FEBRUARY 2018

FSBC Working Paper

How We Built the Future of Shared Mobility Within Four Weeks

Alexander Kirn, Alexander Gmelin, Philipp Sandner, Moritz Strube

A lot has been written about blockchain technology, quickly spreading from the financial industry to other sectors, especially mobility services. For shared mobility services it is seen as one of the enabling technologies of the future. At INVERS, we think that mobility should no longer be determined by ownership and we were, therefore, curious to find out how blockchain technology can be applied to shared mobility services in practice. With help from the Blockchain Center at the Frankfurt School of Finance & Management, we set out to build a prototype of a future shared mobility service which is purely based on blockchain technology.

As we envisioned of what the future of shared mobility might be like, we were inclined to take it one step further and not only use a new underlying technology like blockchain, but also highlight two other emerging trends we are seeing in the industry – scooter sharing and chatbots.

As we start digging deeper into the blockchain technology, let's first take a closer look at these two emerging trends and how they can have a strong impact on the way we perceive individual mobility in the near future.

Frankfurt School Blockchain Center
www.fs-blockchain.de
contact@fs-blockchain.de

Follow us
www.twitter.com/fsblockchain
www.facebook.de/fsblockchain

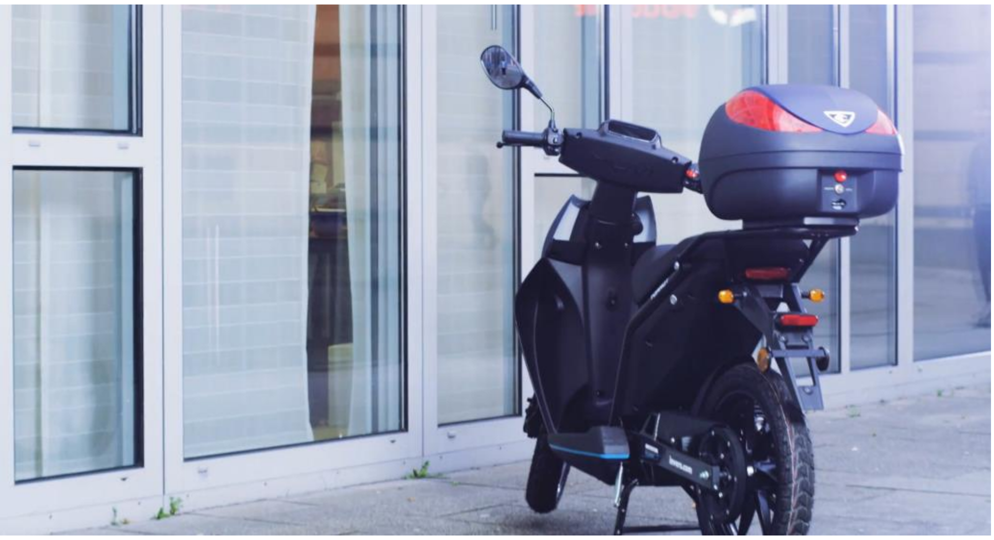
Frankfurt School of
Finance & Management gGmbH
Adickesallee 32-34
60322 Frankfurt am Main
Germany

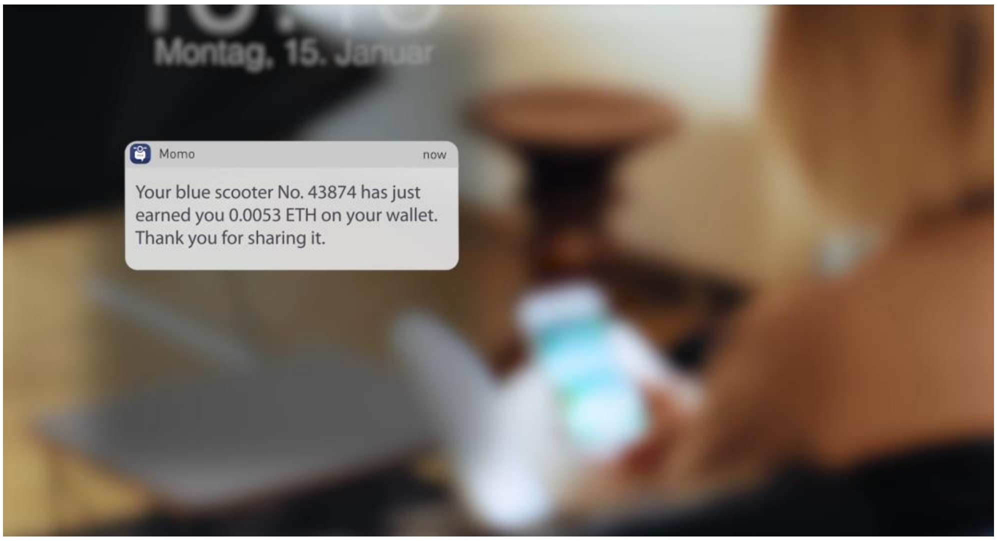
The emerging trend of scooter sharing

In the past few years, we have witnessed the strong growth of new mobility services in cities with ride-hailing like Uber, Didi or mytaxi, and carsharing like car2go, Maven and DriveNow. Just recently, an emerging new trend topped all growth statistics in Europe. Scooter sharing has been largely embraced by users in major cities all over Europe. Operators almost quadrupled their fleet sizes from 2,000 in 2016 to over 8,000 scooters in 2017[1]. It is expected that the scooter sharing market will soon outgrow the free floating carsharing market in terms of fleet size.

ILLUSTRATION

P2P mobility: Ethereum-based scooter sharing





Your blue scooter No. 43874 has just earned you 0.0053 ETH on your wallet. Thank you for sharing it.

Why is there such strong growth? With scooter sharing, users can be part of the electric revolution as more than 92% of shared scooters are electric today [2]. Finding a parking spot is usually quick and easy. In short, shared scooters are environmentally friendly, convenient and easy to use, create less congestion, are cheap and fun to ride, while at the same time they present an attractive business opportunity for operators.

Early on, we were convinced by this trend and supported the first scooter sharing operators in 2015 with our Shared Mobility OS technology stack to run their operations. Still today, we believe that this is just the beginning.

Consequently, we decided to put a scooter at the heart of our vision for the future of inner city mobility. From our experience with many scooter sharing operators (more than half of Europe's shared scooters run on INVERS shared mobility OS), we have identified key success factors in this market. We want to highlight a few of them to point out how blockchain technology can create a competitive advantage over existing approaches, with substantial benefits for operators as well as users.

Seamless user-experience

Registration. The user experience typically starts with a registration process for the respective mobility service. A lot of progress has been made in this area including an all-digital verification process as opposed to visiting the operator's office. However, it still takes a few minutes and a reliable data connection, which can be cumbersome for some users. Blockchain technology makes it possible to establish a secure digital identity online, which can include a driver's license and can then be used securely and instantaneously across multiple services such as other mobility offerings.

User interface. Sharing a vehicle must be easier than owning a vehicle to make way for mass adoption. Therefore, accessing a vehicle should be intuitive with only little action required. While most smartphone apps in our industry have improved significantly in terms of usability as well as their overall user experience in the last years, we decided to go beyond a dedicated user app – using a contextual chatbot instead. Chatbots not only provide a user interface which allows the user to just express his wishes and the technology takes care of the rest. Furthermore, they pose a radically new and promising approach to user interaction, are highly flexible and can be setup and adopted much quicker than a dedicated app. They can either be combined with a chat interface or with an invisible interface through speech recognition like Alexa, Siri, etc.

High availability of scooters

For users to flock to such a shared scooter mobility service and to meet their expectations and needs, high availability must be maintained. Therefore, with more users, the fleet must grow accordingly. Blockchain technology can facilitate growth of fleets and reduce costs for operators.

Asset-light growth of vehicle fleet. For operators, the greatest capital expense is its fleet of vehicles. Thus, capital constrained scooter sharing operators often grow slower, sometimes not meeting market demand. Blockchain-technology can reduce this bottleneck by allowing operators to easily add scooters owned by third parties to a fleet. They could be provided by small scooter rental operators or even private individuals as part of a Peer2Peer (P2P) sharing model. P2P sharing models are already in existence today but they are complicated. Many parties are involved and therefore require an intermediary for establishing trust and handling the complex billing and revenue splitting between those parties. With blockchain technology as a trusted and secure infrastructure, most of these functions can be performed directly, hence faster, without the need for a costly intermediary. Another inefficiency for today’s P2P models is insurance, as the renter has to purchase an insurance policy for the trip at a significant premium from the P2P operator – even though the vehicle and the renter might already have coverage. Blockchain technology can enable the user to bring his own insurance which can then also be securely verified. Of course, this requires insurance companies to come up with appropriate plans and to make them accessible via blockchain technology. Given the fact that blockchain is seen as a potential disruptor for the entire insurance industry and the current market dynamics with new business models like usage-based insurance policies as well as insurtech startups on the horizon, this is a likely development.

Reduction in payment processing fees. Tariffs for scooter sharing are usually priced in the same range as public transport. Due to the low amounts they are considered as micropayments. Based on the current payment infrastructure and the growing penetration of credit cards, these micropayments can be very costly for operators, not to forget the processing latencies of up to a few days. To reduce costs, operators can debit an accumulated number of micropayments per user by lengthening the billing interval while increasing the risk of a shortfall. This is only a viable option for frequent users.

Blockchain transactions are not only significantly cheaper regardless of the debited amount but also processed within seconds. With our approach, it is the vehicle that collects the appropriate amount before the user is granted access to it. This eliminates the risk of not getting paid. While credit card transaction fees on micropayments can be as high as 10% for a single ride, transaction cost for blockchain based payments will likely be close to zero in the future. For a business in which a 10% margin is already best in class, it is needless to say that this is a significant improvement.

Stable and always-on service

The highly distributed, decentralized architecture of blockchain technology eliminates single points of failure and contributes to an always-on mobility service. This leads to a shift in focus towards the devices connected to the blockchain and their data connection. Only with a proven and robust technology stack that scales to many scooters as well as users, one of the most important success criterions can be achieved: reliability! This is key for every shared mobility service. Users rely on mobility services in their everyday lives to go to work, run errands, visit friends and they will only adopt these services in the long term when their trust is not being disappointed. The service has to simply work.

The solid foundation for our vision: Shared Mobility OS

When thinking of a blockchain project that involves vehicles, whether it be for shared mobility services or anything else, the first question to ask is:

How do you get a live copy of a physical object in the digital space? We call this copy “the digital twin”.

To realize the digital twin, a robust, high-performance interface to the vehicle is necessary. With INVERS Shared Mobility OS, we have already built the right solution. More than 25 years of experience in this market, highly reliable hard- and software architecture, profound knowledge of vehicle data interfaces as well as the latest technology developments in the mobile world make Shared Mobility OS the leading platform for shared mobility services around the globe. So far, over 250 mobility services with more than 60.000 vehicles worldwide have been realized on Shared Mobility OS.

Shared Mobility OS is a suite of tools comprising of end-user apps, backend software, and telematics for operators to run a shared mobility service. The backend helps the operator manage its fleet and telematics with end-user support provided across multiple channels. The entire technology stack is built on an API centric architecture which facilitates adding or replacing of components. The comprehensive toolset and more importantly the API centricity of Shared Mobility OS enabled us to realize our outlook into the future of shared mobility.

How was it done?

In the following parts we will discuss in detail how we adapted Shared Mobility OS in 4 weeks to run an entirely blockchain-based scooter sharing rental process. We focused on the scenario that a user has seen a scooter in the street and is now in front of it, ready to use it immediately.

Building the app and chatbot, we developed a Decentralized App for scooter sharing. A Decentralized App (ÐApp) is a proprietary or public web app with a blockchain-based backend. Sharing a scooter with our ÐApp requires an existing Ethereum address for the user with the necessary funds (ETH) for paying the ride. This can be achieved by installing the status app (status.im) on the mobile phone of the user. Status is a chat-based mobile Ethereum wallet, currently available only in test release for iOS and Android, not yet in the app stores.

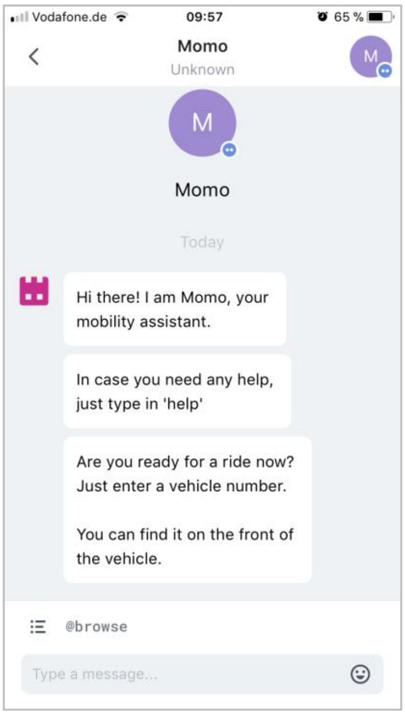
After installation, the ĐApp offers the following options:

- Booking a scooter to start a rental
- Returning a scooter and ending the rental

Booking a scooter unlocks it and withdraws the funds for the trip, based on the planned duration. After returning the scooter, the scooter is locked. With the used duration the actual price for the trip is calculated and any overpaid amount is being refunded. We chose such a prepaid model, as it ensures liquidity of the user. However, a postpaid model would also be a possible and user-friendly approach. The current version of the ĐApps does not include a search and selection of a scooter, as this was not in scope for the present project. This functionality can be easily added at a later stage. For now, we assume that the user simply approaches a scooter and enters its number to book it.

Figure 1

Beginning of the user dialogue



Hi there! I am Momo, your mobility assistant.

In case you need any help, just type in 'help'

Are you ready for a ride now? Just enter a vehicle number.

You can find it on the front of the vehicle.

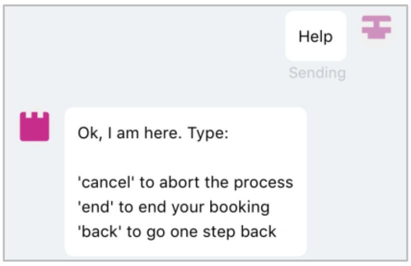
The amounts in ETH are only exemplary and do not represent the actual prices for scooter sharing. For current exchange rates see, e.g., coinmarketcap.com

Our chatbot interface is called Momo, and it is chat-based. Momo is an acronym and stands for: My Own Mobility Organizer. We gave the bot a little playful and human touch as it supports the user getting mobilized. The dialogue initially starts with a greeting and offering to help the user. The user interacts with Momo by entering appropriate commands at the bottom as plain message text, see the field "Type a message" (see Figure 1).

The user interface provides help for the user, as shown in the following screenshot (see Figure 2).

Figure 2

Available help for the user



Help

Sending

Ok, I am here. Type:

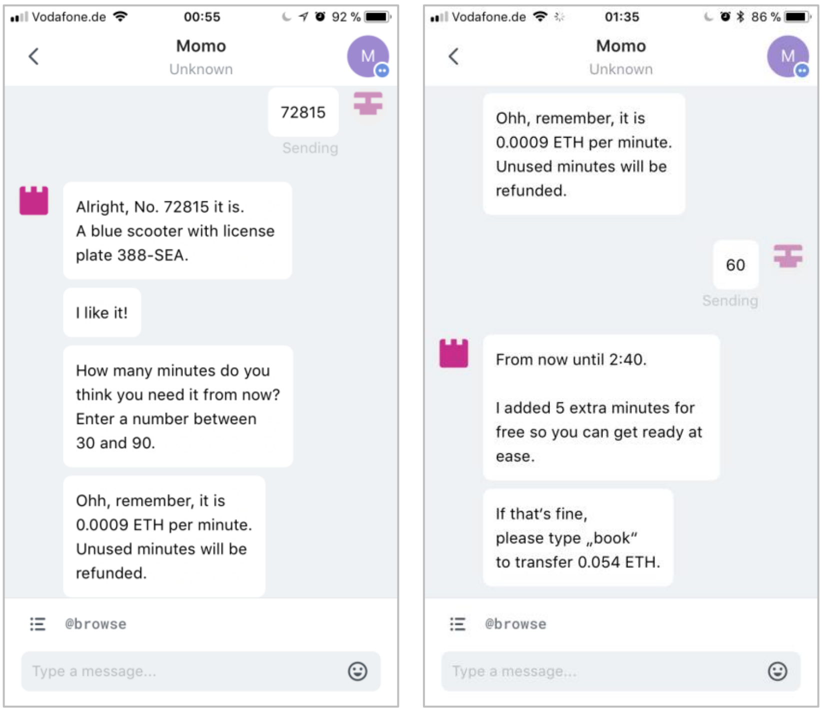
'cancel' to abort the process
'end' to end your booking
'back' to go one step back

The user can book a scooter by entering the vehicle number found in front of the scooter. The user is then asked for the duration in minutes he plans to ride the scooter. The price per minute is provided in Ether (see Figure 3, left panel).

After entering a valid duration in minutes, the cost of the trip is calculated based on a price per minute in Ether (ETH). The first five minutes are considered as preparation time for the user and are provided free of charge. The user is asked for confirmation by entering "book" (see Figure 3, right panel). "book" will trigger a transaction including upfront payment.

Figure 3

Dialog for booking a scooter



72815

Sending

Alright, No. 72815 it is. A blue scooter with license plate 388-SEA.

I like it!

How many minutes do you think you need it from now? Enter a number between 30 and 90.

Ohh, remember, it is 0.0009 ETH per minute. Unused minutes will be refunded.

Ohh, remember, it is 0.0009 ETH per minute. Unused minutes will be refunded.

60

Sending

From now until 2:40.

I added 5 extra minutes for free so you can get ready at ease.

If that's fine, please type „book" to transfer 0.054 ETH.

As mentioned above, a further extension of the prototype could also include exact payment based on the actual duration of the trip in minutes or even seconds, without necessity of refunding unused minutes. This can be enabled by state channel systems like Perun (www.perun.network).

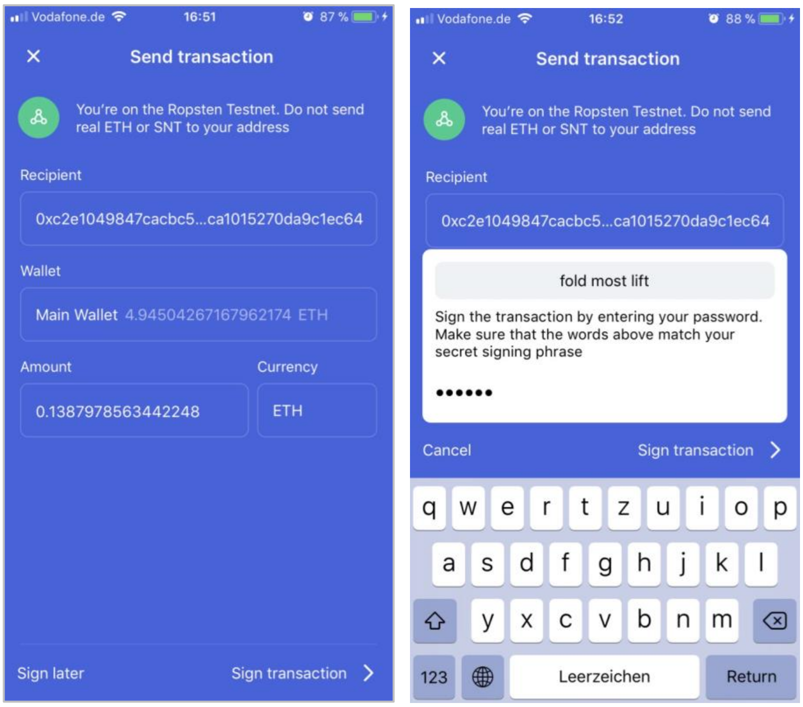
Based on the planned duration of the trip, the appropriate amount of Ethers including gas is calculated. Gas has nothing to do with the fuel of the scooter, since it is electric. Instead, in this context gas is referred to as the internal price for running a transaction or contract in Etherum. It is the fee paid to miners.

To start the trip, the total amount must be paid, i.e. transferred from the wallet of the user to the scooter, which has its own wallet. The price for gas is deducted by Status to make sure that the transaction is mined in time.

To confirm payment, the user has to virtually sign the transaction. This can be done within the Status app (see Figure 4, left panel). The field „recipient" contains the address of the wallet of the scooter to which the funds are transferred. With Ethereum, an address starts with 0x… and is sometimes also referred to as the wallets public key. The field „wallet“ shows the available funds, „amount“ and „currency“ show the amount which is about to be transferred.

Figure 4

Signing the payment transaction



Send transaction

You're on the Ropsten Testnet. Do not send real ETH or SNT to your address

Recipient: 0xc2e1049847cacbc5...ca1015270da9c1ec64

Wallet: Main Wallet 4.94504267167962174 ETH

Amount: 0.1387978563442248

Currency: ETH

Send transaction

You're on the Ropsten Testnet. Do not send real ETH or SNT to your address

Recipient: 0xc2e1049847cacbc5...ca1015270da9c1ec64

fold most lift

Sign the transaction by entering your password. Make sure that the words above match your secret signing phrase

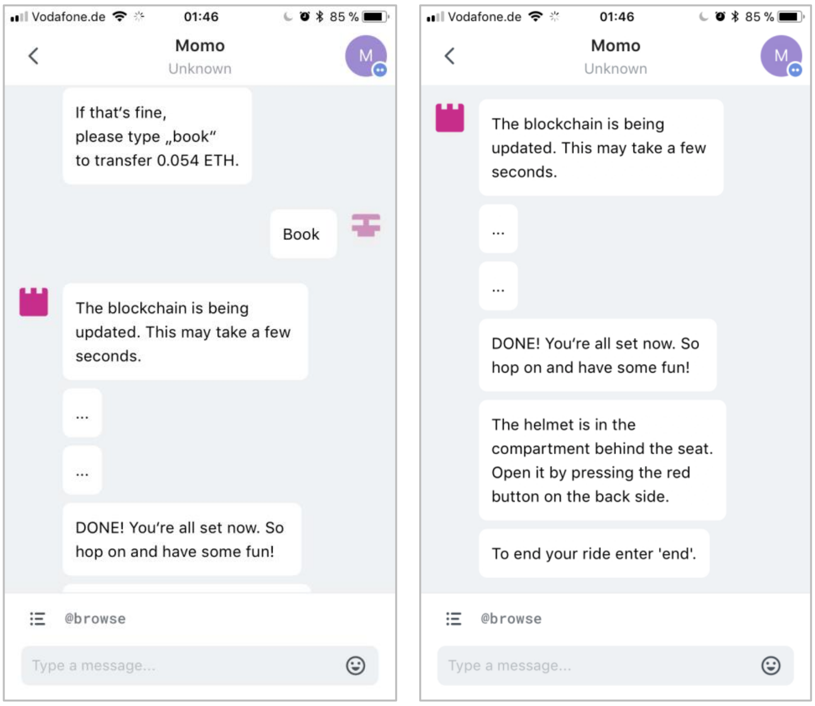
••••••

Signing the transaction requires the user to enter the appropriate password according to his secret key (see Figure 4, right panel). The key was generated by the user as part of the initial setup process of the Status app. It is done by entering a secret signing phrase, consisting of twelve dictionary words. This secret signing phrase enables the creation of a key pair, which provides a public key for generating an address and a secret key for signing transactions.

After successfully signing the transaction, the user has to wait for the confirmation of the successful mining of the transaction (see Figure 5, left panel). On average, every 15 seconds a block is mined on the Ethereum Blockchain. The actual duration depends on the amount the user pays for gas. A higher willingness to pay for gas leads to faster execution of the transaction. An established status channel will display the waiting time for mining.

Figure 5

Dialog for booking a scooter (continued)



If that's fine,
please type „book“
to transfer 0.054 ETH.

Book

The blockchain is being updated. This may take a few seconds.

...

...

DONE! You're all set now. So hop on and have some fun!

The blockchain is being updated. This may take a few seconds.

...

...

DONE! You're all set now. So hop on and have some fun!

The helmet is in the compartment behind the seat. Open it by pressing the red button on the back side.

To end your ride enter 'end'.

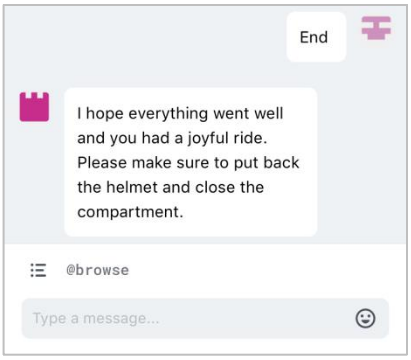
Once the transaction is mined the scooter will be unlocked and the user can start the trip. The user gets some hints about using the scooter and is informed about how to end a trip (see Figure 5, right panel).

Entering “end” ends the trip, see Figure 6. This is confirmed by the user interface. Depending on the duration of the trip, possible surplus payments are refunded.

This completes the walk-through from the user perspective. In the following section, we will describe the ĐApp from an implementation point of view.

Figure 6

Dialog for returning a scooter



I hope everything went well and you had a joyful ride. Please make sure to put back the helmet and close the compartment.

How did we build it?

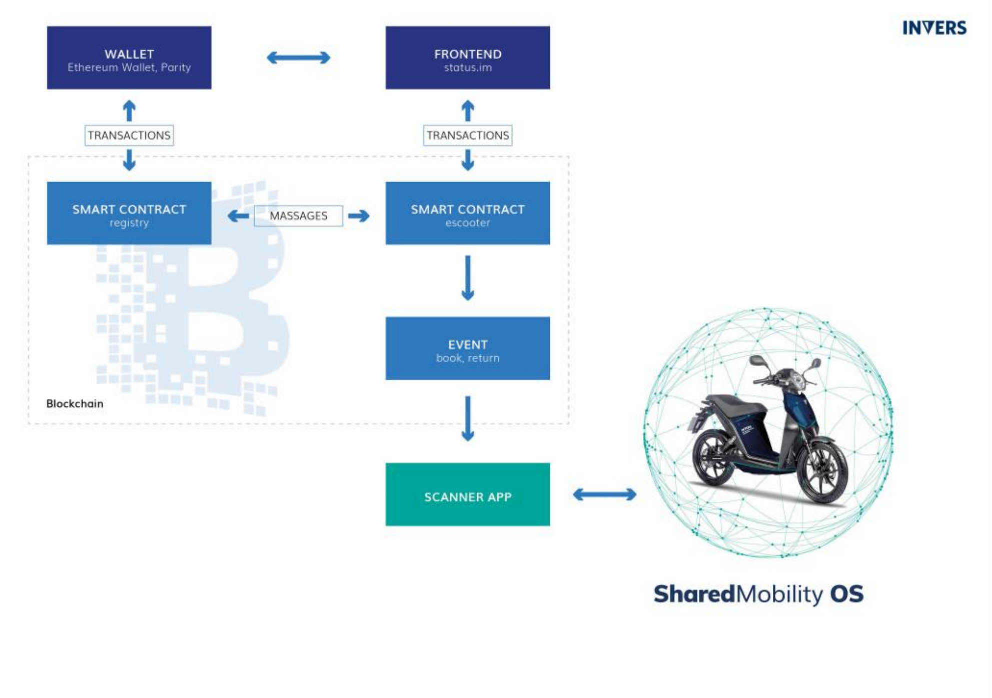
Figure 7 shows the architecture of the the ĐApp. The project consists of the following parts, which will be discussed in detail in the following subsections.

- a frontend for users to booking and returning scooters
- a wallet for administrators to manage accounts, funds and smart contracts
- smart contracts for booking and returning scooters
- transactions enabling the interaction between users and smart contracts
- messages enabling the interaction between smart contracts
- events (log entries) triggered by transactions
- an app watching for events and sending requests to the scooter via Cloudboxx API as part of Shared Mobility OS
- a scooter equipped with a Cloudboxx and integrated in Shared Mobility OS

All these components are described more in detail in the following subsections.

Figure 7

Architecture of the ĐApp



Frontend

We used Status to provide a mobile ĐApp interface. The interface is described in detail in section “How does it look like?”. Status is the gateway to the Ethereum ecosystem for Android and iOS. Status combines a messenger and a browser. It acts as a platform for everyday smartphone users to begin using ĐApps, and a simple way for ĐApp developers to reach new users. In 2017, we are still only at the dawn of the Ethereum ecosystem. But given time, Status will evolve into a seamless way to interface with a full suite of decentralized applications and services.

Status aims to provide its users a platform to socially interact with smart contracts, that’s why it’s based on messaging. Globally, messaging applications now boast even more active users than social networks. With Status you can chat with your friends or different ĐApps.

The Status app consists of a wallet, a messaging interface and a ĐApp interface. With the wallet application users can manage their balance, send Ether and request Ether from their contacts. The messaging interface is used to communicate with other Status users or group chats, but also to interact with ĐApps. These can be either web-based applications or chat bots. Last but not least the so called 'Discover' interface can be used to explore new ĐApps, find other Status users or group chats, where you can join a community. A local P2P exchange will be introduced with the beta release of Status, where users can find other users nearby to exchange fiat to ETH or other tokens and vice-versa.

Wallet

A wallet is a gateway to decentralized applications on the Ethereum blockchain. It allows a user to hold and secure Ether, as well as write, deploy and use smart contracts. Popular wallets are the Ethereum Wallet (www.ethereum.org) or Parity (www.ethereum.org). We used Parity for deploying the smart contracts in the project to

- create accounts (addresses),
- edit and compile smart contracts,
- deploy smart contracts to the blockchain,
- send transactions, and
- inspect events.

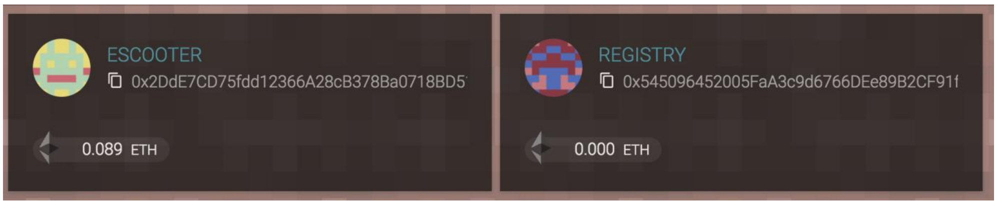
The following Parity screenshots are from an administrator or developer perspective. The user only uses the Status app which also provides a wallet functionality.

Smart contracts

Smart contracts are programs written in the Solidity programming language[3]. There are currently two smart contracts, `Registry.sol` to register users and `EScooter.sol` representing one escooter (see Figures 8-10):

Figure 8

Smart contracts (Screenshot from Parity)



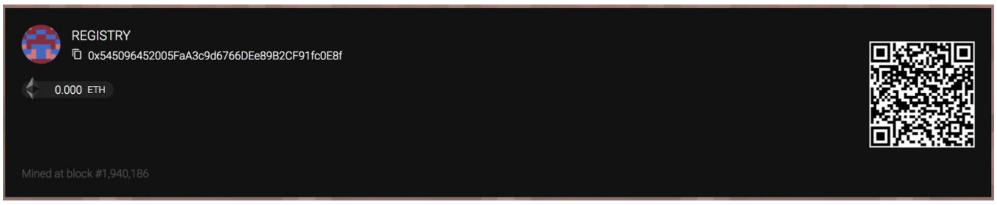
ESCOOTER
0x2DdE7CD75fdd12366A28cB378Ba0718BD5
0.089 ETH

REGISTRY
0x545096452005FaA3c9d6766DEe89B2CF91f
0.000 ETH

Registry.sol

Figure 9

Registry contract (Screenshot from Parity)



REGISTRY
0x545096452005FaA3c9d6766DEe89B2CF91fc0E8f
0.000 ETH
Mined at block #1,940,186

Each user has to register to rent a scooter to make sure that only authorized people have access to the scooters. The smart contract `Registry.sol` makes this registering possible. For this it provides functions to register a user, remove a user from the register and check if a user is registered.

EScooter.sol

Figure 10

EScooter contract (Screenshot from Parity)

ESCOOTER
0x2DdE7CD75fdd12366A28cB378Ba0718BD510eAfa
0.089 ETH
Mined at block #1,940,237

Each scooter is represented by an own smart contract deployed to an own address. The interface of the contract enables booking and returning of the scooter by a user. Only the owner of the contract can assign a registry to a scooter. Only a user registered in the assigned registry can book a scooter, and this only if the scooter is not already booked and enough funds are transferred with the booking.

Transactions

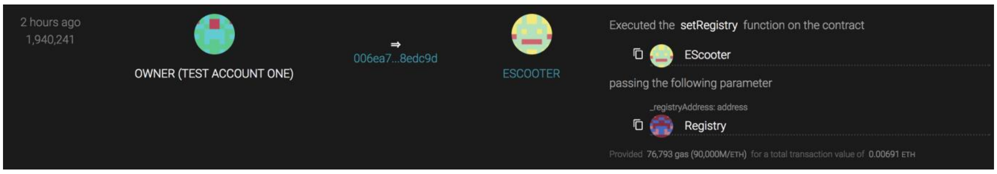
Transactions are signed data packages that store a message to be sent from an externally owned account (representing a human user, external actor) to another account on the blockchain (human user or smart contract).

Owner transactions

The contract owner (administrator) triggers transactions with a client like Parity. These transactions are calls to set the registry and register authorized users (see Figure 11, 12).

Figure 11

Set registry transaction (Screenshot from Parity)



2 hours ago
1,940,241

OWNER (TEST ACCOUNT ONE)

⇒
006ea7...8edc9d

ESCOOTER

Executed the **setRegistry** function on the contract

EScooter

passing the following parameter

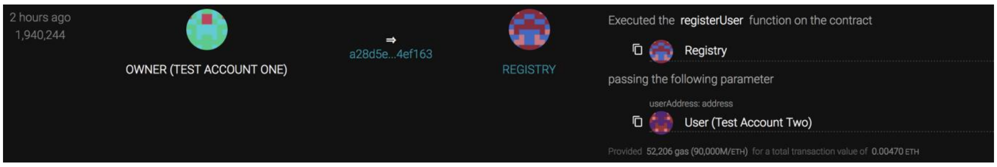
_registryAddress: address

Registry

Provided 76,793 gas (90,000M/ETH) for a total transaction value of 0.00691 ETH

Figure 12

Register user transaction (Screenshot from Parity)



2 hours ago
1,940,244

OWNER (TEST ACCOUNT ONE)

⇒
a28d5e...4ef163

REGISTRY

Executed the **registerUser** function on the contract

Registry

passing the following parameter

userAddress: address

User (Test Account Two)

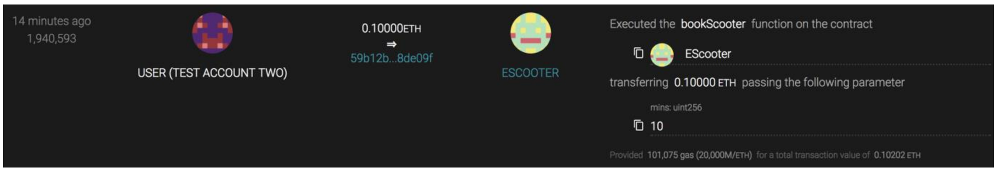
Provided 52,206 gas (90,000M/ETH) for a total transaction value of 0.00470 ETH

User transactions

The user triggers transactions with the user interface. These transactions are calls to book and return a scooter (see Figure 13, 14).

Figure 13

Booking transaction (Screenshot from Parity)



14 minutes ago
1,940,593

USER (TEST ACCOUNT TWO)

0.10000ETH
⇒
59b12b...8de09f

ESCOOTER

Executed the bookScooter function on the contract

EScooter

transferring 0.10000 ETH passing the following parameter

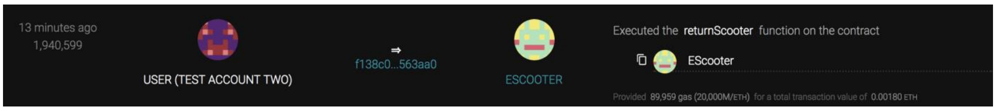
mins: uint256

10

Provided 101,075 gas (20,000M/ETH) for a total transaction value of 0.10202 ETH

Figure 14

Return transaction (Screenshot from Parity)



13 minutes ago
1,940,599

USER (TEST ACCOUNT TWO)

⇒
f138c0...563aa0

ESCOOTER

Executed the returnScooter function on the contract

EScooter

Provided 89,959 gas (20,000M/ETH) for a total transaction value of 0.00180 ETH

Messages

Smart contracts are able to send “messages” to other contracts. They can be conceived of as function calls. Essentially, a message is like a transaction, except it is produced by a contract and not an external actor. The scooter contract sends a message call to the registry contract to check if a user requesting to book a scooter is registered. The registry returns true, if the user is registered, false otherwise. Only if the user is registered booking is possible.

Events

If a user sends a transaction to book or return a scooter corresponding events are generated. Events are log entries generated by message calls (transactions) in the blockchain. Events are written as logs to the blockchain and are accessible to external services, but not to smart contracts.

Scanner app

An external service called the scanner app is watching for booking (“Booking”) or trip ending (“Return”) events. This app is written in node.js and uses the web3.js JavaScript API to access the blockchain. Each scooter is represented by mapping a QNR to the address of the smart contract belonging to the scooter. The QNR is an internal unique identifier for a vehicle interface, in this case the Cloudboxx API, in Shared Mobility OS.

The application is watching for events containing the topics “Booking” and “Return”. To define event filters, search terms have to be hashed using the sha3 function provided by web3. In case of an event the Cloudboxx API is called with appropriate HTTP calls.

Cloudboxx API

The scanner app provides an endpoint for unlocking and locking the scooter. A scooter is identified uniquely by a QNR. When the user finishes the trip, he uses the ĐApp to return the scooter and end the rental. This calls a function of the scooter contract and creates an event which triggers a lock request to INVERS Shared Mobility OS and eventually the Cloudboxx in the scooter.

Security considerations

The contracts are deployed to a public Ethereum blockchain testnet called Ropsten offering access to all smart contracts, transactions and events stored in this public blockchain. While there is no risk of theft of funds – which are provided for free in the testnet – it has to be prevented that unauthorized users obtain access to scooters and steal them.

General security provided by the blockchain protocol:

- smart contracts, owner and users are identified by addresses
- no actions by adversaries are possible using these addresses without knowing the associated private key, which has to be kept secret

- all events contain secure information about the source of the event to prevent generation of fake events by adversaries to get access to a scooter

Therefore, the following security measures are implemented:

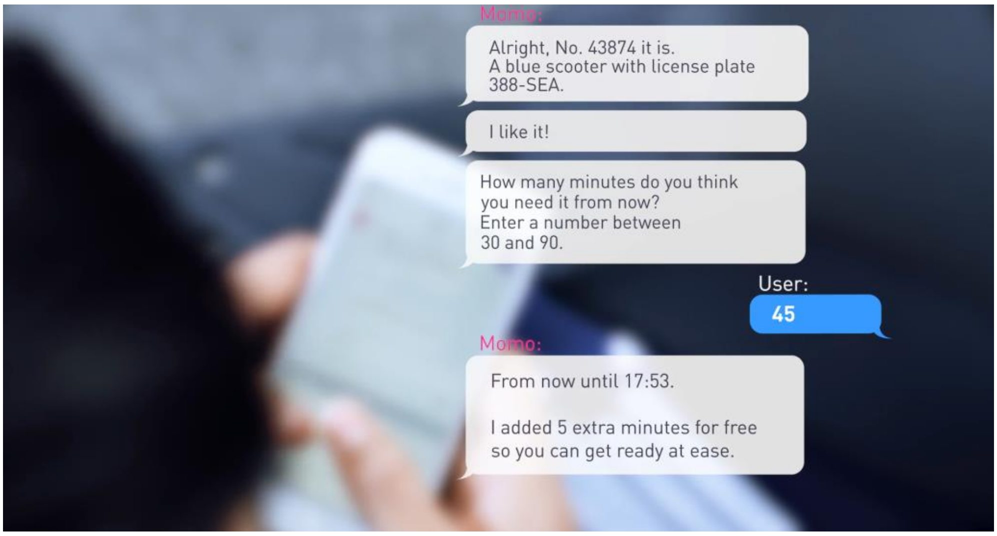
- access to sensible smart contract functions is restricted to the contract owner
- no data to access the INVERS Cloudboxx is stored in the blockchain
- addresses for authorized users can be registered
- only transactions for registered addresses are accepted by the smart contract for renting a scooter
- a whitelist containing the QNRs restricts access to scooters
- the scanner app only accepts events generated by a smart contract belonging to a scooter

While there is no risk of theft of funds in the current setting of the project deployment in the Ropsten testnet, the deployment of the project to the production blockchain requires further security measures.

Conclusion

By now, it is indisputable that shared mobility services are here to stay. Some segments in shared mobility – like scooter sharing – are growing exponentially. And yet, the absolute size of the shared mobility market is still negligible compared to the ownership mobility market. Shared mobility technology has contributed to making these services more accessible, attractive and economical for both users and operators. At INVERS, we believe that we are still just at the beginning.

New emerging technologies like blockchain and chatbots will further accelerate growth and provide new opportunities for new and existing shared mobility operators. We believe that blockchain technology will drive growth for P2P sharing models as it makes the transactions between many parties more secure and efficient. By building the prototype, we learned that blockchain technology is still in its infancy. For instance, the processing of transactions requires in average 15 seconds on the Ethereum network (much faster than the 10 minutes it would need with Bitcoin for instance). To make use of its full potential, a larger ecosystem of blockchain based services will be necessary – such as a blockchain-based driver's licence or blockchain-based insurance plans.



Momo:

Alright, No. 43874 it is.
A blue scooter with license plate 388-SEA.

I like it!

How many minutes do you think you need it from now?
Enter a number between 30 and 90.

User:

45

Momo:

From now until 17:53.

I added 5 extra minutes for free so you can get ready at ease.

To take advantage of these opportunities, shared mobility operators need a reliable and flexible technology platform which can adapt to evolving technologies, like Shared Mobility OS.

Today, Shared Mobility OS enables already a broad variety of mobility services such as Carsharing, Ride Pooling, Automated Car Rental, Scooter Sharing etc. as its technological foundation. With the present project, we demonstrated that it can also enable a future blockchain-based service and this can be done within only four weeks.

Alexander Kirn is CEO of INVERS GmbH. You can contact him via LinkedIn (https://www.linkedin.com/in/alexanderkirn/) or via alexander.kirn@invers.com; **Alexander Gmelin** is Entrepreneur in Residence at INVERS and leading the blockchain projects there. You can contact him via his LinkedIn account (https://www.linkedin.com/in/agmelin) or by email (alexander.gmelin@invers.com). **Prof. Dr. Philipp Sandner** is head of the Frankfurt School Blockchain Center. You can contact him via email (email@philipp-sandner.de), alternatively via LinkedIn (https://www.linkedin.com/in/philippsandner/) or follow him with his Twitter account (@philippsandner). **Moritz Strube** is working at ChainLabs providing IT development services and consulting in the area of blockchain with focus on Ethereum and IOTA. You can contact him via mail (moritz.strube@chainlabs.digital) or follow him on Twitter (@mostwi).

INVERS is pioneer and market leader for shared mobility technologies. It develops a full technology stack for shared mobility operators called INVERS Shared Mobility OS comprising user apps, a backend for operators and telematics. INVERS' customers operate carsharing, automated rental, scooter-sharing or ride-pooling services. They operate more than 60.000 vehicles with over 1 million active users in more than 30 countries worldwide. Founded in 1993, INVERS has grown to more than 90 employees with its headquarter in Siegen, Germany and offices in Cologne and Vancouver.

[1] https://www.innoz.de/sites/default/files/howebock_global_scootersharing_market_report_2017.pdf

[2] https://www.innoz.de/sites/default/files/howebock_global_scootersharing_market_report_2017.pdf

[3] https://solidity.readthedocs.io/en/develop/