

Principles of Decentralized Ledgers

Coursework

Scalable Blockchain Application

Dr. Arthur Gervais

2019

1 Introduction

By now you've become much more experienced in the blockchain space. You also have likely realized that current blockchains don't scale — to be precise, current Proof of Work blockchains only scale towards about 10 transactions per second, for the whole network! Visa for example is reported to support up to 57'000 transactions per second.

In this coursework, you are expected to build a scalable blockchain application, that takes advantage of existing 2nd layer solutions. There are several technologies that you can choose from, it's important to choose one that is *permissionless*, and which doesn't introduce a new *custodian*. Below are two choices that currently are mainnet ready, feel free to consult me if you have another solution in mind that you'd like to build upon.

- Lightning (payment channels): <https://github.com/lightningnetwork/lnd>
- Liquidity (commit-chain based hubs): <https://docs.liquidity.network>

2 Schedule and Guidelines

The coursework is scheduled according to the following guidelines.

- Pick a subject
- The project is assessed individually
- Implement the application
- Write a short report (max 4 pages)
- Report + Code due by 6th of March 2019

- You can present your work at the end of the course if you wish so

You're free to choose one of the following topics (cf. Section 3), or a topic of your interest. If you plan to pursue your own topic, please do speak briefly to me in advance.

3 Topics

- Airdrop token tool
- Spinning wheel (<https://www.lightningspin.com/>)
- Streaming service, pay per second
- Tipping bot for Telegram, Whatsapp, Wechat or similar
- Tipping Browser (Chrome or Firefox) extension for websites (reddit, youtube, fb, and others)
- Satoshi Dice (should be provably fair)
- Collaborative drawing (e.g. <https://satoshis.place/>, <https://canvas.liquidity.network>)
- Point of Sale for Android
- File Selling website (e.g. <https://github.com/ElementsProject/filebazaar>)
- Wordpress publisher plugin (e.g. <https://github.com/ElementsProject/wordpress-lightning-publisher>)
- Donation button (e.g. <https://github.com/ElementsProject/nanotip/>)
- Hourglass Pyramid Scheme (e.g. <https://exitscam.me/play>)
- CryptoKitty Game (<https://dappradar.com/app/3/cryptokitties>)
- Any app that you would want to do checkout the following for inspiration (<https://dappradar.com/rankings>)

These topics are explained more in-depth on the Wednesday lecture when coursework 3 starts.

4 Grading

The grading will be mostly influenced by

- a clear description of the adversarial threat model
- description of why you need a blockchain

- code quality - simplicity is preferred over complexity
- secure programming considerations are a plus
- building a web/GUI interface is optional but a big plus

The report should contain the following information:

1. Topic description and what's your architectural choice on how to implement it (application logic description).
2. Threat model description, i.e. what an adversary is supposed to be able to do.
3. At least one screenshot showcasing your application.

5 Conclusion

This exercise should expose you to the art of writing and interacting with a scalable blockchain application that can potentially reach mainstream adoption.