contract-driven platform enables fact tracking, non-repudiation, auditability and tamper-resistant storage of information among distributed participants without a central authority. Extending smart contracts to the concept of decentralized autonomous organizations (DAOs) [122] facilitates the ad hoc integration and coordination of collaborations and transactions. Finally, it allows for heterogeneous interoperability by using the interoperability design principles detailed in Section 2.1.4 in the context of the M2X ecosystem. Therefore, this chapter addresses the identified knowledge gap by answering the following research questions:

RQ-1: How to enable interactions, transactions and collaborations as well as reliable value transfer among entities of the M2X ecosystem?

- RQ-1.1: What are the goals and requirements of a M2X interaction, transaction and collaboration platform?
- RQ-1.2: What are selected system-engagement processes of a M2X interaction, transaction and collaboration platform?
- RQ-1.3: What are the modalities of a M2X economy?

In this chapter, we discuss the value exchange, transaction, interaction and collaboration within the M2X ecosystem as well as governance structures that are applicable. Section 3.1 presents an architecture for M2X interactions, transactions and collaborations based on the running cases introduced in the previous Section 1.1.2. Similarly to the Internet, the entities in the M2X ecosystem are heterogeneous, owned by different entities and distributed around the globe. Therefore, we detail system-engagement processes as well as a decentralized governance model. While we created a set of explicit and implicit rules for transactions, interactions and collaborations among humans, a similar toolset for autonomous smart devices is missing. Hence, Section 3.2 introduces a set of modalities that we deem to be important aspects of the M2X economy. Finally, Section 3.3 discusses the findings of this chapter.

Please note, the subsequent sections of this work are based on the following publications of the author in collaboration with varying co-authors: [5][36][37].

3.1 V2X Interaction and Transaction Platform

In Section 1.1.2 we introduce a set of vehicle-focused running cases rooted in the application field of the V2X economy. This section outlines the conceptual and technical

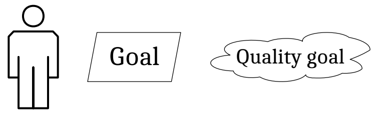


Figure 3.1: Selection of AOM Notation Elements

foundations that enable a V2X platform for the provision of services and goods in such a V2X ecosystem. Conceptually, the V2X economy is a subset of the more general M2X economy, limiting the focus on interactions, transaction and collaboration among vehicles (V2V), vehicles and infrastructures (V2I) as well vehicles and humans (V2H). When focusing on the V2X economy, we limit the scope to machines and infrastructure components related to vehicular use cases and activities. However, this simplification does not prevent the application of this section's finding to the general M2X economy since it covers the essential conceptual and technical functionalities and challenges.

Next, Section 3.1.1 captures the requirements of a V2X system and details the system design. In Section 3.1.2, the requirements are used to derive a corresponding system architecture. Afterwards, Section 3.1.3 details the system engagement process and the governance infrastructure.

3.1.1 System Design and Requirements

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 [58], i.e., goal models. The produced goal model is used in subsequent Section 3.1.2 to derive the system architecture. The resulting system architecture and specifications serve as implementation guidelines.

3.1.1.1 AOM Goal Modeling

In system development and software engineering, good requirements follow certain characteristics. According to [123][124], requirements address one issue only and are completely specified without missing information. Moreover, they have to be consistent and do not contradict themselves, or in correlation with other requirements. Finally, a requirement must also be atomic and without conjunctions [125].

The AOM methodology is a sociotechnical 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 3.1 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 the form of sticky men, whereas functional requirements are depicted as parallelograms. Note that in the specific context of this work, a sticky man does not exclusively represent human entities but rather all kinds of entities, e.g., also vehicles (or smart machines in general), agents and infrastructure. Functional requirements are referred to as goals. Non-functional requirements are depicted as clouds and refer to quality goals of the modeled 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 [126]. 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.

The following Section 3.1.1.2 introduces the top-level goal model our system, followed by Section 3.1.1.3 focusing on the non-functional goals of the AOM goal model.

3.1.1.2 Top-Level AOM Goal Model

Figure 3.2 presents the top-level AOM goal model of the system using the modeling method described above. The main value proposition is to provide a V2X platform and the corresponding interaction, transaction and collaboration model, thereby representing the root of the goal model. The complex main value proposition is split into four sub-goals representing the four main components.

First, a component for managing the V2X platform. This functional goal includes managing certain aspects of the platform itself, e.g., creating, updating, deleting a new platform, as well as the management of the underlying smart contracts. Each platform operates a master smart contract and several sub-smart contracts. While the master contract is in charge of platform management and controlled by the hardware vendor, the sub-contracts each offer service provision for a specific service.

The second functional goal enables V2X interaction. That mostly covers on-, and off-chain supply and demand administration. Entities may register offers or requests on-chain in order to attract business, or collaboration partners, but for other use cases a local supply-demand management off-chain is more suitable, e.g., road-space negotiation. Supervising on-, and off-chain auctions is equivalent to the on-, and off-chain supply and demand management. Besides that, plug-ins and decentralized applications (dApps) of the ecosystem might use platform's smart contracts for service enactment and have to be integrated as well in this context.

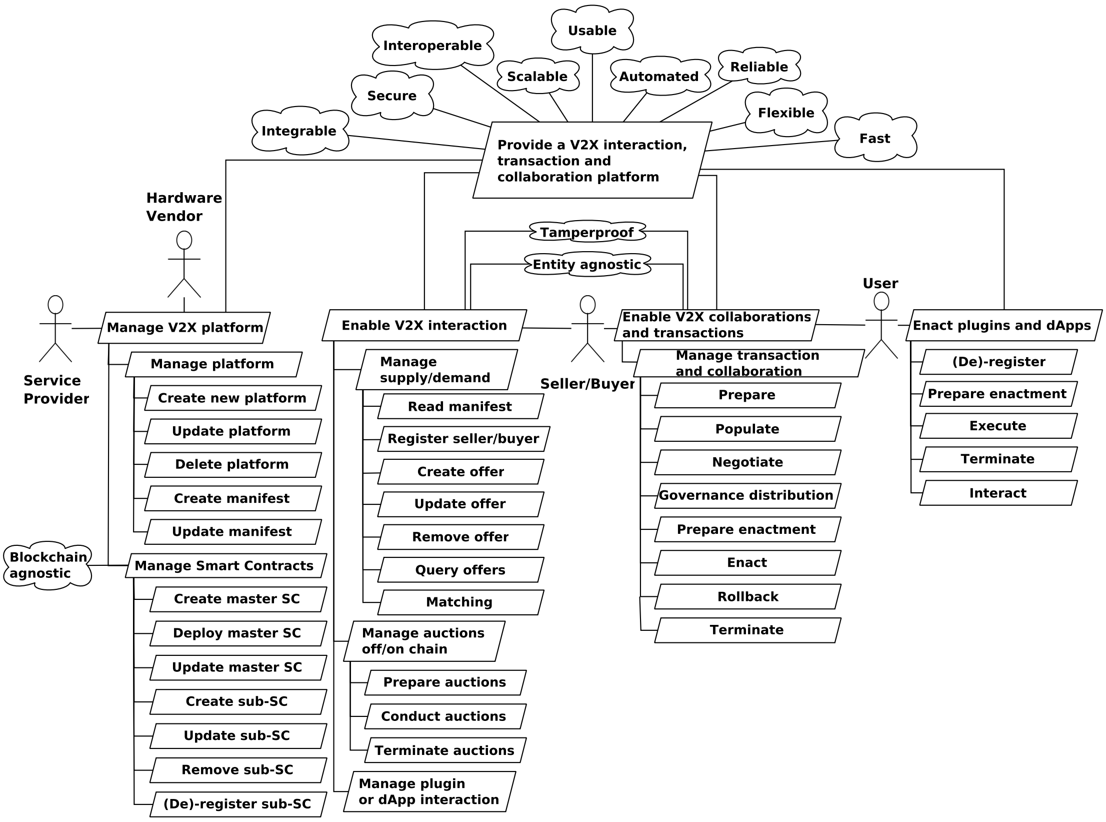


Figure 3.2: Top-Level Goal Model Representation V2X Platform – Based on [5] and [37]

The third functional requirement, representing the third main component, enables V2X transactions and collaborations via the blockchain. The most important part here is the transaction management via a smart contract collaboration lifecycle (detailed later on in Section 3.1.3.1). Finally, the fourth functional requirement focuses on the enactment of various plug-ins and dApps. Applications and plug-ins have to be registered, prepared for enactment, executed and terminated. Moreover, they have to interact with various entities of the ecosystem depending on the use case. Since nowadays most blockchains offer Turing-complete smart contract support, the variety of applications and plug-ins in our ecosystem is quite vast.

3.1.1.3 Non-Functional Requirements

Besides the four sub-goals of the top-level AOM goal model, we further identify thirteen quality goals, nine of them attached to the main value proposition and subsequently inherited to all refining sub-goals. A *scalable* system design is necessary to provide services to a large number of users. A further property that supports to achieve this scalability is the non-functional requirement *automated*, that refers to a high degree of process automation eliminating the need for human interaction or intervention, e.g., tedious and repetitive tasks. *Flexible* digital collaboration is a highly dynamic process that involves the enactment of a multitude of variations of activities, participating partners as well as the exchange of diverse data [127]. Thus, we must allow diverse collaboration scenarios and permit the inter-organizational harmonization of heterogeneous concepts and technologies between participating entities. Another key property of the system is being easy to use (*Usable*) for business collaboration. According to Norta et al. [125], easy usability also includes the support of proper *error avoidance* in order to "anticipate and prevent common errors that occur during a collaboration configuration. Closely related is *error handling*, to help with system support a user to recover from errors. *Learnability* refers to how quickly users master using the system" [125].

Interoperable hardware and software design is another consequence of the previous quality goals as well as easy integration (*integrable*). It is crucial to interoperate at runtime with information systems supporting other business functions. Furthermore, a *secure* service provision is crucial in terms of operational security, e.g., protect user accounts and personal data from unauthorized access, secure data transfer within the system between entities, or preventing data- and information leaks as well as preventing accidents. According to [128], "security is a composite of the attributes of confidentiality, integrity, and availability", where confidentiality ensures the absence of unauthorized disclosure of information, the availability the readiness for correct service and integrity the absence of unauthorized system manipulations. A *reliable* enactment of all interactions and transactions facilitates the previous goals as well. Finally, since cars and similar vehicles move much faster than humans, a *fast* service provision is essential for most tasks.

Further quality goals are assigned to sub-goals. Data communicated internally as well as externally has to be protected against unauthorized tampering (*tamperproof*) in order to protect business collaborations, but also ensure the safety of participating entities. Finally, we assign two additional quality goals that ensure a *blockchain-agnostic* as well as *entity-agnostic* design. The solution should be neither limited to a specific blockchain, nor vehicle hardware of a specific vendor, or infrastructure provider.

The presented goal model is used in the following Section 3.1.2 to derive the system architecture. We do not list all details of the further refined AOM goal model in this work due to space constraints and in order to focus on the most relevant system components and features.

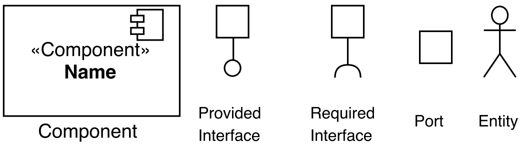


Figure 3.3: UML-Component Diagram Notation Elements

3.1.2 System Architecture

The abstract system 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 and provides a specific set of services [129][130]. Dedicated services and components may also consist of other underlying sub-services [131].

In the following, a technology-agnostic UML-component-diagram representation is used to illustrate the system architecture [60][61]. The UML notation elements used to model the architecture are presented in Figure 3.3. 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 3.3. 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 [60][61]. Once more, sticky men are used to depict entities and their interactions with the system.

The remainder of this section first introduces an abstract high-level overview of the system architecture and components. Further illustrations present selected sub-components of the architecture.

3.1.2.1 High-Level Architecture

The highest architecture abstraction levels of our system are depicted in Figure 3.4 and Figure 3.5. Figure 3.4 presents a five layer model of the V2X system consisting of the *Hardware Layer*, the *Firmware Layer*, the *Network and Integration Layer*, the *Application Layer* and the final *Blockchain Layer*. Figure 3.5 presents a more technical UML component diagram that can be mapped to the layer model of Figure 3.4. The representation is divided into two distinct packages, i.e., the *Blockchain* package and the *Vehicle* package. In UML, packages are used "to group elements, and provide a

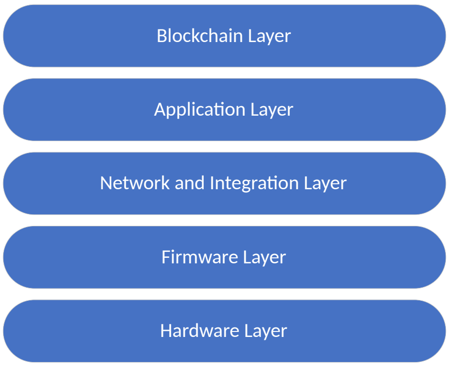


Figure 3.4: V2X System Model – Layer Structure

namespace for the grouped elements" [61]. In the context of this architecture illustration, packages are used to provide a separation of concerns between the blockchain part and the vehicle-related system components.

The *Vehicle*-package consists of five main element groups that match the layer model of Figure 3.4. The five element groups are: First, the *TPD*-component and the *Hardware and sensor*-component that constitute the *Hardware Layer*. Second, the vehicle's manufacturer operating system (OS) (*Vendor OS*) and the ROS (Robot Operating System) [132] and/or the Apollo [133] module that represent the *Firmware Layer*. This layer manages the underlying hardware of the vehicle as well as its fundamental functionalities, e.g., driving, navigating, and so on. ROS provides a set of libraries and tools which help software developers to create robot applications in general. However, ROS is also often used in the context of self-driving and autonomous vehicles. While ROS is a general platform for robot applications, Apollo specifically focuses on software development for autonomous driving systems[1]. Third, the *Network and Integration Layer* maps to the *Communication Manager*-component consisting of the vehicles *OBU*-component and the *V2X Integration Interface*-component. As previously mentioned in Section 2.2.1, the OBU enables data exchange with other OBUs or RSUs, usually via short-range wireless- or radio communication. The *V2X Integration Interface* provides a set of standardized integration interfaces for VANET applications to utilize the OBU, the vehicles hardware sensors and to interact with vehicles OS. Depending on the application, the interaction with the vendor's OS may be mere information exchange, but for more complex applications (e.g., road-space negotiations) this may be navigation requests to the OS such as: "Yield to vehicle X as a result of a successful road-space negotiation, but only if the maneuver can be performed safely". Next, the *Application Unit Manager*-component represents the *Application Layer* consisting of the applications available/installed on the vehicle (AU-1 to AU-3). As mentioned in Section 2.2.1 an AU may reside in the same

[1] Note that Apollo and ROS can be substituted with an arbitrary alternative solution that enables self-driving and autonomous driving capabilities.

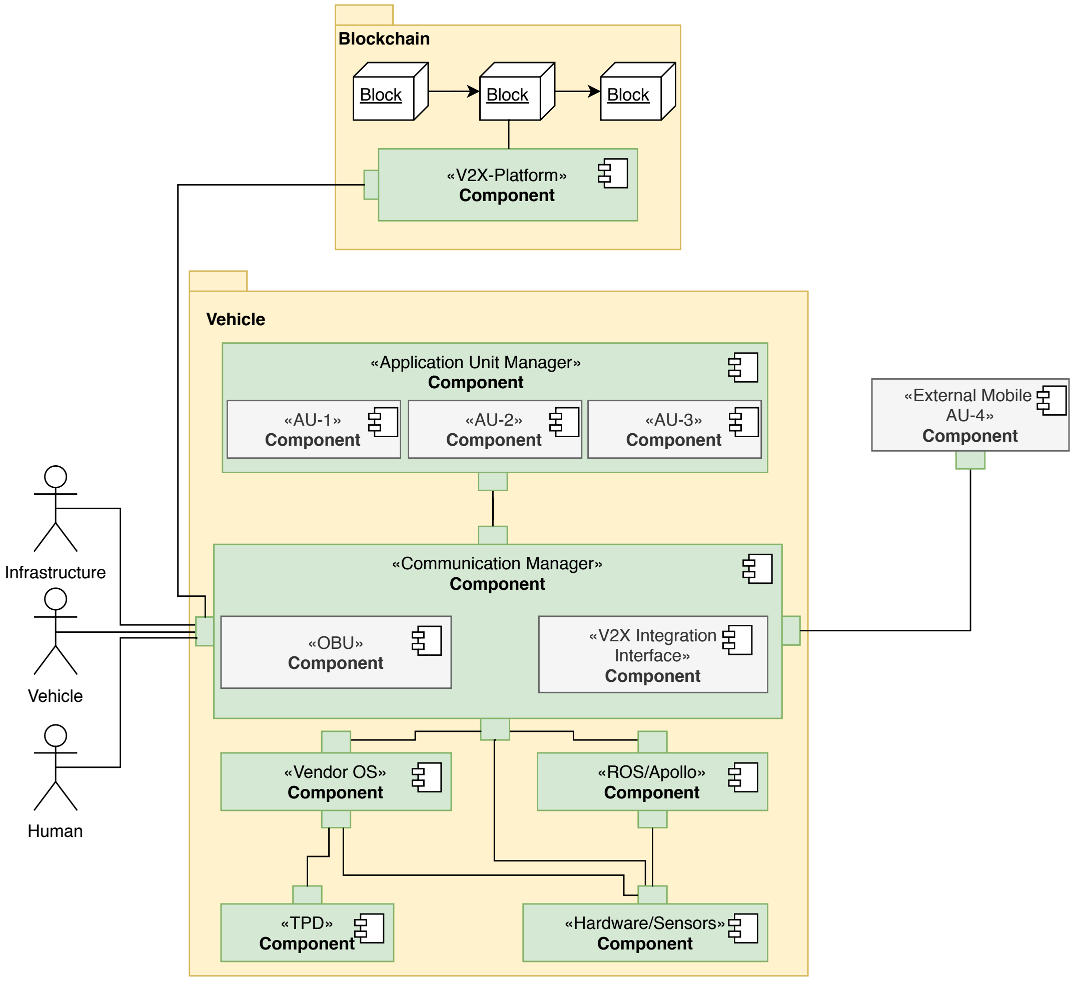


FIGURE 3.5: High-Level Architecture Overview of the V2X System Model – Based on [5] and [37]

physical device as the OBU, e.g., AU-1 to AU-3, or in a separate mobile device that is regularly removed from the vehicle (e.g., smartphones) as illustrated by AU-4. The later case uses the *Communication Manger Unit* to integrate the mobile AU-4 into the system. Finally, the *Blockchain*-package comprises the V2X platform itself and the blockchain utilized to enable service provision thereby representing the *Blockchain Layer*.

3.1.2.2 Selected Architecture Refinements

The following section outlines selected refinements of the high-level architecture presented in Figure 3.5. We simplified certain aspects due to space constraints and to reduce technical complexity.

Figure 3.6 presents a more detailed view of the *Vehicle* sub-system. At the bottom, hardware components such as the TPD, communication devices and hardware sensors are illustrated. Please note that the listed hardware component represent only a small illustrative subset of the actual hardware that is part of an autonomous vehicle. The different hardware connectors and the OBU provide a variety of communication interfaces for external – to the user's smartphone via Bluetooth or WIFI, the V2X platform via the Internet, to other vehicles via the WAVE (Wireless Access for Vehicular Environments) protocol stack [134][135] – as well as internal data exchange, e.g., via the CAN bus. The CAN bus interface is used to query information from the car such as speed, steering, breaks and many more. The TPD and other hardware components connect to the vehicle OS and ROS/Apollo, which again communicate with the upper layer of the *Network and Integration Layer* (represented by the *Communication Manager*-component) as described previously.

The *Communication Manager*-component consists of two further sub-components, the *OBU*-component and the *V2X Integration Interface*-component. The *OBU*-component is merely a communication device, thus we will not further detail its technical structure. On the other hand, the *V2X Integration Interface*-component contains a variety of components relevant for the fundamental functionalities of the V2X system.

Due to the velocity of moving vehicles, most of the time on-chain transactions and interactions are not an option and instead collaboration negotiations and enactments as well as auctions on a local level between nearby vehicles are necessary. The *Interaction, Transaction and Collaboration Management*-component contains all functionalities to do so as well as settings that control the auction preferences of a vehicle. More details on the actual workflow of the off/on-chain auctions are available in Section 3.1.3.2. Finally, the *Reputation System*-components maintains trust scores and reputation of other network participants based on previous interactions, either by the vehicle itself or other trusted network entities.

Moreover, the *V2X Integration Interface*-component further hosts several blockchain clients to connect and operate with the corresponding blockchain platforms, thereby ensuring a blockchain-agnostic architecture. Furthermore, a *Vehicle Wallet*-component exists to enable token transfers to the vehicles as part of the vehicle's TaaS earnings, or to pay for electricity and maintenance.

The *AU Plugin Management Interface*-component is managing the communication of the underlying components with and via the OBU. In addition, it is the integration point for all AUs which are handled by the *Application Unit Manager* and also provides the OBU communication capabilities to all AUs. Since AUs may be developed by different vendors, developers and companies, a standardized integration interface – similar to mobile applications on Android or iOS – is necessary. The *Application Unit Manager* may handle blockchain-enabled applications such as the TaaS application, blockchain-based payment solutions, or road space negotiations. However, not all AUs have to

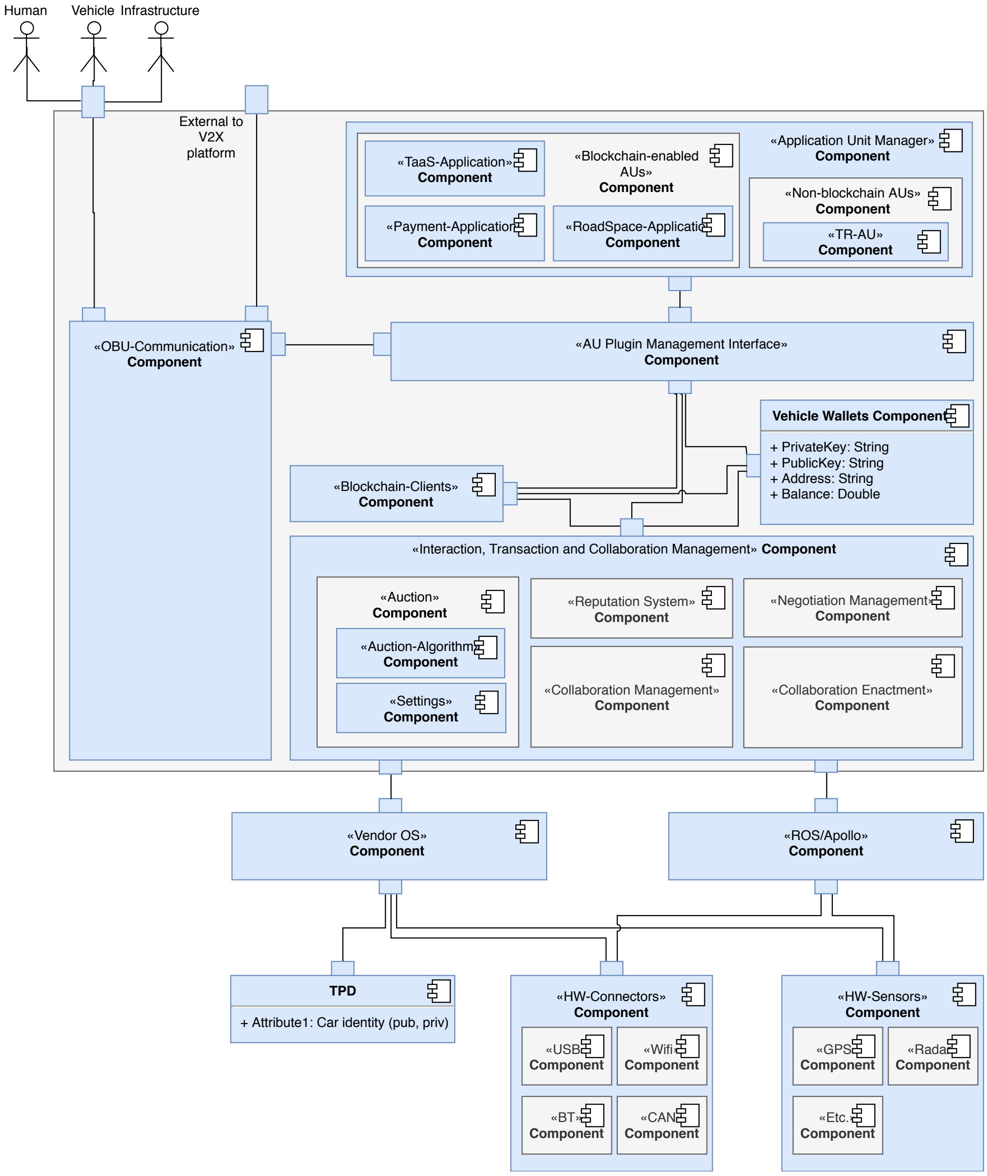

Human
Vehicle
Infrastructure
External to V2X platform
«TaaS-Application» Component
«Blockchain-enabled AUs» Component
«Application Unit Manager» Component
«Non-blockchain AUs» Component
«Payment-Application» Component
«RoadSpace-Application» Component
«TR-AU» Component
«OBU-Communication» Component
«AU Plugin Management Interface» Component
Vehicle Wallets Component
+ PrivateKey: String
+ PublicKey: String
+ Address: String
+ Balance: Double
«Blockchain-Clients» Component
«Interaction, Transaction and Collaboration Management» Component
«Auction» Component
«Auction-Algorithm» Component
«Settings» Component
«Reputation System» Component
«Negotiation Management» Component
«Collaboration Management» Component
«Collaboration Enactment» Component
«Vendor OS» Component
«ROS/Apollo» Component
TPD
+ Attribute1: Car identity (pub, priv)
«HW-Connectors» Component
«USB» Component
«Wifi» Component
«BT» Component
«CAN» Component
«HW-Sensors» Component
«GPS» Component
«Radar» Component
«Etc.» Component

FIGURE 3.6: Refined Illustration of the V2X Vehicle Sub-System – Based on [5] and [37]

communicate with the blockchain or the V2X platform, this applies only to blockchain-enabled or blockchain-dependent applications. The traffic regulation (TR) AU is not dependent on any blockchain-related services or endpoints and hence does not require any interactions with this part of the system. Please note, for simplification reasons external mobile AUs are not illustrated in Figure 3.6.

Finally, Figure 3.7 presents a detailed view on the *V2X platform*-component. At the bottom of the figure, the *Application Unit Manager*-component of Figure 3.6 is presented again. Blockchain-enabled AUs communicate with their on-chain smart contract counterparts, e.g., the TaaS smart contract, or the road space negotiation smart contract. Via the corresponding interfaces such smart contract components can utilize the V2X platforms *Supply/Demand management*-component as well as the *On-Chain Auction*-component. The later one manages on-chain auctions for goods and services that do not occur on a local level as described earlier. Auctions may result based on the supply and demand management that is conducted in the corresponding component. The on-chain supply and demand management of values is not as restricted as for example road space negotiations that can only occur in a specific location, at a specific time among a very limited set of participants. Such services and goods may include battery charges, TaaS offer and demand matching, parking spots, and so on. All transactions that occur on the *V2X Platform* are included into the underlying blockchain.

We expect the M2X ecosystem to be decentralized and distributed consisting of devices and entities of different vendors – i.e., a multi-stakeholder system. As a result, not a single M2X (respectively V2X) platform exists. Instead, different vendors may operate their own platforms that ensure interoperability among each other. To manage each of the platforms, the vendor/operator has access to the *Platform Administration*-component that not only allows to manage the platform itself (user management, payment management, device management, etc.) but also to create so-called master smart contracts for their products and services. Each V2X platform is represented by a master smart contract that in return offers the basic fundamentals of each of the V2X platforms – i.e., supply- and demand management, on-chain auction management, subscription management and the on-chain integration interface for the AU (the corresponding smart contracts for the blockchain-enabled AUs).

Next, in Section 3.1.3 we present the system-engagement processes of the V2X system, outline the collaboration lifecycle, a distributed governance proposal and the interaction workflow of the on/off-chain auction algorithm.

3.1.3 System Engagement Processes

The transaction, interaction and collaboration platform automates and simplifies VANET-based V2X service provision on several levels. A core element of many of the use cases is a smart contract-based negotiation and contract enactment between entities that are the result of collaborating tasks and sub-processes. For example, two vehicles conduct a road-space negotiation auction that results either in a change of positions or is aborted. This process potentially involves payment processing, further local as well as global communication and local match-making between vehicles. On an abstract level, most of the use cases presented earlier in this work follow a similar procedure on the smart-contract level. The same applies for scenarios that involve a price negotiation or auction. In the following, we introduce these two abstract processes in more detail. The processes and

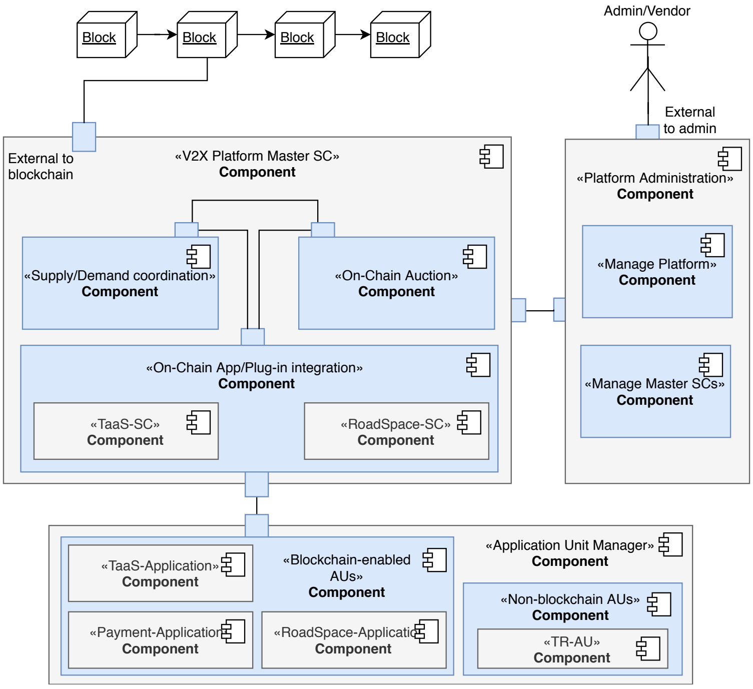


FIGURE 3.7: Refined Illustration of the V2X Platform Sub-System – Based on [5] and [37]

algorithms are represented using Business Process Model and Notation (BPMN) [59] and sequence diagrams. Consequently, Section 3.1.3.1 details the BPMN representation of the abstract collaboration lifecycle and a decentralized governance approach, followed by Section 3.1.3.2 that details an efficient auction mechanism for V2X and M2X use cases.

3.1.3.1 Collaboration Lifecycle Management

In [120, 136–138], Norta presents a conceptual smart contract based collaboration lifecycle as illustrated in Figure 3.8. The abstract nature of the proposed conceptual collaboration lifecycle allows the utilization of the same approach for all V2X services offered within our V2X system that involves V2H, V2V, or V2I interactions, transactions and collaborations. The lifecycle, as illustrated in Figure 3.8, is divided into the following stages: *i.*) preparation, *ii.*) negotiation, *iii.*) governance distribution *iv.*) preparation of collaboration enactment *v.*) collaboration enactment *vi.*) rollback, and *vii.*) termination stage. While Figure 3.8 presents the collaboration among partners from a lifecycle perspective, Figure 3.9 depicts the creation sequence of a distributed governance infrastructure (DGI) from an infrastructure perspective in comparison to the previously described lifecycle. During the preparatory stage, based on pre-configured templates, information regarding the involved entities, such as identifiers and wallet addresses are incorporated into the contract. In addition, the conditions of the requested contract are

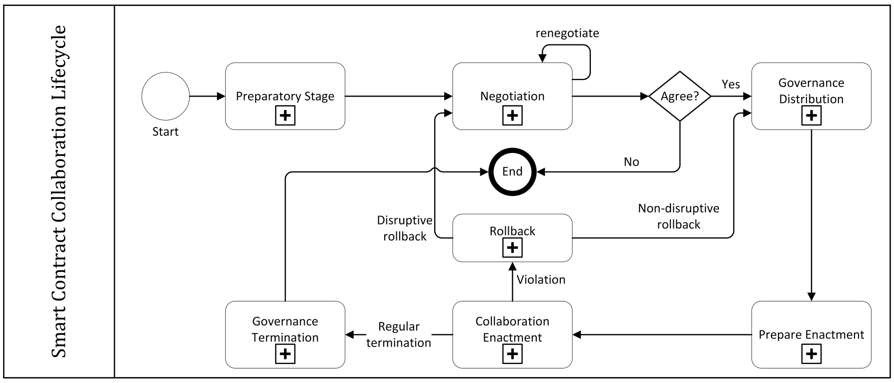


FIGURE 3.8: Smart Contract Collaboration and Negotiation Lifecycle – Based on [120][136][137][138]

formally defined by specifying, e.g, the content and target of the contract. Following the example of the transport service for a human, this might include the departure location, final destination and price. The conditions of the requested cab-ride mainly depend on information such as the travel distance and fuel/energy consumption of the vehicle. In case the vehicle and the user agree on the negotiated conditions, both parties sign the contract and express their approval – if no agreement is reached, a contract rollback is triggered. A smart contract between the involved parties is established and serves as a DGI-coordinating agent. As part of the governance distribution, each participating entity receives a local contract copy containing the respective obligations of each party [120][139], e.g., transporting the user to the correct location. The participants "obligations are observed by monitors and assigned business-network model agents (BNMA) that connect to IoT-sensors" [139] such as the vehicle's GPS-sensor. During the stage of the contract enactment preparation, the required process endpoints (e.g., for payment processing) are provided and prepared. "Once the e-governance infrastructure is set up, technically realizing the behavior in the local copies of the contracts requires concrete local electronic services. After picking these services follows a creation of communication endpoints so that the services of the partners are able to communicate with each other. The final step of the preparation is a liveness check of the channel-connected services" [120]. Afterwards, the contract execution phase is triggered and the vehicle picks up the user. The transportation contract terminates, or expires either after the user arrives at the final destination, or when the contract is prematurely terminated. Failing to transport the user to the agreed-up destination might result in an immediate rollback of the smart contract or invokes some kind of a mediation process that is supervised by a conflict resolution escrow service that is not depicted in Figure 3.8.

The presented lifecycle and governance infrastructure does not only cover trading negotiation but rather all kinds of contract enactments. The user prepares and negotiates a contract with a collaboration partner and executes it in case that both parties agree

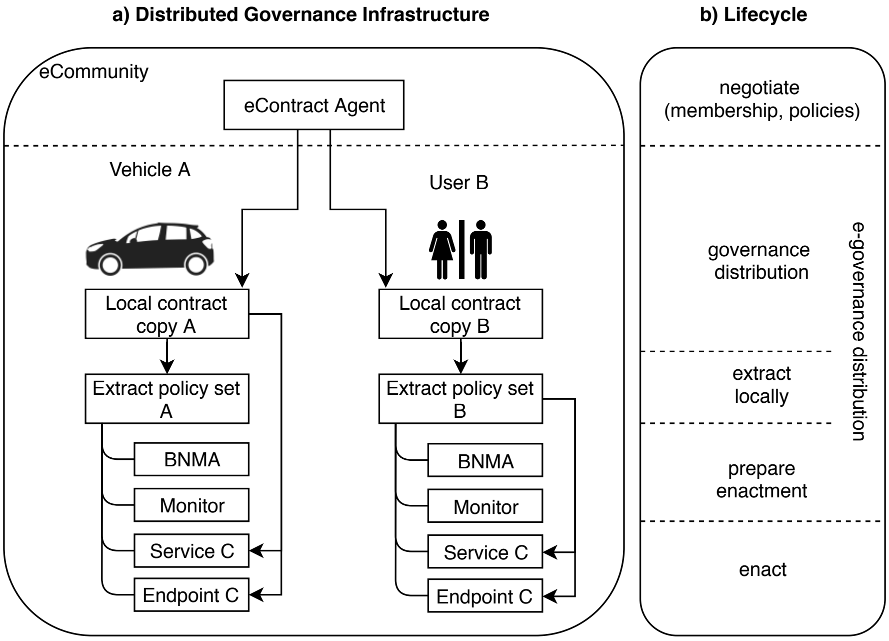


FIGURE 3.9: Distributed Governance Infrastructure – Based on [120] and [140]

on the specifications. That also includes incentives in case the user behaves correctly as well as punishment of bad behavior, e.g., by paying a penalty. A serious violation of the contract from any of the involved parties might result in an early termination or a rollback. While some conflicts may be handled in a calming manner that allows to continue the collaboration enactment, others may cause an early termination of the collaboration.

3.1.3.2 Auctions and Negotiations

A further core concept of our V2X system is to support the exchange and provision of goods and services. When trading goods and services, the buying and the selling party usually have contrary goals in terms of pricing. The seller's goal is to maximize profits while the buyer tries to minimize the costs. Auctions are a common approach to reach a consensus on a certain price between buyer and seller. We designed an auction algorithm based on the concept of Vickrey Auctions [141][142]. During a Vickrey auction, participants exchange sealed bids. Each bidder submits a written and signed bid without having any knowledge of the bids of the other participants. After submitting all bids, the sealed bids are opened and subsequently the highest bidder wins. But instead of paying the price of the highest offer, the price paid is the second-highest bid. Due to space constraints and the technical nature of this paper we will not cover the economical, and game theoretical implications of Vickrey auctions and instead refer the reader to specific supplementary literature, e.g., [141][142][143][144][145].

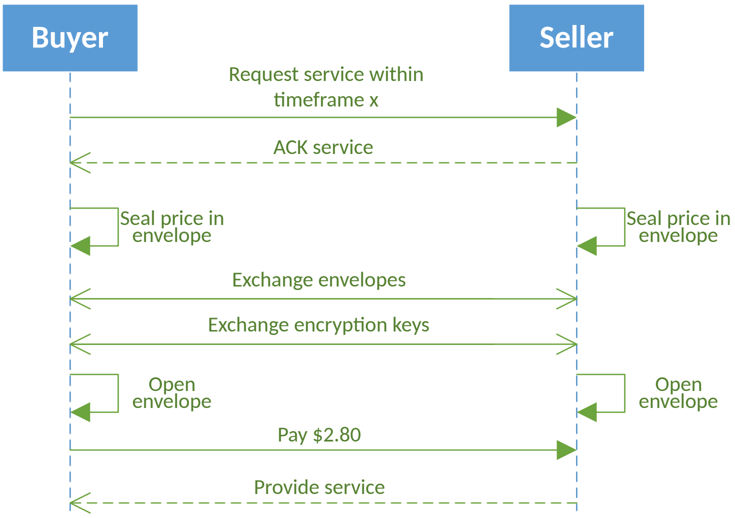


Figure 3.10: V2X On/Off-Chain Auction Algorithm with 1 Buyer and 1 Seller – Based on [5] and [37]

Figure 3.10 and Figure 3.11 present the sequence diagrams of our auction algorithms that are either run locally (off-chain) between auction participants that reside in close proximity to each other, or on-chain when interacting on a global scale. As mentioned in the AOM goal model (Figure 3.2), speed is one of the non-functional goals of our system – hence, only one auction round is conducted.

For the one-to-one auction – illustrated in Figure 3.10 – with only one buyer and one seller, we assume that the buyer is not willing to pay more than $3, and the seller is not selling for less than $2.80. Both participants prepare an encrypted (sealed) and signed bid before exchanging the bids. As soon as both participants received the other party's bid, the encryption keys are exchanged as well. Buyer and seller decrypt the bids and compare the offers. Given the case that the buyer offered more than $2.80 the auction is successful and due to the second-price rule of Vickrey auctions, the buyer pays $2.80 to the seller. In case the buyer offers less than the seller's minimum price the auction ends without a deal.

In case multiple buyers and sellers participate in an auction, the workflow is very similar as illustrated in Figure 3.11. In the given scenario buyer one is willing to pay a price of $1.80, buyer two offers a price of $3.20 and buyer three is offering $3.50. The seller is not selling for less than $2. We conduct a single auction round and the buyers as well as the seller all submit their bid in an encrypted and signed envelope that is distributed and send to all registered participants. As soon as all participants received the bids, the encryption keys are exchanged as well and the sealed bids are decrypted. Buyer three wins the auction and pays the seller the price of buyer two that offered $3.20.

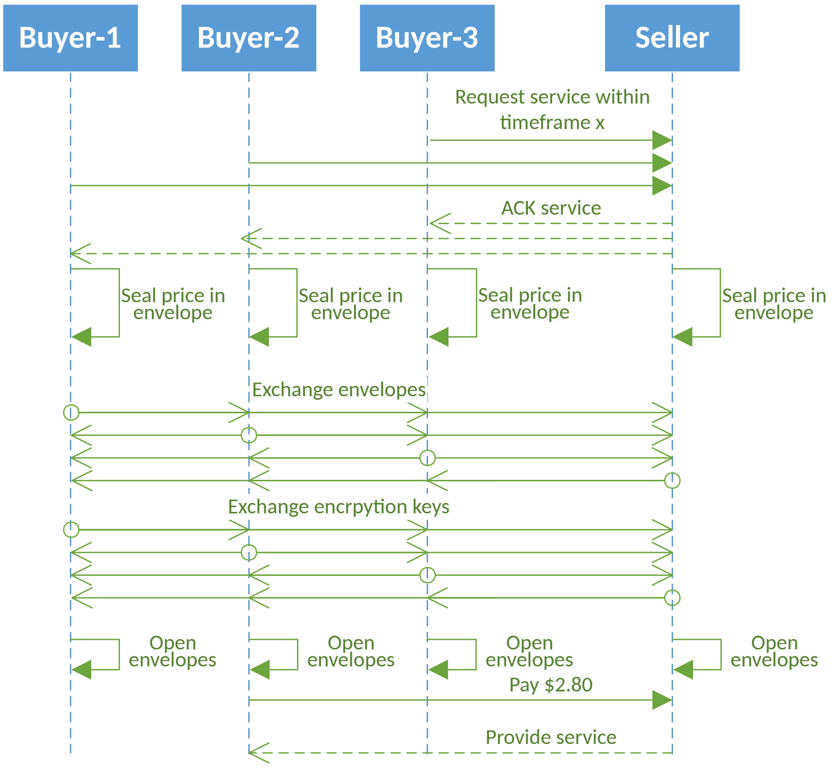


Figure 3.11: V2X On/Off-Chain Auction Algorithm with Many Buyers and Many Sellers – Based on [5] and [37]

In case we have multiple sellers, the sequence diagram is almost identical and the bidding process follows the same procedure. Except in the end, the highest bidder is paying the second-highest price to the seller with the highest minimum price, and so on – as long as the paid price is higher than the matched seller's minimum price.

Finally, not all use cases benefit from a dynamic auction algorithm as described above. For example, in the context of continuous data streams, e.g., traffic jam information, a subscription-based pricing model is more suitable. Alternatively, other applications such as smart parking spots might benefit from a fixed-price structure, e.g., $1 per hour. However, also subscription and fixed-price payment models can be combined with a dynamic auction algorithm, or adaptive price ranges based on demand, e.g., higher toll road fees during rush hours.

3.2 Modalities

The previous sections demonstrate the general viability and capabilities of a blockchain-driven M2X/V2X platform using the running case of autonomous self-driving vehicles in the context of TaaS and road space negotiations where agents act as independent and autonomous participants. Blockchain technology and smart contracts specifically are an essential part the enables the V2X economy that offers a wide range of new business, and transaction models. Those new models require a variety of context-specific process modalities such as machine-readable mechanisms to evaluate the reliability of business partners, or the seamless and accountable transfer of values. In order to detect deviations from agreed-upon contracts a machine-readable, complete and non-reputable logging of all interactions and transactions is essential. Such mechanisms allow all involved parties to prove contract violations and to enforce corresponding sanctions. Enabling trusted transactions among participants who do not trust each other based on a pre-defined set of rules is a key enabler for this concept [146]. Moreover, the distributed and decentralized structure of a blockchain makes an ideal platform for autonomous agents within the M2X ecosystem.

The following section details how blockchain technology supports and guarantees those modality concepts. We introduce a set of six modalities that we identified throughout our work on applications and systems within the research field of the M2X economy. However, we do not claim completeness and encourage further research in this direction.

3.2.1 Accountability and Logging

To achieve non-reputable accountability a comprehensive logging mechanism for all digital processes and their corresponding parameters is necessary. Even though such data-logging is already happening in most IoT systems and use cases, data is usually only stored locally, or within a specific system instead of being globally available to all involved or interested parties. Hence, a unified overall perspective with access mechanisms across process and a shared semantic structure as well as protection against local malfunctions or manipulations are needed.

While the overall perspective on specific business processes is context-dependent and potentially unique to each process, the protection against malfunctions and manipulations is rather generic and a prime example for the application of blockchain technology. Similarly to the tamperproof logging of financial transactions in Bitcoin, processes, events and parameters are logged, signed by the originator and stored in a tamperproof and auditable manner. In a blockchain system, such data sets can be stored directly within a block or using tree-like hash structures such as Merkle trees. The viability of this approach has already been demonstrated in blockchain-based, or blockchain-enabled filesystems such as IPFS and Filecoin [147][148].

Moreover, the non-reputable accountability of contractual agreements is another important aspect of every business process. An agreed-upon transportation destination, or the agreed-upon amount of requested/delivered electricity for a specified price are just two examples where non-reputable accountability is crucial and can be ensured via comprehensive logging mechanisms. The same applies to timely delivery of specified quantities of goods or services with certain quality parameters.

As briefly described in Section 3.1.3.1, as part of the governance distribution, monitoring mechanisms are deployed to observe the compliance of each party with the corresponding obligations. Deviations from the contractual agreement are logged. A transportation vehicle may exceed the agreed-upon battery charging level, or the battery charging stations might not be able to deliver sufficient amounts of electricity caused by a shortage in produced electricity in the corresponding solar power plant due to bad weather. In order to handle such contract deviations a proper conflict resolution management has to be in place. We briefly introduced some conflict resolution concepts in Section 3.1.3.1. The blockchain offers a distributed, replicated, synchronized and tamperproof data structure to all involved entities that allows for non-reputable and accountable data logging that cannot be manipulated by a single entity or a minority of malicious users.

3.2.2 Privacy

Protecting privacy-sensitive data and accountable, all-encompassing logging of interactions and transactions is – at first glance – contradicting. However, using a large number of partial identities, e.g., in combination with pseudonyms, which can only be mapped to specific entities and their corresponding devices if they wish so, as well as concepts such as blind signatures [149] and anonymous certificates [150] allow to combine tamperproof logging, accountability and protection of privacy-sensitive personal data at the same time. The concepts of fully homomorphic encryption and zero knowledge proof systems provide further promising mechanisms to process sensitive data in a privacy-preserving manner [151][152][153].

For many use cases, only the involved contract parties have to be aware of the other party's identity. For logging and accountability purposes in the context of payment solutions, anonymized or pseudonymised identities are often sufficient and even allows to build reputation systems based on such data. Nevertheless, minimizing the amount of data that has to be logged is not only desired but actually crucial. Research on Big Data security and privacy revealed many different techniques to correlate data, de-anonymize entities, and so on [154][155]. Especially in the context of location data – as might be used as part of transportation services provided by vehicles – allow for unexpectedly simple and manifold profiling methods [156].

In the context of autonomous agents or autonomous smart devices, the privacy concerns become less critical since we can assume that their "personal" data may be less critical from a privacy perspective compared to personal data of human entities within the M2X ecosystem.

3.2.3 Trust

In business collaboration processes, trust is often based on, or the result of previous interactions evaluated via objective parameters pertaining to the specific context of a business arrangement, e.g., was the ordered product delivered in sufficient quantity and quality? Furthermore, subjective trust may be based on reviews of entities, services, or goods using comparison metrics [157][158][159].

In the TaaS context of human-operated vehicles (e.g., Uber and Lyft), trust is expressed using a review and evaluation system based on pre-defined performance parameters. Similar metrics could apply in the future for self-driving and autonomous vehicles that offer transportation services for persons and goods. In the context of battery charges, the electricity supplier (charging stations) as well as the consumer (vehicles) are interested in an objectively measurable parameter of how much electricity is delivered/consumed. For the consuming party, a timely delivery of sufficient quantities is essential to offer further services to its own customers, while the supplier favors customers with a superb and flawless payment history.

While many M2X systems are equipped with sensors and actuators in order to either sense, or interact with their environment, such information is often not sufficient for complex M2X use cases. External data feeds are required, that provide machines with data that they cannot sense or generate on their own, e.g., gas prices, weather forecasts, or traffic information. In the context of blockchain technology, such services are often referred to as oracle services [160]. With respect to oracle services, trust is a sensitive topic and essential to the proper service provision. In order to judge the data-feed quality of an oracle, the record of previously provided information is an indicator for the quality of information provided in the future. A FOAM [161] radio beacon or a GPS sensor that provides false localization data is either faulty or not trustworthy. Similar assumptions apply to arbitrary domain-specific oracle services.

Comprehensive logging – as described in Section 3.2.1 – enables the setup of trust infrastructures. However, bootstrapping a trust system is challenging especially in decentralized systems without a trusted central authority. Nevertheless, different solutions from other research fields exist and can be applied in this context [162][163]. Moreover, we can further enhance such existing mechanisms by introducing a staking system (similar to PoS as introduced earlier in Section 2.1.2.2), where stakes represent “bounties”. Each oracle service provider deposits a stake and in case the provided oracle data-feed is proven unreliable or faulty by an auditor, the service provider loses its stake. The consensus mechanisms of the underlying blockchain system ensure that participating entities cannot make false accusations or manipulate data in a malicious manner in order to game the system.

3.2.4 Market Behavior

This modality pertains to the autonomous negotiation of supply and demand. The goal of economically viable and stable market behavior is to ensure stable and meaningful price structures for all market participants and at the same time, avoid risks and issues that are well known from high-frequency trading on exchanges [164]. This means, for example, that costs must be taken into account that arise from the provision of large quantities that are ultimately not purchased. Likewise, actually consumed service units must be billed reliably and timely. However, the respective microeconomic situations depend on the respective application and its context. For this purpose, concepts such as reservations, options and capacities must be developed and mapped to standardized structures of blockchain smart contracts.

Closely linked to economically viable and stable market behavior is the choice of proper price models as previously discussed in Section 3.1.3.2. For example, when purchasing localization data with a high-frequency, payment and renegotiation of every single query is not economically feasible. Pricing structures with time unit-based usage (e.g. per month) or with larger unit contingents (e.g., price per 1000 queries) and corresponding monitoring mechanisms are more suitable. On the other hand, in the context of TaaS, specifically tailored negotiations for each service provision seem to be feasible. Depending on the use case and the service, price models need to be adapted accordingly.

3.2.5 Interoperability

Traditional IT platforms tend to deliberately forced, or functional lock-in effects that lead to the formation of self-contained data- and service silos such as Facebook, Google, or Amazon. In the context of traditional IT platforms several Facebooks make little sense – neither from a network economic, nor from a profit, or monopoly-oriented perspective of a corporation. Similar applies to the M2X ecosystem of decentralized and autonomous smart devices where a one-stop platform is also desirable, but not a manufacturer-focused platform with deliberately forced, or functional lock-ins. Instead, similarly to the blockchain interoperability approaches described in Section 2.1.4, an interoperability hub/layer that implements the compatibility of different manufacturer platforms is a desirable and viable option. Only the resulting interoperability of smart devices enables the exploitation of economies of scale and increased efficiency. A blockchain-based interaction, transaction and collaboration platform as described in previous sections not only enables an interoperable platform for autonomous smart devices (e.g., vehicles), but also further reduces dependency on intermediaries. Furthermore, a blockchain-based solution enables the decentralized settlement of value added in the form of crypto tokens; these can be created entirely without central instances or intermediaries and exchanged directly P2P [165]. Technically, such an interaction, transaction and collaboration platform could be realized by so-called relay chains. Relay chains as Polkadot [100] offer a communication interface (hub) over which different heterogeneous blockchain platforms can interact with each other. Thus, for example, the specific blockchain-based services of a manufacturer can also be made accessible outside their own platform. This not only enables the cross-platform interaction of autonomous smart devices described above, but also increases customer reach for manufacturers and service providers.

Besides the software interoperability, interoperability on the hardware level is indispensable as well. In the context of battery charging services for vehicles and electricity trading in general this concerns the line voltage, the frequency, or the compatibility of the respective connection method for the consumer: Different charger standards for various electric car manufacturers exist. Finally, producers and consumers have to carry out control measurements at the same intervals and log them as described in Section 3.2.1.

3.2.6 Environment Integrity

Environment integrity is the trust of individual entities in the proper configuration and execution of their runtime environment and partner services required for their own service provision. However, ensuring the “correct functioning” of a smart device is difficult to prove and rather abstract. However, for practical purposes it can be broken down into the sub-processes of configuration, maintenance and update. Each of these three sub-processes is ensured by establishing a secure boot environment provided by the system vendor. The correct calibration and potentially even self-calibration of sensors of smart devices is a sub-category of environment integrity [166][167].

Typically, cryptographically secured chains of trust are used to couple, for example, the update of a driver function to the keys stored in the boot environment. This approach is heavily dependent on the cooperation of many manufacturers and has led to multiple problems in the past, e.g., Windows driver signatures [168]. The problem can be solved by blockchain technology and thus by flexible, decentralized trust chains. Cap and

Leiding [169] propose such a solution in the context of modular resources for web pages, thereby preventing the execution of malicious Javascript files.

Environment integrity may also exceed the limits of physical environments as demonstrated by the example of autonomous and self-driving vehicles. In addition to software updates (e.g. control software), many configuration parameters (e.g. road maps, traffic regulations) need to be updated in a trustworthy manner, sometimes on a very short notice (e.g. traffic jam information or fuel prices); hardware (engines, brakes, etc.) also needs to be maintained regularly and in a trustworthy manner.

3.3 Discussion

In this chapter, we outline a M2X/V2X platform for value exchange, transactions, interactions and collaborations in combination with the corresponding governance structure based on the V2X running cases. Despite the comprehensive description and analysis, the presented solution only provides a glimpse on the economy due to its social and technical complexity. While the presented models and architectures may serve as an initial step, they neither represent a reference architecture, nor a fully-fledged production-ready solution. Similar applies to the exploratory overview on modalities that we identify during on-going research work in the field of blockchain-based M2X use cases. The list of modalities is neither complete, nor comprehensive. Further research is required to enhance the existing list, advance the modality specifications and create a corresponding conceptual framework that abstracts the modalities from specific use cases.

More than ten years passed since the introduction of blockchain technology as a concept in the context of Bitcoin. However, the current state of the technology, surrounding toolsets as well as its suitability as the foundation of complex ecosystems is a topic of on-going research and development. DLT-enabled electronic smart contracts provide the foundation to govern business transactions in a M2X-compatible manner using a computerized transaction protocol. Smart contracts allow for the automated, globally-available orchestration and choreography of heterogeneous sociotechnical systems with a loosely coupled, P2P-like network structure. In addition, a blockchain-based smart contract-driven platform also enables fact tracking, non-repudiation, auditability and tamper-resistant storage of information among distributed participants without a central authority. Extending smart contracts to the concept of DAOs facilitates the ad hoc integration and coordination of collaborations and transactions. Finally, it allows for heterogeneous interoperability by using the interoperability.

Despite the benefits of a blockchain technology stack, a variety of disadvantages exist, e.g., a limitations in quantity and quality of development tools, missing integration into existing legacy IT-systems, a lack of industry-standards, unclear specifications, unfinished and inconsistent documentation, fast iteration cycles in software development (hence, often changing APIs, etc.), and many more. In addition, technical limitations and issues such as consensus algorithms, scalability and privacy of data are still being researched. Moreover, despite some initial concepts, most blockchain platforms require a steady Internet connection and barley allow for asynchronous data processing of offline-nodes [170]. Finally, while a blockchain-based solution provides a promising foundation, alternative non-DLT-based approaches such as centralized databases, or general distributed system concept may be suitable as well despite their own drawbacks.

The M2X modalities presented in Section 3.2 require further additions and refinements. The concrete starting point for autonomously acting vehicles allows the formulation of requirements that can be generalized and applied to other applications. Similarly, the idea of autonomous market participants opens up a series of further conceptual questions that need to be considered more closely from the point of view of similar applications.

Finally, as a sociotechnical concept, the M2X economy also raises sociotechnical questions besides the technical and conceptual implications, (dis)advantages and limitations. Especially the continuous automation of tasks and jobs that were previously performed by humans is a controversial discussion due to resulting layoffs. However, automation of existing tedious work in combination with a restructured economic value system may foster the development of a society that benefits from a progressing automation.

Chapter 4

Validation, Authentication and Identities in Decentralized Networks

In Chapter 3 we outline scenarios for the M2X economy, described the interaction and transaction platform based on the V2X running cases as well as the corresponding system engagement processes, discussed the modalities of the M2X economy and finally focused on the decentralized governance infrastructure and conflict resolution mechanisms. All topics discussed in that chapter share the need for a proper identity solution. Especially, in the context of hardware devices, humans and software agents that/who all require a digital representation of their "real-world" identity to conduct digital business transaction, or enact digital collaborations – a digital representation mapping to the analogue identity is necessary. The concept is often referred to as "digital twins" in the context of IoT applications [171]. In order to enable secure business collaborations, interactions and transactions within a digital economy a digital representation is required to establish and enable trust, reputation mechanisms, perform verifiable and accountable transactions and establish reliable as well as auditable data provenance.

Identity management in the M2X ecosystem is a multi-stakeholder issue that involves not only its users, but also OEMs, infrastructure providers, regulators and various service providers. A single central authority that governs the identity management for all stakeholders is unlikely and poses the risk of a single point of failure. Moreover, identity data silos raise privacy concerns and suffer from interoperability issues, i.e., lock-in effects. Moreover, as previously argued in Chapter 3, a centralized infrastructure and architecture that powers the M2X economy is neither desirable nor facilitating the full potential of the ecosystem. In addition, an identity infrastructure that relies on a centralized certificate authority (CA) is not an option either – especially given the underlying security issues and implications, e.g., [172][173][174]. Thus, a centralized identity solution is not an option and a decentralized and interoperable solution that fosters an open M2X ecosystem is required. However, decentralized identity-, authentication- and validation systems and mechanisms pose challenges and issues too. These have to be solved accordingly before providing a reliable foundation. Therefore, this chapter addresses the identified knowledge gap by answering the following research questions:

RQ-2: How to identify, authenticate and validate entities in a decentralized M2X ecosystem?

- RQ-2.1: What are the challenges and issues of validation and authentication in decentralized networks?
- RQ-2.2: What are the specifics of a protocol for validation and authentication in decentralized networks?
- RQ-2.3: What are the security guarantees and implications of binding an identity to a blockchain system?

The chapter is structured as follows: Section 4.1 discusses issues and challenges of decentralized identity, authentication and validation mechanism in decentralized networks. Afterwards, Section 4.2 introduces a blockchain-based protocol for validation and authentication that is applicable to the M2X ecosystem. Next, Section 4.3 proposes a mechanism to de-incentivize sybil node attacks and provide an exact price tag on the costs of a sybil node attack, thereby providing a novel mechanism to quantify security guarantees of a decentralized identity solution within decentralized networks. Finally, Section 4.4 discusses the findings of this chapter.

Please note, the subsequent sections are based on the following publications of the author in collaboration with different co-authors: [68][175][176][177].

4.1 Challenges and Issues

Generally, digital user identity, validation and authentication methods are based on an asymmetric key concept where the key material represents the digital identity of an entity. Most of these systems rely on a centralized database, or a controlling entity to manage and store these keys. However, such “information storage presents a single point of compromise from a security perspective. If this system is compromised it poses a direct threat to users digital identities” [178]. A further disadvantage of centralized data silos and identity management is a lack of interoperability among those data silos that results in a favorable lock-in effect from the provider perspective and several disadvantages from the end-user perspective.

Public key infrastructures (PKIs) are not only the most common system for distributing and managing public keys, but also for ensuring a correct association between a public key and its owner. The hierarchical trust model for certificate authentication (commonly used by CAs and web browsers) relies on hierarchically structured central authorities [179], whereas the PGP Web of Trust (WoT) uses a decentralized approach [180]. Instead of relying on central CAs, each user acts as an authority itself and ensures a number of bindings between users and their public keys. In recent years, several security incidents have proven that CAs are vulnerable due to their centralized structure [174][181][182]. On the other hand, the decentralized PGP WoT does not provide sufficient certainty that the information stated in a public key is correct since users do not carefully verify other users (missing incentives). In addition, it is trivial for malicious users to generate large numbers of key pairs and create structures that look like carefully verified keys without much effort.

The centralized system of CAs and its decentralized counterpart the PGP WoT are the two most commonly used types of PKIs. Both have different advantages and disadvantages which are described in the following sections. In addition, a new generation of blockchain-based and DID-inspired identity solutions emerged in recent times.

4.1.1 Certificate Authorities

CAs are institutions or organizations inside a network that are treated as trustworthy by definition. They can sign individuals, organizations or another CA's certificates. Users who decide to trust a certain CA (and their decisions), also trust all individuals signed by this CA. The result is a tree-like, hierarchical structure with the initial CA (Root-CA) at the top of the system. This tree-like structure is also one of the major weaknesses of CAs since it introduces a single point of failure. The whole trust-system collapses as soon as a Root-CA gets compromised, or untrustworthy for any reason. There have been several security incidents involving CAs in recent years. In 2011, an attacker issued certificates for domains of large IT-companies such as Google, Yahoo and others using an access to DigiNotar's (a Dutch CA) systems [174]. As a consequence, DigiNotar's root certificate was removed from most browsers and the company went bankrupt. Another incident involved Trustwave Holdings which operates a CA and issued a subordinate root certificate to a customer which enabled the customer to issue certificates on its own. The customer's identity was never revealed due to a Non-Disclosure Agreement [181].

Another certificate incident of 2011 involved the CA Comodo [182], where a compromised reseller account was used to issue arbitrary certificates.

Similar security problems might also occur in case a national authority forces a CA to cooperate and grant access to the CA's root certificates for surveillance reasons.

4.1.2 Web of Trust

Instead of relying on a central authority, each PGP WoT user acts as an authority itself and ensures a number of bindings between users and their public keys. A successful verification of a public key results in a unidirectional signature between the public key of the verifier (Alice) and the verified user's (Bob) key. Such a signature is interpreted as a trust relation; Alice successfully verified the authenticity of Bob's public key and therefore Alice decides to trust Bob's public key. Users may decide to trust a key if it is signed by somebody they trust, or if there exists a chain of trusted signatures from their key to the target key. Typically, PGP's WoT regards the following as criteria for trust: *i.*) Number of signatures on the subject under investigation. *ii.*) Centrality of a node in the entire set of subjects. *iii.*) Timeline with information when signatures have been made – typically, signatures should appear during a longer period. *iv.*) Number of asymmetric trust relations within the WoT.

In contrast to the centralized CA system, the PGP WoT has no central point of failure. Nevertheless, it suffers from several other downsides such as missing incentives for key verification and a lack of punishments in order to motivate its users to adhere to the verification rules and contribute to the well being of the system. Furthermore, it is trivial for malicious users to generate large numbers of keys and connect them in such a way, that the resulting network looks like a group of trustful users. Finally, as shown in a previous study [183], about 40% of the PGP WoT's email addresses are dead (not reachable) which raises questions regarding the trustworthiness of signatures related to these unreachable email addresses.

A decentralized alternative to the centralized PKI systems are so-called decentralized PKIs (DPKIs) [184][185][186]. In many DPKIs, a blockchain replaces the centralized key management and distribution of identity keys in form of a key-value storage. DPKIs aim to reduce the risk of centralized single-point-of-failure-systems by making data more tamper-resistant, preventing MITM attacks and minimizing the control of untrusted third parties over the system. The Certcoin project [185][187] is such a DPKI system that uses "the consistency guarantees provided by cryptocurrencies such as Bitcoin and Namecoin [188] to build a PKI that ensures identity retention" [185]. Certcoin does not require a central authority and uses the Bitcoin blockchain and the resulting advantages (decentralized, difficult to manipulate, distributed replication, fault tolerance, redundancy, transparency, etc.). The protocol provides methods for public key registration, update, revocation, recovery, verification and lookup. Certcoin has been implemented [1], but not been used in practical scenarios. Moreover, further development ceased to exist.

Finally, identity provision in decentralized networks is prone to sybil attacks. This type of attack is a common issue in large-scale P2P systems, where hostile, or faulty computing elements threaten the security of the whole network. Single malicious entities may be able to present multiple identities, thereby controlling a substantial fraction of the system, consequently undermining its functionality and security [50]. Several techniques focus on preventing sybil nodes from joining a network at all [189][190]. Other approaches attempt to detect them while they are already part of the network [191][192]. One of the key enablers of sybil attacks is the absence of a mechanism that prevents attackers from setting up arbitrary numbers of (virtual) nodes.

4.1.3 Self-Sovereign Identities

Taking the idea of blockchain-based DPKIs in combination with the concept of decentralized identifiers (DIDs), as introduced in Section 2.3, enables a user-centric, self-sovereign and interoperable identity ecosystem that aims to prevent most of the aforementioned security flaws. "Self-sovereign identity puts end-users not the organizations that traditionally centralize identity in charge of decisions about their own privacy and disclosure of their personal information and credentials" [178]. Similarly to some DPKI solutions, self-sovereign identity systems that are based on DIDs utilize distributed ledgers – or blockchains – as distributed storage system which replace centralized and incompatible data silos with a cooperative shared storage resource. The result is a user-controlled identity provision model where users control access and sharing of their data based on a need-to-know-basis using the concepts of DIDs, DID documents and verifiable claims.

In recent times, academic as well as business projects introduced and developed first prototypes of self-sovereign identity solutions. Among them the Horcrux protocol [178], Sovrin [193], uPort [194], SelfKey [195], Blockstack [196] and others [197].

While DID-based self-sovereign identity solutions are not specifically focused on humans, most existing solutions rely on them as a main use case. However, as demonstrated in the following sections, the concept is also applicable to the general M2X ecosystem. We

[1] https://github.com/cfromknecht/certcoin

introduce a DID- and blockchain-based identity solution for validation and authentication in decentralized networks that aims to prevent many of the downsides of centralized CAs, the decentralized PGP WoT.

4.2 Authentication and Validation in Decentralized Networks

The combination of DIDs, DID documents and verifiable claims allows to establish decentralized and distributed networks for identity provision as well as authentication and validation in decentralized networks. While DIDs globally and uniquely identify entities, DID documents and verifiable claims describe the entity and provide mechanisms to the DID subject to authenticate itself and prove its association with the DID.

The following section introduces Authcoin, a challenge-response based protocol for authentication and validation in decentralized networks that is able to: *i*) prove control over an asymmetric key pair (validation) linked to a DID document and *ii*) to produce verifiable claims that can be used to authenticate an entity. By documenting the communication process of the bidirectional validation and authentication mechanism on a blockchain system, a transparent and auditable as well as tamper-resistant log is created that makes it difficult for adversaries to introduce malicious identities/keys into a network. While combining Authcoin and the concept of DIDs is most suitable in the context of its application in the M2X ecosystem, the protocol itself is more abstract and only relies on the concept of public and private keys. Hence, in subsequent sections, we spare the association of DIDs, DID documents and corresponding key pairs to ease the illustration and instead only refer to the asymmetric key pairs. The same applies to all formal models of the protocol in later sections. Alternatively to DIDs, describing entities as in PGP [198] is also compatible with Authcoin. Finally, while this work specifically focuses on the M2X economy and its entities, Authcoin is also applicable in scenarios exclusively focusing on interactions among humans.

The following sections provide a general overview on Authcoin itself, its challenge-response-mechanism and further important concepts. The general Authcoin workflow as illustrated in Figure 4.1 is described step-by-step. First, Section 4.2.1 discusses the generation of a new key pair and its DID and DID document association. Afterwards, the formal key validation as well as validation and authentication are introduced in Section 4.2.2- 4.2.3. Next, Section 4.2.4 and Section 4.2.5 deal with key revocation, expiration and recovery. More details on the design of appropriate challenges and an overview on Authcoin's validation and authentication requests (VARs) are provided in Section 4.2.6. Finally, we discuss the underlying data storage layer in Section 4.2.7.

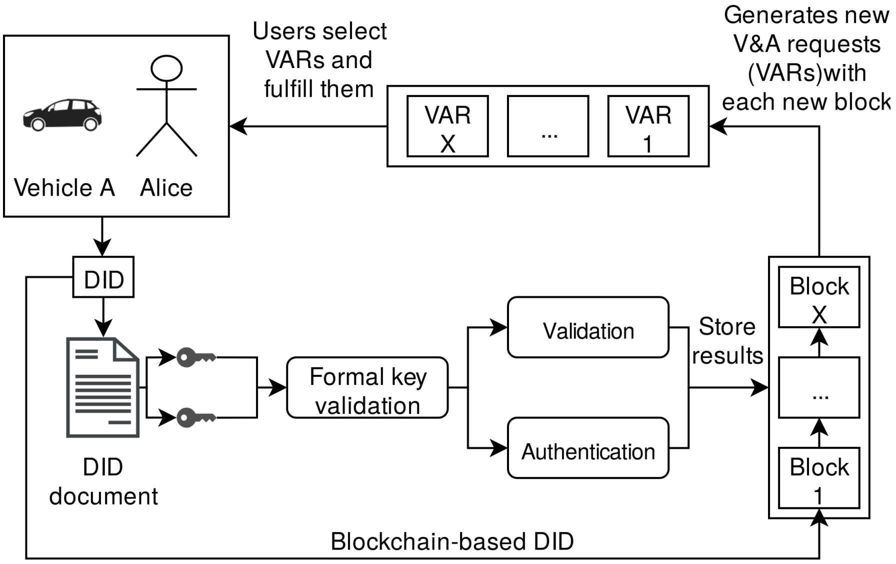


FIGURE 4.1: General Overview on Authcoin's Workflow – Based on [68]

4.2.1 Establishing an Initial Key Binding

The first step, according to Figure 4.1, is to create a new asymmetric key pair. An initial binding between the generated key pair and its owner is established by associating the key with an identifier. In the context of DIDs, the public key of the key pair is associated with a DID document. DID documents associated with key pairs and a DID may belong to a human, machine, or software agent entity. In the context of a PGP system for human use, basic information such email, or the user's name are associated with the key pair.

All accumulated information are collected and stored as a transaction on the blockchain as illustrated in Figure 4.1 and described later in Section 4.2.7.

4.2.2 Validation

Before the actual validation and authentication (V&A), each involved public key is automatically checked for formal validity. The protocol validates the following properties: Is the key well-formed (syntax and data format)? Is the key length sufficient? Is the key still valid or already expired? Has the key been revoked? In case all involved keys pass the formal validation, the actual V&A process starts. If necessary, the mentioned example properties can be extended.

Authcoin's general validation process is similar to the domain validation process deployed by the automated CA Let's Encrypt [199] as illustrated in Figure 4.2. In the context of Let's Encrypt, the domain owner runs the Let's Encrypt-Client on the domains machine, afterwards the client contacts the Let's Encrypt-Server (LES) and asks for a challenge.

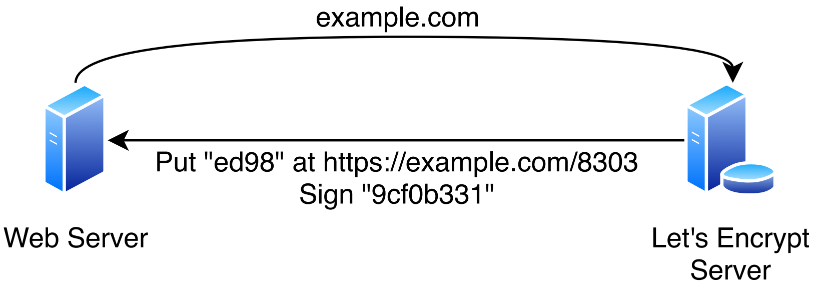


FIGURE 4.2: Domain Validation Process as Deployed by Let's Encrypt – Based on [199]

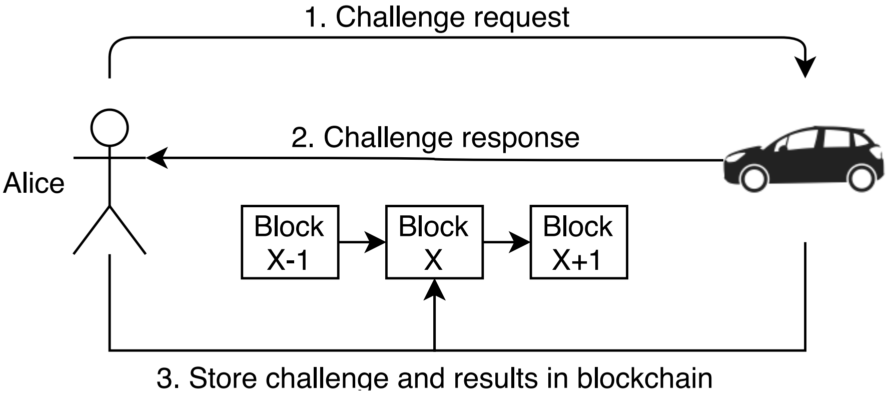


FIGURE 4.3: General Validation and Authentication Process as Deployed by Authcoin – Based on [68]

Usually, the challenge is to provide a certain resource under a specific URI and sign it with the private key that corresponds to the public key which is validated. In Figure 4.2, the client is asked to provide the resource "ed98" at `https://example.com/8303` and sign it with the private key. The client software fulfills the challenge as requested by the LES, which checks if the challenge's outcome is satisfying. In case the validation succeeds, the domain owner has proven that he/she has access to the domain (domain validation), has access to the public and private key (key validation) and that the certificate (key pair) corresponds to the tested domain.

In the context of Authcoin, a similar validation process is performed as presented in Figure 4.3. Alice needs to verify the other parties ownership of the public key (validation process). To do so, Alice sends a challenge ("this is a challenge") encrypted with vehicle's public key to the vehicle which is asked to fulfill the challenge by signing the response with its private key and send it back to Alice. Alice checks the results and can deduce (in case that the process finished successfully) the following three facts from the challenge and response: *i.*) The vehicles has access to the communication account used for the validation (account validation). *ii.*) The vehicle has access to the public and private key (key validation). *iii.*) The key pair corresponds to the tested communication account (binding). The validation requests and results of the validation processes are stored as part of the blockchain. Both, Alice and the vehicle, independently post the challenge and response to the blockchain.

It is important to keep in mind that in the example above the identity of the owner is not verified, only that the corresponding entities have access to the specific key pairs. The authentication process is addressed later in Section 4.2.3. An important difference between the domain validation process described earlier and the Authcoin validation example, is that the later process is performed in both directions (bidirectional). Alice sends a challenge to the vehicle and receives (hopefully) a matching response. In return the vehicle does the same when receiving Alice's challenge in order to also verify Alice's public key. As a result, it is more difficult for malicious users to introduce fake keys into the system and maintain introduced malicious keys. A major advantage of Authcoin's bidirectional validation procedure is that it can be performed in an automated manner and even on a large scale. In consequence, keys are validated on a regular basis resulting in an improved overall security of the network.

4.2.3 Authentication

After the successful key validation, the problem of authentication is addressed. Similar to the validation procedures described before, Authcoin relies on a challenge-response-mechanism for authentication. Involved parties may either rely on a direct exchange of data that allows to authenticate the other party, or using (third-party) verifiable claims as described earlier in Section 2.3.2.

In the context of the M2X ecosystem, machines, devices and agents may prove their identity by reproducing a cryptographically secured proof signed by the device's manufacturer or an eligible party. A proof, e.g., a verifiable claim, concerning a vehicle's VIN with a signature of the manufacturer or certified car mechanics could be used for vehicles. Often, authentication of smart devices, infrastructure components and software agents is reduced to reproducing and exchanging (in a challenge-response manner) a verifiable claim issued by the claim issuer. Which entity is sufficiently trustworthy to be accepted to issue verifiable claims depends on the context of the specific M2X use case. For vehicles, a verifiable claim might be a digital proof representing the vehicle's VIN. Similar approaches may apply to other machines and hardware components as well as software agents. For security-critical applications, proofs and identities may reside within TPDs.

It is important to note that for the V&A mechanisms discussed above a signature between two keys is only created in case that the validation process (validation signature), or authentication process (authentication signature) is successful. A failed validation or authentication is also documented on the blockchain. In addition, Authcoin requires validations and authentications to be performed in both directions (bidirectional) instead of unidirectional. Bidirectional V&A results in more frequent examinations of each key, identity, or certificate and makes it more difficult for malicious entities to stay undetected. Another advantage of Authcoin's V&A approach is a lower threshold for users to participate in V&A since it is not necessarily required to meet the verification target in person even though this is still possible and just a different type of challenge. Finally, automating the validation process provides each user with the unique possibility to validate all existing entities of the Authcoin-system on its own without relying on any transitive relations. Nevertheless, it is possible for authentications if desired by the user.

As already demonstrated through the examples above, Authcoin is not fixed to a specific type of challenge. Instead it is meant to be as flexible and extensible as possible and utilize a flexible challenge-response concept. Therefore, the results of future research on challenges can easily be integrated in Authcoin, especially new challenges which are more secure, more reliable and harder to manipulate.

The chosen challenges influence the overall security and reliability of the system. As a result, adapting the requirements for the deployed challenges leads to a different level of provided security. In some scenarios, deploying only validation mechanisms might be sufficient for a given purpose. In other scenarios, it might be necessary to combine different challenges based on different identifiers in order to provide a maximum level of security and reliability. Many other scenarios lie in between these two extremes.

A further security improvement utilizes biometric identifiers – only for the human entities of the M2X ecosystem – which are more difficult to fake. Commonly used biometric identifiers are fingerprints, eyes (retina or iris recognition), voice, face (facial recognition systems), or DNA. Biometric identifiers can either be used to derive a new key pair from the identifier [200], or are included in the new key pair in addition to the traditional identifiers. Adding biometric identifier to the key pair establishes additional bindings between the key owner and the key pair. However, biometric identifiers have disadvantages of their own and are only applicable to a subset of all M2X ecosystem entities.

4.2.4 Revocation and Expiration

Currently, Authcoin's key revocation is handled as known from PGP [201]. A key is revoked by posting a key revocation certificate to the blockchain. Future versions of Authcoin might extend the protocol with a more sophisticated approach using a combination of offline and online key pairs, where the offline key pair can be used to revoke, update or replace the online key pair. A signature is revoked by adding a signature revocation certificate to the blockchain. Revoking a signature expresses a total loss of trust in the signed key.

Both, keys and signatures, have an expiration date. For security reasons, the lifetime of key pairs used with Authcoin is limited to a maximum of 12 months. Afterwards, a new key pair has to be created, but users can also decide to use shorter lifespans. Signatures either expire after a user-defined timespan (max. 12 months), or when the signing key or the key which got signed expire. An expired key cannot longer be used for V&A in the context of Authcoin. Using the key outside Authcoin is still possible, even though it is not recommended.

4.2.5 Key Recovery

Thus far, Authcoin does not support any key recovery mechanisms. Therefore, a lost private key cannot be recovered. An alternative approach is utilized by Certcoin, which deploys a shared secret solution [202][203]. A user's private key is shared among a number of trusted entities and at least two of them are required to restore the secret key [185][187]. An advantage of this solution is the availability of a key recovery mechanism, but it comes with the downside of handling additional keys and the requirement of sufficient trusted persons. Furthermore, for non-technical users, the concept of public-key-cryptography alone is complicated enough; adding the concept of shared secrets demands too much from non-security experts.

4.2.6 Automated Validation and Authentication Requests

Decentralized and distributed networks lack central authentication authorities, thereby leaving them easy targets for sybil attacks. Authcoin implements several restrictions and mitigation concepts in order to prevent malicious users from harming the system and identifying them as soon as possible. Similar to the PGP WoT and comparable solutions, Authcoin cannot prevent the participation of sybil nodes. Nevertheless, due to the adaptable challenge-response mechanism and the transparent and auditable storage of information, it is much more difficult for an adversary to keep malicious nodes undetected. Moreover, Section 4.3 details a concept of de-incentivizing sybil node attacks and estimating a clear price per sybil node identity to derive security assumptions regarding an identity of the network.

The first line of defense is the challenge-response mechanism which can – if used correctly – detect and identify sybil nodes since they might not be able to pass the proposed

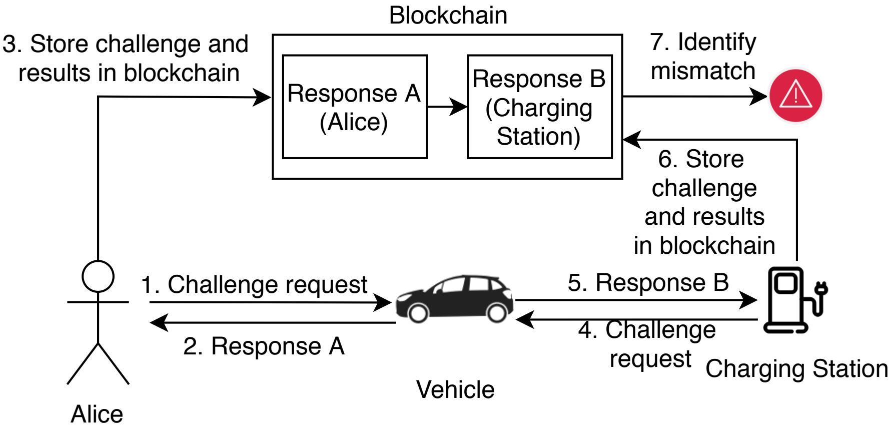


FIGURE 4.4: Identify Mismatch on the Blockchain – Based on [68]

challenges successfully. Identifying a V&A mismatch is illustrated in Figure 4.4. Designing challenges as tamper-resistant and secure as possible makes it more difficult for sybil nodes to stay undetected. Deploying a mandatory bidirectional authentication process is an additional burden for malicious users. It might still be possible to create a collective of sybil nodes with signatures between the participating entities, but as soon as nodes outside this collective interact with the collective the probability of exposure increases. As a result of limiting keys lifespan to a maximum of 12 months, maintaining such sybil collectives is also time-consuming. In the PGP WoT, it is trivial to create an arbitrary number of keys with unlimited life span, connect them among each other and subsequently create outgoing signatures to legitimate nodes and also might receive some signatures from unreliable verifiers, which finally results in a permanent incorporation of the sybil collective in the "web of trust". Moreover, the transparent nature of Authcoin makes it also easier to identify unreliable verifiers who do not take the V&A process seriously and identify them (and their actions) as not trustworthy.

Another approach for detecting and mitigating malicious nodes are validation and authentication requests (VARs) which are automatically and randomly created during the mining process as illustrated in Figure 4.1. The number of generated VARs depends on the number of existing and still valid (not expired, revoked, etc.) keys in the system. An automated VAR expresses the desire of the system to validate and/or authenticate a randomly chosen entity inside the system. VARs are publicly stored as part of the blockchain and can be fulfilled by Authcoin's users. Deploying such an automated request mechanism results in several benefits compared to existing solutions: Firstly, the approach of automated VARs makes it easier to break into sybil collectives "by accident" and expose them as such in case they fail the validation, or authentication process. Identifying one sybil node leads to questioning all other nodes claiming to have successfully validated and/or authenticated the sybil node, therefore identifying these nodes either also as sybil nodes or at least as unreliable verifiers. Due to VARs and the bidirectionality of authentications, it is also possible to increase the number of V&As for each key, resulting in higher probabilities of detecting malicious users.

In order to make it more difficult to misuse the VAR-mechanism, we introduced some restrictions: VARs cannot be issued by users manually, instead they are generated automatically once a new block is added to the blockchain. It prevents malicious users from creating any desired number of requests on their own and fulfill them afterwards with keys under their control. In order to avoid similar tactics for the automatically generated VARs, a hash-based selection algorithm decides whether a specific user is eligible to fulfill the VAR. The selection algorithm simply calculates the hash of the concatenation of the VAR and the potential verifier's key. If the binary presentation of the result starts with a certain combination (e.g. a 1 or a 0; 10 or 11, etc.) the user is allowed to fulfill the VAR. This decision seems to be random, but that is exactly the intention. The algorithm is used to increase the effort for malicious users to fulfill VARs with other keys under their control. Of course this approach may be undermined, but it increases an attackers cost of not getting exposed. Besides limitations through the selection algorithm, it is also necessary that the key used to fulfill the VAR itself was created before the VAR in order to avoid that attackers create keys to fulfill the VAR after they discovered it.

In future versions of Authcoin, the VAR mechanism might be combined with an incentive system in order to encourage users to fulfill VARs on a regular base (in case that is even necessary; besides that, fulfilling validations does not require the user's interaction at all and can be performed automatically in the background). The incentive system might be part of an overall trust metric concept which not only includes the results of V&As, but also rewards behavior that benefits the system such as fulfilling VARs.

4.2.7 Storing Information

Authcoin utilizes a blockchain-based transaction database as an underlying storage system which is used to keep track of keys, challenges, responses, signatures and all other relevant information. Blockchain-based storage systems provide several desirable properties such as: decentralization (no trusted central authority), distribution of data, fault tolerance, transparency and redundancy. Furthermore, it is not possible to manipulate the blockchain as long as the majority of its users decide to do so. Finally, the design goals of DIDs – as outlined in Table 2.1 – which are enabled by a blockchain-based platform also apply to our identity solution as presented above. For the ease of explanation, this paper assumes that Authcoin has its own, independent blockchain. But it is also possible to utilize existing blockchains for this purpose. Alternative projects such as Ethereum [70] maintain their own, independent and customized chain. Moreover, Authcoin does not rely on a specific consensus algorithm such as PoW or PoS and instead only requires a block creation mechanism.

4.2.8 Protocol Formalization

Preventing design flaws, security and privacy issues as well as incomplete specifications that pose a risk, is a challenging task during the design and development of new security protocols in the field of computer science [204][205][206][207]. In a best-case scenario, issues of a security protocol are inconvenient to users who rely on it, while in other cases, design flaws and errors are fatal. The broken encryption of a wireless network [208] is an example for the first case, whereas a broken security protocol that grants an attacker access to sensible parts of nuclear power plants [209] illustrates a more serious threat.

Formal methods, such as Petri nets [55], π-calculus [210] and communicating sequential processes [211], are utilized for the design, development and analysis of new as well as existing protocols, thereby eliminating, or minimizing the security issues of the targeted protocols [212][213]. Most recently, a formal security analysis on the underlying protocol of the popular Signal Messenger[23] was conducted by Cohn-Gordon et al. [215], resulting in the demonstration of several standard security properties provided by the protocol.

In the following section, a formal model of the Authcoin protocol is created using Colored Petri Nets (CPNs) [56][57] in order to detect and eliminate eventual design flaws, missing specifications as well as security and privacy issues [216]. CPN is a graphical oriented language for the design, specification, simulation as well as the verification of systems and describes the states of a modeled system and the events (transitions) that cause the system to change states. CPN models are represented using a directed bipartite graph that consists of places, transitions, arcs and tokens. Places are denoted as circles and transitions as rectangles. Arcs connect places with transitions, or transitions with places and have inscriptions in CPN-ML expressions [56][57][217][218][219]. CPN ML is an expression programming language for inscriptions which are used to further specify data types and operations of the modeled system. Tokens and their colors represent different data types. The resulting CPN model “of a system describes the states of the system and events (transitions) that can cause the system to change state. By making simulations of the CPN model, it is possible to investigate different scenarios and explore behaviors of the system” [56].

[2] https://whispersystems.org/

[3] The same protocol was later adapted by WhatsApp [214]

Besides it general suitability for system formalization, CPNs are especially well-suited to be applied in the context of blockchain-based systems. Blockchains are discrete state machines where the most recent block represents the current state of the system. With each new block the systems' state transitions to a successor state. Similarly, CPN models represent discrete state machines and change states via transitions. While in CPN data structures are represented in form of colored tokens, many blockchains use tokens for the same reason. Moreover, blockchain transactions can be easily mapped to colored token data structures. CPN uses CPN-ML expressions to specify and implement data types and operations of the modeled system which maps to the functionalities of smart contracts in the context of blockchain technology. Finally, the hierarchical structure of CPN models can be used to formalize dApp components of interleaved smart contracts. Therefore, CPN is well-suited as a formalism of choice for blockchain systems.

In the following, CPN-Tools[4] is used to design, evaluate and verify the CPN models. The result is a formal specification of the protocol that is used to guide further implementation efforts.

4.2.8.1 Modeling Strategy

In order to model the Authcoin protocol using CPN, an appropriate modeling strategy is required, mapping the existing descriptions of the protocol to the corresponding elements of a CPN model. Mapping the informal descriptions and requirements of the Authcoin protocol to a formal model using CPNs, results in a sound model. Based on this model it is possible to consider concurrency conflicts, dependability issues and detect and eliminate eventual design flaws as well as security and privacy issues. To do so, we first outline the modeling strategy used to create the CPN models before presenting the resulting CPN models.

Authcoin organizes and defines the exchange of information between different entities that are modeled as agents. In software engineering, various agent-oriented approaches exist, such as: Tropos [220], Gaia [221], Prometheus [222], MASB [223][224] and MaSE [225]. In [226], Mahunnah et al. introduce a mapping heuristics from agent models to CPN models based on Sterling's and Taveter's [58] methodology for Agent-Oriented Modeling (AOM).

Similarly to Section 3.1.1, we utilize AOM to model the functional and non-functional requirements. However, in case of Authcoin there are two types of AOM models that are necessary to represent the protocol, i.e., the goal model itself and the behavioral model. Again, the AOM goal model is used to capture the functional requirements of the system

[4] http://cpntools.org/

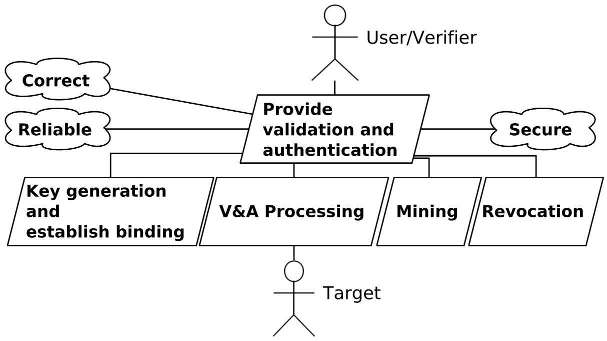

User/Verifier
Correct
Reliable
Provide validation and authentication
Secure
Key generation and establish binding
V&A Processing
Mining
Revocation
Target

FIGURE 4.5: Top-Level AOM Goal Model of Authcoin – Based on [175] and [176]

in the form of goals, as well as non-functional requirements and roles of involved entities. Non-functional requirements represent quality goals of the system [58].

Figure 4.5 illustrates the top level goal model of Authcoin. As previously in Chapter 3, a goal is represented in form of a parallelogram, quality goals in the form of clouds and sticky men represent roles. Functional requirements of the goal model are structured in a tree-like hierarchy with the overall objective of the system at the top. The main objective of Authcoin is to provide a secure and reliable validation and authentication protocol. The main goal is further decomposed into multi-layered sub-goals until the lowest atomic sub-goal is reached. In the context of Authcoin, the main goal is further divided into the following sub-goals: Key generation and establishing a binding, validation and authentication processing, mining and revocations. The three quality goals *secure*, *correct* and *reliable* are attached to the overall main goal of the goal model, meaning they are relevant to all sub-goals and are inherited.

The AOM behavior model refines the previously developed goal model for specific agents and activities. A behavior model in AOM has two parts: An agent behavior model coupled with a behavior interface model [58]. The former describes the rule-based behavior of an agent, while the latter focuses on identifying activities with associated triggers, preconditions and post-conditions [226]. Table 4.1 presents the behavior interface model of the goals depicted in the top level goal model of Figure 4.5. Each activity is listed with its corresponding trigger, optional pre-conditions and its post-conditions. The execution of an activity is either triggered by an event, or by a pre-condition after the occurrence of an event [226]. The *Mining*-activity in Table 4.1 is triggered by receiving input transactions for the next block of the blockchain - to do so, the precondition has to be fulfilled. After the activity's execution, the proposed input transactions are either available on the blockchain or an error occurred and triggered a failure message.

Complete and detailed listings of all remaining goal models and behavior interface models are available in Appendix A.1.2 and Appendix A.1.3. All acronyms, names and abbreviations as well as a description of the token colors are available in Appendix A.1.4.

Activity	Trigger	Pre-Condition	Post-Condition
Key generation and establish binding	User wants to create a new key pair	Identifier list, key expiration date, key type, key length	Key pair, EIR on blockchain, EIR
V&A Processing	Received EIRs for V&A	Verifier EIR, target EIR	V&A results on blockchain or failure message
Mining	Received input for blockchain	Input transactions	CR, RR and SR on blockchain and VARs or failure message
Revocation	User wants to revoke an EIR or a SR	KeyPair, EIR, SR, CR, RR, VARs	Revoked EIR or SR and updated information on blockchain

TABLE 4.1: Exemplary Behavioral Interfaces of Activities for Authcoin – Source: [175] and [176]

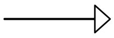
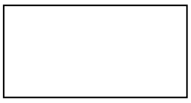
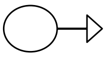
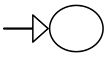
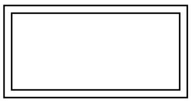
Notation	Name
	Connecting Arc
	Sub-goal or Activity
	Trigger or Precondition
	Postcondition
	Goal

TABLE 4.2: Notation Mapping CPN to AOM – Based on [226])

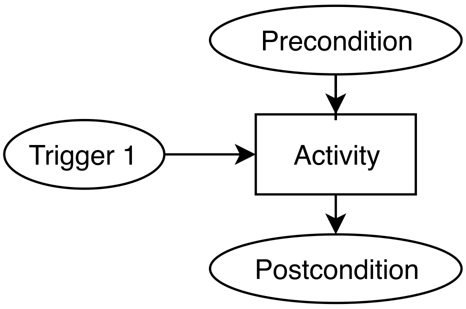


FIGURE 4.6: Mapping a Behavior Interface Model to a CPN Model – Based on [175] and [176]

4.2.8.2 Mapping AOM Models to CPN Models

Mapping the created AOM models to CPN is the final step to derive the top level CPN model of Authcoin. Table 4.2 and Figure 4.6 illustrate the mapping heuristic of AOM goal models to CPN models as well as the mapping of behavior interface models to CPN models.

Directed arcs connect places and transitions representing the protocol execution through activities. Transitions represent activities, or sub-goals and CPN modules, depicted in the form of double-boarded rectangles, illustrate goals derived from the goal model. We refine the CPN modules into smaller sub-elements of the overall model, mapping to

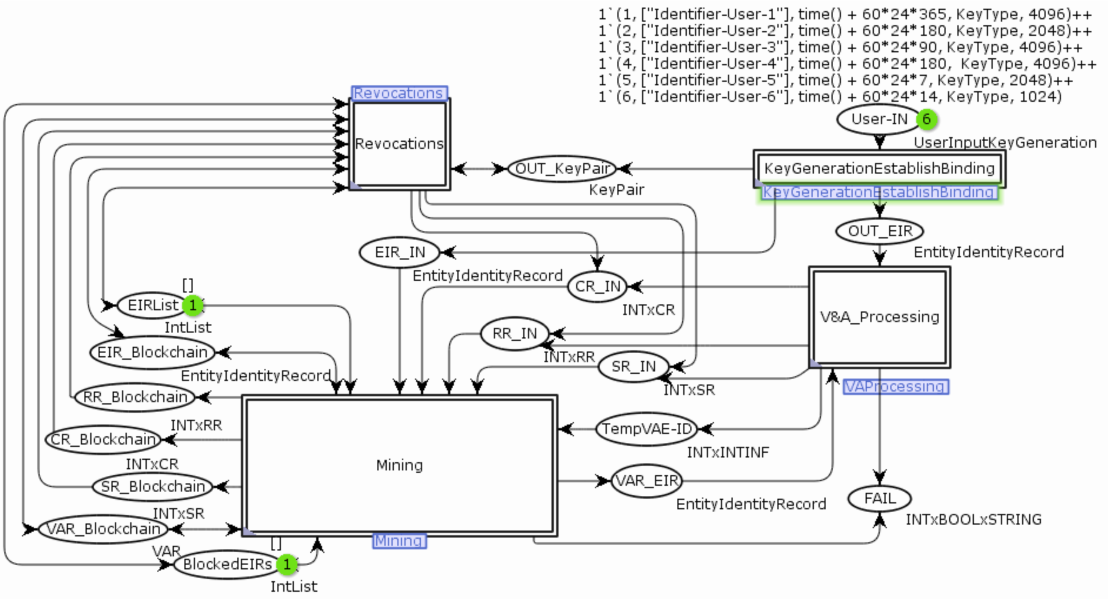

1`(1, ["Identifier-User-1"], time() + 60*24*365, KeyType, 4096)++
1`(2, ["Identifier-User-2"], time() + 60*24*180, KeyType, 2048)++
1`(3, ["Identifier-User-3"], time() + 60*24*90, KeyType, 4096)++
1`(4, ["Identifier-User-4"], time() + 60*24*180, KeyType, 4096)++
1`(5, ["Identifier-User-5"], time() + 60*24*7, KeyType, 2048)++
1`(6, ["Identifier-User-6"], time() + 60*24*14, KeyType, 1024)
Revocations
Revocations
User-IN
6
UserInputKeyGeneration
OUT_KeyPair
KeyPair
KeyGenerationEstablishBinding
KeyGenerationEstablishBinding
OUT_EIR
EntityIdentityRecord
EIR_IN
EntityIdentityRecord
CR_IN
INTxCR
[]
EIRList
1
IntList
V&A_Processing
VAProcessing
RR_IN
INTxRR
SR_IN
INTxSR
EIR_Blockchain
EntityIdentityRecord
RR_Blockchain
INTxRR
CR_Blockchain
INTxCR
SR_Blockchain
INTxSR
VAR_Blockchain
VAR
Mining
Mining
TempVAE-ID
INTxINTINF
VAR_EIR
EntityIdentityRecord
FAIL
INTxBOOLxSTRING
[]
BlockedEIRs
1
IntList

Figure 4.7: Authcoin Top-Level CPN Model – Source: [175] and [176]

the same relation between goals and sub-goals as in AOM. As illustrated in Figure 4.6, places with outgoing arcs either act as triggers, or represent a precondition, whereas places with incoming arcs represent post-conditions of a given activity in AOM [226].

The final result of the mapping process is illustrated in Figure 4.7, presenting the complete and formalized top level CPN model of Authcoin derived from AOM and implemented using CPN-Tools. The top level CPN model consists of the four sub-modules derived from the four sub-goals of the top level AOM goal model. A detailed explanation of Figure 4.7, as well as further depictions and the refined implementation of these sub-modules are available in Section 4.2.8.4. The CPN token color sets, names and abbreviations used in the model are introduced in the following section.

4.2.8.3 Protocol Semantics

Token colors represent data structures of Authcoin data objects that are used to illustrate the data flow through the CPN model. Tokens are transferred and manipulated by CPN transitions and ensure that the data objects adhere to the specified data syntax. Exemplary some acronyms, names and description of token colors of Authcoin's top level module are presented in Table 4.3. The first column specifies the module of the first occurrence of a certain token, name or acronym. The second column specifies the name, followed by a short description in column three. The last column provides information concerning the data types. A complete listing of all acronyms, names and abbreviations as well as a description of the token colors of Authcoin CPN model is available in Appendix A.1.4.

Module	**Token color**	**Description**	**Type**
Top Level	PublicKey	Public key	(KeyFingerprint, Key, ExpirationDateUTC, KeyType, KeyLength)
Top Level	PrivateKey	Private key	String
Top Level	ChallengeRecord	Contains all information of a V&A challenge	(CR_ID, VAE_ID, Timestamp, ChallengeType, Challenge, VerifierEIR_ID, VerificationTargetEIR_ID)
Top Level	ResponseRecord	Contains all information regarding a V&A response	(RR_ID, VAE_ID, Timestamp, CorrespondCR_ID, Response)

Table 4.3: Exemplary Acronyms, Names and Description of Token Colors of Authcoin's Top Level Module – Source: [175]

4.2.8.4 Refined CPN Models

We presents the further refined sub-modules of the top level CPN model presented in Figure 4.7. The sub-module refinements are derived from the AOM model using the same process as outlined for the top-level CPN model in Section 4.2.8.1. For the ease of illustration, only the first level of module refinements is presented and explained in detail. The top level CPN model in Figure 4.7 consists of four sub-modules. Each of the following paragraphs focuses on one of these modules and provides detailed explanations.

For further information on the remaining refined sub-modules and additional explanations, we refer the reader to [175], [176] and [227]. In addition, the CPN-Tool source file of the presented CPN model is available in Appendix A.1.1.

***KeyGenerationEstablishBinding*-Module** The first sub-module of the Authcoin top level CPN model is the *KeyGenerationEstablishBinding*-module. The module is illustrated in Figure 4.8 and describes the process of generating a new key pair and establishing a binding between the key pair, the owning entity and the DID document as previously described. The user provides a list of identifiers, an expiration date, a key type and the desired key length as input. The input is processed and results in a new key pair and an EntityIdentityRecord (EIR) that is posted to the blockchain. An EIR contains all identity related information of an entity. Note that the EIR and

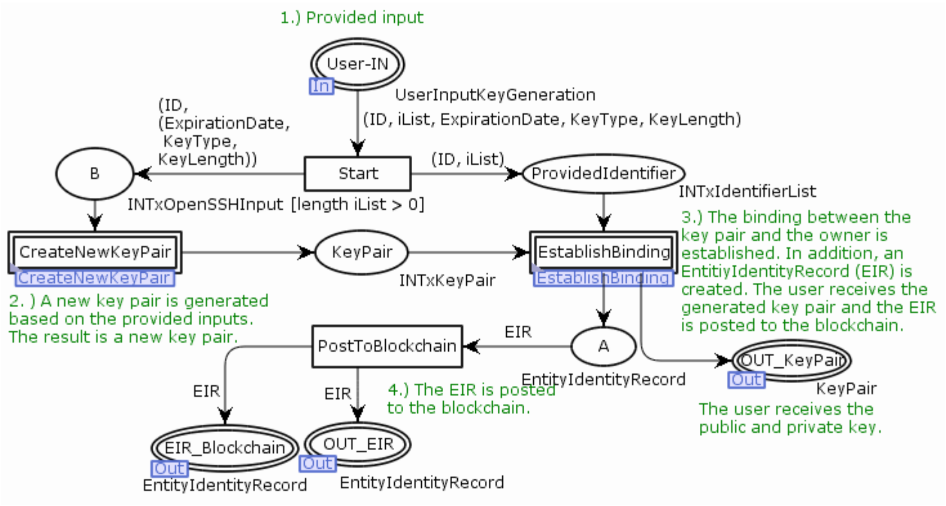
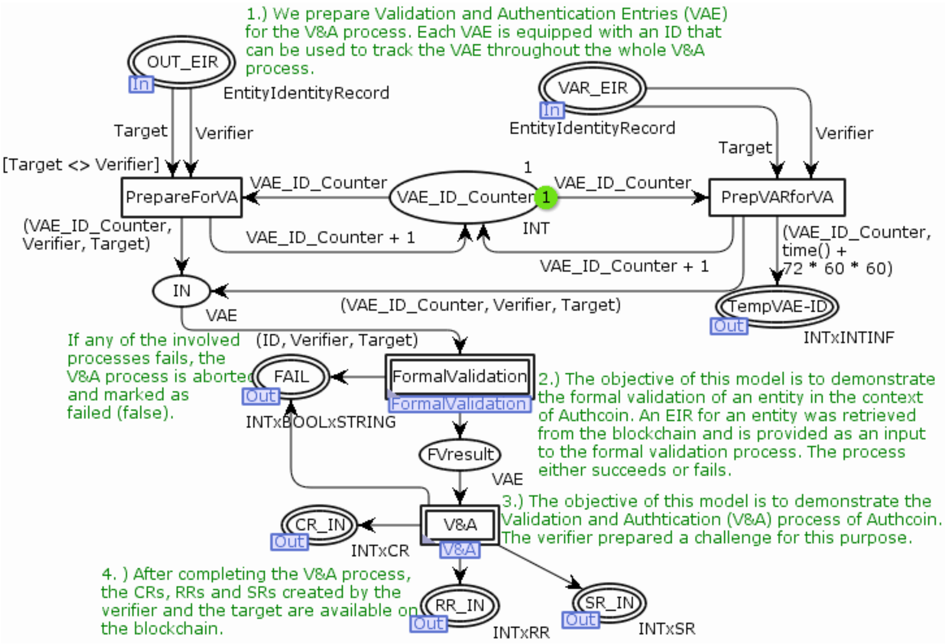

1.) Provided input
User-IN
In
UserInputKeyGeneration
(ID, iList, ExpirationDate, KeyType, KeyLength)
(ID,
(ExpirationDate,
KeyType,
KeyLength))
B
Start
(ID, iList)
ProvidedIdentifier
INTxIdentifierList
INTxOpenSSHInput [length iList > 0]
CreateNewKeyPair
CreateNewKeyPair
KeyPair
INTxKeyPair
EstablishBinding
EstablishBinding
3.) The binding between the key pair and the owner is established. In addition, an EntitiyIdentityRecord (EIR) is created. The user receives the generated key pair and the EIR is posted to the blockchain.
2.) A new key pair is generated based on the provided inputs. The result is a new key pair.
PostToBlockchain
EIR
A
OUT_KeyPair
Out
KeyPair
EntityIdentityRecord
The user receives the public and private key.
EIR
EIR
4.) The EIR is posted to the blockchain.
EIR_Blockchain
Out
EntityIdentityRecord
OUT_EIR
Out
EntityIdentityRecord

FIGURE 4.8: CPN Model of the *KeyGenerationEstablishBinding*-Module – Source: [175] and [176]

the DID document are separate data objects on the blockchain since Authcoin does not necessarily requires to be used in conjunction with DIDs.

***V&A-Processing*-Module** Figure 4.9 illustrates the *V&A-Processing*-module in more detail. A set of EIRs is provided as an input, one for the target and one for the verifier of a V&A procedure. Both EIRs are further processed to create a VAE (V&A Entry) that consists of an ID for the specific V&A process and the target as well as the verifier EIR. The VAE is subsequently further processed in the *FormalValidation*-module. The formal validation procedure is executed for all involved EIRs. It is checked if the public key of each EIR is well-formed, has a sufficient key length and has not been expired or revoked yet. If one test fails, the V&A processing fails. Otherwise the VAE is further processed in the *V&A* module. During the V&A process, the verifier and target exchange challenges (CR – ChallengeRecord), with each other and create the corresponding responses (RR – ResponseRecord). Both entities evaluate the received responses and create corresponding signatures (SR – SignatureRecord) depending on whether they are satisfied with the received response or not. All information is posted to the publicly available blockchain. If any of these steps fail, the whole V&A process is terminated and the specific VAE is marked as failed.

***Mining*-Module** Depending on whether the V&A processing finished successfully, the corresponding information (CRs, RRs and SRs) are posted to the blockchain, as illustrated in Figure 4.10. The actual blockchain-mining process is implemented in a symbolic way, meaning that each input transaction is directly mined into a new block without actually simulating a blockchain consensus algorithm and further block-building



1.) We prepare Validation and Authentication Entries (VAE) for the V&A process. Each VAE is equipped with an ID that can be used to track the VAE throughout the whole V&A process.

If any of the involved processes fails, the V&A process is aborted and marked as failed (false).

2.) The objective of this model is to demonstrate the formal validation of an entity in the context of Authcoin. An EIR for an entity was retrieved from the blockchain and is provided as an input to the formal validation process. The process either succeeds or fails.

3.) The objective of this model is to demonstrate the Validation and Authtication (V&A) process of Authcoin. The verifier prepared a challenge for this purpose.

4.) After completing the V&A process, the CRs, RRs and SRs created by the verifier and the target are available on the blockchain.

FIGURE 4.9: CPN Model of the *V&A-Processing*-Module – Based on [175] and [176]

procedures[5]. In the context of this work, a symbolic implementation is sufficient since Authcoin can be deployed on top of different blockchain architectures with varying mining-concepts. Furthermore, the process of mining a new block does not affect the protocol itself as long as it guarantees that a transaction posted to the blockchain is processed in a block within a given time span. It is only relevant for Authcoin that with each new block, a defined number of VARs is generated.

In our CPN model, every time a new EIR, CR, RR or SR is posted to the blockchain, a new block is produced followed by the creation of new VAR(s) that are also posted to the chain. The creation of new VAR(s) is triggered when a new block is added to the blockchain. The presented CPN model is different in two aspects from description of VARs previously in Section 4.2.6: First, in the model, users do not choose VARs on their own and instead a user is randomly chosen. Second, the model is limited to process only two VARs in order to avoid an endless loop during runtime.

The *ProcessVAR*-module in the *Mining*-module describes the processing steps of a VAR chosen by a user. First, the status of the VAR is updated in order to avoid multiple processing of the same VAR. Afterwards, the EIR of the VAR's target is retrieved. In combination with the verifier's EIR, the V&A process of the *V&A-Processing*-module is triggered and executed. The results of the V&A process are used to update the pending

[5] As mentioned previously, Authcoin does not require a specific consensus algorithms such as PoW and instead only requires a block creation mechanism. In this section, we use the term *mining* loosely equivalent to creating new blocks.

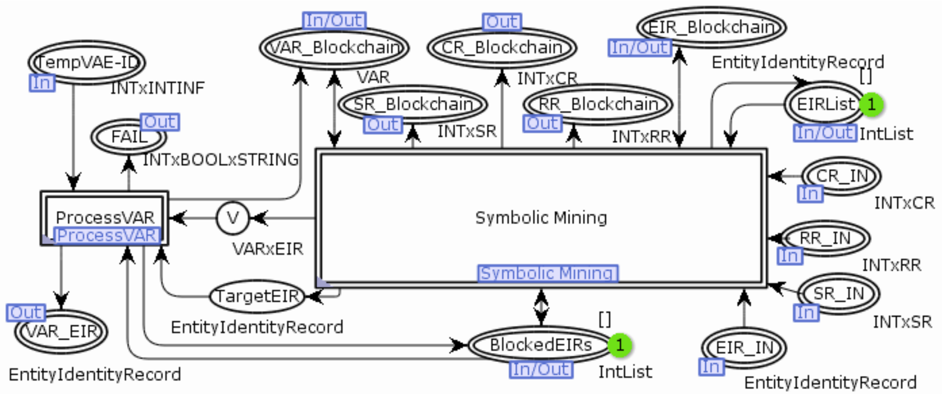


FIGURE 4.10: CPN Model of the *Mining*-Module – Based on [175] and [176]

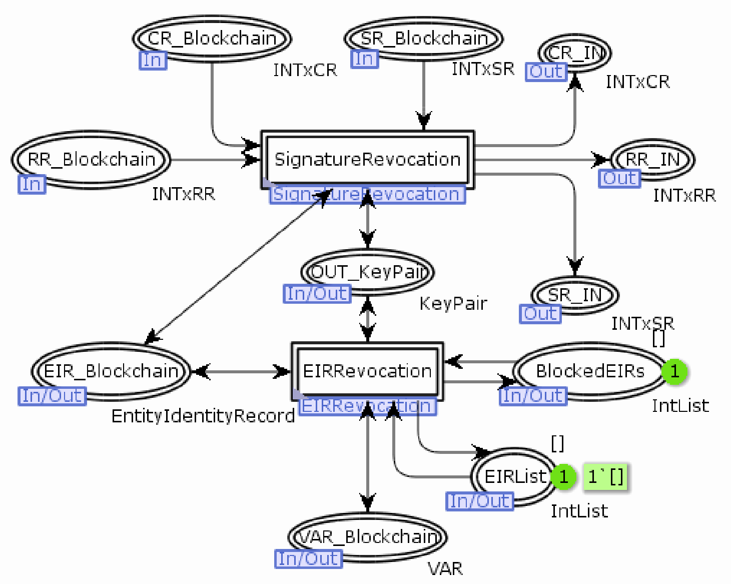


FIGURE 4.11: CPN Model of the *Revocations*-Module – Source: [175] and [176]

VAR and change the status of the VAR to *finished*. The updates are posted to the blockchain.

***Revocation*-Module** The *Revocation*-module is presented in Figure 4.11. In the context of Authcoin, it is possible to revoke either signatures in the form of SRs, or to revoke entity records in the form of EIRs. An EIR can be revoked if it is no longer required, or not trustworthy anymore. SRs can be revoked in case that the signing entity has to remove the expressed trust relationship. The updated revocation information is posted back to the blockchain.

4.3 Costs of Sybil Node Attacks

Previous sections combine the concepts of decentralized identifiers (DIDs), DID documents, verifiable claims as well a blockchain-based challenge-response scheme for validation and authentication in decentralized networks. Even though the protocol presented in Section 4.2 incorporates features to detect sybil node attacks, it is still not able to mitigate them, or quantify the provided security guarantees against sybil node attacks.

Similar applies to a variety of other approaches that focus on preventing sybil node attacks – or at least detecting sybil nodes – in the context of decentralized networks, e.g., SybilGuard [228] and SybilLimit [229] as well as [191], [192] and [230] specifically in MANETs, or [189] and [231] in sensor networks. However, public blockchain platforms such as Bitcoin and Ethereum – which are also decentralized P2P networks – managed to circumvent sybil node attacks all together by establishing economic incentives to reward users participating in the consensus processes.

The following sections present an extended version of the *Unchained* protocol – as proposed in [177] – called *UnchainedX* which utilizes the same economic incentive system to: *i*) de-incentivize sybil node attacks and *ii*) provides an exact price tag on the costs of a sybil node attack, thereby allowing to quantify security guarantees for UnchainedX-protected identities. To do so, UnchainedX binds an identity to a financial stake that is deposited on a blockchain and lost in case a node acts maliciously. As a result, introducing a sybil node is equivalent to investing more financial assets than it would cost to simply create a genuine network identity.

In the following, Section 4.3.1 and Section 4.3.2 present the UnchainedX protocol itself and its parameters. Afterwards, Section 4.3.3 details how difficulty changes of the underlying proof-of-work consensus algorithm are handled.

4.3.1 Protocol Specification

Similarly to the validation and authentication protocol described in Section 4.2, the UnchainedX protocol originally assumed that a simple asymmetric key pair represents an entities identity. As part of this work, we again extend the key pair approach by using a DID and DID document based identity representation instead of only an asymmetric key pair. Whenever we refer to identities in the following sections, we specifically discuss hardware/machine identities. However, binding an UnchainedX identity proof to a human, or a software agent via a DID document is also an option.

An identity, as established in previous sections, uniquely identifies a specific entity. In the context of business collaborations, interactions and transactions as well as for communication purposes devices have to validate each other's identities before exchanging sensitive information. The subsequent sections outline how to establish an UnchainedX identity and the corresponding validation procedure for communicating devices. Neither of these steps require a trusted third party apart from a PoW-based blockchain platform. Future protocol versions might be extended to support PoS systems as well. In the context of this work, the features and functionalities of UnchainedX are illustrated and explained using the Bitcoin blockchain.

4.3.1.1 Creating an Identity Proof

The process of binding an existing DID-based machine identity is illustrated in Figure 4.12. It assumes that the identity already exists outside the UnchainedX context and that a corresponding key pair exists which represents a Bitcoin wallet address. The wallet is equipped with a minimum amount of Bitcoin (more details on pricing later in Section 4.3.2.2) which are used as a deposit in the process of binding the identity to the blockchain.

First, the Bitcoin deposit is made by transferring the tokens to a pre-defined wallet address. The transaction is mined into a new block by the Bitcoin network and is credited to the deposit address (step 2 and step 3). Next, an UnchainedX identity proof is created based on information resulting from the block containing the deposit transaction. The identity proof consists of the block header (block number, block hash, target difficulty), the deposit transaction, the block's Merkle tree to prove that the deposit transaction is part of the block, the index number of the deposit address in the block as well as the public key.

Additionally, a unique *proofID* is created using the Hash Message Authentication Code (HMAC) as illustrated in Equation 4.1 - 4.3. The PoW hash of the block containing the deposit address is used as the key for the HMAC in combination with the index number of the deposit transaction resulting in the *proofID*. Since the *proofID* depends on the PoW hash of the block, predicting a *proofID* is equivalent to predicting the correct hash of the next Bitcoin block and therefore not feasible. As a result, a potential attacker cannot create arbitrary proof IDs.

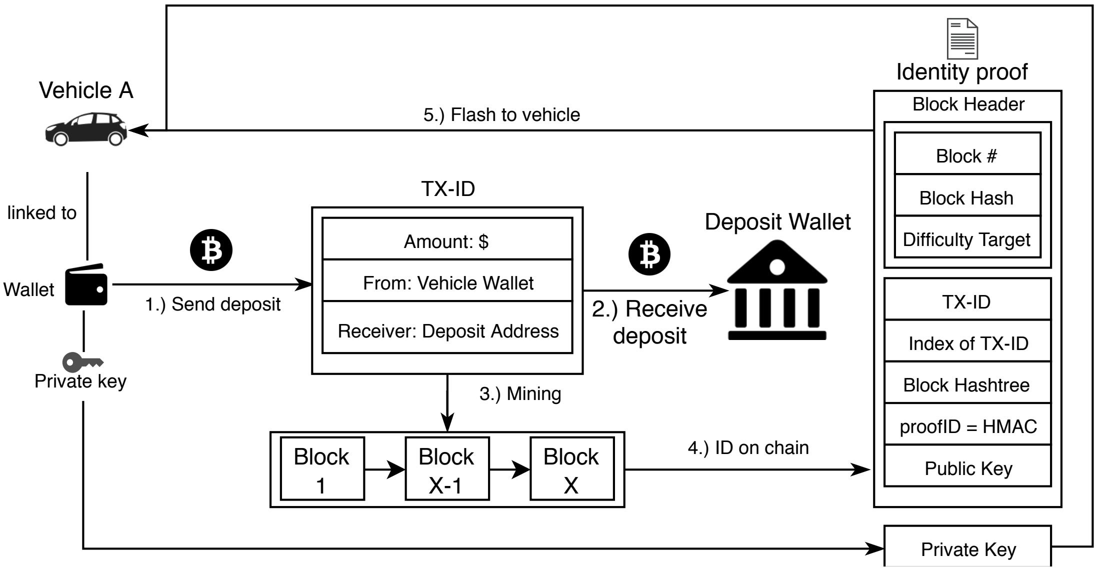


FIGURE 4.12: Creating a New Identity Proof – Based on [177]

$$k_{\text{HMAC}} := \text{Block}_{\text{PoWHash}} \tag{4.1}$$

$$\text{TXindex} := \text{index of deposit TX in Block} \tag{4.2}$$

$$\text{proofID} := \text{HMAC}(k_{\text{HMAC}}, \text{TXindex}) \tag{4.3}$$

Finally, the resulting identity proof is deployed to the machine in combination with its DID-based identity and the corresponding asymmetric key pair. In the context of small IoT devices, this process might be part of the deployment procedure, e.g., a new sensor network. Alternatively, in the context of our V2X scenarios from Chapter 3 the proof might be created and deployed to a vehicle by the manufacturer itself.

4.3.1.2 Identity Proof Validation

Upon first contact or when establishing a communication channel among entities of a decentralized network, entities verify each others' identity in a bidirectional manner to secure and protect sensible network data. In order to ensure that none of the parties is a sybil node, both of them validate the DID-based identity and UnchainedX identity proof. Figure 4.13 details the proof validation among two vehicles. The process consists of a sequence of validation checks and in case one of them fails, the whole proof is considered invalid. In that case, the proof is discarded and no further communication is initiated.

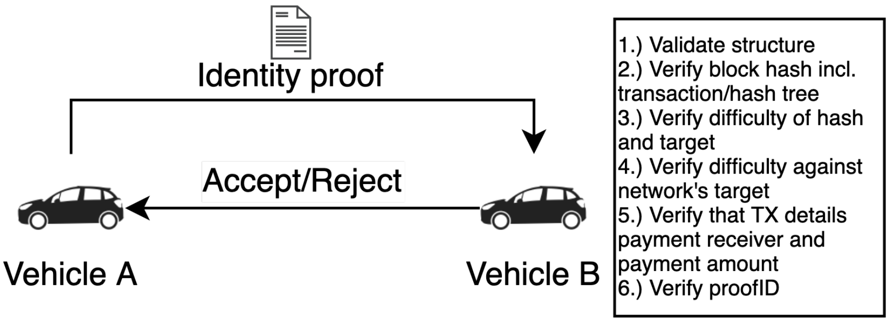

Identity proof
Accept/Reject
Vehicle A
Vehicle B
1.) Validate structure
2.) Verify block hash incl. transaction/hash tree
3.) Verify difficulty of hash and target
4.) Verify difficulty against network's target
5.) Verify that TX details payment receiver and payment amount
6.) Verify proofID

FIGURE 4.13: Overview of the Validation Process – Based on [177]

First, the structure and syntax of the presented proof is validated. As described in the previous Section 4.3.1.1, a valid proof consists of the block header (block number, block hash, target difficulty), the deposit transaction, the block's Merkle tree to prove that our deposit transaction is part of the block, the index number of the deposit address in the block as well as the transactions' corresponding public key. Subsequently, it is verified that the block's height is above $\text{networkParameter}_{\text{height}}$ (more on protocol parameters in Section 4.3.2). Next, nodes verify that the Merkle tree of the block contains the deposit transaction, before comparing the block difficulty to the difficulty target in the block header. In addition, it is verified that the UnchainedX difficulty target is equal or higher than $\text{networkParameter}_{\text{minDifficulty}}$ for the given block height.

Thereafter, the deposit transaction itself is verified to ensure that the deposit was send to the correct deposit address ($\text{networkParameter}_{\text{receiver}}$) and is greater or equal to the minimum deposit amount ($\text{networkParameter}_{\text{amount}}$). Moreover, the transaction has be to signed by and with the corresponding key pair contained in the identity proof. Finally, the *proofID* is compared to the recalculated result of the formula in Equation 4.3 given the provided input parameters from the identity proof.

4.3.2 Protocol Parameterization and Pricing

Previous sections reference several protocol parameters of UnchainedX and the pricing structure of an identity proof. The subsequent Section 4.3.2.1 and Section 4.3.2.2 provide further details on both topics.

4.3.2.1 Network Parameters

Deposit Address: The parameter $\text{networkParameter}_{\text{receiver}}$ specifies the deposit address for the identity proof. Funds of the deposit address should be non-recoverable for a malicious entity orchestrating a sybil node attack. Hence, we have to ensure that the deposit address is not controlled by the attacker itself or a colluding party. Note that a transaction fee-based alternative is not feasible since an attacker could simply mine the corresponding block itself and hence recover its own transaction fee.

While the specific choice of the deposit address(es) is/are use case dependent, we outline two simple approaches. First, funds might be send to an invalid receiving address following a proof-of-burn strategy [232]. Even though the transferred deposit cannot be recovered, it might be considered destructive for the remaining network participants that use the system outside the UnchainedX context. Burning tokens with a limited supply (e.g., Bitcoin) ultimately changes the underlying economic assumptions for all network users and also results in a shortage of available tokens at some point in the future, thereby disabling UnchainedX itself.

Alternatively, deposits can be directed to a pool of (decentralized) network operators (hardware and software level) to support further development of the network as well as covering maintenance costs. Since network operators and similar entities are stakeholders of the network, they have a strong incentive to keep the network secure and prevent sybil attacks – thus, not act maliciously.

Deposit Amount: $\text{networkParameter}_{\text{amount}}$ specifies the minimum deposit size required to create a valid identity proof. The deposit size has to be chosen by the network operator in such a way that it remains affordable for network participants to establish new identity proofs, but at the same time economically disincentivizing malicious entities from creating large numbers of sybil nodes. Networks with a large number of nodes may allow for lower deposits since the network may also tolerate larger quantities of sybil nodes without any tangible network service disruptions. Subsequently, Section 4.3.2.2 further elaborates on the pricing of identity proofs.

An alternative approach is to lower the parameter $\text{networkParameter}_{\text{amount}}$ while at the same time introducing a new parameter $\text{networkParameter}_{\text{lockedAmount}}$ which specifies a deposit that is send back to the identity proof owner. However, the re-transfer is locked up using the `CheckLockTimeVerify` output [233] of a transaction or another type of smart contract[6]. The locktime should be equivalent or higher than the lifetime of an identity proof. When leaving the network, genuine users receive the locked amount back. At the same time, this approach ensures that large scale sybil node attacks lock up significant amounts of financial assets. Finally, a smart contract based lock up method also enables a more sophisticated system where users provide proofs of malicious behavior of nodes, which results in losing the locked up tokens – similar to losing a stake in PoS consensus systems.

[6] Note that smart contracts are not supported by Bitcoin but by many other blockchain platforms.

Minimum Target Difficulty: The target difficulty of PoW-based consensus systems affects the average amount of computing power required to mine a valid block. Since UnchainedX depends on only valid blocks being produced, the minimum target difficulty of a valid block also affects the security guarantees provided by UnchainedX. The $\text{networkParameter}_{\text{minDifficulty}}$ specifies the minimum target difficulty of the underlying PoW-based block that is required to create a valid identity proof. While selecting a predefined difficulty level is the simplest available option, it does not allow for adaptions to the changing difficulty of the underlying PoW blockchain. Difficulty drops may make it impossible to create new identity proofs since the target difficulty of the blockchain is always lower than specified in the protocol. Alternatively, significantly increased difficulties might make it unintentionally cheap for an attacker to create large numbers of sybil nodes. Section 4.3.3 outlines different approaches that allow for dynamic target difficulty adjustments.

Block Height: The $\text{networkParameter}_{\text{height}}$ parameter determines the minimal acceptable block height of an identity proof. The block height is the number of blocks preceding a block on the blockchain. As a result, the genesis block of a chain has a block height of zero [234]. In the context of UnchainedX identity proofs, the block height corresponds to the block height at the start of a network's lifetime, thereby allowing to reject proofs containing lower block heights with potentially lower difficulties.

4.3.2.2 Pricing an Identity Proof

An attacker that aims to introduce a large number of sybil nodes can do so by acquiring a large number of identity proofs, or calculating a fake blockchain block with a valid difficulty level containing an arbitrary number of fake identity proofs. However, mining a block with valid difficulty incurs high opportunity- and energy costs since the attacker has to pay for the hashing power used to create the block. While the energy costs may vary depending on the geographical location, the opportunity cost is easy to quantify and equal to the block reward plus additional transaction fees.

Equations 4.4-4.5 are used to calculate the upper bound of the price $\text{amount}_{\text{max}}$ of an identity proof where it becomes cheaper for an attacker to generate a fake block instead of simply paying for the identities (disregarding energy costs and transaction fees). The equations assume the current Bitcoin block size limit of 1MB, a minimum transaction size of 224B and a block reward of 12.5BTC [232][235][236].

$$\text{amount}_{\text{max}} = \text{block reward} \times \frac{\text{min TX size}}{\text{max block size}} \tag{4.4}$$

$$= 12.5\ \text{BTC} \times \frac{224\text{B}}{1\text{MB}} = 0.0028\ \text{BTC} \tag{4.5}$$

Based on the Bitcoin price of \$8,574 as of the writing of this work (2019-05-31) [237], the resulting maximum for $\text{amount}_{\text{max}}$ is around \$24. In reality, the value $\text{amount}_{\text{max}}$ will likely be lower than 0.0028 BTC in order to make identities affordable and furthermore anticipate volatility with regards to the Bitcoin price. Moreover, identity proof prices may vary depending on the device that the proof is fabricated for, e.g., the proof for a vehicle might be more expensive than the proof for a small temperature sensor based on the economic value of the entity for the overall network.

4.3.3 Difficulty Adjustments

Changes of the target difficulty of the Bitcoin PoW consensus algorithm directly influence the security guarantees provided by UnchainedX as well as the process of creating and validating an identity. Hence, the protocol has to account for such difficulty changes. The Bitcoin network adjusts the PoW target difficulty every 2016 blocks which is roughly equivalent to two weeks given an average target block time of ten minutes. To do so, every UnchainedX entity maintains a list of accepted target difficulties for each 2016 block interval. Receiving an update on a new accepted target difficulty that is greater than the previous value results in retroactively invalidating identities confirmed based on blocks with lower difficulty values. In case a lower target difficulty value is accepted, it might be necessary to retroactively accept discarded peers into the network. In order to update the list of accepted target difficulties, we propose three approaches – each with different advantages and disadvantages.

In the context of the M2X ecosystem, we can safely assume that all participating entities are connected to the Internet most of the time, especially in densely populated areas such as cities. However, in sparsely populated areas, in buildings, or other locations with a lack of connectivity, devices have to validate UnchainedX-bound identities without being able to check the most recent target difficulty level online (even for longer times), e.g., via an oracle or similar services. Therefore, UnchainedX supports offline verification and is able to handle the corresponding difficulty adjustments. Even though two weeks without an Internet connection is rather unlikely for entities of the M2X economy, the mechanism remains necessary for PoW-based blockchains with lower difficulty adjustment periods such as Ethereum.

4.3.3.1 Bundled Oracle and Network Operator Updates

The simplest approach is to provide an oracle service that is queried by the devices or assume a scenario where network operators publish publicly available and auditable signed messages containing the target difficulty for each 2016 block range. Such oracle/-operator messages are appended to each identity proof and used to update the target difficulty list of all receiving nodes. Despite its simplicity, this approach suffers two main disadvantages: First, it is rather centralized and relies either on oracle services or network operators to provide correct difficulty updates. Second, in case a device is not regularly connected to the Internet, to interact with oracles, or a network operator, new nodes might not be able to join the network.

4.3.3.2 Majority Vote

The second option extends the approach proposed in the previous Section 4.3.3.1. Instead of relying on a single oracle or operator, nodes connect to a multitude of providers and attach one or more of their messages to an identity proof and maintain a target difficulty list with multiple difficulty values for each block. In case of a mismatch within the list or on receiving an update message, the majority value is considered the true difficulty target. Supposing no majority, the highest difficulty value is considered valid. As a result, option two mitigates the issue of a single point of failure but it still dependent on a rather centralized, or at least federated system of data sources.

4.3.3.3 Maximum Difficulty

An alternative option that does not rely on any centralized infrastructure is to deploy each device either with an empty list or a pre-loaded history of known accepted target difficulties. The difficulty of each received identity proof is compared to the known target difficulty of the corresponding block based on the maintained list. A mismatch or a difficulty that is to low results in rejecting the incoming proof. A target difficulty that is higher than the value in the maintained list results in an update of the lists' value assuming that the identity proof itself is valid in its entirety. As long as each node is connected to at least one honest node, this approach allows to eventually detect and invalidate forged identities with insufficient difficulty values. However, an attacker that is able to create an identity proof based on a mined block with a target difficulty higher than the difficulty of the underlying Bitcoin blockchain, is able to perform a denial of service attack (DoS attack) on the protocol. Up on receiving the higher target difficulty value, all connected nodes update their list accordingly and start invalidating actually genuine identities which were generated during the timeframe corresponding to the malicious block. Yet, the attack scenario is rather unlikely since mining a block with a target difficulty higher than the difficulty of the Bitcoin chain itself is even more expensive than mining a regular Bitcoin block that is – at the time of writing this thesis – rewarded with 12.5 Bitcoin. Moreover, the DoS attack can be mitigated by combing this approach with one of the remaining methods described previously. Simultaneously, this approach might serve as a fallback solution for the previously described approaches.

4.3.4 Protocol Formalization

Next, we provide a formal model of UnchainedX using the same approach as for the Authcoin protocol in Section 4.2.8. Again, we first present the AOM goal model of the protocol, define the behavioral interfaces of activities before mapping these two to a formal CPN model.

Once more, we utilize the AOM goal model to capture the functional requirements of the system in the form of goals, as well as non-functional requirements and roles of involved entities. Non-functional requirements represent quality goals of the system. Figure 4.14 illustrates the top level goal model of the protocol. The main objective of UnchainedX is to de-incentivize and price the cost of sybil node attacks. The main goal is further decomposed into multi-layered sub-goals until the lowest atomic sub-goal is reached. In the context of UnchainedX, the main goal is further divided into the following sub-goals: *Create deposit transaction*, *Mine transaction*, *Create identity proof* and *Validate identity proof*. The five quality goals *secure*, *correct*, *tamperproof*, *entity agnostic* and *automated* are attached to the overall main goal of the goal model, meaning they are relevant and inherited to all sub-goals and are inherited. The quality goal "reliable" pertains to the two sub-goals of *Mine transaction* and *Validate identity proof*. Furthermore, we list three different roles. The *user* – either a human, or machine -, the *mining* entity that performs the PoW calculations of the underlying blockchain and the validator who validates an identity proof once received.

Subsequently, the AOM behavior model refines the previously developed goal model for specific agents and activities. Table 4.4 presents the behavior interface model of the goals depicted in the top level goal model of Figure 4.5. Each activity is listed with its corresponding trigger, optional pre-conditions and its post-conditions. The *Create Deposit Transaction*-activity is triggered after providing the required network input parameters as well as a machine identity and a wallet. Afterwards, as part of the *Mining*-activity, the resulting deposit transaction is mined into a new block. During the *Create Identity Proof*-activity, the block, the machine identity and the provided wallet

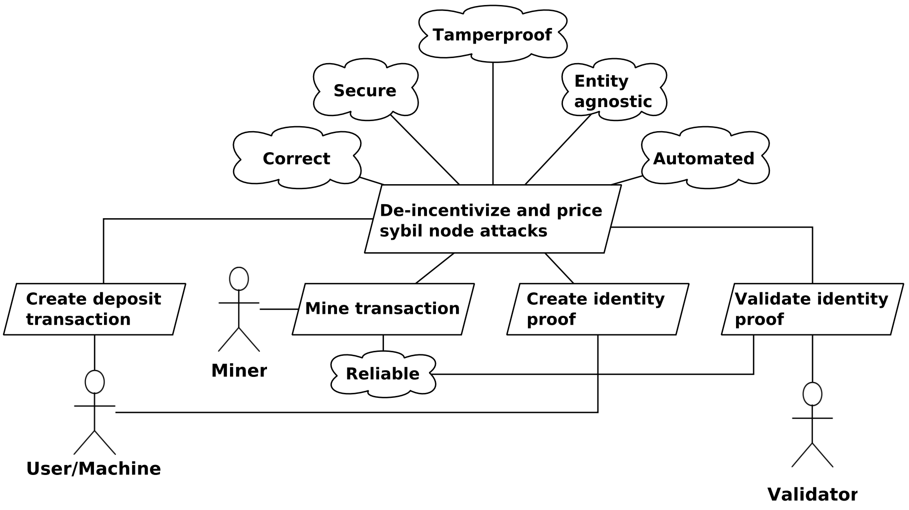


Figure 4.14: UnchainedX Top-Level AOM Goal Model

Activity	Trigger	Pre-Condition	Post-Condition
Create Deposit Transaction	User wants to create a deposit transaction	Network parameters, machine identity and machine wallet	Network parameter, machine identity, machine wallet, deposit transaction
Mining	Received deposit transaction	Deposit transaction, previous block hash, deposit wallet and blockchain difficulty target	Block, previous block hash, blockchain difficulty target, deposit wallet
Create Identity Proof	Deposit transaction mined into block and user wants to create new identity proof	Block with deposit transaction, machine identity and machine wallet	Identity proof, machine identity, machine wallet
Validate Identity Proof	Incoming identity proof	Identity proof, network parameter, machine identity and machine wallet	Boolean statement whether the provided identity proof is valid, or not

Table 4.4: Behavioral Interfaces of Activities for UnchainedX

are used to create an identity proof for the machine. Finally, the created identity proof is validated as part of the *Validate Identity Proof*-activity which takes an identity proof, the network parameter, the machine identity and the initial wallet to determine whether the identity proof is valid.

Token colors represent data structures of UnchainedX data objects that are used to illustrate the data flow through the CPN model. Exemplary acronyms, names and the description of token colors of UnchainedX's CPN model are presented in Table 4.5. The first column specifies the name, followed by a short description in column two. The last column provides information concerning the data types. A complete listing of all acronyms, names and abbreviations as well as a description of the token colors of the UnchainedX CPN model is available in Appendix A.2.2.

Mapping the created AOM models to CPN is the final step to derive the top level CPN model of UnchainedX. We rely on the same mapping mechanism as presented earlier in Table 4.2 and Figure 4.6 which illustrate the mapping heuristic of AOM goal models to CPN models as well as the mapping of behavior interface models to CPN models. The final result of the mapping process is illustrated in Figure 4.15, presenting the complete and formalized CPN model of UnchainedX derived from the AOM model and implemented using CPN-Tools.

The CPN model consists of four transitions derived from the four sub-goals of the top level AOM goal model. The protocol flow start on the left-hand side of Figure 4.15 with the *Create deposit transaction*-transition. An infrastructure provider, user, or machine that wishes to create a new identity triggers the transition by providing the required network parameters as described in previous sections as well as information relating to the entities identity (in the CPN model, a machine in assumed), i.e., a machine identity consisting of a DID and a public/private key in addition to a wallet that corresponds to the used key pair. In case the wallet balance is sufficient to make a deposit, a matching deposit transaction with the target deposit address is created. Thereafter, the transaction is mined into a new block of the underlying blockchain platform (*Mining*-transition). The resulting block contains a BlockID, the hash of the previous block, the blockchain's difficulty target and a list of included transactions. Once the block is mined, an identity proof is created (*Create identity proof*-transition) according to the specifications described in Section 4.3.1. Finally, the resulting identity proof is checked for validity resulting in a Boolean representation of the validation process's success, or failure.

Token Color	**Description**	**Type**
KeyPair	Key pair	(pubKey, privKey)
Wallet	Blockchain wallet	(Address, Balance)
NetworkParameter, NP	Unchained network parameter	(Difficulty, minBlockHeight, minDeposit, depositAddress)
Difficulty	Minimum PoW difficulty for an identity proof as defined by the network operator	Integer
minBlockHeight	Minimum block height as defined by the network operator	Integer
minDeposit	Minimum deposit to be made for an identity proof as defined by the network operator	Integer
depositAddress	Deposit address as defined by the network operator	String
Transaction, TX	Structure of a deposit transaction	(ID, from, to, amount, pubKey, txSig)
Block	Blockchain block	(ID, prevBlockHash, BlockchainDiffTarget, txList)
IdentityProof, IP	Identity proof	(BlockID, BlockHash, BlockchainDiffTarget, TXID, BlockTXList, proofID, proofSig)
proofID	proofID as specified by the protocol	String
MachineIdentity	Machine entity identity	(DID, KeyPair)
depositWallet	Deposit wallet as defined by the network operator	Wallet
machineWallet	Machine's wallet	Wallet

Table 4.5: Exemplary Acronyms, Names and Description of Token Colors of the UnchainedX CPN Model

4.4 Discussion

While the combination of the concepts of DIDs, Authcoin and UnchainedX provides a suitable identity-, authentication- and validation mechanism for entities participating in the M2X economy, it still requires further research in several directions to overcome limitations and allow for widespread adoption. This section discusses limitations, open issues as well as advantages and disadvantages of the solution proposed in this chapter.

Since we already discussed the general use and suitability of blockchain technology as

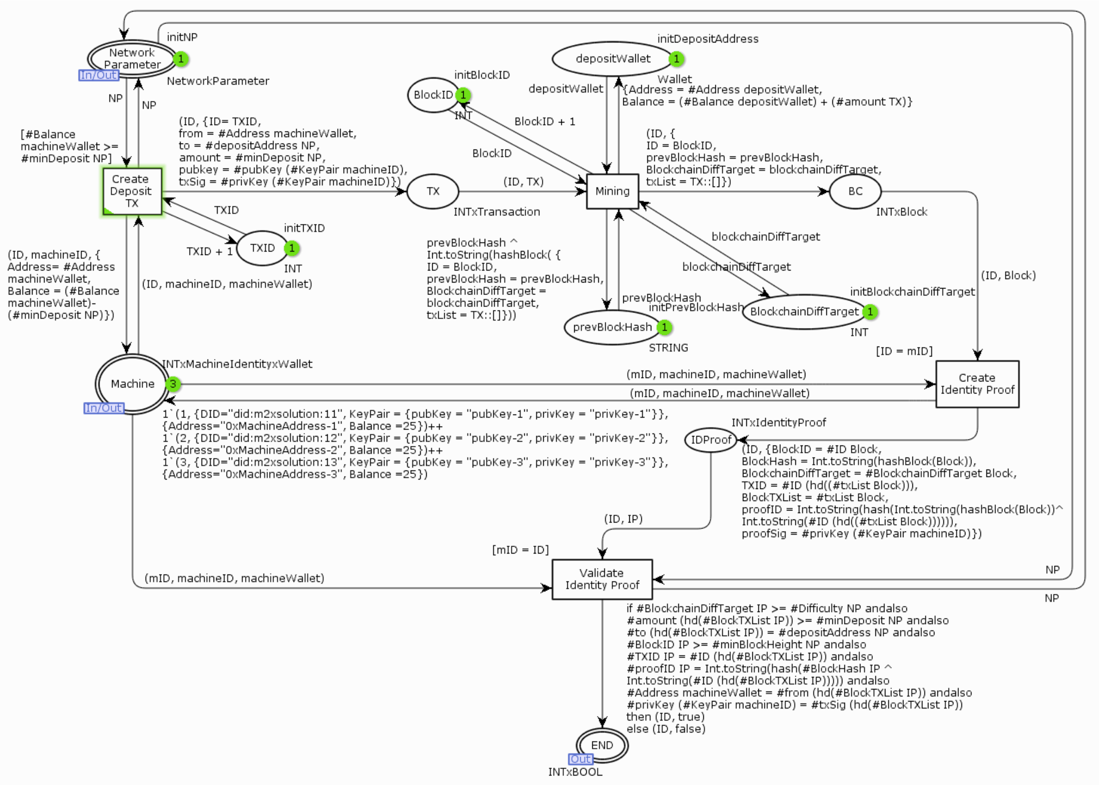

initNP
Network Parameter
1
In/Out
NetworkParameter
NP
NP
[#Balance machineWallet >= #minDeposit NP]
Create Deposit TX
(ID, {ID= TXID,
from = #Address machineWallet,
to = #depositAddress NP,
amount = #minDeposit NP,
pubkey = #pubKey (#KeyPair machineID),
txSig = #privKey (#KeyPair machineID)})
TX
INTxTransaction
TXID
initTXID
TXID + 1
TXID
1
INT
(ID, machineID, {
Address= #Address
machineWallet,
Balance = (#Balance
machineWallet)-
(#minDeposit NP)})
(ID, machineID, machineWallet)
initBlockID
BlockID
1
INT
BlockID + 1
BlockID
initDepositAddress
depositWallet
1
depositWallet
Wallet
{Address = #Address depositWallet,
Balance = (#Balance depositWallet) + (#amount TX)}
(ID, TX)
Mining
(ID, {
ID = BlockID,
prevBlockHash = prevBlockHash,
BlockchainDiffTarget = blockchainDiffTarget,
txList = TX::[]})
BC
INTxBlock
prevBlockHash ^
Int.toString(hashBlock({
ID = BlockID,
prevBlockHash = prevBlockHash,
BlockchainDiffTarget =
blockchainDiffTarget,
txList = TX::[]}))
blockchainDiffTarget
blockchainDiffTarget
prevBlockHash
initPrevBlockHash
prevBlockHash
1
STRING
BlockchainDiffTarget
1
initBlockchainDiffTarget
INT
(ID, Block)
[ID = mID]
INTxMachineIdentityxWallet
Machine
3
In/Out
(mID, machineID, machineWallet)
(mID, machineID, machineWallet)
Create Identity Proof
1`(1, {DID="did:m2xsolution:11", KeyPair = {pubKey = "pubKey-1", privKey = "privKey-1"}},
{Address="0xMachineAddress-1", Balance =25})++
1`(2, {DID="did:m2xsolution:12", KeyPair = {pubKey = "pubKey-2", privKey = "privKey-2"}},
{Address="0xMachineAddress-2", Balance =25})++
1`(3, {DID="did:m2xsolution:13", KeyPair = {pubKey = "pubKey-3", privKey = "privKey-3"}},
{Address="0xMachineAddress-3", Balance =25})
INTxIdentityProof
IDProof
(ID, {BlockID = #ID Block,
BlockHash = Int.toString(hashBlock(Block)),
BlockchainDiffTarget = #BlockchainDiffTarget Block,
TXID = #ID (hd((#txList Block))),
BlockTXList = #txList Block,
proofID = Int.toString(hash(Int.toString(hashBlock(Block))^
Int.toString(#ID (hd((#txList Block)))))),
proofSig = #privKey (#KeyPair machineID)})
(ID, IP)
[mID = ID]
(mID, machineID, machineWallet)
Validate Identity Proof
NP
NP
if #BlockchainDiffTarget IP >= #Difficulty NP andalso
#amount (hd(#BlockTXList IP)) >= #minDeposit NP andalso
#to (hd(#BlockTXList IP)) = #depositAddress NP andalso
#BlockID IP >= #minBlockHeight NP andalso
#TXID IP = #ID (hd(#BlockTXList IP)) andalso
#proofID IP = Int.toString(hash(#BlockHash IP ^
Int.toString(#ID (hd(#BlockTXList IP))))) andalso
#Address machineWallet = #from (hd(#BlockTXList IP)) andalso
#privKey (#KeyPair machineID) = #txSig (hd(#BlockTXList IP))
then (ID, true)
else (ID, false)
END
Out
INTxBOOL

Figure 4.15: UnchainedX CPN Model

a foundation of the M2X economy in the previous Chapter 3 as well as in Section 2.3, we will waive further discussions on that topic. However, relying on a blockchain-based identity solution as described in this work results in certain challenges and limitations. First, the transparent and auditable design of Authcoin challenges ensures on one hand the security of the network. On the other hand, depending on the challenge design they may expose personal information which are meant to be private. Moreover, revealing information in the process of a challenge renders them unavailable to use for subsequent challenges. Therefore, further research on privacy-preserving challenges is required. However, verifiable claims still allow Authcoin users to use them as designed in the context of DIDs, or in an Authcoin context using encrypted challenges thereby combining the advantages of DIDs and Authcoin.

Generally, the design of suitable challenges that are standardized and vendor-independent requires further research. The same applies to the design of user-friendly and tamper-resistant challenges. The development of standards across industries and hardware vendors via consortia is necessary to ensure an interoperability, compatibility and portability of a M2X identity solution. This work simplifies M2X identities by not specifically differentiating between humans, machines and software agents, thereby offering an abstract solution. Future research may result in category-specific implementations of our abstract identity framework.

Finally, in the context of business interactions, transactions and collaborations an identity is the central gatekeeper. Compromising an identity potentially results in subsequent threats to the M2X ecosystem and its purpose. Hence, a thorough security analysis of the used protocols is required. While the protocols are already formalized using CPNs, a formal model does not allow to make statements regarding the security of a protocol. In case of the Authcoin protocol, a security analysis based on the CPN models was performed in [176] and [227] using the ISSRM domain model [238][239] and so-called security risk-oriented patterns SRPs [240]. This resulted in further security improvements of the Authcoin protocol. The analysis, the application of the SRPs and the updated CPN models are not part of this work.

Chapter 5

Evaluation

The following chapter evaluates the results and findings of Chapter 3 and Chapter 4. The different research directions of these chapters require different evaluation approaches for each of their findings. Hence, the following sections utilize a variety of evaluation concepts, which are explained throughout the corresponding sections. Section 5.1 evaluates the V2X transaction, interaction and collaboration platform introduced in Chapter 3, while Section 5.2 and Section 5.3 focus on the proposed solutions for validation, authentication and identities in decentralized networks. The chapter concludes with a discussion of the findings in Section 5.4.

5.1 V2X Platform – Feasibility Evaluation

Due to the exploratory nature of the artifacts produced in Chapter 3, a real-world proof-of-concept, or even a prototype implementation are out of scope of this work while other evaluation methods such as simulations, or quantitative evaluation methods are not applicable. Thus, for the V2X platform a paper-based feasibility evaluation is performed which considers existing technology and on-going research that allow for a simplified and minimal proof-of-concept implementation. The proposed technology stack represents a tentative proposal based on available solutions. The goal is to *i*) demonstrate the general feasibility of a proof-of-concept implementation, *ii*) identify technological gaps that require further research and *iii*) demonstrate the generalizability of the V2X platform as described in this work.

The evaluation is structured as follows: Section 5.1.1 focuses on the hardware and infrastructure components, while Section 5.1.2 reviews pre-existing projects and research for a suitable smart contract platform which serves as the backbone of the V2X platform.

Data sources, data storage and data management are discussed in Section 5.1.3, followed by the identity and authentication mechanisms in Section 5.1.4. Next, Section 5.1.5 outlines the potential directions of future research efforts which are required to bridge the gap between an initial proof-of-concept implementation and the deployment of a production-ready system. Finally, Section 5.1.6 focuses on the generalizability of the V2X platform.

5.1.1 Hardware and Infrastructure

Despite the lack of availability of autonomous vehicles as well as the lack of widespread adoption of highly connected vehicles, a minimal testbed for the V2X platform with vehicles is feasible. The concept of VANETs, as introduced in Section 2.2, provides the conceptual network architecture, the communication technology and Internet connectivity. Even though a broad deployment of TPDs in vehicles and RSUs along the roadside is missing, closed testing areas exist. To compensate for the lack of suitable vehicles with sufficient hardware and a VANET infrastructure, existing vehicles may be equipped with external hardware devices as limited sensing, computing and connectivity platforms. The AutoPi[1] OBDII (On-Board Diagnostics 2 – a vehicle's self-diagnostic and reporting protocol) dongle is based on an embedded Raspberry Pi and allows to extend the sensing, computing and communication capabilities of all OBDII-compatible vehicles. Communication via 3G/4G and Bluetooth is supported in addition to the diagnostic data provided by the car (e.g., acceleration, steering, braking, etc. – depending on the vehicle), location data is available via the AutoPi's GPS. More sensors can be attached via GPIO sensors [241].

Another option is to utilize the ubiquitousness of mobile phones and their built-in sensors, communication hardware and computing capabilities. These can be connected to a vehicle via a simple OBDII dongle with Bluetooth connectivity. Nevertheless, the AutoPi and the mobile phone solution are limited in their capabilities – especially computing power – and thus only suitable for proof-of-concept implementations. Later hardware solutions require more computing power, more sensors with high-quality sensing capabilities and VANET-compatible communication hardware. Finally, future developments of the technology stack will merge the external hardware device into the vehicle itself.

5.1.2 V2X Smart Contract Platform

The V2X smart contract platform is a multi-stakeholder system that requires publicly available transactions, smart contracts and data as well as their private counterparts

[1] https://www.autopi.io/

to account for privacy concerns – but also to allow for OEM, or service provider specific service platforms. Therefore, a complementary setup of public and permissioned blockchains is required. OEMs and service providers may use Hyperledger Fabric [242] for permissioned on-chain transactions, data storage and smart contract orchestration to facilitate their services.

The public counterpart to the permissioned Hyperledger Fabric blockchain may be Ethereum [70], or the EVM-compatible (Ethereum Virtual Machine) Qtum [139] blockchain. On top of Ethereum, the Robonomics platform [243][244] acts as an Ethereum network infrastructure for cyber-physical systems' integration into limited and simplified M2X use cases and scenarios. The Robonomics platform allows for the distribution, control and provision of services by cyber-physical systems using a supply- and demand based marketplace in combination with a contractual obligation management mechanism.

Orchestration of complex smart contracts requires proper tools for creation, deployment and management of such. While initial research on management tools exist [245], only a few tools are available for productive use. Kaleido[2] provides an enterprise-grade smart contract management system for smart contract source code management, compilation and deployment in combination with smart contract registry and gateway APIs. As of the writing of this thesis, Ethereum, Quorum and Hyperledger are supported.

Interoperability among blockchains is not only important to integrate the permissioned and public blockchains, but also to account for a large variety of blockchain platforms used by different ecosystem entities. In Section 2.1.4 we already discussed the concept of blockchain interoperability as well as different approaches to achieve it, e.g., Hardjono et al. [98] propose a set of design principles for interoperable blockchains and demonstrate how the MIT project Tradecoin [99] is designed using this interoperability model. In the context of this paper-based feasibility evaluation for a minimal proof-of-concept, the Polkadot [100] blockchain is selected to enable cross-blockchain transfers of any type of data, or asset. In Polkadot, relay-chains act as hubs between different blockchains where the relay-chains themselves are distinct blockchains that relay messages between chains, but may also track the state of connected chains, thereby essentially acting as an interoperability layer.

While most blockchain platforms allow for value transfer via tokens, real-world business transactions are still settled using fiat-currencies. Consequently, a mechanism for the settlement of fiat-cryptocurrency payments is a necessity. The Corda [246] blockchain platform recently introduced a proof-of-concept[3] that links bank collateral accounts to tradable asset-backed tokens on the Corda blockchain [247][248]. Moreover, Corda

[2] https://kaleido.io

[3] https://github.com/corda/cash-issuer

demonstrated an implementation of the Corda Settler[4] to handle off-ledger settlements in other non-Corda payment systems. However, a Polkadot integration does not exist yet. Alternatively, for proof-of-concept demonstration payment and non-payment related transactions could be handled by separate systems.

Business interactions, transactions and collaborations are subject to legal compliance. Additionally, automated business enactment without human intervention or supervision requires special care. Hence, verifying the soundness of utilized smart contracts before their enactment is imperative in order to eliminate concurrency-, or dependability conflicts [128]. Both, Ethereum and Qtum, use Solidity as a smart contract programming language. Rudimentary tool-support for Solidity exists, e.g., Securify[5], Smartcheck[6], or MythX[7], but is still in its infancy compared to major programming languages. Alternatively, the Tezos [249] blockchain platform uses the Michelson programming language for smart contract development which is designed to facilitate formal verification [250]. The high-level smart contract programming language Liquidity can be written and compiled to Michelson [251].

In Section 3.1.3.1 we discussed the academic research concepts of a smart contract collaboration and negotiation lifecycle based on [120][136], a distributed governance infrastructure based on [120][140][137] and conflict-related exception- and compensation management during decentralized collaborations [138]. While the conceptual framework exists, a practical implementation is missing. Moreover, privacy concerns of mapping business transactions, interactions and collaborations onto a blockchain system are not addressed in this work.

5.1.3 Data Sources, Data Storage and Data Management

Analogous to the on-chain transaction information stored on the blockchain platform itself, the selection of data sources and the stored data itself is separated into publicly available and private data sets which can be either stored on-chain, or off-chain.

While smart and autonomously acting machines are equipped with sensors themselves to sense their environment, the perceived information is limited in quantity, quality and pertains to a specific geographical location. Further data sources are required for complex cross-organizational M2X applications. Ensuring the integrity of external resources is the topic of on-going scientific research [169]. The Ocean Protocol project [252] offers a decentralized protocol in combination with a marketplace that allows to buy/sell

[4] https://github.com/corda/corda-settler

[5] https://securify.chainsecurity.com/

[6] https://tool.smartdec.net/

[7] https://mythx.io/

data. As a result, machines may monetize their gathered data on the marketplace and purchase further data which they cannot collect themselves. Similarly, the decentralized oracle network ChainLink [253] provides reliable, tamper-proof and blockchain-agnostic inputs and outputs for complex smart contract orchestrations.

On-chain mass data storage and management of the V2X platform is handled by the InterPlanetary File System (IPFS) [147]. IPFS is a P2P hypermedia protocol and distributed file system. It utilizes a content-addressed block storage model that allows to address data sets via hyperlinks, and a generalized Merkle DAG (directed acyclic graph) to build versioned file systems [147]. Alternatively, StorJ [254] can be used for distributed cloud storage. While IPFS and StorJ are designed for storing files, BigchainDB [255] is a suitable solution for complex operations on large data sets and is best suited for storing, indexing and querying structured data – thus it provides complementary features to IPFS and StorJ. BigchainDB is a decentralized database with additional blockchain-enabled features such as immutability as well as the creation, control and transfer of digital assets. The BigchainDB network is designed to host nodes owned and controlled by different entities or organization and therefore well suited for a multi-stakeholder ecosystem that requires interoperable data containers instead of classical data silos.

5.1.4 Identity and Authentication

In Chapter 4 we explain the necessity for an identity solution as well as a validation and authentication mechanisms in M2X scenarios which enable collaboration, interactions and transactions among entities of our ecosystem. The solution presented in Chapter 4 is demonstrated via proof-of-concept implementations and provides – despite its early stage of development – the general mechanisms required for a M2X identity solution. However, a mechanism to safely store and use key pairs is required. Moreover, recovery and backup options are missing as well.

Secret sharing [202][203] is a common approach to recover lost keys. To do so, a user's (private) key is shared among a number of trusted entities and at least two of them are required to restore the secret key. An advantage of this solution is the availability of a key recovery mechanism, but it comes with the downside of handling additional keys and the requirement of sufficient trusted persons. Furthermore, for non-technical users, the concept of public-key-cryptography alone is complicated enough; adding the concept of shared secrets demands to much from non-security experts [256][257].

The M2X ecosystem with different OEMs, service providers and consumers, users, infrastructure providers, etc. is a multi-stakeholder system. As a result, multi-signature transactions in which multiple entities sign a transaction before broadcasting it to the blockchain network are also needed. A variety of multi-signature wallets exist, e.g., the Gnosis Wallet[8]. Storing the private keys corresponding to the multi-signature transactions on multiple servers further increases security [258].

5.1.5 Existing Gaps

The previous sections focused on the paper-based feasibility evaluation for a minimal proof-of-concept based on existing technologies and recent research results. Nonetheless, the current state-of-the-art does not support a full-scale V2X (or M2X) service platform and application deployment, instead only isolated applications and minimal proof-of-concept use cases can be demonstrated.

Specifically for the V2X ecosystem, a broad adoption of connected, or even autonomous vehicles with advanced sensing, computing and communication capabilities is desirable. Likewise, the deployment of VANETs and the corresponding infrastructure (e.g., RSUs) is required as well. General 5G coverage accounts for the increased bandwidth requirements of M2X applications and reduces the number of required RSUs. Merging the currently used external hardware boxes (as described in Section 5.1.1) into the vehicle itself is beneficial and reduces system complexity, the need for extra hardware and fosters standardization and compatibility.

Instead of acting as an external hardware with external software, OEMs can integrate the same functionality as part of the vehicle's software stack, e.g., via the AUTOSAR Adaptive [259] platform. AUTOSAR Adaptive is a standard that defines interfaces required to develop future automotive ECUs (Electronic Control Units) running on multi-core microprocessors. These interfaces allow automotive OEMs to implement autonomous driving functionalities and IoT capabilities as well as further services on their vehicles.

Even though blockchain technology as a concept was already introduced in 2008, its software development, its management tools and the technology itself have still not matured. Smart contract programming languages such as Solidity are subject to frequent and often incompatible changes. The available software development tools are rudimentary and not comparable to tool-support of major programming languages like C++ and Java. Albeit first tendencies to blockchain interoperability are researched, production-ready solutions, or even an industry-standard are missing. The relay-chain approach of Polkadot might sound promising – however, instead of having a single relay-chain hub for all existing blockchain platforms, we might end up in a fragmented ecosystem with many relay-chain hubs that all aim to become the most popular platform while still being not interoperable among each other. The issue of blockchain interoperability is closely

[8] https://wallet.gnosis.pm/

related to platform interoperability of V2X platforms operated by different OEMs, infrastructure operators, or service providers. Only a network of interoperable platforms allows for an open ecosystem that facilitates the M2X economy. Further challenges arise in the context of blockchain scalability which is currently limited.

Advancements of the reasoning and decision making proficiency is obligatory to extend existing IoT and CPS applications to a status where smart devices are capable of acting as autonomous business entities. Additionally, until that point, (smart) devices are prone to human actors exploiting the limited capabilities of machines in the M2X ecosystem.

A conflict resolution management system based on existing research is introduced in previous sections. While the system is well suited for M2M and M2I scenarios that follow standardized interaction patterns with few ambiguities, conflicts that require human mediation services occur in M2H scenarios due to the fundamental differences of machine-readable and precisely defined business contracts and humans understanding of contractual obligations.

Finally, the M2X economy raises a series of legal questions and concerns. Externally owned machines, e.g., by a company, or the government, are less challenging from a legal perspective than machines that own themselves. Assuming that such a legal status could exist, further questions regarding liabilities, compliance and litigation follow. Furthermore, for an economy to exist that bridges between humans and machines, legal obligations and laws have to be available in human-, and machine-processable format.

5.1.6 Generalizability – V2X to M2X

In the previous sections of this chapter, we demonstrate the technical feasibility of the V2X platform using a paper-based evaluation method. Moreover, we present a selection of three V2X running cases in Section 1.1.2 that can be provisioned based on the V2X platform. Extending the same concept to a generalized M2X economy is more complex but attainable and allows for production, transport, trading and provision of services and goods. Nonetheless, we cannot prove that all use cases of the upcoming M2X economy can be implemented via a system similar to the V2X platform presented in this work.

As mentioned in Section 1.1.2.1, the TaaS concept is not limited to human transportation as described in Chapter 3 and instead applies to transportation of goods via drones, ships, planes and trains as well. Similarly, the notion of trading road space as a scarce resource can be extended to further goods, commodities, data and even services. Blockchain-based electricity trading solutions already exist [260][261] as well as sensor data marketplaces [262][263] and may serve as specific illustrations for such exchange platforms with integrated value transfer.

Finally, autonomously acting smart factories that independently ensure sufficient supply of required resources for production, a well-organized production process, supply and demand management as well as the required logistics of transporting resources to the factory and the final products to the customer are among the most complex applications of the M2X economy.

Following the provided examples above we conclude the generalizability of the presented V2X platform to a broader M2X spectrum. Whether all future M2X applications can be provisioned in this manner is subject to future research.

5.2 Authcoin State-Space Analysis

The proposed protocol for validation, authentication and identity provision in decentralized networks is evaluated in two ways. First we presents the state-space analysis results of the corresponding CPN models of the protocol. Second, a minimal proof-of-concept implementation of the protocol is listed in Appendix B.3.6. The proof-of-concept is implemented on the EVM-compatible Qtum [139] blockchain infrastructure does not cover a fully-fledged implementation and rather tests the general feasibility of the concept. Note that a further analysis of the CPN models from a security perspective exists, but is not covered as part of this work. In [227], a risk- and threat analysis based on the Information Systems Security Risk Management (ISSRM) domain model [264][265] is performed on the existing CPN models. Afterwards, the identified risks are mitigated using security risk-oriented patterns [266][267] (SRPs) – a means to mitigate common security- and privacy risks in business-processes.

The following section focuses on evaluating the created Authcoin CPN model by performing state-space analyses. In order to avoid a state-space explosion [57], the model is separated into sub-modules and a full state-space is calculated for each. From the results of the analyses, we derive model properties and explain their implications.

During a state-space analysis, all reachable states and state changes of a given CPN model are calculated and represented in a directed graph, where the nodes correspond to the set of reachable markings and the arcs correspond to occurring binding elements [57]. From this graph, it is possible to deduce properties of the CPN models and the systems presented by the models. The state-space analysis used in this work is generated using built-in functionalities of CPN-Tools. Besides a full state-space analysis, a SCC (Strongly Connected Component) graph is calculated based on the directed graph of

Module	**Loops**	**Home markings**	**Dead markings**	**Dead transitions**	**Live transitions**
Key Generation Establish Binding	No	No	Yes	No	No
Formal Validation	No	No	Yes	Yes	No
Validation & Authentication	No	No	Yes	Yes	No
VAR Creation	No	No	Yes	No	No
Process VAR	No	No	Yes	Yes	No
Revocations	No	No	Yes	Yes	No

Table 5.1: Selected State-Space Analysis Results of the Authcoin CPN Models – Based on [176][175]

the state-space analysis. The nodes of the SCC graph are obtained by making a disjoint division of the nodes in the state-space such that two state-space nodes are in the same SCC if and only if they are mutually reachable, i.e., there exists a path in the state-space from the first node to the second node and vice versa [57]. The SCC graph is used to deduce further model properties, e.g., if the SCC graph has fewer nodes than the state-space graph then at least one cycle exists in the state-space graph of the CPN model.

Since a full computational verification of the whole CPN model is not feasible for this size of models and causes a state-space explosion, all parts of the models are tested independently. As presented in Table 5.1, the CPN model is split into six sub-modules, that are tested with prepared input statements that aim to cover as many execution paths as possible without causing a state-space explosion. The *KeyGenerationEstablishBinding* module is depicted in Figure 4.8, the *FormalValidation* module and the *V&A* module are depicted in Figure 4.9 and the *Revocations* module in Figure 4.11. The *VARcreation* module refers to a slightly modified version of the *SymbolicMining* module illustrated in Figure 4.10. The *ProcessVAR* module consists of the *Mining* module in combination with the *V&A* module, minus the *SymbolicMining* module.

A full state-space is calculated for all modules listed above, followed by the calculation of the SCC graph. During these calculations, all other parts of the CPN models have been disabled. Relevant results and selected properties derived from the analysis are presented in Table 5.1. It is important to keep in mind that the properties of the separated modules might differ from the properties of the whole CPN model itself due to the combination and influences of the different components on each other. Nevertheless, verifying the correct execution and behavior of sub-modules strengthens the assumptions that the overall protocol performs as intended. The complete results of the state-space analyses are available in Appendix A.1.5.

As shown in Table 5.1, none of the tested modules contain any loops. Thus, there are no infinite occurrences of execution paths in the state-space graph which guarantee the termination of each module. Still, it is possible to deduce from the design of the Authcoin protocol that there are loops in the complete model, since the blockchain architecture causes loops when chaining new blocks to the blockchain.

Furthermore, the state-space analysis shows the absence of any home markings. A home marking is a marking that can be reached from any other reachable marking, meaning that it is impossible to have an occurrence sequence that cannot be extended to reach the home marking.

The detected dead markings are caused either by intentionally disabling certain parts of the CPN models or customized input values that prevent a state-space explosion. “A dead marking is a marking in which no binding elements are enabled” [57]. The existence of at least one dead marking guarantees a termination of executable actions at a certain point, thereby preventing infinite runtime. Since all modules contain a dead marking, none of them has a live transition. By definition, “a transition is live if from any reachable marking we can always find an occurrence sequence containing the transition” [57].

All detected dead transitions are caused by intentionally disabling execution paths and prepared input statements. A transition is considered dead if there is no reachable marking that enables the transition. Since all occurrences of dead transitions are artificially enforced, it means that all transitions of all tested modules can be potentially enabled at a certain point during the protocol execution [57].

5.3 UnchainedX Evaluation

An evaluation of the UnchainedX protocol is performed as follows: First, we evaluate the security guarantees – which mainly depend on the target difficulty level as well as the token price of underlying PoW blockchain – by deploying a fictional network. For this evaluation, we chose the Bitcoin and the Ethereum blockchain as the most popular and utilized PoW chains. The combination of evaluation both blockchains covers important corner cases of changing difficulties and token prices, e.g., increasing and decreasing difficulty in combination with sudden drops and raises. The evaluation is performed on a dataset ranging from December 2016 to mid May 2019. While neither future developments of the underlying cryptocurrency price, nor the target difficulty can be reliably predicted, the following analysis provides an intuition on the historical worst case performance of our protocol. A proof-of-concept implementation of UnchainedX on

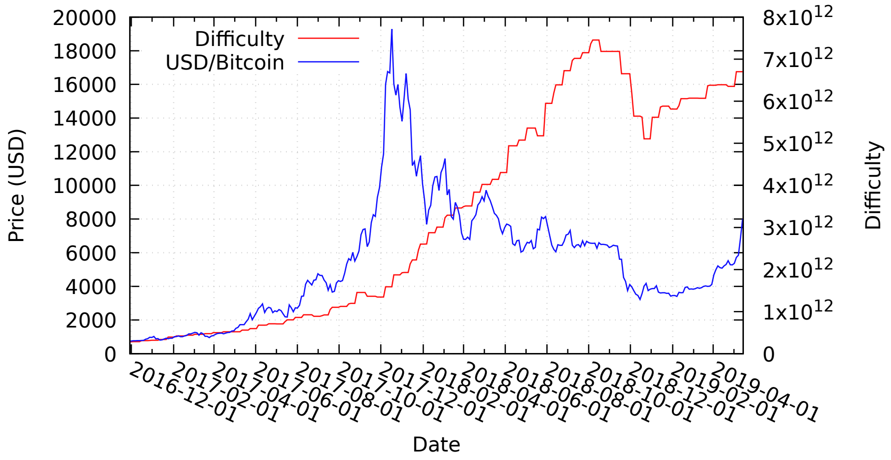


FIGURE 5.1: Average Daily Price of Bitcoin in USD and Block Difficulty Level Between December 2016 and May 2019 – Partially based on [177], Data Source [268]

the Bitcoin and the Ethereum blockchain is implemented and deployed on a Raspberry Pi to demonstrate the general feasibility. Finally, we present the state-space analysis of the UnchainedX CPN model.

5.3.1 Bitcoin Price and Difficulty Analysis

As illustrated in Figure 5.1 the target difficulty level of the Bitcoin blockchain is steadily increasing with minor decreases in between until October 2018, followed by a major decrease until January 2019 before increase again for the rest of the evaluation period. The Bitcoin token price follows a similar pattern, but with a high-price peak at the end of December 2017. The lowest token price appears around the same time as the difficulty drop of January 2019, followed by a recovery of the token price until the end of the evaluation period.

In Section 4.3.2 we discussed the lower bound security guarantees of UnchainedX which depend on the lowest level of the target block difficulty and the lowest token price equivalent per block that occurred during the existence of a particular network. In the context of our hypothetical evaluation network deployed in December 2016, we observe that all nodes joining later than December 2016 have higher security guarantees than the initial bootstrap nodes due to an increased block difficulty and token price. Even though token price and target difficulty decreased heavily between the end of 2017 and the end of 2018, the lowest price and difficulty levels still remained above the deployment levels. However, since the actual minimum price per identity is determined by the network operator the token price and difficulty level only represent a theoretical measurement for security guarantees. Nonetheless, a higher token price results in an increased block price which may result in more expensive identities, thereby raising the costs of sybil node attacks.

While deploying a network at the lowest price of our evaluation period is not an issue, the opposite applies to deploying a network at the Bitcoin price high-point in the beginning of November 2019 – days before the decreasing Bitcoin price as illustrated in Figure 5.1. As a result of the decreasing price, it becomes generally less expensive to introduce new identities to the system for a short period of time. Again, the actual pricing structure depends on the network operator determining the minimum price of an identity. For practical reasons it is likely that most operators pick minimum values below the current token price and difficulty level therefore allowing for price/difficulty declines that do not affect the security guarantees of our example network. However, in case of the substantial price and difficulty decline between the end of 2017 and the end of 2018 as illustrated in Figure 5.1, a pricing and difficulty update is probably required.

Next, we analyze the occurrence of substantial token price declines for the chosen evaluation period by calculating, for each potential deployment date of our network, the highest drop in price and thus security level experienced by the network. Table 5.2 presents the proportion of starting dates that would have lead to a decline on any subsequent day of at least a given percentage. In 0.0% of possible starting dates the security level would have at any later point dropped below 10% of the given date while for below 20% it is only 1.3%.

Drop to	**Affected start dates**
< 10%	0.0%
< 20%	1.3%
< 30%	7.4%
< 40%	16.7%
< 50%	32.4%
< 60%	44.8%
< 70%	46.5%
< 80%	54.5%
< 90%	64.9%
< 100%	79.6%

Table 5.2: Affected starting dates after which the Bitcoin price drops below a certain percentage of the given day's price between December 2016 and May 2019.

Historically, substantial declines occur very rarely. Smaller declines occur more frequently, with almost 65% of possible starting days experiencing drops of at least 10% at some point in the future. While most networks are able to tolerate smaller drops in security level, raising Bitcoin prices can also be an issue, as they can make identities too expensive for regular users. Considering this for networks intended to exist over long time frames, provision of an update mechanism for networkParameter$_{\text{amount}}$ should be made. In case mass adoption occurs, the volatility level of cryptocurrencies and fiat currency is expected to converge. Hence, UnchainedX's level of security is supposed to stabilize similarly.

5.3.2 Ethereum Price and Difficulty Analysis

UnchainedX is blockchain-agnostic and only requires the underlying blockchain platform to utilize a PoW consensus mechanism. Therefore, besides the Bitcoin blockchain, we also analyze the changing difficulty levels and token prices of the Ethereum blockchain and how they would have affected the sybil attack prevention mechanism of our fictional network deployed in December 2016. Figure 5.2 presents the history of the Ethereum token price and the Ethereum block difficulty of the same evaluation period as for the Bitcoin evaluation. Similarly to the Bitcoin token price, the Ethereum token price also increased heavily between December 2016 and the end of December 2017, followed by a large drop until April 2018. Afterwards, a price recovery in May 2018 was followed by a further drop until December 2018 before slowly starting to increase again until the end of the evaluation period.

The Ethereum difficulty increases steadily until October 16, 2017 before a sudden drop due to a difficulty adjusting hard-fork of the Ethereum network [269]. In March 2018, the difficulty surpassed the previous all-time high of October 2017. A significantly lower difficulty level is achieved in the beginning of 2019, followed by a short term recovery and a further drop of the difficulty level in March 2019. Despite heavy reoccurring decreases of the difficulty level, even the reduced difficulty level is far higher than the initial level in December 2016.

Hence, the security guarantee evaluation results are similar to the Bitcoin evaluation of the previous Section 5.3.1. Nodes deployed in December 2016 with the initial difficulty are cheaper and easier to create in terms of identity price and block difficulty. All nodes deployed at later points in time provide higher security guarantees. When focusing on the timeframe briefly before and after the difficulty adjustment as well as the difficulty drop in early 2019, identities created before the adjustment and the drop are less difficult to create than identities created afterwards. The same applies for the price of identities both before and after the price declines of Ether in 2018 as illustrated in Figure 5.2.

Finally, assuming that the PoW blockchain target difficulty levels will not increase indefinitely and remain somehow static (with minor fluctuations) at some point in the future, UnchainedX's lower and upper bounds are likely to converge as well and be less volatile.

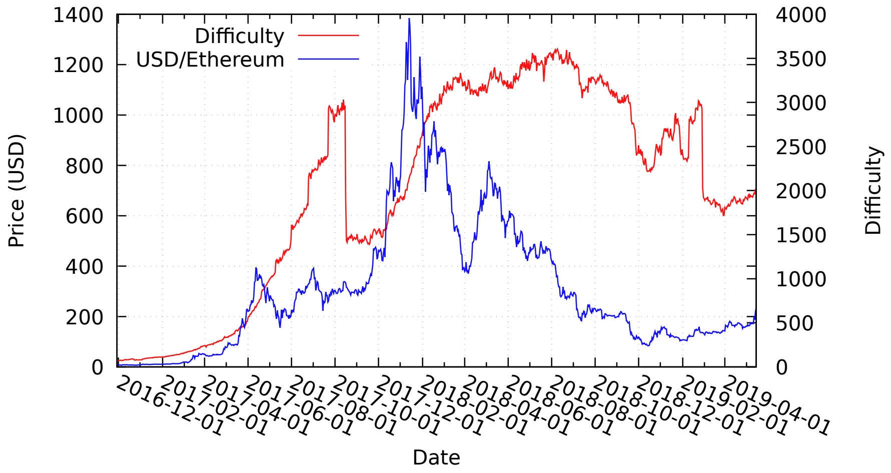


Figure 5.2: Average Daily Price of Ether in USD and Block Difficulty Level Between December 2016 and May 2019 – Partially based on [177], Data Source [270]

5.3.3 Proof-of-Concept Implementation

While Section 5.3.1 and Section 5.3.2 evaluate UnchainedX with regards to security guarantees in the context of different PoW blockchain platforms, the applicability and feasibility within the context of IoT devices has not been covered yet. To do so, a proof-of-concept implementation of the original Unchained protocol based on the Bitcoin and Ethereum blockchain was created and is available on Github (see Appendix B.3.5). The prototypes are relying on public/private key pairs instead of DIDs to simplify the proof-of-concept. However, this does neither affect provided security guarantees, nor the performance of the prototypes. In order to evaluate the prototypes in a common IoT scenario, we choose a Raspberry Pi 3 as an evaluation platform.

For the Bitcoin blockchain, the size of an identity proof varied between 10KB-50KB depending on the size of the block. The time to verify an identity proof was around two seconds without further optimizations. In case of Ethereum, a proof consumes around 50KB-150KB of storage depending on the block size, but takes around 60s to be verified. The slow verification time is caused by a deliberate design choice of the Ethereum hash function Ethash [271] to achieve ASICS resistance. As a result, deploying the protocol on a Bitcoin-like PoW blockchain is more practical. However, other PoW blockchain platforms that do not rely on Ethash or similar algorithms with the same property are suitable as well.

Since the identities are deployed to networks by the network operators or device manufactures, we assume that the proofs itself are not generated on the actual devices – hence,

Loops	**Home markings**	**Dead markings**	**Dead transitions**	**Live transitions**
No	No	Yes	No	No

Table 5.3: State-Space Analysis Results of the UnchainedX CPN Model

we did not conduct any performance benchmarks for creating proofs on the Raspberry Pi 3.

5.3.4 State-Space Analysis

The state-space analysis of the UnchainedX CPN model follows the same approach as the state-space analysis of the Authcoin CPN model in Section 5.2. However, due to a lower level of complexity, the mode is not separated into sub-modules. Instead, the analysis is performed on the complete CPN model. Results and selected properties derived from the analysis are presented in Table 5.3, while the complete state-space analysis is available in Appendix A.2.3. As presented in Table 5.3, the CPN model does not contain any loops. Therefore, no infinite occurrences of execution paths in the state-space graph exist which guarantees the termination of the model. The analysis also shows the absence of home markings which implicates that it is impossible to establish a path that cannot be extended to reach a home marking. The detected dead markings are caused by customized input values that prevent a state-space explosion. The existence of at least one dead marking also guarantees a termination of executable actions at a certain point, thereby preventing infinite runtime. Due to the CPN model containing a dead marking, it does not have a live transition. No occurrences of dead transitions are detected, therefore all transitions of the model can be potentially enabled at a certain point during the protocol execution.

5.4 Discussion

A major limitation of this work is its conceptual nature, which is partially caused by the fact that sufficiently *smart* machines, or devices that are able to autonomously act in a (self-sovereign) business context, are not available yet. Instead of tackling technical details and presenting specific implementations, we introduce an overall concept. Therefore, the paper-based feasibility evaluation of Section 5.1 presents a simplified proof-of-concept based on existing technological solutions. However, as outlined in Section 5.1.5, a substantial technical gap exists. Bridging this technological gap touches not only various related research areas, but also requires further decades of research and development. While we extrapolate the generalizability of the V2X platform to further M2X applications other, yet unknown, or not considered M2X use cases may not be covered.

While the combination of DIDs, Authcoin and UnchainedX provides a suitable identity-, authentication- and validation mechanism for entities participating in the M2X economy, the presented solution still misses a comprehensive integration of the three building blocks. Moreover, cross-organizational integration across different stakeholders, data silos and domains is missing. The same applies to industry-standard implementations within the M2X context. For vehicle-specific use cases, the recently presented vehicle identity standard might be a suitable candidate [272]. Integrating the vehicle identity standard – that uses a DID-structure – with the proposed validation and authentication mechanism of Authcoin and the UnchainedX identity proofs is desirable.

Further limitations of the Authcoin evaluation result from the customized input statements used to prevent a state-space explosion. Even though the input statements are designed to cover as many executions paths of the state-space graph as possible, certain places and transitions with minor relevance we omit intentionally in order to avoid a state-space explosion. For the same reason, the number of iterative executions of the *VARcreation* module has been limited artificially in such a manner, that only one VAR is created. Further limitations result from the modeling process itself: In order to constrain the modeling complexity, we decided to implement the mining process in a symbolic way, artificially limit the number of processed VARs to two. Moreover, the CPN models do not contain limitations with regards to which user can fulfill a VAR, or not in contrast to the protocol description in [68]. Furthermore, due to the sociotechnical nature of the Authcoin protocol, certain aspects of the model are simplified, such as STRING-based placeholder challenges and randomized variables at different places and transitions in order to simulate decisions of external entities. Additional limitations originate from the limited scripting capabilities within CPN Tools, e.g., the implemented hashing functionality does not provide real hashing properties. Similar applies to the symbolic implementation of public-key cryptography which only allows for the symbolic signing of hashed data records.

Comparable constraints apply to the UnchainedX CPN model. These result from the customized input statements of the model as well as the modeling process itself which requires several simplifications. We did not implement an actual consensus algorithm and mining process for the underlying blockchain platform. Moreover, we simplified data structures of the UnchainedX protocol and the blockchain platform, e.g., no Merkle trees, blocks have no nonce, a simplified calculation of UnchainedX's proofID calculation. Again, limitations originate from the restricted scripting capabilities of CPN Tools resulting in the same limitations regarding the hashing-functionalities and public-key cryptography as the Authcoin CPN model.

As a cryptocurrency-based protocol, the UnchainedX concept currently suffers from volatile cryptocurrency prices that complicate the everyday use of such protocols. Moreover, UnchainedX may allow network operators to determine the cost of a sybil node attack, but does neither prevent such an attack, nor help to detect sybil nodes. A combination of Authcoin, UnchainedX and existing sybil node detection methods is desirable.

Chapter 6

Conclusion and Future Work

The following Chapter 6 concludes this work and suggests future research directions in order to extend as well as to explore its key findings further. First, Section 6.1 summarizes the research results and answers the posed research questions. Afterwards, Section 6.2 focuses on future work.

6.1 Conclusion

This thesis focuses on the emerging M2X economy in the context of IS research and makes three contributions: First, it suggests architectural concepts that encompass an interaction-, transaction- and collaboration model for M2X applications, a business collaboration lifecycle and governance structure as well as a set of modalities for these use cases derived through an exploratory research approach. Second, it presents a decentralized self-sovereign identity solution in combination with a validation and authentication mechanism that is suitable for the M2X ecosystem. Sybil attacks are a common issue of large-scale P2P networks, where hostile or faulty computing elements threaten the security of the whole network. Therefore, we present a mechanism to price the costs of a sybil node attack, thereby providing an easy to use metric for the sybil resistance of a decentralized M2X system. As step towards a formal validation of these novel infrastructural concepts, a Colored Petri Net model is provided covering the protocol-driven data exchange of the M2X identity solution. The developed identity protocols are validated using CPN models and proof-of-concept implementations, while specific aspects of the presented M2X identity solution are evaluated using historical data to asses its suitability. Finally, the feasibility of the M2X interactions-, transactions- and collaboration model as well as the identity solution is demonstrated.

6.1.1 Collaboration, Value Exchange and Governance

Based on the TaaS running case from the field of V2X use cases – as a sub-category of M2X – a blockchain-based, platform- and manufacturer-agnostic interaction, transaction and collaboration model is developed that enables a V2X platform for goods and services. We outline and describe the technical foundations and the three running cases of V2X transactions and interactions, e.g., vehicle-to-vehicle (V2V), vehicle-to-human (V2H), or vehicle-to-infrastructure (V2I). Based on the running cases, the requirements and criteria of the proposed solution are identified. Concerning functional and non-functional requirements, we envision a blockchain and manufacturer agnostic and interoperable V2X platform with a plug-in interface for external applications. Subsequently, we derive the service-oriented architecture of the system based on the identified requirements and goals. We present the system architecture using technology-agnostic UML-component diagrams that detail the system's main components and communication interfaces.

A core element of many of the use cases is the smart contract-based negotiation and contract enactment among entities which are the result of collaborating tasks and subprocesses. On an abstract level, most of the examples presented in this work follow a similar workflow. Hence, we decide to integrate a smart-contract negotiation lifecycle that is divided into five stages (preparatory, negotiation, contract execution, rollback and contract expiry stage) that we explain in detail. Furthermore, we propose an auction algorithm for the V2X economy that allows to reach an efficient consensus on a price between buyer and seller. The auction mechanism can be executed on-, or off-chain.

Blockchain technology and smart contracts specifically are an essential part that enables the V2X/M2X economy to offer a wide range of new business- and transaction models. Those new models require a variety of context-specific process modalities. We derive a preliminary set of context-specific process modalities. In consequence, we identify the modalities of *Accountability and Logging*, *Privacy*, *Trust*, *Market Behavior*, *Interoperability* and *Environment Integrity.*

The technical feasibility of the proposed platform is evaluated using a paper-based feasibility evaluation method. We demonstrate that a simplified proof-of-concept implementation is feasible with existing technologies and recent research results. The required hardware and infrastructure components exist and the Ethereum blockchain is suitable to serve as the underlying smart contract platform. External data-feeds may be collected via sensors, or data sharing, and trading platforms. The accumulated data is either stored on-, or off-chain depending on the context and size. While a proof-of-concept implementation is feasible with existing solutions, a technical gap remains with respect to an industry-ready system. Finally, we demonstrated the generalizability of the V2X platform to further applications within the M2X spectrum.

6.1.2 Validation, Authentication and Identities

In order to enable secure business collaborations, interactions and transactions within a digital economy, the digital representation is required to establish and enable trust, reputation mechanisms, perform verifiable and accountable transactions and establish verifiable as well as auditable data provenance. Especially in the context of hardware devices, humans and software agents – that/who all require a digital representation of their "real-world" identity to conduct digital business transaction, or enact digital collaborations – a digital representation mapping to the analogue identity is necessary.

Identity management in the M2X ecosystem is a multi-stakeholder issue that involves not only its users, but also OEMs, infrastructure providers, regulators and various service providers. A single central authority that governs the identity management for all stakeholders is unlikely and poses the risk of a single point of failure. Moreover, identity data silos raise privacy concerns and suffer from interoperability issues, i.e., lock-in effects. Thus, a centralized identity solution is not an option and a decentralized and interoperable solution is required that fosters an open M2X ecosystem. For this purpose, the concept of DIDs as representations for self-sovereign identities is utilized in combination with a blockchain-based authentication- and validation mechanism – inspired by the Authcoin protocol. Authcoin is a challenge-response based protocol for authentication and validation in decentralized networks that is able to: *i*) prove control over an asymmetric key pair linked to a DID document (validation) and *ii*) to produce verifiable claims that can be used to authenticate an entity. By documenting the communication process of the bidirectional validation and authentication mechanism on a blockchain system, a transparent and auditable as well as tamper-resistant log is created that makes it difficult for adversaries to introduce malicious identities/keys into a network. The Authcoin protocol is formalized using Colored Petri Nets and an agent-oriented modeling methodology resulting in a sound CPN model. The utilized modeling strategy is explained, the top level model and the refined sub-modules are illustrated and described. We also present the required protocol semantics by defining the necessary token color sets representing the used data structures. The state-space analysis of the Authcoin protocol's CPN model shows no infinite loops and the reachability of a home marking.

To combat the prevalent issue os sybil attacks in decentralized networks, we present a mechanism to price the costs of a sybil node attack, thereby providing an easy to use metric for the sybil resistance of a decentralized M2X system. A protocol for a decentralized blockchain-based identity system is introduced that allows for offline verification while also raising the cost of introducing high numbers of sybil nodes to a network by using economic disincentives. Circumventing the protocol and introducing a sybil node is equivalent to investing more financial assets than it costs to create a malicious block on the underlying blockchain platform. The protocol uses blockchain technology to bind identities to blockchain-based wallet addresses, i.e., public/private key pairs.

In order to join the network, for each entity an identity proof is created. The proof is derived from a deposit transaction that transferred the deposit from the entity's wallet address to a pre-defined deposit address. Users validate each others' identities using the uniquely generated identity proofs. We discuss the corresponding network parameters, update mechanisms as well as upper- and lower-bounds of security guarantees. Finally, specific aspects of the presented M2X identity solution are evaluated using historical data to assess the suitability of the proposed system based on past events. The evaluation is performed using data from the Bitcoin, and Ethereum platform to demonstrate the security guarantees of binding an identity to a blockchain-system. Once more, the protocol is formalized using CPNs following the same methodology and evaluation approach as for the Authcoin protocol. Again, the state-space analysis shows no infinite loops and the reachability of a home marking. Furthermore, the formal model is suitable to guide future implementations that go beyond the proof-of-concept implementation presented in this work.

6.2 Future Work

Since the M2X economy is still in its infancy, the contribution of this work have to be considered initial research work in this area. Therefore, the limitations and challenges in this work provide opportunities for future research. In addition to the gaps identified in Section 5.1.5, this section briefly discusses a selection of open issues and ideas.

Further research directions are derived from the demarcations of this work listed in Section 1.4.7. This work spared to discuss the privacy and security concerns of the M2X ecosystem. Blockchain-based smart contract platforms enable inter-organizational processes in untrusted environments. However, mapping business processes to smart contracts results in privacy challenges, e.g., [273]. The validation and authentication mechanism utilized by Authcoin also discards the issue of privacy and requires further research on how to integrate, design and validate privacy-preserving challenges without stating personal information on the blockchain. Still, even if no explicit personal information is directly stored on-chain, a lot of information can be deduced from pseudonymous on-chain data as demonstrated on the Bitcoin blockchain [155][274].

In the information system as well as the technical context, data security and physical security of involved entities has to be ensured. As part of business transactions and collaborations, personal information is exchanged, stored and processed. Moreover, as part of collaborations processes, business partners reveal potentially sensitive business information to each other. A privacy-preserving and secure process model is required. While in IT-security the protection goals and the mechanisms that guarantee them are relatively well understood, the potentials risks that may arise from machines, or agents acting autonomously are less clear. For example, malicious market participants who conduct atypical price-, or condition negotiations must be distinguished from genuine participants. Furthermore, illegal price agreements among agents raise interesting questions in the context of antitrust laws.

While M2X devices are equipped with a variety of sensors to sense their surroundings, complex M2X scenarios require information beyond the sensing capabilities of a single machine. Internet connectivity enables machines to access large amounts of data – either for free or after performing a payment. However, determining which data-streams, data-sources, or blockchain oracles are reliable, safe and trustworthy is difficult to predict – especially in the context of an open and decentralized ecosystem. Thus, a mechanism to provide reliable, secure/safe and accountable external data-provision for agents in the M2X ecosystem is necessary.

Finally, several questions arise on a legal and sociological level: Within which boundaries are machines allowed to (autonomously) conduct business? Who is liable for maliciously acting machines, or a machine that fails to deliver a service – the owner, the programmer, the operator, the machine? How does the M2X economy change our economic system? How does it integrate into our daily lives? This is just a small subset of questions that may inspire future research. A particular interesting research direction concerns the influence of a universal M2X economy – where most services and work is provided and done by machines – on the concepts of job and employments for humans.

Appendix A

Protocol Formalization

A.1 Authcoin Protocol

A.1.1 CPN Model

https://owncloud.gwdg.de/index.php/s/qzwMrcD22iNrBYx

A.1.2 Goal Model

https://owncloud.gwdg.de/index.php/s/UDNj5VX3Yeeq9nf

A.1.3 Behavior Interfaces

https://owncloud.gwdg.de/index.php/s/AKKS4r6Pr1Ifzif

A.1.4 Protocol Semantics

https://owncloud.gwdg.de/index.php/s/4hyBwIhiOsv69ah

A.1.5 State-Space Analysis

https://owncloud.gwdg.de/index.php/s/Kq482ryYVUkIC4D

A.2 UnchainedX Protocol

A.2.1 CPN Model

https://owncloud.gwdg.de/index.php/s/PZhGh2kvbnm8EBV

A.2.2 Protocol Semantics

https://owncloud.gwdg.de/index.php/s/9f0t3t7dFQM4ZRB

A.2.3 State-Space Analysis

https://owncloud.gwdg.de/index.php/s/jbY9odGt7F1kpAl

Appendix B

Publications

B.1 Peer-Reviewed

B.1.1 Self-managed and Blockchain-based Vehicular Ad-hoc Networks

Abstract: *Combining Vehicular Ad-hoc Networks (VANETs) and Ethereum's blockchain-based application concepts enables transparent, self-managed and decentralized system which are self-regulating and in no need of a central managing authority.*

Reference: Benjamin Leiding, Parisa Memarmoshrefi, and Dieter Hogrefe. Self-Managed and Blockchain-Based Vehicular Ad-Hoc Networks. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pages 137–140. ACM, 2016. Heidelberg, Germany. doi: 10.1145/2968219.2971409. URL http://doi.acm.org/10.1145/2968219.2971409.

B.1.2 Authcoin: Validation and Authentication in Decentralized Networks

Abstract: *Authcoin is an alternative approach to the commonly used public key infrastructures such as central authorities and the PGP web of trust. It combines a challenge response-based validation and authentication process for domains, certificates, email accounts and public keys with the advantages of a blockchain-based storage system. As a result, Authcoin does not suffer from the downsides of existing solutions and is much more resilient to sybil attacks.*

Reference: Benjamin Leiding, Clemens H. Cap, Thomas Mundt, and Samaneh Rashidibajgan. Authcoin: Validation and Authentication in Decentralized Networks. In *The 10th Mediterranean Conference on Information Systems - MCIS 2016*, Paphos, Cyprus, September 2016.

B.1.3 Mapping Requirements Specifications into a Formalized Blockchain-Enabled Authentication Protocol for Secured Personal Identity Assurance

Abstract: *The design and development of novel security and authentication protocols is a challenging task. Design flaws, security and privacy issues as well as incomplete specifications pose risks for its users. Authcoin is a blockchain-based validation and authentication protocol for secure identity assurance. Formal methods, such as Colored Petri Nets (CPNs), are suitable to design, develop and analyze such new protocols in order to detect flaws and mitigate identified security risks.*
In this work, the Authcoin protocol is formalized using Colored Petri Nets resulting in a verifiable CPN model. An Agent-Oriented Modeling (AOM) methodology is used to create goal models and corresponding behavior models. Next, these models are used to derive the Authcoin CPN models. The modeling strategy as well as the required protocol semantics are explained in detail. Furthermore, we conduct a state-space analysis on the resulting CPN model and derive specific model properties. The result is a complete and correct formal specification that is used to guide future implementations of Authcoin.

Reference: Benjamin Leiding and Alex Norta. Mapping Requirements Specifications Into a Formalized Blockchain-Enabled Authentication Protocol for Secured Personal Identity Assurance. In *The 4th International Conference on Future Data and Security Engineering - FDSE 2017*, pages 181–196, Ho Chi Minh City, Vietnam, 2017. Springer.

B.1.4 Safeguarding a Formalized Blockchain-Enabled Identity-Authentication Protocol by Applying Security Risk-Oriented Patterns

Abstract: *Designing government independent and secure identification- and authentication protocols is a challenging task. Design flaws and missing specifications as well as security- and privacy issues of such protocols pose considerable user risks. Formal methods, such as Colored Petri Nets (CPN), are utilised for the design, development and analysis of such new protocols in order to detect flaws and mitigate identified security risks before deployment. This paper fills the gap, by applying in a novel way a set of security risk-oriented patterns (SRP) to the so-called Authcoin protocol that we*

formalise using CPN. The initial formal model of Authcoin facilitates the detection and elimination of design flaws, missing specifications as well as security- and privacy issues. The additional risk- and threat analysis based on the Information Systems Security Risk Management (ISSRM) domain model we perform on the formal CPN models of the protocol. The identified risks are mitigated by applying SRPs to the formal model of the Authcoin protocol. SRPs are a means to mitigate common security- and privacy risks in a business-process context by applying thoroughly tested and proven best-practice solutions. The goal of this work is to test the utility of SRPs outside of the the usual application domain, to reduce the risks and vulnerabilities of the Authcoin protocol.

Reference: Alex Norta, Raimundas Matulevičius, and Benjamin Leiding. Safeguarding a Formalized Blockchain-Enabled Identity-Authentication Protocol by Applying Security Risk-Oriented Patterns. *Computers & Security*, 86:253 – 269, 2019. ISSN 0167-4048. doi: https://doi.org/10.1016/j.cose.2019.05.017. URL http://www.sciencedirect.com/science/article/pii/S0167404818302670.

B.1.5 Unchained Identities: Putting a Price on Sybil Nodes in Mobile Ad hoc Networks

Abstract: *As mobile ad hoc networks (MANETs) and similar decentralized, self-organizing networks grow in number and popularity, they become worthwhile targets for attackers. Sybil attacks are a widespread issue for such networks and can be leveraged to increase the impact of other attacks, allowing attackers to threaten the integrity of the whole network. Authentication or identity management systems that prevent users from setting up arbitrary numbers of nodes are often missing in MANETs. As a result, attackers are able to introduce nodes with a multitude of identities into the network, thereby controlling a substantial fraction of the system and undermining its functionality and security. Additionally, MANETs are often partitioned and lack Internet access. As a result, implementing conventional measures based on central authorities is difficult. This paper fills the gap by introducing a decentralized blockchain-based identity system called Unchained. Unchained binds identities of nodes to addresses on a blockchain and economically disincentivizes the production of spurious identities by raising the costs of placing large numbers of Sybil identities in a network. Care is taken to ensure that circumventing Unchained results in costs similar or higher than following the protocol. We describe an offline verification scheme, detail the functionalities of the concept, discuss upper- and lower-bounds of security guarantees and evaluate Unchained based on case-studies.*

Reference: Arne Bochem, Benjamin Leiding, and Dieter Hogrefe. Unchained Identities: Putting a Price on Sybil Nodes in Mobile Ad hoc Networks. In *International Conference on Security and Privacy in Communication Systems - SecureComm 2018*, pages 358–374. Springer, 2018.

B.1.6 Enabling the Vehicle Economy Using a Blockchain-Based Value Transaction Layer Protocol for Vehicular Ad-Hoc Networks

Abstract: *The next generation of tightly interconnected vehicles offers a variety of new technological as well as business opportunities. Those vehicles form so called vehicular ad-hoc networks (VANETs) in order to enable vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-human (V2H), or in general vehicle-to-everything (V2X) communication and interaction. A variety of manufacturers started implementing specific use cases, but limited to their own brands and products. However, a platform- and manufacturer-agnostic default standard for interactions and transaction within this new economy is still missing.*

This paper fills the gap in the state of the art by introducing a novel blockchain-based V2X platform that enables a transaction and interaction layer for goods and services required to kick-start the upcoming V2X economy. We present the general functions and features of the system, outline the requirements and goals as well as the architecture of the V2X platform. Moreover, we detail the system engagement processes of the identified stakeholders inside the V2X ecosystem and the theoretical foundations of those interactions and transactions.

Reference: Benjamin Leiding and William V. Vorobev. Enabling the V2X Economy Revolution Using a Blockchain-based Value Transaction Layer for Vehicular Ad-hoc Networks. In *The 12th Mediterranean Conference on Information Systems - MCIS 2018*, Corfu, Greece, 2018.

B.1.7 Blogchain – Disruptives Publizieren auf der Blockchain

Abstract: *Wir stellen ein neues Konzept als Metamodell für das wissenschaftliche Publikationswesen vor. Unser Konzept ist im Kontext eines dreistufigen Phasenmodells digitaler Disruption von Geschäftsprozessen angesiedelt. Die erste Phase besteht dabei aus Technologie ohne Prozessanpassung. Die zweite Phase umfasst eine Prozessanpassung unter der Kontrolle von Intermediären und führt zu unerwünschter aber schwer vermeidbarer Zentralisierung. Die dritte Phase durchbricht schließlich die Vormachtstellung intermediärer Institutionen und nutzt dazu die disruptiven Möglichkeiten der*

Blockchain-Technologie.
Die Anwendung dieser Technologie erlaubt eine Veränderung der Geschäftsprozesse bestehender Zeitschriften, macht die Rolle des Verlags als Intermediär überflüssig und verspricht eine Lösung des Problems der Kostenexplosion in der wissenschaftlichen Literaturversorgung. Wir stellen Ergebnisse einer theoretischen Machbarkeitsstudie vor und präsentieren eine erste Implementierung als Proof-of-Concept. Diese werden als Basis für ein Feldexperiment dienen.

Reference: Clemens H. Cap and Benjamin Leiding. Blogchain – Disruptives Publizieren auf der Blockchain. *HMD Praxis der Wirtschaftsinformatik*, 55(6):1326–1340, 2018. doi: 10.1365/s40702-018-00470-w. URL https://doi.org/10.1365/s40702-018-00470-w.

B.1.8 Disruptives Publizieren mit der Blockchain

Abstract: *Wir stellen ein neues Konzept für das wissenschaftliche Publikationswesen vor. Unsere Vision ist im Kontext eines dreistufigen Phasenmodells digitaler Disruption von Geschäftsprozessen angesiedelt. Die erste Phase besteht dabei aus Technologie ohne Prozessanpassung. Die zweite Phase umfasst eine Prozessanpassung unter der Kontrolle von Intermediären und führt zu unerwünschter aber schwer vermeidbarer Zentralisierung. Die dritte Phase durchbricht schließlich die Vormachtstellung intermediärer Institutionen und nutzt dazu die disruptiven Möglichkeiten der Blockchain-Technologie.*
Die Anwendung dieser Technologie erlaubt eine Veränderung der Geschäftsprozesse bestehender Zeitschriften, macht die Rolle des Verlags als Intermediär überflüssig und verspricht eine Lösung des Problems der Kostenexplosion in der wissenschaftlichen Literaturversorgung. Wir stellen Ergebnisse einer theoretischen Machbarkeitsstudie vor, präsentieren eine erste Implementierung als Proof-of-Concept und diskutieren weitere mögliche Realisierungsformen unseres Ansatzes.

Reference: Clemens H. Cap and Benjamin Leiding. Disruptives Publizieren mit der Blockchain. In *H. Fill, A. Meier (Hrsg.): Blockchain – Grundlagen, Anwendungsszenarien und Nutzungspotentiale*, Springer Vieweg, 2019.

B.1.9 Automated Sensor-Fusion Based Emergency Rescue for Remote and Extreme Sport Activities

Abstract: *Even though technological advances changed and improved our daily life in various ways, the risks and dangers of extreme sport activities (ESAs) still persist* *and the progress of technology had little impact on them. Existing emergency rescue devices for ESAs still require manual activation and do not detect emergency situations autonomously. However, fusing data feeds of simple sensors can easily enhance the functionalities of those devices and allows for the detection of emergency situations and subsequent rescue in the case of injuries. We identify the difficulties and challenges posed by ESAs, the role and potential value of information technology in such activities and example use cases and scenarios. We further present a prototype device for climbers that can detect potentially dangerous fall events.*

Reference: Benjamin Leiding, Arne Bochem, and Luca Hernandez Acosta. Automated Sensor-Fusion Based Emergency Rescue for Remote and Extreme Sport Activities. In *IWCMC 2019 Wireless Sensor Symposium (IWCMC-Wireless Sensors 2019)*, Tangier, Morocco, 2019. IEEE.

B.1.10 Lowering Financial Inclusion Barriers With a Blockchain-Based Capital Transfer System

Abstract: *Transferring money and gaining access to credit across international borders, is still complicated, time consuming and expensive. Existing money transfer systems suffer furthermore from long lines, exchange rate losses, counter-party risks, bureaucracy and extensive paperwork. An estimate two billion adults are unbanked and with no, or limited access to financial services. Providing workable financial services to this population is often tagged as a key step towards eliminating world poverty and bootstrapping local economies. The Everex application focuses on easing the financial inclusion problem by applying blockchain technology for cross-border remittance, online payment, currency exchange and micro lending, without the volatility issues of existing, non-stablecoin cryptocurrencies. Finally, the Everex wallet facilitates a fiat-to-cryptocurrency gateway that eases access to cryptocurrencies, thereby enabling our users to instantly buy and sell tokens without having to visit an exchange. This paper fills the gap in the state of the art by presenting a blockchain-based capital transfer system that aims to lower financial inclusion barriers and provide financial services to the unbanked. We present the advantages of the system, outline the requirements and goals, as well as the architecture of the Everex financial eco-system.*

Reference: Alex Norta, Benjamin Leiding, and Alexi Lane. Lowering Financial Inclusion Barriers With a Blockchain-Based Capital Transfer System. In *CryBlock 2019 - 2nd Workshop on Cryptocurrencies and Blockchains for Distributed Systems (co-located with INFOCOM 2019)*, Paris, France, 2019. IEEE.

B.1.11 Ensuring Resource Trust and Integrity in Web Browsers Using Blockchain Technology

Abstract: *Current web technology allows the use of cryptographic primitives as part of server-provided Javascript. This may result in security problems with web-based services. We provide an example for an attack on the WhisperKey service. We present a solution which is based on human code reviewing and on CVE (Common Vulnerabilities and Exposures) data bases. In our approach, existing code audits and known vulnerabilities are tied to the Javascript file by a tamper-proof Blockchain approach and are signaled to the user by a browser extension. The contribution explains our concept and its workflow; it may be extended to all situations with modular, mobile code. Finally, we propose an amendment to the W3C subresource recommendation.*

Reference: Clemens H. Cap and Benjamin Leiding. Ensuring Resource Trust and Integrity in Web Browsers Using Blockchain Technology. In *Advanced Information Systems Engineering Workshops*, pages 115–125, Cham, 2018. Springer International Publishing.

B.1.12 Dead Letters to Alice - Reachability of E-Mail Addresses in the PGP Web of Trust

Abstract: *Over the last 25 years four million e-mail addresses have accumulated in the PGP web of trust. In a study each of them was tested for vitality with the result of 40% being unreachable. Of the mailboxes proven to be reachable, 46.77% turn out to be operated by one of three organizations. In this article, the authors share their results and challenges during the study.*

Reference: Benjamin Leiding and Andreas Dähn. Dead Letters to Alice-Reachability of E-Mail Addresses in the PGP Web of Trust. In *Baltic Young PhD Conference (BaSoTI 2016) and arXiv:1605.03162*, Tallinn, Estonia, 2016.

B.1.13 Exploring Classroom Response Systems in Practical Scenarios

Abstract: *The increasing number of students per classroom requires new ways of interactions betweens teachers and students. Classroom Response Systems (CRS) aim to solve this problem by enabling feedback for large audiences. We defined and identified requirements of a viable solution and present Tweedback as a example of modern Classroom Response Systems. Tweedback is a web application and provides different types of*

feedback: A chatwall, where the audience can ask questions, a panic-button to provide immediate feedback on the lecturers presentation and multiple choice questions. Tweedback has activeley been used since January 2013. The feedback of our users and our own practical experiences allowed us to identify several issues of Classroom Response Systems and develop suitable solutions.

Reference: Benjamin Leiding, Jonas Vetterick, and Clemens H. Cap. Exploring Classroom Response Systems in Practical Scenarios. In *Baltic Young PhD Conference (BaSoTI 2014)*, Riga, Latvia, 2014.

B.2 Not Peer-Reviewed

B.2.1 Securing the Authcoin Protocol Using Security Risk-oriented Patterns (Master's Thesis)

Abstract: *Designing and developing new security and authentication protocols in the field of computer science is a challenging task. Design flaws and missing specifications as well as security and privacy issues of such protocols pose risks for its users. Formal methods, such as Colored Petri Nets, are utilized for the design, development and analysis of such new protocols in order to detect flaws and mitigate identified security risks. In this thesis, the Authcoin protocol is formalized using Colored Petri Nets in order to detect and eliminate eventual design flaws, missing specifications as well as security and privacy issues. Furthermore, a risk and threat analysis based on the ISSRM domain model is performed on the formal CPN models of the protocol. Subsequently, the identified risks are mitigated by applying security risk-oriented patterns to the formal model of the Authcoin protocol. Security risk-oriented patterns are a means to mitigate common security and privacy risks in processes by applying thoroughly tested and proven best-practice solutions. The goal of this thesis is to reduce the risks and vulnerabilities of the Authcoin protocol using the techniques and approaches mentioned above. In addition, we share the lessons learned during the novel application of security risk-oriented patterns to Colored Petri Nets and evaluate the resulting CPN models using state space analyses.*

Reference: Benjamin Leiding. Securing the Authcoin Protocol Using Security Risk-Oriented Patterns. Master's thesis, University of Göttingen, Göttingen, Germany, 2017.

B.2.2 Enabling the Vehicle Economy Using a Blockchain-Based Value Transaction Layer Protocol for Vehicular Ad-Hoc Networks (Whitepaper Version 1.3)

Abstract: *The next generation of tightly interconnected vehicles offers a variety of new technological as well as business opportunities. Vehicles form so called vehicular ad-hoc networks (VANETs) in order to enable vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-human (V2H), or in general vehicle-to-everything (V2X) communication and interaction. A variety of manufacturers started implementing specific use cases that are limited to their own products. However, a default interaction standard for this new economy is still missing. Chorus Mobility presents a blockchain-based system that enables a manufacturer agnostic platform solution that allows VANET participants to enact and transact any kind of services and goods. This whitepaper fills the gap in the state of the art by introducing a blockchain-based transaction and interaction layer that enables our V2X platform for goods and services required to kick-start the upcoming V2X economy. We present the advantages of the system, outline the requirements and goals, as well as the architecture of the Chorus Mobility V2X platform and eco-system.*

Reference: Benjamin Leiding, and William V. Vorobev. Enabling the Vehicle Economy Using a Blockchain-Based Value Transaction Layer Protocol for Vehicular Ad-Hoc Networks - Whitepaper Version 1.3. URL: `https://bit.ly/2XMby0L`, 2019.

B.3 Supervised Projects

B.3.1 Implementation of a Peer-Reviewing Platform on the Blockchain (Bachelor‘s Thesis)

Author: Fabiola Buschendorf

Abstract: *The blockchain technology offers an infrastructure for decentralized applications, such as cryptocurrencies. The data structure is shared in a peer-to-peer network, in which untrusted members can distribute and process information transparently and verifiable. Modern blockchain platforms allow Turing-complete computation with so-called smart contracts and data storage for application development. Further, desired behavior is rewarded with cryptographic tokens, that can be traded for further services. This thesis takes advantage of those properties and proposes a prototype of an academic* *peer-reviewing platform on the blockchain. Peer-reviewing benefits from decentralization, as large scientific publishing houses possess a monopole on renowned journals, thus they control access to and prices of published scientific articles. Further reports reveal that anonymity and lacking incentivation facilitate fraud within the process and cause poor-quality reviews. In this work, a requirements engineering process identifies associated roles and functionalities of an academic peer-reviewing system. An analysis compares existing blockchain platforms and selects a well suited environment for the system. The architecture of a decentralized web application is designed. Finally, an open-source prototype is implemented using recent tools and frameworks and code examples are explained.*

Link: https://github.com/bleidingGOE/fakechair

B.3.2 Distributed Privacy-Preserving Analyses on the Blockchain (Master's Thesis)

Author: Michael Debono Mrden

Abstract: *Being able to share data between medical institutions is increasingly important for medical research, particularly in areas such as infection prevention and control, oncology, and rare diseases. Furthermore, big data analytics can take advantage of shared data to create more value from additional data. However, sharing sensitive data often encounters bottlenecks because of data privacy, where many times the explicit consent of patients is needed for any of their data to leave a clinical site. An active research area is working on solutions for enabling medical analyses across clinical sites in a privacy-preserving manner, while in another area, blockchain technologies are being embraced to rethink the privacy model in general.*

In this thesis, we examine the current techniques in distributed privacy-preserving analyses in the medical field, as well as privacy-preserving methods on the blockchain, and aim to combine them. We focus on privacy-preserving methods such as differential privacy, Homomorphic Encryption (HE), Secure Multi-Party Computation (SMPC), and Software Guard Extensions (SGX).

Inspired by MedCo, a system that uses HE, that enables groups of clinical sites to privately and securely share patient data and run queries for medical analyses from investigators, and Enigma, a blockchain platform based on Ethereum that improves data privacy through SMPC and SGX, we implement and evaluate a prototype that allows queries for medical analyses to be distributed across clinical sites through a blockchain without compromising privacy.

B.3.3 Distributed Privacy-Focused Driver Scoring System (Bachelor's Thesis)

Author: Simon Niklas Schuler

Abstract: *Modern vehicles have become so advanced and sophisticated at monitoring themselves and the driver that they pose a risk to the privacy of the driver by collecting privacy critical information. Among others especially insurance companies are interested in the data gathered by the on board computer of the vehicle while on the road. This enables the insurance companies to monitor individual drivers and to better calculate the risk of insuring them, thus minimizing the overall cost of insurance. Most current driver surveillance systems are put in place on small scale and by the insurance companies themselves. This creates an environment where the driver is dependent on his/her insurance company because it possesses all collected data. Emerging from this are problems like the possible abuse and leakage of the driver data, the drivers privacy and centralization around the insurance. In this thesis we propose a decentralized and privacy preserving system to address the need of the driver for privacy and safety of his/her data as well as the need of the insurance to monitor the driver to some extent. To achieve privacy for the driver our system implements a scoring mechanism that veils privacy critical driving data from the insurance but still accounts for the need to evaluate drivers in real time. Our system is designed to provide privacy and decentralization on a large scale and it is based on Blockchain technology to achieve important properties needed to overcome the challenges.*

Link: https://github.com/bleidingGOE/Distributed-Privacy-Focused-Driver-Scoring-System

B.3.4 A Modular Implementation of a Decentralized Academic Peer-Review Platform (Master's Thesis)

Author: Luca Hernandez Acosta

Abstract: *The process of scientific publishing has a long history dating back to 1665. With the idea of peer reviewing submitted manuscripts in 1831, one sought to increase the quality of publications, and thus to sort out inferior submissions before publication. Until this day, the peer review process has been deeply manifested in the publishing process. Traditional and established publishers benefit from the peer review process, which is provided free of charge by individuals out of the research community. The current state of*

peer review raises concerns about quality, fairness and costs related to publishing. While open access publication enables global access to research papers without subscription fees, it still fails a fair review process. Peer reviewing is an essential part of scientific publication to validate and evaluate scientific work. However, reviewers are not compensated neither monetary nor in a form of gaining reputation for qualitative reviews. In this thesis we are proposing a decentralized platform for peer reviewing conference or journal submissions. By using distributed technologies like the Blockchain technology and the Interplanetary File System (IPFS) we provide a transparent review process, a reputation option for reviewers and the characteristics of open accessible publications.

Link: https://github.com/HernandezAcosta/Master-Project/tree/master/master-webapp

B.3.5 Unchained Identities: Putting a Price on Sybil Nodes in Mobile Ad hoc Networks (Proof-of-Concept Implementation)

Author: Simon Niklas Schuler

Link 1: https://github.com/bleidingGOE/unchained-cli-btc
Link 2: https://github.com/bleidingGOE/unchained-cli-eth

B.3.6 Authcoin: Validation and Authentication in Decentralized Networks (Proof-of-Concept Implementation)

Authors: Rando Mihkelsaar, Gregor Johannson, Marko Mets, Mart Aarma, and Vladislav Šikirjavõi.

Link 1: https://github.com/bleidingGOE/Authcoin-Qtum
Link 2: https://github.com/bleidingGOE/Authcoin-android
Link 3: https://github.com/bleidingGOE/authcoin-demo-server

B.3.7 Privacy-Preserving Metadata-Queries Using Attribute-Based Encryption (Proof-of-Concept Implementation)

Author: Simon Niklas Schuler

Link: https://github.com/SchulerSimon/metadata-queing-using-abe

Bibliography

[1] Rob van der Meulen. Gartner Says 8.4 Billion Connected “Things” Will Be in Use in 2017, Up 31 Percent From 2016. https://www.gartner.com/newsroom/id/3598917, 2017. (Accessed January 24, 2019).

[2] Amy Nordrum. Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated. https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated, 2016. (Accessed January 24, 2019).

[3] Heather A Horst and Daniel Miller. *Digital Anthropology*. A&C Black, 2013.

[4] Kehua Su, Jie Li, and Hongbo Fu. Smart City and the Applications. In *Electronics, Communications and Control (ICECC), 2011 International Conference on*, pages 1028–1031. IEEE, 2011.

[5] Benjamin Leiding and William V. Vorobev. Enabling the V2X Economy Revolution Using a Blockchain-based Value Transaction Layer for Vehicular Ad-hoc Networks. In *The 12th Mediterranean Conference on Information Systems - MCIS 2018*, Corfu, Greece, 2018.

[6] Per Lynggaard and Knud Skouby. Complex IoT Systems as Enablers for Smart Homes in a Smart City Vision. *Sensors*, 16(11):1840, 2016.

[7] Saurabh Vaidya, Prashant Ambad, and Santosh Bhosle. Industry 4.0 – A Glimpse. *Procedia Manufacturing*, 20:233–238, 2018.

[8] Norbert Wiener. *Cybernetics or Control and Communication in the Animal and the Machine*. MIT Press, 1948.

[9] Yingxu Wang, Witold Kinsner, and Du Zhang. Contemporary Cybernetics and its Facets of Cognitive Informatics and Computational Intelligence. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(4):823–833, 2009.

[10] Dan C Marinescu. *Complex Systems and Clouds: A Self-Organization and Self-Management Perspective*. Morgan Kaufmann, 2016.

[11] Jiafu Wan, Min Chen, Feng Xia, Di Li, and Keliang Zhou. From Machine-to-Machine Communications Towards Cyber-Physical Systems. *Computer Science and Information Systems*, 10(3):1105–1128, 2013.

[12] Jiafu Wan, Di Li, Caifeng Zou, and Keliang Zhou. M2M Communications for Smart City: An Event-Based Architecture. In *2012 IEEE 12th International Conference on Computer and Information Technology*, pages 895–900. IEEE, 2012.

[13] Vojislav B Misic and Jelena Misic. *Machine-to-Machine Communications: Architectures, Technology, Standards, and Applications*. CRC Press, 2014.

[14] Ciprian-Radu Rad, Olimpiu Hancu, Ioana-Alexandra Takacs, and Gheorghe Olteanu. Smart Monitoring of Potato Crop: A Cyber-Physical System Architecture Model in the Field of Precision Agriculture. *Agriculture and Agricultural Science Procedia*, 6:73–79, 2015.

[15] Radhakisan Baheti and Helen Gill. Cyber-Physical Systems. *The Impact of Control Technology*, 12(1):161–166, 2011.

[16] Marco Conti, Sajal K Das, Chatschik Bisdikian, Mohan Kumar, Lionel M Ni, Andrea Passarella, George Roussos, Gerhard Tröster, Gene Tsudik, and Franco Zambonelli. Looking Ahead in Pervasive Computing: Challenges and Opportunities in the Era of Cyber–Physical Convergence. *Pervasive and Mobile Computing*, 8(1):2–21, 2012.

[17] National Science Foundation (NSF). Cyber-Physical Systems (CPS) - NSF 19553. URL: https://www.nsf.gov/publications/pub_summ.jsp?ods_key=nsf19553&org=NSF, 2019. (Accessed May 11, 2019).

[18] Jiafu Wan, Hehua Yan, Hui Suo, and Fang Li. Advances in Cyber-Physical Systems Research. *KSII Transactions on Internet & Information Systems*, 5(11), 2011.

[19] Siddhartha Kumar Khaitan and James D McCalley. Design Techniques and Applications of Cyberphysical Systems: A Survey. *IEEE Systems Journal*, 9(2):350–365, 2015.

[20] László Monostori, Botond Kádár, T Bauernhansl, S Kondoh, S Kumara, G Reinhart, O Sauer, G Schuh, W Sihn, and K Ueda. Cyber-Physical Systems in Manufacturing. *Cirp Annals*, 65(2):621–641, 2016.

[21] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[22] Min Chen, Jiafu Wan, and Fang Li. Machine-to-Machine Communications: Architectures, Standards and Applications. *KSII Transactions on Internet & Information Systems*, 6(2), 2012.

[23] Lucija Ivančić, Dalia Suša Vugec, and Vesna Bosilj Vukšić. Robotic Process Automation: Systematic Literature Review. In *International Conference on Business Process Management*, pages 280–295. Springer, 2019.

[24] Cecilia Gejke. A New Season in the Risk Landscape: Connecting the Advancement in Technology With Changes in Customer Behaviour to Enhance the Way Risk is Measured and Managed. *Journal of Risk Management in Financial Institutions*, 11(2):148–155, 2018.

[25] Institute for Robotic Process Automation in association with Carnegie Mellon University. Introduction to Robotic Process Automation - A Primer. URL: https://irpaai.com/wp-content/uploads/2015/05/Robotic-Process-Automation-June2015.pdf, 2015. (Accessed September 25, 2019).

[26] Jan Mendling, Gero Decker, Richard Hull, Hajo A Reijers, and Ingo Weber. How do Machine Learning, Robotic Process Automation, and Blockchains Affect the Human Factor in Business Process Management? *Communications of the Association for Information Systems*, 43(Art. 19):297–320, 2018.

[27] M Ratia, Jussi Myllärniemi, and Nina Helander. Robotic Process Automation-Creating Value by Digitalizing Work in the Private Healthcare? In *The Proceedings of the 22nd International Academic Mindtrek Conference*, pages 222–227. ACM, 2018.

[28] Jerome Geyer-Klingeberg, Janina Nakladal, Fabian Baldauf, Fabian Veit, WMP van der Aalst, F Casati, R Conforti, M de Leoni, and M Dumas. Process Mining and Robotic Process Automation: A Perfect Match. In *16th International Conference on Business Process Management (BPM)*, pages 124–131, 2018.

[29] Sorin Anagnoste. Robotic Automation Process - The Next Major Revolution in Terms of Back Office Operations Improvement. In *Proceedings of the International Conference on Business Excellence*, volume 11, pages 676–686. De Gruyter Open, 2017.

[30] Rua-Huan Tsaih and Chih Chun Hsu. Artificial Intelligence in Smart Tourism: A Conceptual Framework. *Artificial Intelligence*, 2018.

[31] Stan Franklin and Art Graesser. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 21–35. Springer, 1996.

[32] Volkan Gunes, Steffen Peter, Tony Givargis, and Frank Vahid. A Survey on Concepts, Applications, and Challenges in Cyber-Physical Systems. *KSII Transactions on Internet & Information Systems*, 8(12), 2014.

[33] Jianhua Shi, Jiafu Wan, Hehua Yan, and Hui Suo. A Survey of Cyber-Physical Systems. In *2011 International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6. IEEE, 2011.

[34] Li Da Xu, Wu He, and Shancang Li. Internet of Things in Industries: A Survey. *IEEE Transactions on Industrial Informatics*, 10(4):2233–2243, 2014.

[35] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, 17(4):2347–2376, 2015.

[36] Benjamin Leiding, Parisa Memarmoshrefi, and Dieter Hogrefe. Self-Managed and Blockchain-Based Vehicular Ad-Hoc Networks. In *The Adjunct Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 137–140, Heidelberg, Germany, 2016. ACM. doi: 10.1145/2968219.2971409. URL http://doi.acm.org/10.1145/2968219.2971409.

[37] Benjamin Leiding and William V Vorobev. Enabling the Vehicle Economy Using a Blockchain-Based Value Transaction LayerProtocol for Vehicular Ad-Hoc Networks - Whitepaper Version 1.3. URL: https://bit.ly/2XMby0L, 2019. (Accessed April 20, 2019).

[38] Marco Pavone. Autonomous Mobility-On-Demand Systems for Future Urban Mobility. In *Autonomes Fahren*, pages 399–416. Springer, 2015.

[39] ZJ Chong, Baoxing Qin, Tirthankar Bandyopadhyay, Tichakorn Wongpiromsarn, Brice Rebsamen, P Dai, ES Rankin, and Marcelo H Ang. Autonomy for Mobility On Demand. In *Intelligent Autonomous Systems 12*, pages 671–682. Springer, 2013.

[40] Jeffery B Greenblatt and Susan Shaheen. Automated Vehicles, On-Demand Mobility, and Environmental Impacts. *Current Sustainable/Renewable Energy Reports*, 2(3):74–81, 2015.

[41] Michael James Lighthill and Gerald Beresford Whitham. On Kinematic Waves II. A Theory of Traffic Flow on Long Crowded Roads. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 229(1178):317–345, 1955.

[42] Dianhai Wang and Chunguang Jing. Application of Traffic-Wave Theory in Urban Traffic Analysis. In *IVEC2001. Proceedings of the IEEE International Vehicle Electronics Conference 2001. IVEC 2001 (Cat. No. 01EX522)*, pages 143–147. IEEE, 2001.

[43] Carlos F Daganzo. A Behavioral Theory of Multi-Lane Traffic Flow. Part I: Long Homogeneous Freeway Sections. *Transportation Research Part B: Methodological*, 36(2):131–158, 2002.

[44] Arne Kesting, Martin Treiber, Martin Schönhof, and Dirk Helbing. Adaptive Cruise Control Design for Active Congestion Avoidance. *Transportation research. Part C, Emerging technologies*, 16(6):668–683, 2008.

[45] Perry Robinson MacNeille, Joseph Wisniewski, and Nunzio DeCia. Vehicle-to-Vehicle Cooperation to Marshal Traffic, 2018. US Patent 9,928,746.

[46] Dwip Banerjee, Rabindranath Dutta, and Kamal Patel. Method for Highway Congestion Management Using Dynamic Paid Upgrade for Automobiles to Use Fast Lanes, January 30 2003. US Patent App. 09/915,666.

[47] Maria Börjesson, Jonas Eliasson, Muriel B Hugosson, and Karin Brundell-Freij. The Stockholm Congestion Charges - 5 Years On. Effects, Acceptability and Lessons Learnt. *Transport Policy*, 20:1–12, 2012.

[48] Maria Börjesson and Ida Kristoffersson. The Gothenburg Congestion Charge. Effects, Design and Politics. *Transportation Research Part A: Policy and Practice*, 75:134–146, 2015.

[49] LTA - Government of Singapore. Electronic Road Pricing (ERP). URL: https://www.lta.gov.sg/content/ltaweb/en/roads-and-motoring/managing-traffic-and-congestion/electronic-road-pricing-erp.html, 2019. (Accessed April 08, 2019).

[50] John R Douceur. The Sybil Attack. In *International Workshop on Peer-to-Peer Systems*, pages 251–260. Springer, 2002.

[51] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design Science in Information Systems Research. *MIS Quarterly*, 28(1):75–105, 2004.

[52] Alan Hevner, Jan vom Brocke, and Alexander Maedche. Roles of Digital Innovation in Design Science Research. *Business & Information Systems Engineering*, 61(1):3–8, 2019.

[53] Martin Maguire. Socio-Technical Systems and Interaction Design: 21st Century Relevance. *Applied Ergonomics*, 45(2):162–170, 2014.

[54] John Venable, Jan Pries-Heje, and Richard Baskerville. FEDS: A Framework for Evaluation in Design Science Research. *European Journal of Information Systems*, 25(1):77–89, 2016.

[55] Carl Adam Petri. *Kommunikation mit Automaten*. PhD thesis, Technical University of Darmstadt, 1962.

[56] Kurt Jensen. Coloured Petri Nets. In *Discrete Event Systems: A New Challenge for Intelligent Control Systems, IEE Colloquium on*, pages 5–1. IET, 1993.

[57] Kurt Jensen, Lars Michael Kristensen, and Lisa Wells. Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. *International Journal on Software Tools for Technology Transfer*, 9(3-4):213–254, 2007.

[58] Leon Sterling and Kuldar Taveter. *The Art of Agent-Oriented Modeling*. MIT Press, 2009.

[59] Michele Chinosi and Alberto Trombetta. BPMN: An Introduction to the Standard. *Computer Standards & Interfaces*, 34(1):124–134, 2012.

[60] Grady Booch, Ivar Jacobson, James Rumbaugh, et al. The Unified Modeling Language. *Unix Review*, 14(13):5, 1996.

[61] Object Management Group. OMG Unified Modeling Language (OMG UML) - Superstructure - Version 2.1.2. *OMG Specification*, 2007.

[62] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. URL: https://bitcoin.org/bitcoin.pdf, 2008. (Accessed March 23, 2019).

[63] Konstantinos Christidis and Michael Devetsikiotis. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access*, 4:2292–2303, 2016.

[64] Aafaf Ouaddah, Anas Abou Elkalam, and Abdellah Ait Ouahman. 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*, pages 523–533. Springer, 2017.

[65] Oliver Bussmann. The Future of Finance: FinTech, Tech Disruption, and Orchestrating Innovation. In *Equity Markets in Transition*, pages 473–486. Springer, 2017.

[66] Quoc Khanh Nguyen. Blockchain - A Financial Technology for Future Sustainable Development. In *Green Technology and Sustainable Development (GTSD), International Conference on*, pages 51–54. IEEE, 2016.

[67] Feng Tian. An agri-food supply chain traceability system for china based on rfid & blockchain technology. In *2016 13th international conference on service systems and service management (ICSSSM)*, pages 1–6. IEEE, 2016.

[68] Benjamin Leiding, Clemens H. Cap, Thomas Mundt, and Samaneh Rashidibajgan. Authcoin: Validation and Authentication in Decentralized Networks. In *The 10th Mediterranean Conference on Information Systems - MCIS 2016*, Paphos, Cyprus, September 2016.

[69] Patrick McCorry, Siamak F Shahandashti, Dylan Clarke, and Feng Hao. Authenticated key exchange over bitcoin. In *International Conference on Research in Security Standardisation*, pages 3–20. Springer, 2015.

[70] Gavin Wood. Ethereum: A Secure Decentralized Generalised Transaction Ledger. http://gavwood.com/paper.pdf, 2014. (Accessed March 23, 2019).

[71] Bitcoin ABC. Bitcoin ABC and the Block Size Limit. URL: https://www.bitcoinabc.org/2018-09-07-bitcoin-abc-and-the-block-size-limit/, 2018. (Accessed April 11, 2019).

[72] Bitcoin SV. Bitcoin SV [BSV] Mines World-Record 128MB Blocks. URL: https://bitcoinsv.io/2019/04/04/bitcoin-sv-bsv-mines-world-record-128mb-blocks/, 2019. (Accessed April 11, 2019).

[73] Electric Coin Company. Zcash - Frequently Asked Questions. URL: https://z.cash/support/faq/, 2019. (Accessed April 11, 2019).

[74] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In *2017 IEEE International Congress on Big Data (BigData Congress)*, pages 557–564. IEEE, 2017.

[75] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.

[76] Sunny King and Scott Nadal. PPcoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake. URL: https://peercoin.net/whitepapers/peercoin-paper.pdf, 2012. (Accessed March 23, 2019).

[77] Parity Technologies. Aura - Authority Round – Parity Tech Documentation. URL: https://wiki.parity.io/Aura.html, 2019. (Accessed March 24, 2019).

[78] Péter Szilágyi. Clique PoA Protocol. URL: https://github.com/ethereum/EIPs/issues/225, 2017. (Accessed March 24, 2019).

[79] Arthur Gervais, Ghassan O Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. On the Security and Performance of Proof of Work Blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 3–16. ACM, 2016.

[80] Nicolas Houy. It Will Cost You Nothing to 'Kill' a Proof-of-Stake Crypto-Currency. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2393940, 2014. (Accessed March 23, 2019).

[81] Adam Back. Hashcash - A Denial of Service Counter-Measure. URL: ftp://sunsite.icm.edu.pl/site/replay.old/programs/hashcash/hashcash.pdf, 2002. (Accessed March 24, 2019).

[82] Karl J O'Dwyer and David Malone. Bitcoin Mining and its Energy Footprint. *IET Conference Proceedings*, pages 280–285(5), 2014. URL https://digital-library.theiet.org/content/conferences/10.1049/cp.2014.0699.

[83] Alex De Vries. Bitcoin's Growing Energy Problem. *Joule*, 2(5):801–805, 2018.

[84] Jon Truby. Decarbonizing Bitcoin: Law and Policy Choices for Reducing the Energy Consumption of Blockchain Technologies and Digital Currencies. *Energy Research & Social Science*, 2018.

[85] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol. In *Annual International Cryptology Conference*, pages 357–388. Springer, 2017.

[86] Craig Calcaterra and Wulf A. Kaal. Semada's Proof of Stake Protocol. *University of St. Thomas (Minnesota) Legal Studies Research Paper*, (18-10), 2018.

[87] Vlad Zamfir. Introducing Casper "The Friendly Ghost". URL: https://blog.ethereum.org/2015/08/01/introducing-casper-friendly-ghost/, 2015. (Accessed March 24, 2019).

[88] Igor Barinov, Viktor Baranov, and Pavel Khahulin. POA Network - Whitepaper. URL: https://github.com/poanetwork/wiki/wiki/POA-Network-Whitepaper, 2018. (Accessed March 24, 2019).

[89] Andreas M. Antonopoulos and Gavin Wood. Mastering Ethereum. URL: https://github.com/ethereumbook/ethereumbook/blob/develop/02intro.asciidoc, 2018. (Accessed March 24, 2019).

[90] Hyperledger. Hyperledger Fabric 1.4 - Transaction Flow. URL: https://hyperledger-fabric.readthedocs.io/en/release-1.4/txflow.html, 2018. (Accessed March 24, 2019).

[91] Nick Szabo. Smart Contracts. URL: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html, 1994. (Accessed April 2nd, 2019).

[92] Nick Szabo. The Idea of Smart Contracts. URL: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_idea.html, 1997. (Accessed April 2nd, 2019).

[93] LM Goodman. Tezos - A Self-Amending Crypto-Ledger (White Paper). URL: https://tezos.com/static/white_paper-2dc8c02267a8fb86bd67a108199441bf.pdf, 2014. (Accessed April 2nd, 2019).

[94] Marco Conoscenti, Antonio Vetro, and Juan Carlos De Martin. Blockchain for the Internet of Things: A Systematic Literature Review. In *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–6. IEEE, 2016.

[95] CoinMarketCap. CoinMarketCap - List of All Cryptocurrencies. URL: https://coinmarketcap.com/all/views/all/, 2019. (Accessed April 2nd, 2019).

[96] Dylan Yaga, Peter Mell, Nik Roby, and Karen Scarfone. Blockchain Technology Overview. Technical report, National Institute of Standards and Technology, 2018.

[97] Iuon-Chang Lin and Tzu-Chun Liao. A Survey of Blockchain Security Issues and Challenges. *International Journal of Network Security*, 19(5):653–659, 2017.

[98] Thomas Hardjono, Alexander Lipton, and Alex Pentland. Towards a Design Philosophy for Interoperable Blockchain Systems. *arXiv preprint arXiv:1805.05934*, 2018.

[99] Alex Lipton, Thomas Hardjono, and Alex Pentland. Digital Trade Coin: Towards a More Stable Digital Currency. *Royal Society Open Science*, 5(7):180155, 2018.

[100] Gavin Wood. Polkadot: Vision for a Heterogenous Multi-Chain Framework - Draft 1 (White Paper). URL: https://polkadot.network/PolkaDotPaper.pdf, 2016. (Accessed April 2nd, 2019).

[101] Vitalik Buterin. Chain Interoperability. URL: https://static1.squarespace.com/static/55f73743e4b051cfcc0b02cf/t/5886800ecd0f68de303349b1/1485209617040/Chain+Interoperability.pdf, 2016. (Accessed April 24, 2019).

[102] Jae Kwon and Ethan Buchman. Cosmos - A Network of Distributed Ledgers - White Paper. URL: https://github.com/cosmos/cosmos/blob/master/WHITEPAPER.md, 2019. (Accessed April 24, 2019).

[103] Matthew Spoke and Nuco Engineering Team. Aion: Enabling the Decentralized Internet - White Paper. URL: https://aion.network/media/en-aion-network-technical-introduction.pdf, 2017. (Accessed April 24, 2019).

[104] 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.

[105] Roberto Baldessari, Bert Bödekker, Matthias Deegener, Andreas Festag, Walter Franz, C Christopher Kellum, Timo Kosch, Andras Kovacs, Massimiliano Lenardi, Cornelius Menig, et al. Car-2-Car Communication Consortium - Manifesto. 2007.

[106] Sherali Zeadally, Ray Hunt, Yuh-Shyan Chen, Angela Irwin, and Aamir Hassan. Vehicular Ad Hoc Networks (VANETS): Status, Results, and Challenges. *Telecommunication Systems*, 50(4):217–241, 2012.

[107] Jingjing Wang, Chunxiao Jiang, Zhu Han, Yong Ren, Robert G Maunder, and Lajos Hanzo. Taking Drones to the Next Level: Cooperative Distributed Unmanned-Aerial-Vehicular Networks for Small and Mini Drones. *IEEE Vehicular Technology Magazine*, 12(3):73–82, 2017.

[108] Peng-Yong Kong, Ming-Tuo Zhou, and Jaya Shankar Pathmasuntharam. A Routing Approach for Inter-Ship Communications in Wireless Multi-Hop Networks. In *2008 8th International Conference on ITS Telecommunications*, pages 89–94. IEEE, 2008.

[109] Yingsi Liang, Hui Liu, and Dinesh Rajan. Optimal Placement and Configuration of Roadside Units in Vehicular Networks. In *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, pages 1–6. IEEE, 2012.

[110] Raya, Maxim and Hubaux, Jean-Pierre. Securing Vehicular Ad Hoc Networks. *Journal of Computer Security*, 15(1):39–68, 2007.

[111] Miad Faezipour, Mehrdad Nourani, Adnan Saeed, and Sateesh Addepalli. Progress and Challenges in Intelligent Vehicle Area Networks. *Communications of the ACM*, 55(2):90–100, 2012.

[112] Jun Luo and Jean-Pierre Hubaux. A Survey of Inter-Vehicle Communication. Technical report, 2004. URL: https://pdfs.semanticscholar.org/97cb/31247602189ca18c357c73b0d1ed84d0fa9d.pdf (Accessed March 20, 2019).

[113] Felipe Cunha, Leandro Villas, Azzedine Boukerche, Guilherme Maia, Aline Viana, Raquel AF Mini, and Antonio AF Loureiro. Data Communication in VANETs: Protocols, Applications and Challenges. *Ad Hoc Networks*, 44:90–103, 2016.

[114] L. Wischhof, A. Ebner, and H. Rohling. Information dissemination in self-organizing intervehicle networks. *IEEE Transactions on Intelligent Transportation Systems*, 6(1):90–101, March 2005. ISSN 1524-9050. doi: 10.1109/TITS.2004.842407.

[115] Reed Drummond, Manu Sporny, Dave Longley, Christopher Allen, Ryan Grant, and Markus Sabadello. Decentralized Identifiers (DIDs) - Data Model and Syntaxes for Decentralized Identifiers v0.14. https://w3c-ccg.github.io/did-spec/, 2019. (Accessed June 16, 2019).

[116] Credentials Community Group (W3C). DID Method Registry - A registry for Decentralized Identifier Methods. https://w3c-ccg.github.io/did-method-registry/, 2019. (Accessed May 26, 2019).

[117] Manu Sporny, Dave Longly, and David Chadwick. Verifiable Credentials Data Model 1.0. https://www.w3.org/TR/verifiable-claims-data-model/, 2019. (Accessed May 26, 2019).

[118] Mike P Papazoglou and Benedikt Kratz. A Business-Aware Web Services Transaction Model. In *International Conference on Service-Oriented Computing*, pages 352–364. Springer, 2006.

[119] Tommy Roxenhall and Pervez Ghauri. Use of the Written Contract in Long-Lasting Business Relationships. *Industrial Marketing Management*, 33(3):261–268, 2004.

[120] Alex 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.

[121] Morten Olsen. How Firms Overcome Weak International Contract Enforcement: Repeated Interaction, Collective Punishment and Trade Finance. *IESE Business*

School Working Paper No. WP-1111-E, 2016. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2219243.

[122] Vitalik Buterin et al. A Next-Generation Smart Contract and Decentralized Application Platform - Whitepaper. URL: https://github.com/ethereum/wiki/wiki/White-Paper, 2019. (Accessed July 14, 2019).

[123] Alan M. Davis. *Software Requirements: Objects, Functions, and States*. Prentice-Hall, Inc., 1993.

[124] 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. ISBN 9781559373951.

[125] Alex Norta, Paul Grefen, and Nanjangud C Narendra. A Reference Architecture for Managing Dynamic Inter-Organizational Business Processes. *Data & Knowledge Engineering*, 91:52–89, 2014.

[126] James Marshall. Agent-Based Modelling of Emotional Goals in Digital Media Design Projects. *International Journal of People-Oriented Programming (IJPOP)*, 3(1):44–59, 2014.

[127] Alex Norta. *Exploring Dynamic Inter-Organizational BusinessProcess Collaboration: Privacy Protecting Concepts for ChoreographingeSourcing in B2B with Service-Oriented Computing*. VDM Verlag, 2008.

[128] Algirdas Avizienis, J-C Laprie, Brian Randell, and Carl Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33, 2004.

[129] Thomas Erl. *Service-Oriented Architecture: Concepts, Technology, and Design*. Pearson Education India, 1900.

[130] Randall Perrey and Mark Lycett. Service-Oriented Architecture. In *2003 Symposium on Applications and the Internet Workshops, 2003. Proceedings.*, pages 116–119. IEEE, 2003.

[131] Michael Rosen, Boris Lublinsky, Kevin T Smith, and Marc J Balcer. *Applied SOA: Service-Oriented Architecture and Design Strategies*. John Wiley & Sons, 2012.

[132] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: An Open-Source Robot Operating System. In *ICRA Workshop on Open Source Software*, volume 3, page 5. Kobe, Japan, 2009.

[133] Apollo. Apollo Open Platform. URL: http://www.apollo.auto/, 2018. (Accessed April 21, 2019).

[134] Yunxin Jeff Li. An Overview of the DSRC/WAVE Technology. In *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, pages 544–558. Springer, 2010.

[135] Roberto A Uzcátegui, Antonio Jose De Sucre, and Guillermo Acosta-Marum. WAVE: A Tutorial. *IEEE Communications Magazine*, 47(5):126–133, 2009.

[136] Alex Norta. Creation of Smart-Contracting Collaborations for Decentralized Autonomous Organizations. In *International Conference on Business Informatics Research*, pages 3–17. Springer, 2015.

[137] Alex Norta. Establishing Distributed Governance Infrastructures for Enacting Cross-Organization Collaborations. In *International Conference on Service-Oriented Computing*, pages 24–35. Springer, 2015.

[138] Alex Norta, Anis Ben Othman, and Kuldar Taveter. Conflict-Resolution Lifecycles for Governed Decentralized Autonomous Organization Collaboration. In *Proceedings of the 2015 2nd International Conference on Electronic Governance and Open Society: Challenges in Eurasia*, pages 244–257. ACM, 2015.

[139] Patrick Dai, Neil Mahi, Jordan Earls, and Alex Norta. Smart-Contract Value-Transfer Protocols on a Distributed Mobile Application Platform - White Paper. URL: https://qtum.org/user/pages/01.home/Qtum%20whitepaper_en%20v0.7.pdf, 2017. (Accessed April 30, 2019).

[140] Lea Kutvonen, Alex Norta, and Sini Ruohomaa. Inter-Enterprise Business Transaction Management in Open Service Ecosystems. In *2012 IEEE 16th International Enterprise Distributed Object Computing Conference*, pages 31–40. IEEE, 2012.

[141] Benny Moldovanu and Manfred Tietzel. Goethe's Second-Price Auction. *Journal of Political Economy*, 106(4):854–859, 1998.

[142] William Vickrey. Counterspeculation, Auctions, and Competitive Sealed Tenders. *The Journal of Finance*, 16(1):8–37, 1961.

[143] Lawrence M Ausubel and Paul Milgrom. The Lovely but Lonely Vickrey Auction. *Combinatorial Auctions*, 17:22–26, 2006.

[144] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords. *American Economic Review*, 97(1):242–259, 2007.

[145] David Lucking-Reiley. Vickrey Auctions in Ppractice: From Nineteenth-Century Philately to Twenty-First-Century e-Commerce. *Journal of Economic Perspectives*, 14(3):183–192, 2000.

[146] Clemens H Cap. Bitcoin – Das Open-Source-Geld. *HMD Praxis der Wirtschaftsinformatik*, 49(1):84–93, 2012.

[147] Juan Benet. InterPlanetary File System (IPFS) - Content Addressed, Versioned, P2P File System - White Paper. URL: https://github.com/ipfs/papers/raw/master/ipfs-cap2pfs/ipfs-p2p-file-system.pdf, 2014. (Accessed October 12, 2019).

[148] Protocol Labs. Filecoin: A Decentralized Storage Network - White Paper. URL: https://filecoin.io/filecoin.pdf, 2017. (Accessed April 30, 2019).

[149] David Chaum. Blind Signatures for Untraceable Payments. In *Advances in Cryptology*, pages 199–203. Springer, 1983.

[150] N Zhang, Q Shi, and M Merabti. Anonymous Public-Key Certificates for Anonymous and Fair Document Exchange. *IEE Proceedings-Communications*, 147(6): 345–350, 2000.

[151] Craig Gentry. *A Fully Homomorphic Encryption Scheme*. PhD thesis, 2009.

[152] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-Fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pages 1219–1234. ACM, 2012.

[153] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, Transparent, and Post-Quantum Secure Computational Integrity. *IACR Cryptology ePrint Archive*, 2018:46, 2018.

[154] Bardi Matturdi, Xianwei Zhou, Shuai Li, and Fuhong Lin. Big Data Security and Privacy: A Review. *China Communications*, 11(14):135–145, 2014.

[155] Dorit Ron and Adi Shamir. Quantitative Analysis of the Full Bitcoin Transaction Graph. In *International Conference on Financial Cryptography and Data Security*, pages 6–24. Springer, 2013.

[156] Thomas Mundt, Frank Krüger, and Till Wollenberg. Who Refuses to Wash Hands? Privacy Issues in Modern House Installation Networks. In *2012 Seventh International Conference on Broadband, Wireless Computing, Communication and Applications*, pages 271–277. IEEE, 2012.

[157] Beverley A Sparks and Victoria Browning. The Impact of Online Reviews on Hotel Booking Intentions and Perception of Trust. *Tourism Management*, 32(6): 1310–1323, 2011.

[158] Jordi Sabater and Carles Sierra. Review on Computational Trust and Reputation Models. *Artificial Intelligence Review*, 24(1):33–60, 2005.

[159] Audun Jøsang, Roslan Ismail, and Colin Boyd. A Survey of Trust and Reputation Systems for Online Service Provision. *Decision Support Systems*, 43(2):618–644, 2007.

[160] John Adler, Ryan Berryhill, Andreas Veneris, Zissis Poulos, Neil Veira, and Anastasia Kastania. Astraea: A decentralized blockchain oracle. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1145–1152. IEEE, 2018.

[161] Foamspace Corp. FOAM - The Consensus Driven Map of the World (Whitepaper). URL: https://www.foam.space/publicAssets/FOAM_Whitepaper.pdf, 2018. (Accessed July 9th, 2019).

[162] Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma, Steven Lim, et al. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. *IEEE Communications Surveys and tutorials*, 7(1-4):72–93, 2005.

[163] Rakesh Babu Bobba, Laurent Eschenauer, Virgil Gligor, and William Arbaugh. Bootstrapping Security Associations for Routing in Mobile Ad-Hoc Networks. In *GLOBECOM'03. IEEE Global Telecommunications Conference (IEEE Cat. No. 03CH37489)*, volume 3, pages 1511–1515. IEEE, 2003.

[164] Didier Sornette and Susanne Von der Becke. Crashes and High Frequency Trading. *Swiss Finance Institute Research Paper*, (11-63), 2011.

[165] Clemens H. Cap and Benjamin Leiding. Blogchain – Disruptives Publizieren auf der Blockchain. *HMD Praxis der Wirtschaftsinformatik*, 55(6):1326–1340, 2018. doi: 10.1365/s40702-018-00470-w. URL https://doi.org/10.1365/s40702-018-00470-w.

[166] Emiliano Miluzzo, Nicholas D Lane, Andrew T Campbell, and Reza Olfati-Saber. CaliBree: A Self-Calibration System for Mobile Sensor Networks. In *International Conference on Distributed Computing in Sensor Systems*, pages 314–331. Springer, 2008.

[167] Jose M Barcelo-Ordinas, Messaoud Doudou, Jorge Garcia-Vidal, and Nadjib Badache. Self-Calibration Methods for Uncontrolled Environments in Sensor Networks: A Reference Survey. *Ad Hoc Networks*, 88:142–159, 2019.

[168] Peter Bright. Microsoft Withdraws Bad Windows 7 Update That Broke Future Windows 7 Updates. URL: https://arstechnica.com/information-technology/2014/12/microsoft-withdraws-bad-windows-7-update-that-broke-future-windows-7-updates/?comments=1&post=28121607, 2014. (Accessed July 16, 2019).

[169] Clemens H. Cap and Benjamin Leiding. Ensuring Resource Trust and Integrity in Web Browsers Using Blockchain Technology. In *Advanced Information Systems Engineering Workshops*, pages 115–125, Cham, 2018. Springer International Publishing.

[170] Martin Arrivets. MONET: Mobile Ad Hoc Blockchains - Whitepaper Version 1.0. URL: http://bit.ly/monet-whitepaper, 2018. (Accessed July 16, 2019).

[171] Roland Rosen, Georg Von Wichert, George Lo, and Kurt D Bettenhausen. About the Importance of Autonomy and Digital Twins for the Future of Manufacturing. *IFAC-PapersOnLine*, 48(3):567–572, 2015.

[172] Jake A Berkowsky and Thaier Hayajneh. Security Issues With Certificate Authorities. In *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pages 449–455. IEEE, 2017.

[173] Axel Arnbak and Nico ANM van Eijk. Certificate Authority Collapse: Regulating Systemic Vulnerabilities in the HTTPS Value Chain. In *Telecommunications Policy Research Conference 2012*. TPRC, 2012. doi: http://dx.doi.org/10.2139/ssrn.2031409. URL https://ssrn.com/abstract=2031409.

[174] Nicole van der Meulen. DigiNotar: Dissecting the First Dutch Digital Disaster. *Journal of Strategic Security*, 6(2):46–58, 2013. URL https://pdfs.semanticscholar.org/46b1/f178f969116c85f735c36bd3905a6c429d56.pdf.

[175] Benjamin Leiding and Alex Norta. Mapping Requirements Specifications Into a Formalized Blockchain-Enabled Authentication Protocol for Secured Personal Identity Assurance. In *4th International Conference on Future Data and Security Engineering - FDSE 2017*, pages 181–196, Ho Chi Minh City, Vietnam, 2017. Springer.

[176] Benjamin Leiding. Securing the Authcoin Protocol Using Security Risk-Oriented Patterns. Master's thesis, University of Göttingen, Göttingen, Germany, 2017.

[177] Arne Bochem, Benjamin Leiding, and Dieter Hogrefe. Unchained Identities: Putting a Price on Sybil Nodes in Mobile Ad hoc Networks. In *International Conference on Security and Privacy in Communication Systems*, pages 358–374. Springer, 2018.

[178] Asem Othman and John Callahan. The Horcrux Protocol: A Method for Decentralized Biometric-Based Self-Sovereign Identity. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2018.

[179] Radia Perlman. An Overview of PKI Trust Models. *IEEE network*, 13(6):38–43, 1999.

[180] Rohit Khare and Adam Rifkin. Weaving a Web of Trust. *World Wide Web Journal*, 2(3):77–112, 1997.

[181] Tom Espiner. Trustwave Sold Root Certificate for Surveillance, 2012. URL https://www.zdnet.com/article/trustwave-sold-root-certificate-for-surveillance/. (Accessed July 3rd, 2019).

[182] Comodo Security Solutions, Inc. Comodo Report of Incident, 2011. URL https://www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html. (Accessed July 3rd, 2019).

[183] Benjamin Leiding and Andreas Dähn. Dead Letters to Alice-Reachability of E-Mail Addresses in the PGP Web of Trust. *Baltic Young PhD Conference (BaSoTI 2016) and arXiv:1605.03162*, 2016.

[184] Bo Qin, Jikun Huang, Qin Wang, Xizhao Luo, Bin Liang, and Wenchang Shi. Cecoin: A Decentralized PKI Mitigating MitM Attacks. *Future Generation Computer Systems*, 2017.

[185] Conner Fromknecht, Dragos Velicanu, and Sophia Yakoubov. A Decentralized Public Key Infrastructure with Identity Retention. *IACR Cryptology ePrint Archive*, 2014:803, 2014.

[186] Stephanos Matsumoto and Raphael M Reischuk. IKP: Turning a PKI Around With Decentralized Automated Incentives. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 410–426. IEEE, 2017.

[187] Conner Fromknecht, Dragos Velicanu, and Sophia Yakoubov. Certcoin: A Namecoin Based Decentralized Aauthentication System - 6.857 Class Project. Technical report, 2014.

[188] Namecoin. Namecoin - A Decentralized Open Source Information Registration and Transfer System. URL: https://namecoin.info/, 2019. (Accessed July 3rd, 2019).

[189] Udaya Suriya Raj Kumar Dhamodharan and Rajamani Vayanaperumal. Detecting and Preventing Sybil Attacks in Wireless Sensor Networks Using Message Authentication and Passing Method. *The Scientific World Journal*, 2015, 2015.

[190] RincyMedayil John, Jacob P Cherian, and Jubilant J Kizhakkethottam. A Survey of Techniques to Prevent Sybil Attacks. In *2015 International Conference on Soft-Computing and Networks Security (ICSNS)*, pages 1–6. IEEE, 2015.

[191] Sohail Abbas, Madjid Merabti, David Llewellyn-Jones, and Kashif Kifayat. Lightweight Sybil Attack Detection in MANETs. *IEEE Systems Journal*, 7(2): 236–248, 2013.

[192] Athichart Tangpong, George Kesidis, Hung-yuan Hsu, and Ali Hurson. Robust Sybil Detection for MANETs. In *2009 Proceedings of 18th International Conference on Computer Communications and Networks*, pages 1–6. IEEE, 2009.

[193] Sovrin Foundation. Sovrin: A Protocol and Token for Self-Sovereign Identity and Decentralized Trust - Whitepaper. URL: https://sovrin.org/wp-content/uploads/Sovrin-Protocol-and-Token-White-Paper.pdf, 2018. (Accessed July 3rd, 2019).

[194] Christian Lundkvist, Rouven Heck, Joel Torstensson, Zac Mitton, and Michael Sena. uPort: A Platform for Self-Sovereign Identity - Whitepaper. URL: http://blockchainlab.com/pdf/uPort_whitepaper_DRAFT20161020.pdf, 2016. (Accessed July 3rd, 2019).

[195] SelfKey Foundation. SelfKey - Whitepaper. URL: https://selfkey.org/wp-content/uploads/2019/03/selfkey-whitepaper-en.pdf, 2017. (Accessed July 3rd, 2019).

[196] Muneeb Ali, Jude Nelson, Ryan Shea, and Michael J Freedman. Blockstack: A Global Naming and Storage System Secured by Blockchains. In *Proceedings of the 2016 USENIX Conference on Usenix Annual Technical Conference*, pages 181–194. USENIX Association, 2016.

[197] Chris Reed, Uma M Sathyanarayan, Shuhui Ruan, and Justine Collins. Beyond BitCoin - Legal Impurities and Off-Chain Assets. *International Journal of Law and Information Technology*, 26(2):160–182, 2018.

[198] Simson Garfinkel. *PGP: Pretty Good Privacy*. O'Reilly Media, Inc., 1995.

[199] Internet Security Research Group (ISRG). Let's Encrypt - How it Works. URL https://letsencrypt.org/how-it-works/. (Accessed June 16, 2019).

[200] Vance Bjorn. Cryptographic Key Generation Using Biometric Data, March 7 2000. US Patent 6,035,398.

[201] Jon Callas, Lutz Donnerhacke, Hal Finney, David Shaw, and Rodney Thayer. Rfc 4880 - openpgp message format, 2007. URL https://tools.ietf.org/html/rfc4880. (Accessed June 12, 2019).

[202] Adi Shamir. How to Share a Secret. *Communications of the ACM*, 22(11):612–613, 1979.

[203] George Robert Blakley et al. Safeguarding Cryptographic Keys. In *Proceedings of the National Computer Conference*, volume 48, 1979.

[204] Nancy Cam-Winget, Russ Housley, David Wagner, and Jesse Walker. Security Flaws in 802.11 Data Link Protocols. *Communications of the ACM*, 46(5):35–39, 2003.

[205] Ulf Carlsen. Cryptographic Protocol Flaws: Know Your Enemy. In *Computer Security Foundations Workshop VII, 1994. CSFW 7. Proceedings*, pages 192–200. IEEE, 1994.

[206] F. Javier Thayer Fábrega, Jonathan C. Herzog, and Joshua D. Guttman. Strand Spaces: Why is a Security Protocol Correct? In *Security and Privacy, 1998. Proceedings. 1998 IEEE Symposium on*, pages 160–171. IEEE, 1998.

[207] Serge Vaudenay. Security Flaws Induced by CBC Padding - Applications to SSL, IPSEC, WTLS... In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 534–545. Springer, 2002.

[208] Adam Stubblefield, John Ioannidis, and Aviel D. Rubin. A Key Recovery Attack on the 802.11b Wired Equivalent Privacy Protocol (WEP). *ACM Transactions on Information and System Security (TISSEC)*, 7(2):319–332, 2004.

[209] Chris Brook. Nuclear Power Plant Disrupted by Cyber Attack. https://threatpost.com/nuclear-power-plant-disrupted-by-cyber-attack/121216/, 2016. (Accessed May 07, 2019).

[210] Robin Milner, Joachim Parrow, and David Walker. A Calculus of Mobile Processes, I. *Information and Computation*, 100(1):1–40, 1992.

[211] Charles Antony Richard Hoare. Communicating Sequential Processes. In *The Origin of Concurrent Programming*, pages 413–443. Springer, 1978.

[212] Martín Abadi and Andrew D. Gordon. A Calculus for Cryptographic Protocols: The Spi Calculus. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 36–47. ACM, 1997.

[213] Federico Crazzolara and Glynn Winskel. Events in Security Protocols. In *Proceedings of the 8th ACM Conference on Computer and Communications Security*, pages 96–105. ACM, 2001.

[214] moxie0. WhatsApp's Signal Protocol Integration is now Complete. URL: https://signal.org/blog/whatsapp-complete/, 2016. (Accessed July 3rd, 2019).

[215] Katriel Cohn-Gordon, Cas Cremers, Benjamin Dowling, Luke Garratt, and Douglas Stebila. A Formal Security Analysis of the Signal Messaging Protocol. URL: https://eprint.iacr.org/2016/1013.pdf. (Accessed May 07, 2019).

[216] Yongyuth Permpoontanalarp and Apichai Changkhanak. Security Analysis of the TMN Protocol by Using Coloured Petri Nets: On-the-fly Trace Generation Method and Homomorphic Property. In *2011 Eighth International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 63–68. IEEE, 2011.

[217] Salah Aly and Khaled Mustafa. Protocol Verification and Analysis Using Colored Petri Nets. URL: http://facweb.cs.depaul.edu/research/techreports/tr04-003.pdf, 2003. (Accessed October 31, 2019).

[218] Kim Edwards. *Cryptographic Protocol Specification and Analysis Using Coloured Petri Nets and Java*. PhD thesis, Queen's University Kingston, 1999.

[219] Alex Norta and Lea Kutvonen. Safeguarding Trusted eBusiness Transactions of Lifecycles for Cross-Enterprise Collaboration. Technical report, 2012.

[220] Paolo Giorgini, John Mylopoulos, and Roberto Sebastiani. Goal-oriented Requirements Analysis and Reasoning in the Tropos Methodology. *Engineering Applications of Artificial Intelligence*, 18(2):159–171, 2005.

[221] Michael Wooldridge, Nicholas R. Jennings, and David Kinny. The Gaia Methodology for Agent-oriented Analysis and Design. *Autonomous Agents and Multi-agent Systems*, 3(3):285–312, 2000.

[222] Lin Padgham and Michael Winikoff. Prometheus: A Methodology for Developing Intelligent Agents. In *International Workshop on Agent-Oriented Software Engineering*, pages 174–185. Springer, 2002.

[223] B. Moulin and L. Cloutier. Soft computing. chapter Collaborative Work Based on Multiagent Architectures: A Methodological Perspective, pages 261–296. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1994. ISBN 0-13-146234-2.

[224] Bernard Moulin and Mario Brassard. A Scenario-based Design Method and an Environment for the Development of Multiagent Systems. In *Australian Workshop on Distributed Artificial Intelligence*, pages 216–232. Springer, 1995.

[225] Scott A. DeLoach. Analysis and Design using MaSE and agentTool. Technical report, DTIC Document, 2001.

[226] Msury Mahunnah, Alex Norta, Lixin Ma, and Kuldar Taveter. Heuristics for Designing and Evaluating Socio-technical Agent-Oriented Behaviour Models with Coloured Petri Nets. In *Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE 38th International*, pages 438–443. IEEE, 2014.

[227] Alex Norta, Raimundas Matulevičius, and Benjamin Leiding. Safeguarding a Formalized Blockchain-Enabled Identity-Authentication Protocol by Applying Security Risk-Oriented Patterns. *Computers & Security*, 86:253 – 269, 2019. ISSN 0167-4048. doi: https://doi.org/10.1016/j.cose.2019.05.017.

[228] Haifeng Yu, Michael Kaminsky, Phillip B Gibbons, and Abraham Flaxman. Sybilguard: Defending Against Sybil Attacks Via Social Networks. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 267–278. ACM, 2006.

[229] Haifeng Yu, Phillip B Gibbons, Michael Kaminsky, and Feng Xiao. Sybillimit: A Near-Optimal Social Network Defense Against Sybil Attacks. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 3–17. IEEE, 2008.

[230] Bin Xiao, Bo Yu, and Chuanshan Gao. Detection and Localization of Sybil Nodes in VANETs. In *Proceedings of the 2006 Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks*, pages 1–8. ACM, 2006.

[231] James Newsome, Elaine Shi, Dawn Song, and Adrian Perrig. The Sybil Attack in Sensor Networks: Analysis & Defenses. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, pages 259–268. ACM, 2004.

[232] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten. SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies. In *2015 IEEE Symposium on Security and Privacy*, pages 104–121, May 2015. doi: 10.1109/SP.2015.14.

[233] Peter Todd. BIP 65 - OP_CHECKLOCKTIMEVERIFY, 2014. URL https://github.com/bitcoin/bips/blob/6295c1a095a1fa33f38d334227fa4222d8e0a523/bip-0009.mediawiki. (Accessed June 2nd, 2019).

[234] Bitcoin Project. Bitcoin Developer Guide. https://bitcoin.org/en/developer-guide#proof-of-work, 2019. (Accessed June 2nd, 2019).

[235] Mining - Bitcoin Wiki. URL https://en.bitcoin.it/w/index.php?title=Mining&oldid=64115#Reward. (Accessed June 2nd, 2019).

[236] Transaction - Bitcoin Wiki. URL https://en.bitcoin.it/w/index.php?title=Transaction&oldid=63712. (Accessed June 2nd, 2019).

[237] CoinMarketCap. Bitcoin Price, Charts, Market Cap and other Metrics. URL https://coinmarketcap.com/currencies/bitcoin/. (Accessed June 2nd, 2019).

[238] Éric Dubois, Patrick Heymans, Nicolas Mayer, and Raimundas Matulevičius. A Systematic Approach to Define the Domain of Information System Security Risk Management. In *Intentional Perspectives on Information Systems Engineering*, pages 289–306. Springer, 2010.

[239] Raimundas Matulevičius. *Fundamentals of Secure System Modelling*. Springer, 2017.

[240] Naved Ahmed and Raimundas Matulevičius. Securing Business Processes Using Security Risk-Oriented Patterns. *Computer Standards & Interfaces*, 36(4):723–733, 2014.

[241] AutoPi.io. AutoPi: 2nd Generation Technical Specifications. URL: https://www.autopi.io/hardware-dongle/generation-two, 2019. (Accessed October 13, 2019).

[242] Tamas Blummer et al. An Introduction to Hyperledger - V1.1. URL: https://www.hyperledger.org/wp-content/uploads/2018/08/HL_Whitepaper_IntroductiontoHyperledger.pdf, 2018. (Accessed October 12, 2019).

[243] Sergey Lonshakov, Aleksandr Krupenkin, Aleksandr Kapitonov, Evgeny Radchenko, Alisher Khassanov, and Aleksandr Starostin. Robonomics: Platform for Integration of Cyber Physical Systems Into Human Economy. URL: https://static.robonomics.network/docs/whitepaper/Robonomics-whitepaper-en.pdf, 2018. (Accessed October 12, 2019).

[244] Aleksandr Kapitonov, Ivan Berman, Vitaly Bulatov, Sergey Lonshakov, and Aleksandr Krupenkin. Robonomics Based on Blockchain as a Principle of Creating Smart Factories. In *2018 Fifth International Conference on Internet of Things: Systems, Management and Security*, pages 78–85. IEEE, 2018.

[245] Markus Knecht and Burkhard Stiller. SmartDEMAP: A Smart Contract Deployment and Management Platform. In *IFIP International Conference on Autonomous Infrastructure, Management and Security*, pages 159–164. Springer, 2017.

[246] Mike Hearn. Corda: A Distributed Ledger - Version 0.5. URL: https://www.corda.net/content/corda-technical-whitepaper.pdf, 2016. (Accessed October 12, 2019).

[247] Todd McDonald. Corda and Settlement: Let's Get Atomic. URL: https://medium.com/corda/corda-and-settle<ment-lets-get-atomic-1cee1d896a9b, 2018. (Accessed October 13, 2019).

[248] David Nicol. Settlement Assets on Corda - Where to Start. URL: https://medium.com/corda/settlement-assets-on-corda-where-to-start-b01dd6c842a1, 2018. (Accessed October 13, 2019).

[249] L.M Goodman. Tezos - A Self-Amending Crypto-Ledger – White Paper. URL: https://tezos.com/static/white_paper-2dc8c02267a8fb86bd67a108199441bf.pdf, 2014. (Accessed March 24, 2019).

[250] Why Michelson? URL: https://www.michelson-lang.com/why-michelson.html. (Accessed October 13, 2019).

[251] OCamlPro SAS. Liquidity Documentation. URL: http://www.liquidity-lang.org/doc/, 2018. (Accessed October 13, 2019).

[252] Ocean Protocol Foundation, BigchainDB GmbH and Newton Circus. Ocean Protocol: A Decentralized Substrate for AI Data and Services - Technical Whitepaper (Version 2019-APR-15). URL: https://oceanprotocol.com/tech-whitepaper.pdf, 2019. (Accessed October 12, 2019).

[253] Steve Ellis, Ari Juels, and Sergey Nazarov. ChainLink: A Decentralized Oracle Network - v1.0 – Whitepaper. URL: https://link.smartcontract.com/whitepaper, 2017. (Accessed October 12, 2019).

[254] Storj Labs, Inc. Storj: A Decentralized Cloud Storage Network Framework – Whitepaper. URL: https://storj.io/storjv3.pdf, 2018. (Accessed October 12, 2019).

[255] BigchainDB GmbH. BigchainDB 2.0: The Blockchain Database - Version 1.0 – Whitepaper. URL: https://www.bigchaindb.com/whitepaper/bigchaindb-whitepaper.pdf, 2018. (Accessed October 12, 2019).

[256] Alma Whitten and J Doug Tygar. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In *USENIX Security Symposium*, volume 348, pages 169–184, 1999.

[257] Steve Sheng, Levi Broderick, Colleen Alison Koranda, and Jeremy J Hyland. Why Johnny Still Can't Encrypt: Evaluating the Usability of Email Encryption Software. In *Symposium On Usable Privacy and Security*, pages 3–4, 2006.

[258] Adiseshu Hari and TV Lakshman. The Internet Blockchain: A Distributed, Tamper-Resistant Transaction Framework for the Internet. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, pages 204–210. ACM, 2016.

[259] AUTOSAR GbR. AUTOSAR: Adaptive Platform 19.03. URL: https://www.autosar.org/standards/adaptive-platform/adaptive-platform-1903/, 2019. (Accessed October 12, 2019).

[260] Janusz J Sikorski, Joy Haughton, and Markus Kraft. Blockchain Technology in the Chemical Industry: Machine-to-Machine Electricity Market. *Applied Energy*, 195:234–246, 2017.

[261] Jiawen Kang, Rong Yu, Xumin Huang, Sabita Maharjan, Yan Zhang, and Ekram Hossain. Enabling Localized Peer-to-Peer Electricity Trading Among Plug-In Hybrid Electric Vehicles Uusing Consortium Blockchains. *IEEE Transactions on Industrial Informatics*, 13(6):3154–3164, 2017.

[262] Paolo Missier, Shaimaa Bajoudah, Angelo Capossele, Andrea Gaglione, and Michele Nati. Mind my Value: A Decentralized Infrastructure for Fair and Trusted IoT Data Trading. In *Proceedings of the Seventh International Conference on the Internet of Things*, page 15. ACM, 2017.

[263] Zhiqing Huang, Xiongye Su, Yanxin Zhang, Changxue Shi, Hanchen Zhang, and Luyang Xie. A Decentralized Solution for IoT Data Trusted Exchange Based-on Blockchain. In *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, pages 1180–1184. IEEE, 2017.

[264] Éric Dubois, Patrick Heymans, Nicolas Mayer, and Raimundas Matulevičius. *A Systematic Approach to Define the Domain of Information System Security Risk Management*, pages 289–306. Springer, 2010.

[265] R. Matulevičius. *Fundamentals of Secure System Modelling*. Springer International Publishing, 2017.

[266] N. Ahmed and R. Matulevičius. Securing Business Process Using Security Risk-oriented Patterns. *Computer Standards and Interfaces*, 36:723 – 733, 2014.

[267] N. Ahmed and R. Matulevičius. Presentation and Validation of Method for Security Requirements Elicitation from Business Processes. In *Information Systems Engineering in Complex Environments, Selected Extended Papers from CAiSE Forum 2014*, 2015.

[268] Blockchain.info. Bitcoin Charts and Graphs. URL https://www.blockchain.com/charts. (Accessed May 18, 2019).

[269] Ethereum Team. Byzantium HF Announcement. https://blog.ethereum.org/2017/10/12/byzantium-hf-announcement/, 2017. (Accessed May 29, 2019).

[270] Etherscan. Ethereum Charts and Statistics. https://etherscan.io/charts, 2017. (Accessed May 18, 2019).

[271] James Ray and Vitalik Buterin et al. Ethash Design Rationale. URL https://github.com/ethereum/wiki/wiki/Ethash-Design-Rationale. (Accessed June 11, 2019).

[272] Mobility Open Blockchain Initiative - VID Working Group. MOBI Vehicle Identity Standard - Version 1.0. URL: https://dlt.mobi/wp-content/uploads/2019/07/MOBI-Vehicle-Identity-Standard-v1.0-Preview.pdf, 2019. (Accessed October 26, 2019).

[273] Julius Köpke, Marco Franceschetti, and Johann Eder. Balancing Privity and Enforceability of BPM-Based Smart Contracts on Blockchains. In *International Conference on Business Process Management*, pages 87–102. Springer, 2019.

[274] Dorit Ron and Adi Shamir. How Did Dread Pirate Roberts Acquire and Protect his Bitcoin Wealth? In *International Conference on Financial Cryptography and Data Security*, pages 3–15. Springer, 2014.