

CS3010 Project Definition: Decentralised Taxi App

Scott Street
159027651

October 2017

1 Subject

When the Internet was in its infancy, if you wanted to use it for a specific application, you might have written a protocol. That way, anybody who wanted use the Internet for that purpose would have a common way of doing it - and if a new person came along and wanted to join in, they could just write their software to conform with the standard.

In the past 15-or-so years however, the landscape has changed. Companies now favour their own proprietary systems, where they can have complete control, and ultimately gain the most profits. Specifically companies such as Uber have taken an industry which was once fairly well distributed, and put the control in their own hands - they decide who can be a driver, they manage the fares, and how much they pay their drivers.

But recent developments with distributed networks threaten to disrupt this comfy business model. Technologies such as Ethereum allow "trustless" applications, where activities of a single node are verified by the entire network. It's an area which is yet to be explored to its full potential, but all the features needed to be able to implement feature-rich apps are there. The logic of applications running on such a system has to be rethought, but with Uber as an example, there would be no central authority to take a cut of profits. The entire system would be self-regulating.

2 Deliverable

A ride-sharing webapp accessible with an Ethereum network-enabled browser, designed in such a way that no single entity has control over the running of the system. Drivers will be able to advertise their location (published to blockchain), and riders will be able to send job proposals (containing pickup and drop-off locations) to these drivers on a peer-to-peer basis. This protects the privacy of the rider by ensuring that only the chosen drivers are able to see the rider's location. When a driver initially advertises their location, they are required

to provide a deposit to the network, which will be returned in completion of a trip. This gives the driver a stake in wanting to complete a journey, and should reduce spam on the network.

Drivers are then able to issue a response to a proposal, by either rejecting or quoting a price for the journey. This allows drivers to choose which journeys they take, and prevents drivers from having to travel a long distance to a pickup location, compared with if the allocation was done randomly. Should the rider choose to accept the quote, then both rider and driver form an agreement via a smart contract on the Ethereum network. This includes the passenger offering up the cost of the journey, plus an additional deposit equal to the amount the driver provided previously.

At this point, the fare for the journey, a deposit from the rider, and a deposit from the driver are all held by a smart contract. This acts as an incentive for the driver and rider to successfully complete the physical journey. When this is done, and both parties are in agreement that it is completed, then the deposits can be returned and the fare paid to the driver.

All monetary transactions will be executed with cryptocurrency on the Ethereum network, so as to minimise fees and prevent the transaction from being intercepted by a third party.

As the vast majority of interactions between riders and drivers will be based on no existing knowledge of the other party, a reputation system will be used to form a layer of trust. Based on previous journeys, and the ratings given to both rider and driver on completion of each, future riders will be able to make informed decisions about which drivers to send job proposals. And in the same fashion, drivers will be able to decide which riders' proposals to accept.

3 Originality

Although ride-sharing apps aren't a new thing, nearly all existing solutions are controlled by a central authority who take a cut of the profits. This means users are at the mercy of the company when it comes to fares, and drivers must be approved, potentially opening the way for discrimination.

This project eliminates these problems by taking control away from any one part of the system. All transactions take place in a peer to peer nature, with the network being the only intermediary. This ensures that the two parties involved have full control over the process, whilst at the same time preventing one from cheating the other.

4 Timetable

The following proposed timetable will be used to track progress over the course of the project. The work is broken down into fortnightly blocks. Through the entirety, a project diary will be kept to keep track of key decisions. This is to be used as the basis for much of the final report.

Date	Planned Activity
02/10/17	Begin writing a formal project definition. Decide on project objectives, and have an idea of what features will be included. Which features would the system not work without.
16/10/17	Finish project definition. Begin mapping out interactions of users with the system and other users as a diagram. Create protocol documentation - similar to RFC. This is to be used to test against. Project definition due 20th October
30/10/17	Start implementing said protocol, with aim of creating fully function implementation (not including user interface). Test against RFC-style document.
13/11/17	Finish initial protocol implementation.
27/11/17	Develop testing suite for protocol implementation.
11/12/17	Fix any issues with implementation, and complete testing. Begin TP1 progress report.
25/12/17	Continue TP1 progress report.
08/01/18	Exams scheduled in this period, therefore expecting a slow down in project work. TP1 progress report due 19th January.
22/01/18	Continuation of development based on progress report. Begin writing of final report.
05/02/18	Research into how to develop the user interface. Review of existing mobile Ethereum clients.
19/02/18	Development of user interface.
05/03/18	Addition of identified "stretch" features.
19/03/18	Finalising development and report writing.
02/04/18	Report writing.
16/04/18	Finalising report and considering how to present the project during demos. Final submission due 27th April.