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

Research Paper - Version 0.3
May 8, 2018

Benjamin Leiding and Will Vorobev

Chorus Technology
Email: hello@chorus.technology

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 Technology presents a blockchain-based system that enables a manufacturer agnostic platform solution that allows VANET participants to enact and transact any kind of provision of service and goods. This whitepaper fills the gap in the state of the art by introducing a blockchain-based transaction layer library for (semi)-autonomous vehicles that enables the upcoming V2X economy. We present the advantages of the system, outline the requirements and goals, as well as the architecture of the Chorus V2X platform and eco-system. In addition, we present a Chorus Technology prototype that demonstrates how our system can help mitigating traffic jams and at the same time provides a mean to car insurance companies to incentivize their customers to practice good driving behavior.

Keywords: Blockchain, Autonomous Vehicles, V2X Communication, Chorus Technology, Self-Driving Cars, VANET, Smart Contract

1 Introduction

Despite steadily growing public transport networks and systems, especially in most first world countries, cars and similar vehicles are still the default standard for urban transportation. In the US, "about 86 percent of all workers commuted to work by private vehicle, either driving alone or carpooling" [35] even though in recent years the numbers remained relatively stable after decades of consistent increase - similar applies to other industrial countries [36][56] even though the overall percentage of vehicle commuters in Europe is lower than in the US [10].

While it was normal for the last few decades to own a vehicle and commute on a day-by-day basis, the future will be radically different due to the progressing evolution of self-driving cars and autonomous vehicles. The car-sharing economy that developed in recent years in combination with autonomous cars results in a so called *passenger economy* [25]. Users no longer own cars, instead just hop on an autonomous cars, pick a destination and get delivered without any human interaction. An Intel report estimates the size of this economy to be around US$ 7 Trillion in 2050 [25].

Despite some recent setback, e.g. Uber and Tesla accidents [**?**][**?**][**?**], academic researchers as well as companies from the private sector make fast progresses in the research area of self-driving cars [**?**][**?**]. It took less than 15 years from the first DARPA Grand Challenge (a prize competition for autonomous vehicles) to self-driving cars operating on public streets on a regular base (Tesla, Waymo, Uber, etc.) [**?**][**?**][**?**]. Besides the cars, several projects are also already working on system solutions for trucks, rovers, drones, ships and even airplanes [**?**][**?**][**?**][9]. But progressing automation and driverless transport that enables the passenger economy is only a small aspect of the potential of these new technologies. During a talk[1] at the 2013 Turing Festival in Edinburgh, Mike Hearn did not only described a vision where most users don't own cars anymore and instead use services provided by autonomous vehicles that own itself, but also the potentials of a vehicle-to-vehicle (V2V) as well as a vehicle-to-infrastructure (V2I) economy. Autonomous vehicles (AVs) may own themselves, offer services and goods to earn money, and pay money to acquire services that they cannot provide on their own, e.g., car renting a parking lot, paying for a charged battery, using toll roads, or simple service check ups. The idea of V2V and V2I or in general V2X (vehicle-to-everything) will fuel various new business fields.

Certainly, traditional payment systems such as paper money or fiat currencies in general are not suited to be part of this new economy. There are slow, depend on third parties (e.g., banks) and suffer from bureaucratic overhead. Blockchain technology and cryptocurrencies offer a promising alternative payment solution that comes with several additional advantages that we will discuss later on. The blockchain technology, also referred to as distributed ledger system, is most noticeably known for providing the foundation of the peer-to-peer (P2P) cryptocurrency and payment system Bitcoin [38], but nowadays there a various different platforms out there, e.g., [24][48][62]. Several companies already started to prototype applications that combine vehicles and blockchains. Porsche is researching different payment-related applications for vehicles [49] whereas Ford focuses on traffic marshaling [32]. As expected in the early days of a new technology, companies focus on selective solutions for a selection of very specific problems or use cases and the resulting solutions are only compatible with their own products. What is currently missing in the new business field of V2X economy is an industry standard that can easily be integrated with self-driving and (semi)-autonomous cars or even nowadays cars.

[1] `https://www.youtube.com/watch?v=MVyv4t0OKe4`

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

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

2 Technical Background and Supplementary Literature

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

2.1 Blockchain Technology

As the name suggests, a blockchain consists of a chronologically ordered chain of blocks. Every block consists of a certain number of validated transactions and each of those block links to its predecessor by a hash reference. As a result, changing the content of one block also changes all succeeding blocks and hence breaks the chain. All blocks are stored on and verified by all participating nodes. While the initial Bitcoin blockchain only supported a very limited set of scripting instructions, the next generation of blockchain platforms, e.g., Ethereum [62], Qtum [11], or Tezos [24], provide Turing-complete programming languages on the protocol-layer level in order to enable smart contract capabilities. Smart contracts are, "orchestration- and choreography protocols that facilitate, verify and enact with computing means a negotiated agreement between consenting parties" [11]. Hence, the parties participating in the enactment of a smart contract establish binding agreements and deploy applications using such smart contracts in order to provide blockchain-based applications. Those application are as versatile as smart contracts itself and enable services including the finance sector [39][52], academic and business authentication and identity

solutions [4][8][27][34][53], reputation systems [6] as well platforms for Internet-of-Things (IoT) applications [7][46].

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

2.2 Autonomous Vehicles

During the last 15 years research on autonomous and self-driving cars progressed a lot and nowadays such cars already operating on public streets on a regular base, e.g., Tesla, Waymo, Uber, and so on. The ultimate goal of most manufacturers and researchers is to develop the first fully self-driving and autonomous vehicle. In order to clarify some definitions, this short section provides a short introduction to the most relevant terms and concepts.

An autonomous car, also referred to as a driverless car or robotic car, is able to navigating and interact with its environment without human input based on information provided by its sensors [20][59]. To do so, modern cars are equipped with radar- and laser sensors, lidars, GPS devices, cameras and several further sensing devices. Based on these information, the vehicle interprets the surrounding world and deduces appropriate action strategies such as avoiding obstacles (other vehicles, humans or a house) on the way to the supermarket [14][63]. As the technology developed over time, vehicles were equipped with more and more sensors, resulting in different driving capabilities. The SAE [51] defined six levels of driving autonomy to categorize the varying capabilities and progresses of several approaches:

- **Level 0 (No automation):** No driving autonomy, the driver has to perform all driving tasks and interactions.
- **Level 1 (Driver assistance):** The vehicle is controlled by the driver, but is supported by some basic driver assistance functionalities.
- **Level 2 (Partial automation):** The vehicle is able to perform some specific tasks (acceleration or steering) without driver input. Nevertheless, the driver must be fully engaged in driving task and monitor all decision of the car and the environment at all time. The user has to be able to intervene at any given moment.
- **Level 3 (Conditional automation):** The driver is still a necessity, but is not required to monitor the vehicle or the environment at any given moment. But given a notification by the vehicle, the driver has to be able to take back control over the car in case the vehicle encounters a situation that it cannot deal with on its own.
- **Level 4 (High Automation):** The vehicle is able to perform all driving tasks without human intervention in most driving scenarios. The driver can take control whenever desired.

– **Level 5 (Full Automation):** The vehicle is able to perform all driving tasks without human intervention in all driving scenarios. The driver can take control whenever desired as long as a steering wheel is still part of the vehicle.

The Chorus interaction- and transaction layer library supports and enables a varying number of services for vehicles of each automation level, whereas the most sophisticated applications require SAE level 5 and simpler plug-ins may only require SAE level 0.

2.3 Vehicular Ad-Hoc Networks - VANETs

Communication between vehicles, road infrastructure and Internet-based services is a key enabler of upcoming generation of vehicles. So called vehicular ad-hoc networks provide an abstract concept that models the different components that are required for V2V, V2I and V2X communication. As illustrated in Figure 1, the basic components of VANETs are vehicles, on-board-units (OBUs), application-units (AUs) and road-side-units (RSUs).
RSUs are placed along the road side or in dedicated locations such as at crossroads. Usually, RSUs provide short range communication based on IEEE 802.11p radio technology but can also be equipped with other network devices in order to provide communication within the infrastructural network [1]. An OBU is typically mounted onto a vehicle and used for exchanging information with RSUs or other OBUs. Short range wireless communication or other radio technologies are usually used to exchange information [3].
Closely linked to the OBU is the AU. They might even reside in the same physical unit or is mobile and might be regularly removed from the vehicle (e.g smartphones). The AU provides an execution environment for applications that utilize the OBU's communication capabilities [1][3].
Communication in VANETs occurs either inside a vehicle between AUs and OBU, wirelessly between different vehicles (V2V), vehicles and infrastructure (V2I) or vehicles and the infrastructure via broadband (V2B) [18]. For authentication purposes, each network participant is equipped with a unique public/private key pair which usually resides in a tamper-proof-device (TPD). In blockchain terms, the TPD is similar to an external hardware wallet.

3 The Chorus Vision

3.1 Use Cases

Work-In-Progress

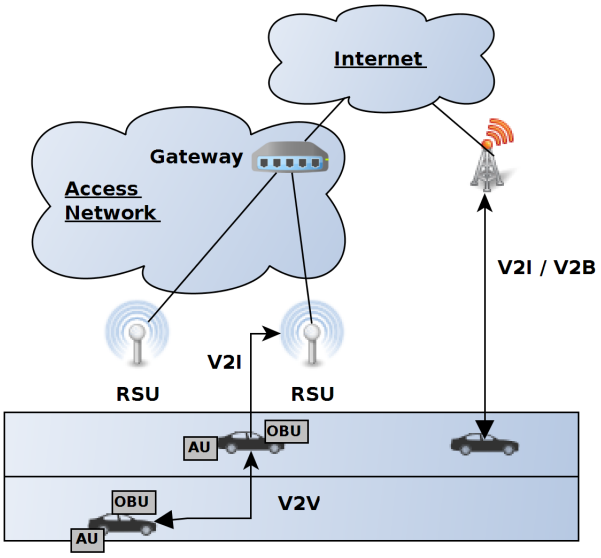

Fig. 1. General VANET architecture (Based on: [3] and [28])

3.2 Vehicle to Human

3.3 Vehicle to Vehicle

3.4 Vehicle to Infrastructure

4 System Design and Architecture

The vision of Chorus outlined in Section 3 is now analyzed from technical perspective as part of the following section. In order to identify, structure and formalize the critical requirements and stakeholders on an abstract level, we use one part of an Agent-Oriented Modeling (AOM) method [57], i.e., goal models. Section 4.1 introduces AOM goal models and the Chorus specific goal model. The produced goal model is used in subsequent Section 4.2 to derive the Chrous system architecture . The resulting system architecture and specifications serve as implementation guidelines.

4.1 Functional Goals, Quality Goals, Stakeholders and Requirements

In system development and software engieerng, good requirements follow certain characteristics. According to [13][21] requirements address one issue only and are completely specified without missing information. Moreover, they have to be consistent and do not contradict itself, or in correlation with other requirements. Finally, a requirement must also be atomic and without conjunctions [41].

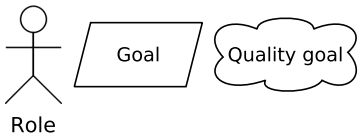

Fig. 2. Selection of AOM notation elements.

The AOM methodology is a socio-technical requirements-engineering approach used to model complex systems that consist of humans, devices, and software agents. An AOM goal model enables both, technical- and non-technical stakeholders, to capture and understand the functional- and non-functional requirements of a complex system. Figure 2 depicts the three main elements that an AOM goal model comprises in order to capture the system requirements and goals. Roles of involved entities are represented in form of sticky men, whereas functional requirements are depicted as parallelograms. Note that in the specific context of this whitepaper, a sticky man does not exclusively represent human entities but rather all kinds of entities, e.g., also vehicles or infrastructure. Functional requirements are also referred to as goals. Non-functional requirements are depicted as clouds and refer to quality goals of the modeled software system. The AOM goal model follows a tree-like hierarchy with the root value proposition of the modeled system at the top. Subsequently, this main goal is decomposed into sub-goals where each sub-goal represents an aspect for achieving its parent goal [33]. 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 4.1.1 introduces the top-level goal model our system, followed by Section 4.1.2 focusing on the non-functional goals of the AOM goal model.

4.1.1 Top-Level AOM Goal Model

The presented AOM goal model is similar to the model presented by the authors in [26], since implementation of a V2X system is a specific use case of the more abstract and general M2M (machine-to-machine) interaction platform represented in that paper. Meanwhile, Figure **??** 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 interaction and transaction layer library for (autonomous) vehicles, 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, one 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 (e.g., Tesla), the sub contracts each offer service provision for a specific service, interaction, transaction or application.

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 partners, but for the 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 basically equivalent to the on- and off-chain supply and demand management. Besides that, plug-ins and dApps of the Chorus eco-system might use platform smart contracts for service enactment and have to be integrated as well in this context.

The third functional requirement, that represents the third main component, enables V2X transaction via the blockchain. The most important part here is the transaction management via a smart contract lifecycle that will be detailed later on in Section 5.1.

Finally, the fourth functional requirement focus on the enactment of various plug-ins and decentralized applications (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 eco-system depending on the use case. Since nowadays most blockchains offer Turing-complete smart contract support, the variety of applications and plug-ins in our eco-system is quite vast.

4.1.2 Non-Functional Requirements

Besides the four sub-goals of the top-level AOM goal model, we further identify thirteen quality goals of the main value proposition that are inherited to all refining sub-goals. A *scalable* system design is necessary to provide Chorus services to a large quantity of users and customers. 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, e.g., tedious and repetitive tasks. *Cost efficiency* is another important quality goal. *Flexible* digital collaboration is a highly dynamic process that involves the enactment of diverse activities, the participation of diverse partners, and the exchange of diverse data [40]. 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 being easy to use (*Usable*) for business collaboration. According to [41], 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" [41].

Moreover, we assign two additional quality goals that ensure a *blockchain agnostic* as well as *entity agnostic* design. Chorus should be neither limited to a specific blockchain nor vehicle hardware of a specific vendor. *Interoperable* hardware and software design is another consequence of the previous quality

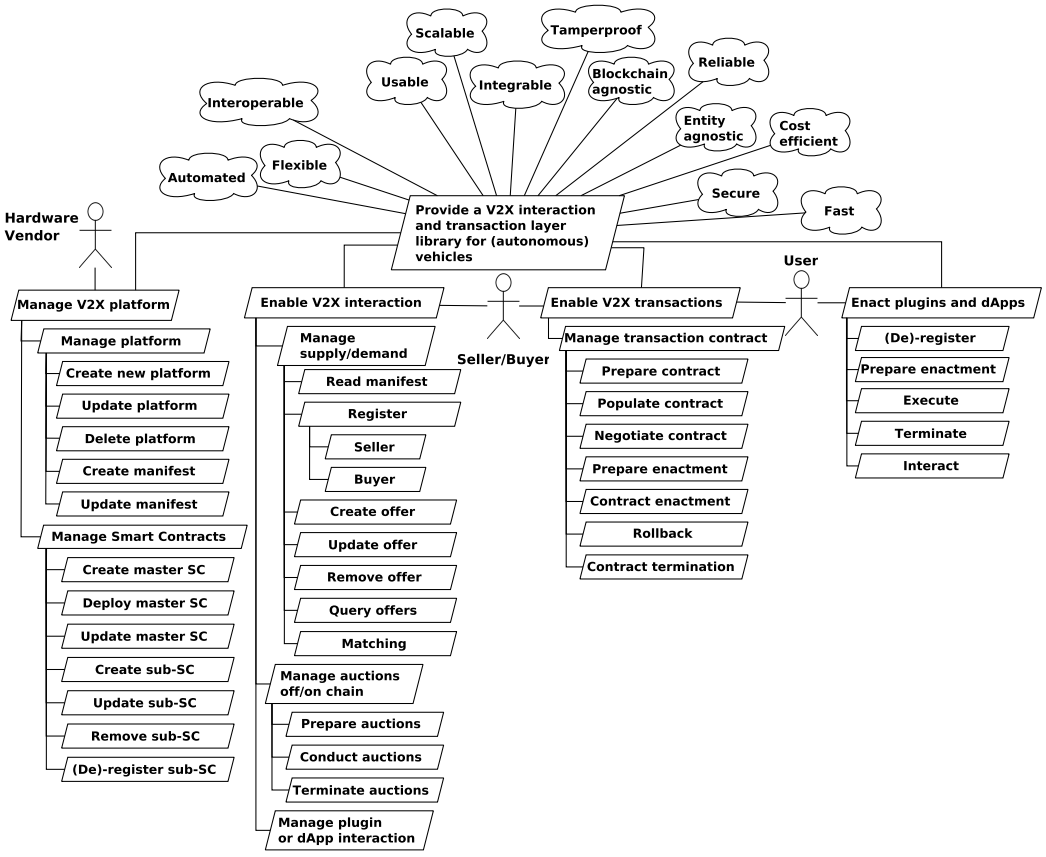

Fig. 3. Chorus - Top-level goal model representation (Source: Based on [26]).

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 also 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. A *reliable* enactment of all Chorus-based interactions and transaction facilitates the previous goals as well. Data communicated internally as well as externally has to protected against unauthorized tampering (*tamperproof*) in order to protect business collaborations, but also ensure the safety of participating entities. Finally, since cars and similar vehicles move much faster than humans, a *fast* service provision is essential for most tasks.

The presented goal model is used in the following Section 4.2 to derive our system architecture. We do not list all details of the further refined AOM goal model in this whitepaper due to space constraints and in order to focus on the most relevant system components and features.

4.2 System Architecture

The abstract system- and business architecture is derived from the functional- and non-function requirements of the AOM goal model presented earlier. The services are powered by a service-oriented architecture (SOA) that is comprised of different designated components. Each of these components is self-contained, well-defined components and provides a specific set of services [17][47]. Dedicated services and components may also consist of other underlying sub-services [50].

In the following, a technology-agnostic UML-component-diagram representation is used to illustrate the system architecture [5][55]. The UML notation elements used to model the architecture are presented in Figure 4. 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 4. 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 [5][55]. Once more, sticky men are used to depict entities and their interactions with the system.

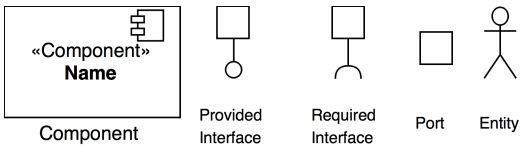

Fig. 4. UML-component diagram notation elements.

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

4.2.1 High-Level Architecture

The highest architecture abstraction level of our system is depicted in Figure 5. The representation is divided into two distinct packages, e.g, the Blockchain package and the Vehicle-System package. In UML, packages are used to group elements, and provide a namespace for the grouped elements [55]. 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, as well as the Chorus mobile smartphone application. The vehicle-system package consists of three main components, e.g., the Chorus firmware, the vehicles manufacturer OS and the component managing the underlying hardware of the vehicle, and several smaller components that will be detailed in Section 4.2.2.

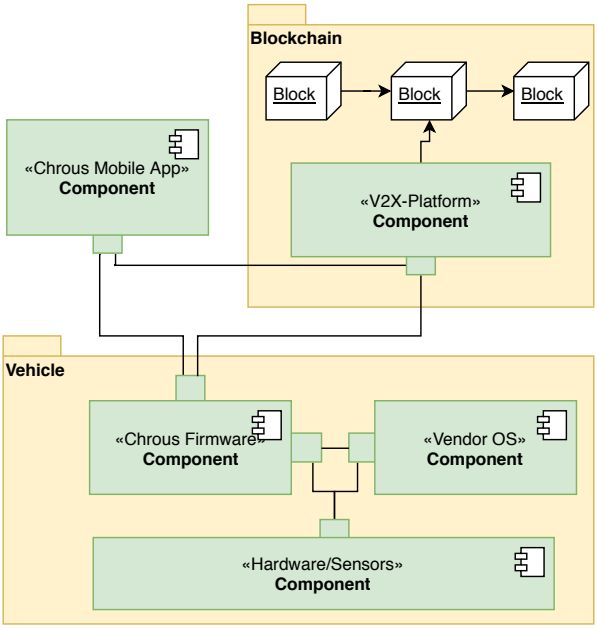

Fig. 5. High-level architecture of the Chorus V2X system.

The blockchain package comprises the V2X platform itself and the blockchain utilized by Chorus to enable our services. The following Section 4.2.2 explains each of the units illustrated in Figure 5 in detail.

4.2.2 Selected Architecture Refinements

The following illustration represent visualization of the Chorus longterm architecture. We simplified certain aspects due to space constraints and to reduce technical complexity.First, Figure 6 shows the different components of the *Chorus Mobile App* component, e.g., the user interface (UI) component, the settings/preferences component, the wallet component, the communication component and the chorus application management component. The UI-component is the gatekeeper for the user and used to control all functionalities from the user side and also to interact with the system. The user can set preferences and change settings, visualize wallet balances, activate plug-ins or Chorus applications (we only listed two as an example) such as the "Good-driving-behavior-plug-in. The wallet component holds the users public and private key pair that represent the wallet address. The user can transfer tokens from and to his/her wallet to use Chorus-compatible services or applications. The user's smartphone is connected to the vehicle component and utilizes the communication component to communicate with the vehicle component as well as the V2X platform. Besides these functional components, Figure 6 also contains the Android OS component and the underlying hardware (GPS, Wifi, USB, bluetooth, etc.) of the smartphone for easier visualization purposes.

Next, Figure 7 presents a more detailed view of the *vehicle* sub-system. In the beginning, this sub-system is located in a dedicated box that is connected to the user's smartphone via bluetooth or Wifi, the V2X platform via the Internet, to other vehicles via the WAVE protocol stack [29][60] as well as the vehicle via OBDII - hence we have a dedicated communication component on the left as well as the right for each of these two functionalities. The OBDII interface is used to query information from the car such as speed, steering, breaks and many more. The information are used by Chorus applications and plug-ins to provide their services. The logic of the Chorus applications mostly resides in the Chorus plug-in component of Figure 7 while the Chorus application component in Figure 6 is mostly used to control the applications and enable them on the dedicated Chorus box. The Chorus component in Figure 7 also contains the Chorus firmware as well as the blockchain client(s).

Due to the speed of moving cars, most of the time on-chain auctions are not an option and instead auctions or negotiation on a local level between nearby vehicles are necessary. The auction components contains all functionalities to do that as well as settings that control the auction preferences of the vehicles. More details on the actual workflow of the off/on-chain auctions are available in Section 5.2.

In the future, when most people do not own cars themselves any more, users can transfer tokens to the vehicles wallet to pay for taxi services and the vehicle uses these earnings to pay for electricity or maintenance. Hence, the vehicle also has a separate wallet for this purpose.

Similar to the previous Figure 6, at the bottom we list hardware-focused components such as the different sensors that may reside in the vehicle or the Chorus box, different hardware connectors and the tamperproof device (TPD) that con-

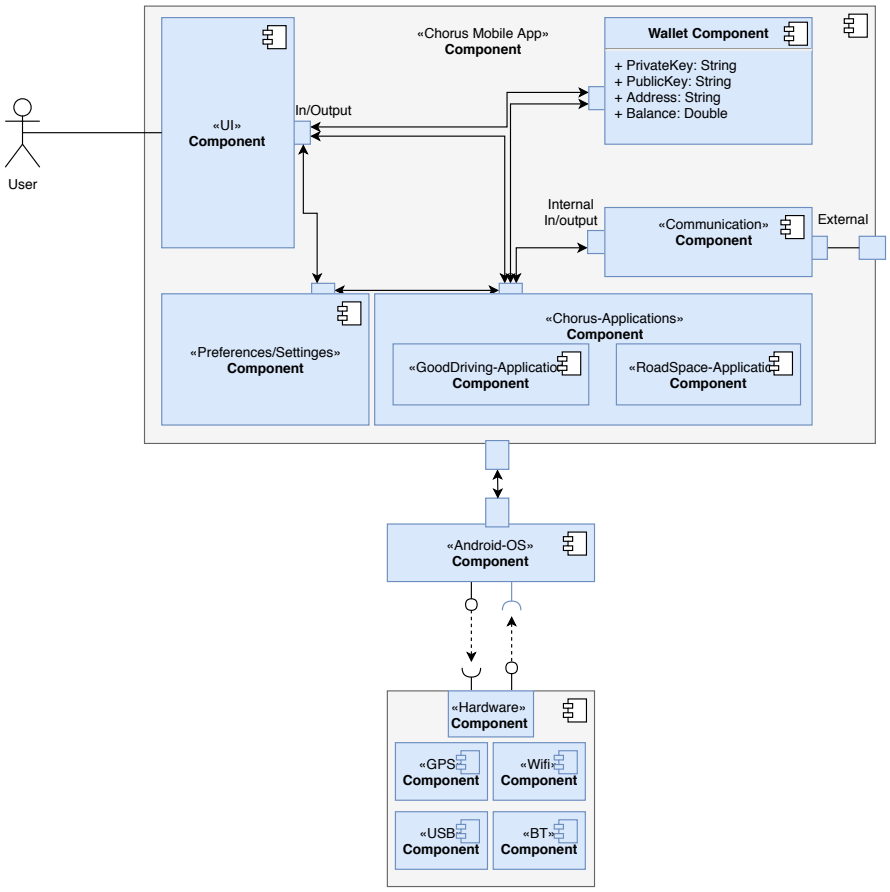

Fig. 6. Refined illustration of the *Chorus Mobile App* component.

trols the car's identity, authenticates exchanged messages and data packets and so on.

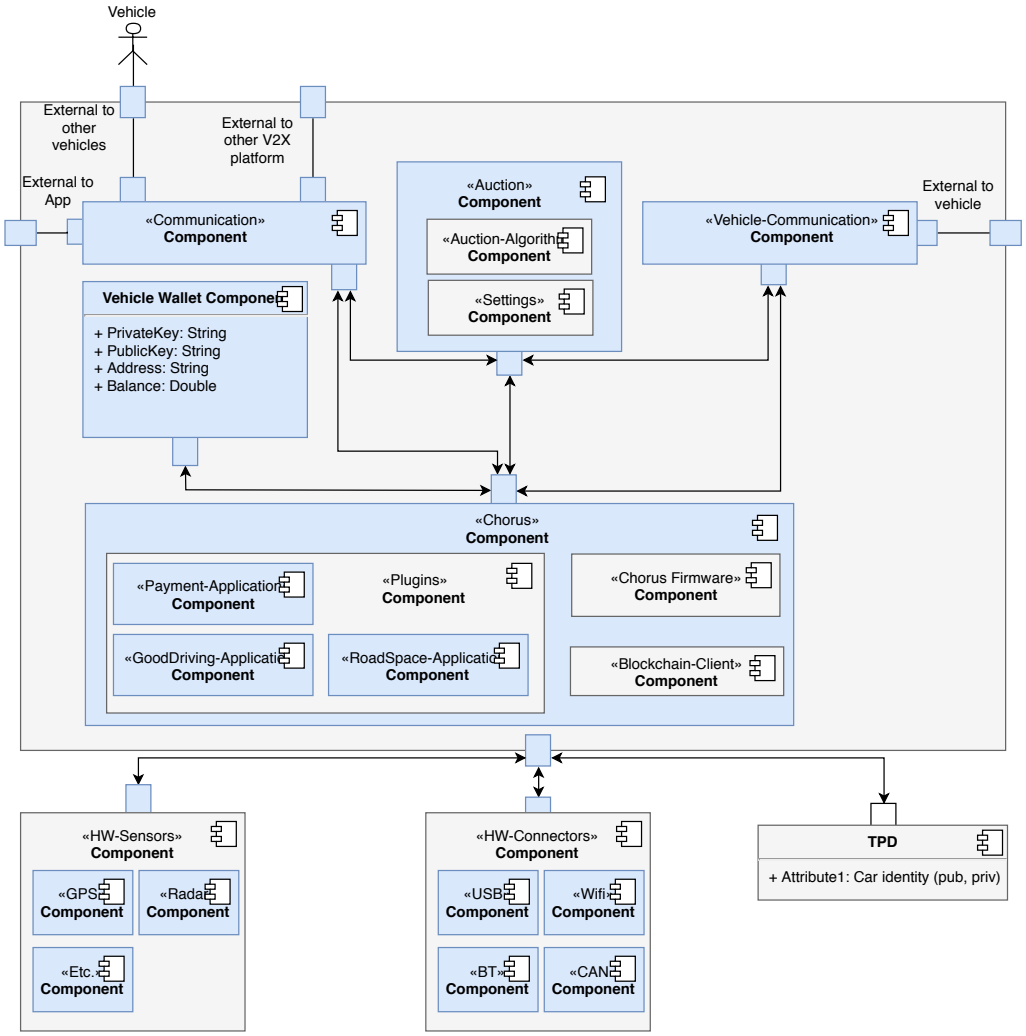

Fig. 7. Refined illustration of the *Vehicle* sub-system.

Finally, Figure 8 presents a more detailed view of the *V2X platform* sub-system. As before for all the other sub-system, the V2X platform also contains a communication component managing the communication between the platform and the blockchain and vehicles. In addition, the respective platform management entity (Chorus or a vehicle manufacturer that is running their own Chorus compatible platform) has access to an administration interface. The adminis-

trator uses this interface to maintain the platform and the corresponding smart contracts as well as the user management and much more. Similar to the auction component in the previous Figure 7, the V2X platform has a component that takes care of on-chain auctions using the same auction algorithms. Auctions may result based on the supply and demand management that is conducted in the corresponding component. Finally, similar to the application/plug-in components in the mobile application and the vehicle, we also have a pendant of these component in the V2X platform. Here, the component is mainly used for application/plug-in related interaction and transaction enactment.

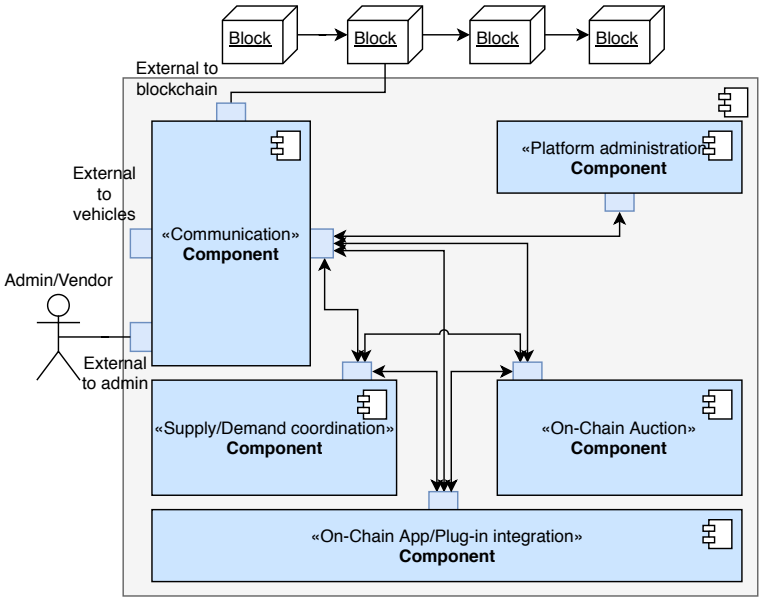

Fig. 8. Refined illustration of the *V2X-Platform* sub-system.

Next, in Section 5 we present the system-engagement processes of the the Chorus V2X system using and outline the interaction workflow of the on/off-chain auction algorithms.

5 System Engagement Processes

The Chorus transaction layer library automates and simplifies VANET-based V2X service provision on several levels. A core element of many of the Chorus use cases is smart contract-based negotiation and contract enactment between entities that are the result of collaborating tasks and subprocesses. For example, as described in Section 2 two vehicles conduct a road-space negotiation auction that results either in a change of positions or is aborted. This process potentially

also 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 paper follow at some point a similar procedure on smart-contract level. 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 are represented using Business Process Model and Notation (BPMN) [44] and UML sequence diagrams [45].

Consequently, Section 5.1 details the BPMN representation of the generalized contract negotiation lifecycle, followed by Section 5.2 that details the different auction mechanisms of our platform. Finally, Section 5.3 covers the Chorus token value proposition.

5.1 Smart Contract Negotiation Lifecycle Management

The abstract smart contract negotiation lifecycle, as illustrated in Figure 9, is divided into the following stages: a) preparatory, b) negotiation, c) contract execution d) rollback and e) a contract expiry stage. During the preparatory stage, 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 formally defined by specifying, e.g, the content and target of the contract. Following, the example of the cab service for a human, this might include the start location, final destination, price, and so on. 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. After signing the agreement, the contract execution phase is triggered and the the vehicle picks up the user.

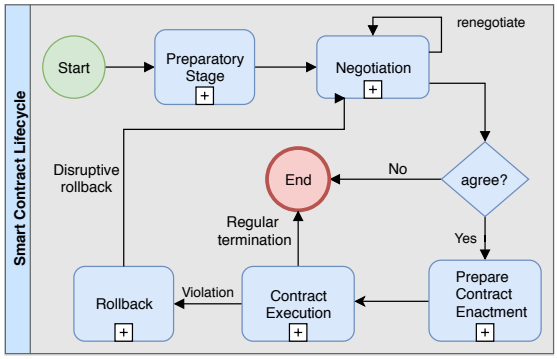

Fig. 9. Smart contract negotiation lifecycle (Based on [11]).

The transportation contract terminates, or expires either after the defined user arrives the final destination, or when the contract is prematurely terminated. Failing to transport the user to the agreed final destination might result in an immediate rollback of the smart contract or invokes some kind of an mediation process that is supervised by a conflict resolution escrow service that is not depicted in this illustration.

The lifecycle presented in Figure 9 not only cover trading negotiation but rather all kind of contract enactments, also the insurance use case from Section 6 that we implement as a prototype. You prepare and negotiate a contract with your insurance company and execute in case you agree on the specifications. That also covers what happens in case you behave correctly, e.g., reward, as well as punishing bad behaviors, e.g., paying a penalty. A serious violation of the contract from any of the involved parties might result in an early termination of the contract or even a rollback.

5.2 Auction Algorithms

A further important core concept of the Chorus transaction library is to support the exchange and provision of goods on services between entities (V2X). 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 his/her 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 two auction algorithms based on the concept of so called Vickrey Auction [37][61]; An algorithm for the scenario with exactly one buyer and one seller, as well as an algorithm that can be used in scenarios with multiple buyers and multiple sellers. Figure 10 illustrates one-to-one auctions and Figure 11 many-to-many auctions.

During a Vickrey auction, participants exchange sealed bid. Each bidder submits a written and signed bid without knowing the bid of the other participants. In the end, the highest bidder wins but instead of paying the price of this highest bid, the price paid is the second-highest bid. Due to space constraints and the technical nature of this whitepaper we will not cover the economical and game theoretical implications concepts of Vickrey auctions and instead refer the reader to specific supplementary literature, e.g., [2][16][30][37][61].

Figure 10 presents a sequence diagram of our one-to-one auction algorithm. We assume that the buyer is not willing to pay more than US$3, and the seller is not selling for less than US$2.80. As mentioned in our AOM goal model (Figure 3, speed is one of the non-functional goals of our system - hence, only one auction round is conducted. Both participants prepare an encrypted (sealed) and signed bid before exchanging the bids. As soon as both participants received the other parties bids, 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 US$2.80 the auction is successful and due to the second-price rule of Vickrey auctions, the buyer pays US$2.80 to the seller. In case the buyer offers less than the seller's minimum price the auction ends without a deal.

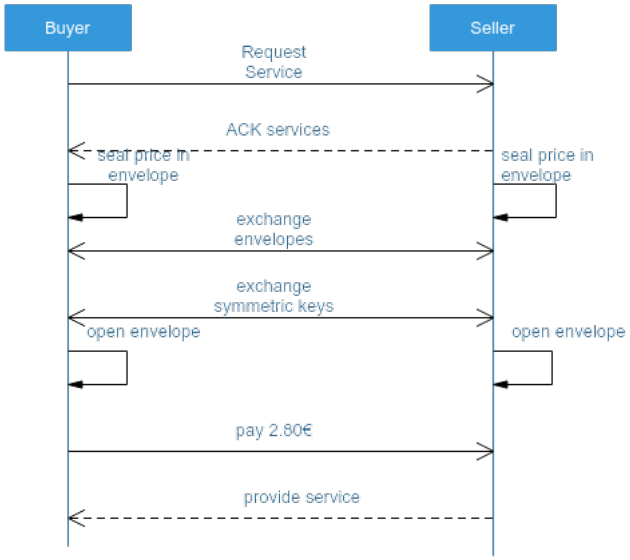

Fig. 10. Chorus auction algorithm - 1 buyer and 1 seller.

In case multiple buyers and sellers participate in an auction the workflow is very similar as illustrated in Figure 11. We assume a scenario where buyer one is willing to pay a price of US$1.80, buyer two offers a price of US$3.20 and buyer three offering US$3.50. The seller is not selling for less than US$2.0. Again, we only 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 US$3.20. 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 payed price is higher than the matched seller's minimum price.

5.3 Token Economics and Value Proposition

Work-In-Progress

6 Prototype and Feasibility Study

Intro

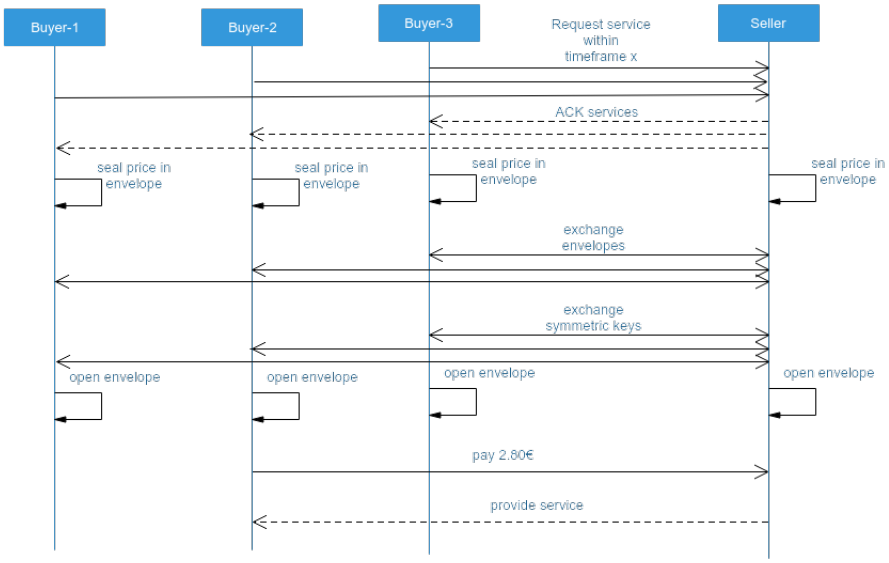

Fig. 11. Chorus auction algorithm - Many buyers and many sellers.

6.1 Prototype Architecture

Work-In-Progress

Fig. 12. System architecture of the Chorus prototype.

while current localization is done via hardware in the user's smartphone (GPS, Glonass, etc.) future version of the Chorus platform will also incorporate alernative localization service such as FOAM [19]. FOAM provides .....

6.2 Eco-System and Economics

Work-In-Progress

6.3 Evaluation

Work-In-Progress

7 Analysis of Related Work

In recent time, other project and teams proposed solutions and systems that are related or overlap with the functionalities of our solution. The following Section 7.1 briefly highlights some academic projects, whereas Section 7.2 outlines competitors from the private sector.

7.1 Academic Competitors

The authors of [28] outline a blockchain-based solution for a variety of services within self-managed vehicular ad-hoc networks (VANETs) such as traffic management, toll payment systems, traffic regulation enforcement and more. In [26], the same authors envision a more holistic and abstract machine-to-machine (M2M) economy enabled by a platform that allows trading of goods and service between any kind of machines - ranging from toll road management to automated food supply- and demand management between fridges and the supermarket and several other scenarios. A special case of this research was independently published by the authors of [54]. They propose and implement a M2M electricity market for chemical industry where energy producers and consumers are trading electricity with each other via a blockchain platform.

7.2 Non-Academic Competitors

The alternative blockchain implementation IOTA [48] is probably one of the most noticeably known IoT focused blockchains and widely used. IOTA is offering an IoT market place [22] for IoT devices where sensor data can be bought and sold using blockchain technology, but has no smart contract capabilities and is more a general purpose IoT storage chain than a service provision platform.

Oaken Innovations[2] implemented a prototype of a blockchain-based toll road system [43] as well as a blockchain-based water meter application [42]. Despite support from different car manufacturers their architecture currently lacks scalability and is dependent on in-vehicle hardware nodes resulting in integration overhead.

Swarm City[3] is a blockchain-based and decentralized commerce platform work running on the Ethereum network [12][58]. It has a built in marketplace and reputation system and focuses on enabling transactions between humans without third-party interaction. By putting all computation logic on the Ethereum blockchain, the scalability and computation power of Swarm City couple dApps is rather limited and not suitable to be applied in the context of VANETs.

Flying Carpet[4] is a blockchain-based, decentralized and autonomous P2P transportation network. The project is in its early stages and the MVP use case focuses on decentralized charging and docking station for drones or UAVs.

[2] `https://www.oakeninnovations.com/`
[3] `https://swarm.city/`
[4] `http://www.flyingcarpet.network/`

The DAV project [5] outlines a solution that allows vehicles to discover, communicate and transact with one another using cryptocurrencies [9]. Their use cases are similar to some of the Chorus examples described earlier, but rather than providing a coherent transaction layer protocol with an integrable library, they focus on use case based solutions.

In 2018, the Ford Motor Company got a patent approval for a traffic marshaling via a blockchain system [32] that is similar to the ideas described in the academic papers of [28]. Despite having a patent, the solution is focused on a single use case and not suitable to be adapted to a general service provision platform for VANETs.

XAIN[6] created and implemented their own Ethereum-based low-energy blockchain with a customized proof-of-work algorithm [31]. Despite having worked on a jointed project with car manufacturer Porsche, their vision and longterm strategy remains unclear.

Slock.it[7] is another project focusing on specific use case rather than a general solution concept. They are working on a universal sharing network that enables anyone to rent, sell or share their property without a middleman [23]. Even though their solution potentially also includes cars, they do not consider any other of the use cases presented in this whitepaper.

DOVU[8] [15] is currently testing a system for drivers to log mileage in leased or borrowed vehicles in the UK. In general, DOVU is working on a vehicle-focused ubiquitous rewards system that can be used in various scenarios of VANETs.

8 Conclusion and Future Work

This whitepaper presents the Chorus Technology blockchain-based transaction and interaction layer for (semi)-autonomous vehicles that enables a V2X platform for goods and services. We outline and describe the technical foundations of this new economy, the longterm vision and benefits as well as the different use cases and scenarios of V2X transactions and interactions, e.g., vehicle-to-vehicle (V2V), vehicle-to-human (V2H), or vehicle-to-infrastructure (V2I).

Based on the use cases and scenarios we identify the requirements and criteria that a blockchain-based V2X transaction and interaction layer protocol must satisfy. With respect to functional and non-functional requirements, Chorus aims to develop a blockchain- as well as manufacturer agnostic and interoperable V2X platform that enables interaction and transaction between participating entities and a plug-in interface for external applications. Subsequently, we derive the service-oriented architecture of the our system based on the identified requirements and goals. We present the system architecture using technology-agnostic UML-component and sequence diagrams that detail the systems main components and communication interfaces. In order to ensure widespread adoption,

[5] `https://dav.network/`
[6] `https://www.xain.io/`
[7] `https://slock.it/`
[8] `https://dovu.io/`

special focus will be given in the future to the API design and library integration for car manufacturers.

A core element of many of the Chorus use cases is smart contract-based negotiation and contract enactment between entities that are the result of collaborating tasks and subprocesses. On an abstract level, most of the use cases presented in this paper follow a similar workflow on the smart-contract level. Hence, we decided to integrate an abstract smart contract negotiation lifecycle that we describe. The lifecycle is divided into the different stages (preparatory, negotiation, contract execution, rollback and contract expiry stage) that we explain in detail. Furthermore, we designed two auction algorithms for the V2X economy that allow to reach an efficient consensus on a certain price between buyer and seller. Our auction algorithms are based on the concept of so called Vickrey Auction, and we envisioned one algorithm for one-to-one interaction as well as a second workflow for scenarios with multiple buyers and sellers. Finally, we present the Chorus token value proposition and the surrounding token economy eco-system that fuels the V2X platform.

Next, we present a Chorus prototype implementation. We demonstrate how our protocol can help mitigating traffic congestions and at the same time provides a mean to car insurance companies to incentivize their customers to practice good driving behavior.

Future releases will focus on the longterm vision of Chorus Technology and the development of the abstract transaction and interaction layer as well as the API and library integration. Besides that, we will also continue to focus on further research aspects of the upcoming V2X economy that will facilitate future developments of Chorus Technology.

References

1. Al-Sultan, Saif and Al-Doori, Moath M and Al-Bayatti, Ali H and Zedan, Hussien: A Comprehensive Survey on Vehicular Ad Hoc Network. Journal of Network and Computer Applications 37, 380–392 (2014)
2. Ausubel, L.M., Milgrom, P., et al.: The lovely but lonely vickrey auction. Combinatorial auctions 17, 22–26 (2006)
3. Baldessari, R., Bödekker, B., Deegener, M., Festag, A., Franz, W., Kellum, C.C., Kosch, T., Kovacs, A., Lenardi, M., Menig, C., et al.: Car-2-Car Communication Consortium - Manifesto (2007)
4. Bochem, A., Leiding, B., Hogrefe, D.: Unchained Identities: Putting a Price on Sybil Nodes in Mobile Ad hoc Networks. In: Security and Privacy in Communication Networks (SecureComm 2018). Singapore (August 2018)
5. Booch, G., Jacobson, I., Rumbaugh, J., et al.: The Unified Modeling Language. Unix Review 14(13), 5 (1996)
6. Calcaterra, C., Kaal, W.A., Vlad, A.: Semada Technical Whitepaper - Blockchain Infrastructure for Measuring Domain Specific Reputation in Autonomous Decentralized and Anonymous Systems. URL: `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3125822` (2018), (Accessed April 18, 2018)
7. Christidis, K., Devetsikiotis, M.: Blockchains and Smart Contracts for the Internet of Things. IEEE Access 4, 2292–2303 (2016)

8. Civic Technologies, Inc.: CIVIC - Whitepaper. URL: https://tokensale.civic.com/CivicTokenSaleWhitePaper.pdf (2017), (Accessed May 01, 2018)
9. Copel, N., Ater, T.: DAV - Decentralized Autonomous Vehicles (Whitepaper). URL: https://dav.network/whitepaper.pdf (2017), (Accessed April 27, 2018)
10. Cortright, J.: Sprawl Tax: How the US Stacks Up Internationally . URL: http://cityobservatory.org/sprawl-tax-how-the-us-stacks-up-internationally/ (2016), (Accessed April 26, 2018)
11. Dai, P., Mahi, N., Earls, J., Norta, A.: Smart-Contract Value-Transfer Protocols on a Distributed Mobile Application Platform. URL: https://qtum.org/uploads/files/a2772efe4dc8ed1100319c6480195fb1.pdf (2017), (Accessed May 01, 2018)
12. David, C., Ponnet, S., De Wachter, K., Adriaenssen, B., Thuy, M.: Arcade.city - Blueprin for a New Economy - Whitepaper Version 1.2. URL: https://drive.google.com/file/d/0B9RSMdR2vWssV2JJX0t6dmN6SUk/view (2016), (Accessed May 04, 2018)
13. Davis, A.M.: Software Requirements: Objects, Functions, and States. Prentice-Hall, Inc. (1993)
14. Dokic, J., Müller, B., Meyer, G.: European roadmap smart systems for automated driving. European Technology Platform on Smart Systems Integration (2015)
15. DOVU.IO: DOVU - How the World's First Mobility Cryptocurrency Will Transform Data Consumption and Distribution Across the Transport Ecosystem - Whitepaper. URL: https://dovu.io/whitepaper.pdf (2017), (Accessed May 04, 2018)
16. Edelman, B., Ostrovsky, M., Schwarz, M.: Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. American economic review 97(1), 242–259 (2007)
17. Erl, T.: Service-Oriented Architecture: Concepts, Technology, and Design. Pearson Education India (2005)
18. Faezipour, M., Nourani, M., Saeed, A., Addepalli, S.: Progress and Challenges in Intelligent Vehicle Area Networks. Communications of the ACM 55(2), 90–100 (2012)
19. Foamspace Corp: FOAM - Whitepaper. URL: https://www.foam.space/publicAssets/FOAM_Whitepaper_May2018.pdf (2018), (Accessed May 08, 2018)
20. Gehrig, S.K., Stein, F.J.: Dead reckoning and cartography using stereo vision for an autonomous car. In: Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on. vol. 3, pp. 1507–1512. IEEE (1999)
21. 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)
22. IOTA Foundation: IOTA Sensor Marketplace. URL: https://data.iota.org/m (2017), (Accessed April 25, 2018)
23. Jentzsch, C.: Decentralized autonomous organization to automate governance - Whitepaper. URL: https://download.slock.it/public/DAO/WhitePaper.pdf (2016), (Accessed May 04, 2018)
24. L. M. Goodman: Tezos - A Self-Amending Crypto-Ledger (White paper). URL: https://www.tezos.com/static/papers/white_paper.pdf (2014), (Accessed April 27, 2018)
25. Lanctot, R.: Accelerating the Future: The Economic Impact of the Emerging Passenger Economy. URL: https://newsroom.intel.com/newsroom/wp-content/uploads/sites/11/2017/05/passenger-economy.pdf (2017), (Accessed April 27, 2018)

26. Leiding, B.: The Machine-to-Machine Economy Revolution - A blockchain-based trading platform for the Internet of Things (2018), unpublished
27. Leiding, B., Cap, C.H., Mundt, T., Rashidibajgan, S.: Authcoin: Validation and Authentication in Decentralized Networks. In: The 10th Mediterranean Conference on Information Systems - MCIS 2016. Cyprus, CY (September 2016)
28. Leiding, B., Memarmoshrefi, P., Hogrefe, D.: Self-Managed and Blockchain-Based Vehicular Ad-Hoc Networks. In: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct. pp. 137–140. ACM (2016)
29. Li, Y.J.: An overview of the dsrc/wave technology. In: International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness. pp. 544–558. Springer (2010)
30. Lucking-Reiley, D.: Vickrey auctions in practice: From nineteenth-century philately to twenty-first-century e-commerce. Journal of economic perspectives 14(3), 183–192 (2000)
31. Lundbaek, L.N., Beutel, D.J., Huth, M., Kirk, L.: XAIN - Practical Proof of Kernel Work and Distributed Adaptiveness - Yellow Paper Version 1.2. URL: `https://www.xain.io/pdf/XAIN_Yellow_Paper.pdf` (2017), (Accessed May 04, 2018)
32. MacNeille, P.R., Wisniewski, J., DeCia, N.: Vehicle-to-Vehicle Cooperation to Marshal Traffic (Mar 27 2018), uS Patent 9,928,746
33. Marshall, J.: Agent-Based Modelling of Emotional Goals in Digital Media Design Projects. International Journal of People-Oriented Programming (IJPOP) 3(1), 44–59 (2014)
34. McCorry, P., Shahandashti, S.F., Clarke, D., Hao, F.: Authenticated key exchange over bitcoin. In: International Conference on Research in Security Standardisation. pp. 3–20. Springer (2015)
35. McKenzie, B.: Who Drives to Work? Commuting by Automobile in the United States: 2013. American Community Survey Reports (2015)
36. Mieke Berends-Ballast et al.: Transport and Mobility 2016. URL: `https://www.cbs.nl/en-gb/publication/2016/25/transport-and-mobility-2016` (2016), (Accessed April 26, 2018)
37. Moldovanu, B., Tietzel, M.: Goethe's second-price auction. Journal of Political Economy 106(4), 854–859 (1998)
38. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System. URL: `https://bitcoin.org/bitcoin.pdf` (2008), (Accessed May 01, 2018)
39. Nguyen, Q.K.: Blockchain - A Financial Technology for Future Sustainable Development. In: Green Technology and Sustainable Development (GTSD), International Conference on. pp. 51–54. IEEE (2016)
40. Norta, A.: Exploring Dynamic Inter-Organizational BusinessProcess Collaboration: Privacy Protecting Concepts for ChoreographingeSourcing in B2B with Service-Oriented Computing. VDM Verlag (2008)
41. Norta, A., Grefen, P., Narendra, N.C.: A Reference Architecture for Managing Dynamic Inter-Organizational Business Processes. Data & Knowledge Engineering 91, 52–89 (2014)
42. Oaken Innovations: Project Oaken - Water Meter Acorn. URL: `https://github.com/Oaken-Innovations/water-meter-acorn` (2016), (Accessed May 04, 2018)
43. Oaken Innovations: Project Oaken - Tesla Tollbooth Acorn. URL: `https://github.com/Oaken-Innovations/tesla-tollbooth-acorn` (2017), (Accessed April 27, 2018)
44. Object Management Group: Notation (BPMN) Version 2.0. OMG Specification (2011), (Accessed May 04, 2018)

45. Object Management Group: OMG Unified Modeling LanguageTM (OMG UML), Superstructure - Version 2.4.1. OMG Specification (2011), (Accessed May 04, 2018)
46. Ouaddah, A., Elkalam, A.A., Ouahman, A.A.: Towards a Novel Privacy-Preserving Access Control Model Based on Blockchain Technology in IoT. In: Europe and MENA Cooperation Advances in Information and Communication Technologies, pp. 523–533. Springer (2017)
47. Perrey, R., Lycett, M.: Service-Oriented Architecture. In: Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium on. pp. 116–119. IEEE (2003)
48. Popov, S.: The Tangle - Version 1.4.2. URL: https://iota.org/IOTA_Whitepaper.pdf (2018), (Accessed April 22, 2018)
49. Porsche AG: Porsche Introduces Blockchain to Cars. URL: https://newsroom.porsche.com/en/themes/porsche-digital/porsche-blockchain-panamera-xain-technology-app-bitcoin-ethereum-data-smart-contracts-porsche-innovation-contest-14906.html (2018), (Accessed April 27, 2018)
50. Rosen, M., Lublinsky, B., Smith, K.T., Balcer, M.J.: Applied SOA: Service-Oriented Architecture and Design Strategies. John Wiley & Sons (2012)
51. SAE International: Automated Driving - Levels of Driving Automation as Defined in SAE International Standard J3016. URL: https://web.archive.org/web/20170903105244/https://www.sae.org/misc/pdfs/automated_driving.pdf (2014), (Accessed May 01, 2018)
52. SALT Technology, Ltd.: SALT - Blockchain-Backed Loans - Whitepaper. URL: https://membership.saltlending.com/files/abstract.pdf (2017), (Accessed April 25, 2018)
53. SelfKey Foundation: SelfKey - Whitepaper. URL: https://selfkey.org/whitepaper/ (2017), (Accessed April 27, 2018)
54. Sikorski, J.J., Haughton, J., Kraft, M.: Blockchain Technology in the Chemical Industry: Machine-to-Machine Electricity Market. Applied Energy 195, 234–246 (2017)
55. Specification, O.A.: OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2. Object Management Group (2007)
56. Statistics New Zealand: Car, bus, bike or train: What were the main means of travel to work (2006)
57. Sterling, L., Taveter, K.: The Art of Agent-Oriented Modeling. MIT Press (2009)
58. Swarm.City: Swarm.city - Token Exchange Whitepaper. URL: https://github.com/swarmcity/sc-token/blob/master/token-exchange-miniwhitepaper.md#token-exchange-whitepaper (2017), (Accessed May 04, 2018)
59. Thrun, S.: Toward robotic cars. Communications of the ACM 53(4), 99–106 (2010)
60. Uzcátegui, R.A., De Sucre, A.J., Acosta-Marum, G.: Wave: A tutorial. IEEE Communications magazine 47(5) (2009)
61. Vickrey, W.: Counterspeculation, auctions, and competitive sealed tenders. The Journal of finance 16(1), 8–37 (1961)
62. Wood, G.: Ethereum: A Secure Decrentralized Generalised Transaction Ledger. URL: http://gavwood.com/paper.pdf (2014), (Accessed May 01, 2018)
63. Zhu, W., Miao, J., Hu, J., Qing, L.: Vehicle detection in driving simulation using extreme learning machine. Neurocomputing 128, 160–165 (2014)