BIOTS Project - weShare

Roman Gruber[a], Philip Jost[b], Tobias Buner[c], Noé Heim[d], Jonas Wyss[e], and Irfan Bunjaku[f]

[a]rgruber@student.ethz.ch, student BSc. Physics
[b]phjost@student.ethz.ch, student BSc. Computer Science
[c]bunert@student.ethz.ch, student BSc. Computer Science
[d]noheim@student.ethz.ch, student BSc. Computer Science
[e]wyssjon@student.ethz.ch, student BSc. Physics
[f]ibunjaku@student.ethz.ch, student BSc. Computer Science

ETH Zürich, 30.04.2018

All authors contributed equally to this report.

Abstract. The goal of the weShare project is to enable individuals to share empty seats when driving and receiving a benefit in doing so. On the other hand the other individuals can get a ride by a driver. In doing so the overall occupancy rate[1] of cars should increase and thus the number of cars on the road and the amount of greenhouse gases emitted by personal motorized transportation decreased.
By building the project on the blockchain the transparency of the system should be greater and thus the trust in the system increased. The basic mechanism of the system is a state machine on the blockchain inside a smart contract which directs the interaction between the passenger and driver.

[1]The occupancy rate is the number of passengers divided by the number of available seats.

Contents

1 Introduction

The average expenses for owning a car in Switzerland are around CHF 10'500 per annum or around CHF 0.7 per km, assuming an annual driving distance of 15'000 km. At an average available household income of slightly below CHF 84'000 p.a., owning a car would cost 12.5% of the available income according to the TC Switzerland [1]. By sharing rides, the total need for cars would decrease and thus some individuals might relinquish their car. The area covered by roads in Switzerland is around 88% of the total area used for mobility infrastructure. Since 1970 the number of cars in Switzerland has almost doubled from 2 to 4 million (2015) at an average occupancy of 1.56 persons per car (2015, [2]). By increasing the average occupancy of cars, the number of cars on the road would decrease and thus reduce pollution. Personal transportation in cars is responsible for 67% of the total CO_2 pollution caused by transportation in Switzerland (2015, [2]). Furthermore, the strain on roads could be reduced and thus money saved which could be invested otherwise. To quote Alex Stephany in his book "The Business of Sharing"; "The Sharing Economy is the value in underutilized assets and making them accessible online to a community, leading to a reduced need for ownership of those assets." [3]
Two different approaches to directly reduce the usage of cars could be used; car sharing and ride sharing. Car sharing is where entire vehicles belong to groups or companies and individuals can rent the cars. Ride sharing is when the cars are in the possession of the individuals. The good that is shared are the rides; drivers can pick up passengers and deliver them to the desired destination. The weShare idea is based on ride sharing on the blockchain rather than car sharing. The general goal is to increase the convenience for individuals. For example if someone travels abroad, but doesn't want to rent a car since the individual only needs a taxi service, the individual could in theory look for a driver in the host country via weShare. A general problem of taking a ride with a stranger is trust; many were told as children not to enter the cars of strangers for fear of abduction or other threats. This is something most adults still don't feel quite comfortable with, since - ignoring the previously mentioned - one still has to trust the driving skill of a stranger. This is a problem we aim to tackle with the help of the blockchain which introduces a new level of transparency and safety of information to the users of the sharing system.

2 The problem to be solved

The Paris Agreement aims to fight the climate change and keep the temperature-increase below 2°C and even to reduce the increase to 1.5 °C [4]. To do so it is inevitable to reduce the CO_2 emissions caused by mankind. Transportation is with 22% the second largest cause of CO_2 pollution [5]. Since a certain amount of transportation is unavoidable or one doesn't want to relinquish the comfort of transport, it seems to be the most satisfying solution to find an efficient way to maintain the flexibility that one desires, while using the given resources in a way that lowers the CO_2 emissions and in an ideal scenario would also be positively incentivized - for example financially.
The average weight of a car is about 1850 kg [6], while the average European adult weighs about 70 kg [7], so it is easy to see that the marginal cost in terms of weight of an extra person in a car is very low. Hence we strive to connect

people who are heading to - more or less - the same destination, so that the occupancy rate is increased. In our opinion a ride sharing approach is a first step in the right direction.
The drivers should not do this for a living, but will - with our scheme - share their own transportation costs with other passengers by sharing the seats in their own cars and thus will decrease the greenhouse gas emission per person per kilometer traveled by car.
Another aspect of the ride sharing idea is, that it may reduce the amount of traffic since there will be less cars on the road, reducing traffic congestions and thus decreasing the cost of the latter for the economy. (In Switzerland the traffic congestion costs are according to the ARE CHF 1.25 billion. [8])
It is however very difficult to quantitatively analyze if this will lead to less traffic, since it could make private transportation cheaper and hence motivate people to use private transport more instead of public transport which could reduce the benefit of the system.

3 A solution

3.1 A brief summary

In matching drivers and passengers many problems arise. Some of which are discussed in the following. A possible ranking system is discussed along with trust and possible financial benefits for the participants. A token system is also discussed. Finally a description of the current code and its working is given.

3.2 Finding people who need a ride - discovery

The discovery of both drivers and passengers should feel intuitive and simple. A mechanism determining acceptable detours for the driver must be implemented to assure a pleasant experience. It should also be possible for the user to decide how big of a detour is acceptable. The length of the route could be calculated based on third-party tools such as Google Maps.
If multiple drivers or passengers are available the driver/passenger should be able to choose the one who fits him/her best. A reputation system could support the driver/passenger in his/her decision. This is discussed further in the following.
Fully decentralizing the driver/passenger discovery proves difficult because of the unicast nature of the Internet. By using something in the likes of how IPFS and BitTorrent use bootstrapping and distributed hash tables to discover peers [9] it would be possible to omit introducing a third party taking a commission to cover server costs. While there would still be a need for a central starting point for the bootstrapping, the cost can be lowered in this way.

3.2.1 Trust in strangers and transactions

It is crucial to the ride sharing system that the payment method is accepted and seen as secure [10]. Otherwise people will not use the system as they deem it unsafe and will turn to other systems which fit their needs. As long as trade has been in existence the oracle for the transactions were the people. If person A wants to trade a good 'a' and is in need of 'b' which is owned by person B he needs to find the other person and initiate the trade. Assuming B wants

‘a’ both parties have to make sure that the respective good is of appropriate quality and that the other person delivers their good. A possibly hierarchical organization of a society ensures that both parties fulfill their deed adequately. As the peer-to-peer sharing with small amounts is non-hierarchical - the law still applies -, it is however very impractical to involve it, since the transaction volume is mostly small. This makes it prone to scams. It is thus necessary to ensure protection of the participants from the latter. A possible way is to include a reliable reputation system, which allows the user to get information about the other participants e.g. potential trade partners.
It is of great importance that the passengers and drivers are well behaved and the car sharing is a pleasant experience for both parties. It thus has to be ensured that there is some mechanism for both parties to get information about their peer in advance to starting the ride [11].

3.2.2 Reputation systems

A possible reputation model to start with could be a point based system, in which participants are awarded points for “good” deeds and detracted points for bad behavior. Upon reaching a threshold of points additional rights are unlocked. The Stack Exchange network is already using such a system with great success, discussions of this system can be found on `https://meta.stackexchange.com/`. Upon joining the network the user would have a neutral rating.
In a Stack Exchange like system drivers could be rewarded with additional reputation by increasing the occupancy rate of their cars. This could be implemented as a conditional multiplier; as the cars have only a limited amount of space, if the driver tries to over-occupy his car, the passengers can report this and if they all agree the driver receives a reputation penalty. Environmentally friendly driving could also be incentivized with a multiplier, however this leads to the problem of how to oraclize this. Generally good driving could be rewarded over a rating the passenger gives the driver after the ride.
It should also be possible for the driver to rate the passenger as to prevent bad behavior of passengers. Possible implementations are a reputation penalty for common misbehavior such as rudeness. Common misbehaviors could be listed and scale of the gravity for each added with room for users to decide themselves. The reputation system itself is also prone to abuse, which can be very difficult to prevent, for more detailed description of possible attacks on the reputation system and how to prevent them see Chen et al. [11]. A possible solution is the automatic comparison to prior behavior and the comparison to the other passengers if there are any as stated in [11].
When the system is already established to a certain degree, it would be possible that before being able to be the driver, the user has to earn a certain amount of reputation as a passenger. However this has a certain trade-off that a consequence of setting a hurdle for being a driver might lead to a lack of the latter thus making the system less interesting for users and potentially leading to the starvation of the system.

3.2.3 Impact of a reputation system on the pricing mechanism

To increase the importance of the reputation system an impact on the pricing mechanism of the rides could be included in such a way that desired behavior is rewarded with higher prices for good drivers and lower prices for well behaved passengers. This could also be done in the form of a multiplier. Boundaries for the multipliers could be added such that if the cost of the ride is high not the whole multiplier is applied but only a fraction of it. Furthermore bad behavior could be punished by the same mechanism as for good behavior but inversely applied. An impact of reputation could however lead to incentives to attack and exploit the reputation system.

3.3 Blockchain and Smart Contracts

To avoid third parties and centralized services the blockchain was used. The idea is based on a smart contract for every user. Two people connect over a platform peer-to-peer and if they find a match, meaning one person gives the other a ride, the system interacts with the blockchain - more precisely with the smart contract. From then on the software changes the own state and the state of the partner into other states after specific interactions.[2]

3.3.1 State machine

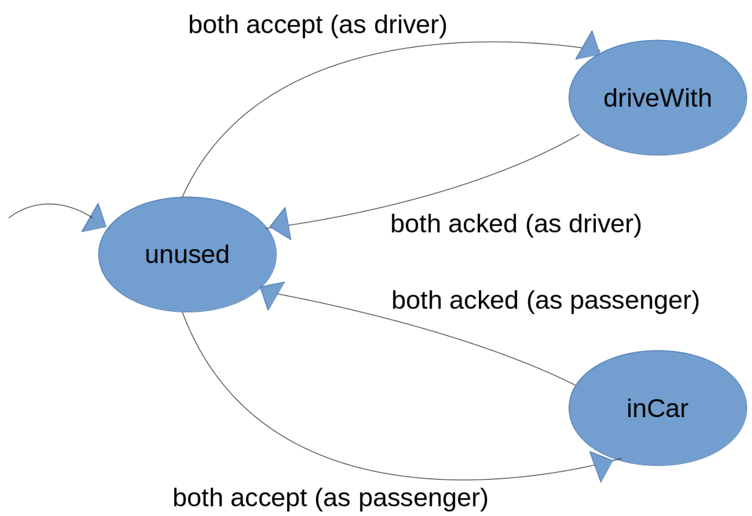

Figure 1: A schematic of the state machine. The state machine depicted is the current version how it is implemented in the code. When entering the application and selecting the desired destination the `unused` state is entered. After a successful discovery and pairing and if both parties accept, the `driveWith` and `inCar` state is entered respectively. If the ride was successful and both acknowledge, the payment process is initiated and the parties reenter the `unused` state.

After the work from the BIOTS week, the project is still in its infancy. Therefore most things are simplified and thus should be extended further in a full implementation. In the current version of the code three states are used (see figure 1):

- `0` for `unused`
- `1` for `driveWith`

[2] The code can be found in the repository `https://github.com/ETHBiots2018/weShare`.

- 2 for `inCar`

To get started it was assumed that there is only the driver and one passenger. Later there could be multiple passengers. The contract gets an address called `PartnerAddress` which points to the address of the other contract. This is part of the initiation or also called discovery, where they communicate over a front-end - at the moment the web front-end (see the GitHub repository), later possibly on a mobile app - and the driver can see which route the passenger wants to travel. If the driver wants to take the passenger with him, he calls the `dAccept` function with the shared address of the passenger which was exchanged before in the discovery process. The passenger can call `pAccept` the same way no matter who called the function first. However after the second has called the accept function a state transmission for both is triggered; the driver gets into the state `driveWith` and the passenger into the state `inCar`. In reality the passenger isn't already in the car, he needs to be picked up first, but to simplify we reduced the number of states which could be used to extend those principles. As soon as they arrive at their destination, they can tell the app that they arrived, which triggers the `dAck` or `pAck` function for the smart contract. Similarly to the previous function calling it doesn't matter who called it first. Just at the moment that the second called the function, they both have agreed to to have successfully arrived at the target location which then triggers the token transfer (see figure 2).

3.3.2 Summary of the implementation

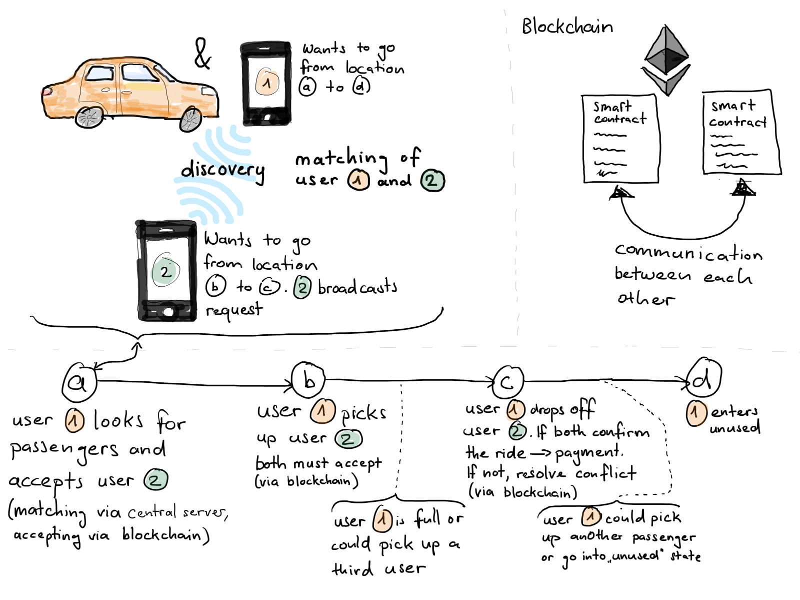

Figure 2: Sketch of the code with some additional steps which are not yet implemented.

Our software prototype is built on top of the ValidityLabs EthDapp-Boilerplate and includes a web front-end and smart contracts to deploy to the blockchain. These two interact with each other in a classical UI-to-API style. The driver and passenger each have an instance of the web front-end and a smart contract assigned to them. Button presses in each UI can trigger state transitions in the smart contract. Once the discovery process is finished and a driver and passenger have agreed to share a ride, their smart contract can also trigger state updates in their respective driver/passenger counterpart smart contract. As of the current development status we are using Metamask [12] to pay for the invocations of smart contract functions that web3 [13] calls. Metamask and web3 both connect to a local network Ethereum client set up by testrpc, where our smart contracts were deployed to using truffle[3]. Once in a release-ready state the software solution will interact with the real Ethereum Blockchain from a mobile application, but Ethereum Blockchain enabled mobile client solutions are still under heavy development as of right now.

3.3.3 Possible improvements and additions

Due to the team having only limited time and no prior knowledge of the Ethereum stack the current features of the software are minimal: As of now the software solution only covers scenarios where everything happens exactly as expected. The driver can only accept a single passenger and there is no mechanism to cancel a ride yet. Regulations on what happens with the money when a user cancels mid-ride are also to be implemented.
By gathering sensor data from the car and the smartphones of the driver and passengers the mobile application can get a pretty good understanding of the ride. Additionally this would allow the application to gather information about the driving behavior. This has however privacy draw backs; an optional implementation - the user can choose him/herself - is thus necessary.
It's hard for the driver to accept additional passengers once the initial ride has begun due to his limited ability to operate his smartphone while driving. We gave thoughts to solving this by voice control or maybe having the passengers which are already in the car help out the driver. The passengers could scout for possible additional passengers and provide all the information to the driver orally. Once they agree on picking up someone the passengers could add people to the ride and the driver would only have to give his final approval by one button press. Computer algorithms would already filter users who have suitable routes to share a part of the ride. Assisting the driver as a passenger could then be rewarded with the reputation system or in the form of currency.

3.4 Tokens and their value

For the token two main systems were considered:

- tokens which operate as reputation combined with transfers of actual money to cover for gasoline and other costs of the driver
- two independent tokens, which create an isolated market (the tokens replace the real world currency)

Both of the token systems considered have their advantages and drawbacks. There is a plethora of other systems one can come up with, but only the two mentioned above will be discussed further as they seem to a well suited first and maybe second step for a ride sharing platform on the blockchain.

[3]Ganache [14] was not available back when the hackathon was held

3.4.1 Token economy

The first system - the combination of reputation tokens and money - works similar to a free market; the better the product (the driving) is, the more customers one gets. The idea was to implement it via a rating/reputation-system. If a person has a high rating they get shown as the very top to the people looking for a ride their area. This gives them the opportunity to choose who they want to take along. At the end of a ride the driver would get instantly paid with real-world money to cover expenses. The token would be created in form of a reputation-system token, which gives the benefit of being shown atop the list of available drivers. In combination with the money transfer it becomes an incentive to be a "good" driver.
The second system - the two independent tokens - is more community-driven and less financial-driven than the first system. The value of the currency token is generated through the value of a ride which can be bought with it. The currency token can be earned by taking people on a ride (through trade) and by driving eco-friendly (multiplier). That way there is no possibility that one just gets driven around. This may exclude people from taking part in the ride sharing. By introducing a token trading system, everyone could still take part in this system. Both these systems have a similar benefit in an economical perspective; they reduce the costs of transportation for everyone partaking in the ride sharing.

3.4.2 Pricing mechanism for rides

A possible approach to the pricing in the first token system is that the gas gets split between the driver and passenger for the time the passenger is in the car. It might be sensible that the passenger(s) have to pay the majority if not all of the gasoline costs. Additionally a small bonus may be introduced to increase the incentive for the driver and cover possible additional expenses of the driver. One can see the tokens-only system as an extension to the one-token-system, which could be introduced later in a hypothetical product cycle. The second system is a more complex than the first, since one has to earn tokens to be able to get a lift. It is a goal to make it possible that a person who drives "well"[4] earns more currency-tokens than a "bad" driver to incentivize "good" driving. This effect should not be as big as to render "good" drivers financially unattractive; the benefit of having a "good" driver should outweigh the marginal cost the latter. The cost of transportation is not reduced directly for the driver in a system where currency tokens can't be traded for real-world money, but through the possibility of buying a ride. It would thus be beneficial to introduce a currency token for real-world money trading system. All in all both systems have a similar financial benefit and also give a similar incentive to drive eco-friendly. The difference lies in it's implementation. The first system is a purely economical implementation, the second is more tailored towards bigger communities and a somewhat established system.
It must be noted that both systems to incentivize "good" driving somewhat rely on a forced price. It would be possible to let the driver and passenger determine the price via a bidding mechanism (face-to-face or digital), where the reputation system is used to reduce the information asymmetry - on both sides - between

[4]Various aspects could be considered in determining the driving quality; passenger satisfaction and eco-fieldly driving are two which readily come to mind.

the driver and the passengers.

3.4.3 A sketch of an app

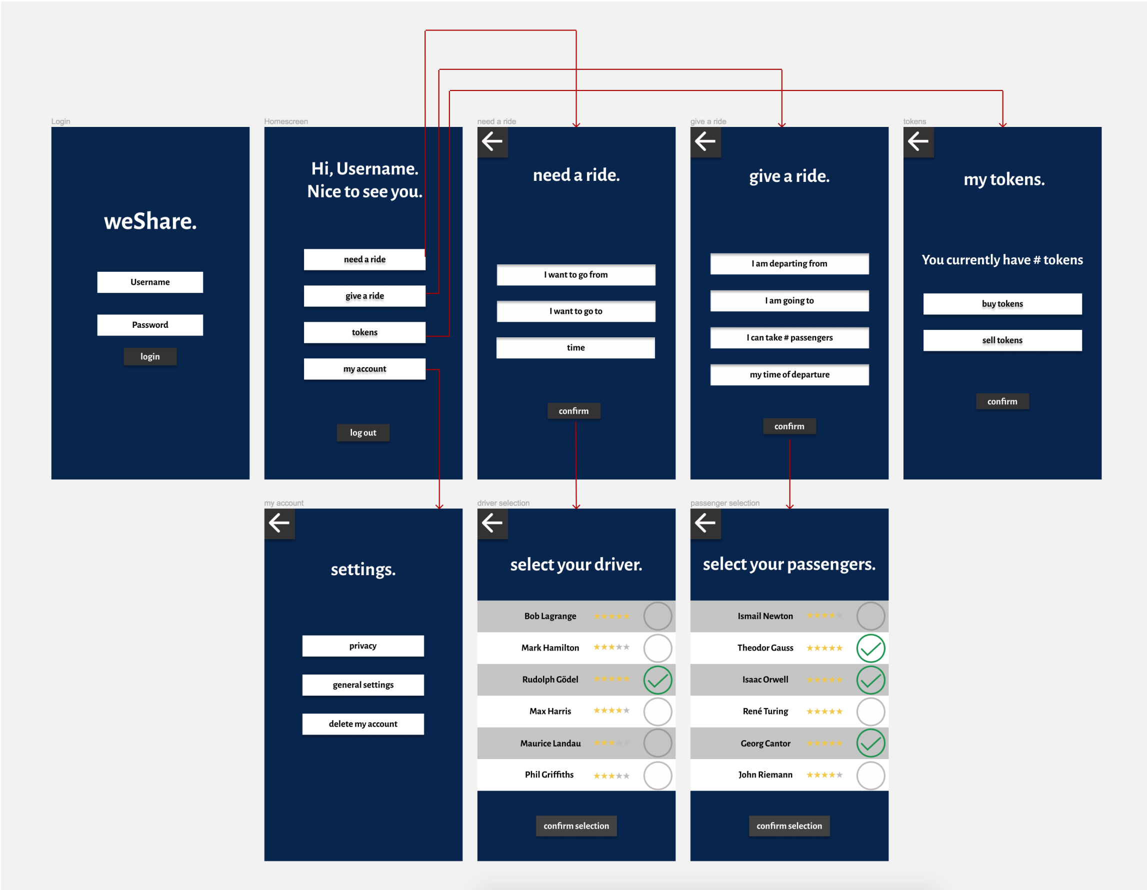

Figure 3: A sketch of some elements a possible mobile application might contain. The red arrows indicate the succession of UI-pages upon an action of the user.

In figure 3 a draft of an app can be seen. The red arrows indicate the succession of UI-pages upon an action of the user. It is important to note that important elements for a complete application are missing. Some of the most important thereof are further informations about the driver and passenger which the user should be able to access, such as a more detailed reputation and review system and the location at which the passenger desires to be picked up. Also the subcategories of the settings page should be implemented. To ensure transparency a page where the workings of the service are explained needs to be added.

3.5 Benefits of the system

3.5.1 Traffic, Greenhouse gases & financial benefits

If the amount of cars on the road is reduced by filling the cars, there will automatically be less traffic. This leads to shorter driving times and the cars can drive more efficiently since they perform less "stop and go" motions and therefore will use less gasoline and have less wear and tear.
Furthermore less cars will lead to less burned fuel, which hence will reduce the

amount of greenhouse gases emitted.
Both the driver and passenger can have a financial benefit as the driver earns money in the form of tokens (or real-world money) and the passenger doesn't have to pay for public transportation or a personal car.

3.5.2 Blockchain

The fact that the communication between the driver and the passenger is running entirely in the Ethereum Blockchain provides data and status storage, processing of transactions and logic in a decentralized manner. Therefore no intermediate (centralized) data center infrastructure is needed. The application thus is very close to pure decentralization.
This is not the only benefit of using the blockchain in our model. There is no third party involved in the whole process. This results in the transaction between the driver and the passenger being a more personal one. There is no upper control instance which both parties have to trust, like other providers of car sharing need. Even creating an account in a centralized database is not needed by design. If there is something like an account, it would be the smart contract of the corresponding user residing on the blockchain. However, that smart contract has to be created before the application can be used. There is no personal information on that contract. It is entirely in control of the user. If the user decides to delete his/her account for whatever reason, there is no obstacle-rich process to remove his/her data from the system. Try that on Facebook. They can't (or won't) give guarantee that all your data is removed, even after 90 days of processing your request! [15] On the blockchain, if the self destruct function is implemented correctly, only the owner of the contract (which is the user in our case) can remove it from the blockchain. It gives the control over user data back to the user itself, as it should be. Notice, this is not implemented yet in our codebase.
A problem that arises is, that bad drivers have to be prevented from simply deleting their bad-rated accounts and creating new ones, in order not to be affected by the drawbacks of their bad behavior.
Simply because there is only Ether and no real money involved, we don't need an instance in between to handle the payment between the passenger and the driver. So, no need to introduce complicated payment systems and interfaces. It's just balance that is decremented in one smart contract and incremented in another. The blockchain enables that the payment system consists of 4 lines of code.

4 Discussion and Evaluation

There is a security hole in the current implementation, such that in the smart contracts (`contracts/Customer.sol` [16]) there are 2 methods - `decrementBal()` and `incrementBal()` - with the reserved word `public`. Once the contract hash of a user is known by an attacker, these functions can be called and thus the balance manipulated to arbitrary amounts. The own balance can be manipulated too in the same way.
There is no process implemented which handles the disagreement of one party with the payment. In this case there will simply be no payment and the smart contracts get stuck in the state they are. A real implementation would need a

conflict solving instance for such problems.
In the current version the whole discovery process is not yet implemented. Therefore there is no ”discovery” in that sense. Both, the passenger and the driver, need to know the others account-hash beforehand to initiate a drive. Ethereum account hashes, consisting of 20 bytes in hexadecimal symbols, are not the easiest to remember for human beings. Thus a system similar to DNS to provide a lookup for usernames (or avatars) into hashes and vice versa would immensely increase the usability of the application. Since it’s very little data, the username can be stored as a property in the corresponding contract as well.
In case of an update of the software that includes an updated version of the smart contracts - say new methods or the state diagram has changed - all existing smart contracts have to be destroyed or marked as outdated and new ones have to be created. This is a process that is not addressed in this document. On the other hand updating the application on the smartphone that interacts with the contract via API won’t be a problem.
Creating an account (a smart contract) in the first place happens currently by hand. A nice and easy to use interface for this, integrated in the app, should be the goal.
Various other weaknesses of the current implementation, although caused by a lack of time to implement, are the following:

- There is by now no possibility to remove a smart contract from the blockchain. If a user wants to remove all his data, we need to implement a self destruct method only callable by the contract owner.
- Initially, when creating the smart contract, each user gets 100 ”balance” on his/her contract. There must be a possibility to refill and receive balance from a contract. At the moment there is no real flow of balance (or another currency) in the ecosystem of our application.
- The initial balance when creating a new account must be decreasing, else it would be possible to create an arbitrary amount of ”balance”.
- The user’s identity must be verified somehow initially, else a user can create as many accounts as desired.

There are also downsides in using the blockchain as a processing and storing medium. Every request has to be mined, which costs gas and can take long.
The team lost too much time building the dev environment only to demonstrate the code with instantly mined transactions. Only later a demonstration video was recorded where one could have fast forwarded the waiting period to mine a block. It turned out that the majority of the teams used the Robsten/Rinkeby Test Network in their live demo and it worked out great for them.

5 Outlook

The weShare approach is just in the starting phase. There is a huge potential in the world of car sharing. Most projects have in common that there is a centralized organization. Existing car- and ride share organizations like Uber, Lyft, BlaBlaCar e.t.c can take big cuts and commissions [17]. While writing this report, we came across HireGo, who has a very similar solution based on Ethereum but for car sharing rather than ride sharing [18].
The code is of course a rough sketch. It shows only a skeleton rather than a

working-in-the-real-world structure. But code is there to be built on.
The main focus is a peer-to-peer platform with its own currency in form of tokens on the blockchain. To improve the system there are many possibilities. First and foremost the basic concepts have to work in connection with a large enough community which uses it.
Tokens were already discussed in a previous section, however much remains to be added. Every addition to the system increases its complexity and thus also its intransparency. Combined with the Internet of Things to measure for example things like CO_2 emissions, there is a plethora of opportunities. This requires a lot of managing and balancing of the whole system. This is another point to work on; the whole token exchange and how the tokens play together.
In addition to the isolated car sharing topic, there are ways to extend the whole idea. The project is based on the idea that people take other people with them in their own car. But if people don't own a car they can only use it by getting picked up. Therefore an extension in form of sharing the cars physically could be an interesting addition. So if one doesn't use one's car one can rent it to other users for tokens.
Further possible implementations include an increased seamlessness between different applications used. For example if desired by the user it would be possible to implement a direct communication between the calendar and its entries (i.e. locations) and the weShare application. As weShare should increase the transparency, this would have to be adequately declared to the user and added optionally.

6 Appendix

6.1 Licensing

6.1.1 This text document

6.1.2 The licensing of the code written during the ETH BIOTS Hackathon 2018

The software code which is part of this report is open source and available at `https://github.com/ETHBiots2018/weShare`.

6.2 Contribution & other statements

All authors mentioned on the front page have contributed equally to this report. This project report was written as part of the spring 2018 course 'Blockchain And the Internet of Things (851-0591-01L)' run by M. Dapp, S. Klauser, and D. Helbing.

References

[1] "Touring Club Schweiz, Kosten eines Musterautos." https://www.tcs.ch/de/testberichte-ratgeber/ratgeber/kontrollen-unterhalt/kilometerkosten.php Accessed: 22.03.2018.

[2] Bundesamt für Statistik, "Mobilität und verkehr - Taschenstatistik," 2017. BFS-number: 836-1700.

[3] A. Stephany, *The Business of Sharing.* Palgrave Macmillan, London, 1st ed., 2015.

[4] "The Paris Agreement." https://unfccc.int/process/the-paris-agreement/what-is-the-paris-agreement Accessed: 25.04.2018.

[5] "Main sources of carbon dioxide emissions." https://whatsyourimpact.org/greenhouse-gases/carbon-dioxide-emissions Accessed: 22.03.2018.

[6] "Average car-weight USA." https://www.nytimes.com/2004/05/05/business/average-us-car-is-tipping-scales-at-4000-pounds.html Accessed: 25.04.2018.

[7] S. Walpole, D. Prieto-Merino, P. Edwards, J. Cleland, G. Stevens, and I. Roberts, "The weight of nations: an estimation of adult human biomass," *BMC public health*, 2012. https://bmcpublichealth.biomedcentral.com/track/pdf/10.1186/1471-2458-12-439 Accessed: 22.03.2018.

[8] M. Keller and P. Wüthrich, "Neuberechnung Staukosten Schweiz 2010-2014," 2016. https://www.are.admin.ch/are/de/home/medien-und-publikationen/publikationen/verkehr/neuberechnung-staukosten-schweiz-2010-2014.html, Accessed: 28.04.2018.

[9] J. Benet, "Ipfs - content addressed, versioned, p2p file system." Unpublished paper. https://github.com/ipfs/papers/blob/master/ipfs-cap2pfs/ipfs-p2p-file-system.pdf Accessed: 27.04.2018, 2018.

[10] P. A. Samuelson and W. D. Nordhaus, *Volkswirtschaftslehre.* No. 4, FinanzBuch Verlag GmbH, 2010. Page no. 67.

[11] R. Chen, F. Bao, and J. Guo, "Trust-based service management for social internet of things systems," *IEEE transactions on dependable and secure computing*, vol. 13, no. 6, pp. 684–696, 2016.

[12] validitylabs, "Ethdapp-boilerplate." https://github.com/validitylabs/EthDapp-Boilerplate Accessed: 24.04.2018, Commit: 62f29fc352ab691e54821c7573ebb2361b9b3d8f.

[13] Ethereum, "Ethereum javascript api." https://github.com/ethereum/web3.js/ Accessed: 24.04.2018.

[14] trufflesuite, "Personal blockchain for ethereum development." http://truffleframework.com/ganache/ Accessed: 24.04.2018.

[15] "How do I permanently delete my [Facebook] account?." https://www.facebook.com/help/224562897555674 Accessed: 26.03.2018.

[16] "Source code of weShare." https://github.com/ETHBiots2018/weShare/ Accessed: 14.03.2018.

[17] E. Huet, "Uber raises uberx commission," September 2015. https://www.forbes.com/sites/ellenhuet/2015/09/11/uber-raises-uberx-commission-to-25-percent-in-five-more-markets/#1876d5e73214, Accessed: 27.04.2017.

[18] "HireGo, decentralised car sharing platform." https://www.hirego.io/ Accessed: 22.03.2018.