

FinalProject Peer2Peer Systems and blockchains

Development of a DAPP for Smart Auctions

Lorenzo Bellomo, 531423

Project Structure

The project is organized as follows:

- *Root folder*: It contains only the report file and a folder DAPP, all the code is in this folder.
- *DAPP folder*: It contains the configurations files needed (*bs-config.json* for lite-server, *package.json* for node, *truffle-config.json* for truffle). It additionally holds file *ropsten.json*, which contains the private credentials for ropsten deployment. Then other folders are explained just in the next points.
- *contracts, migrations, test and build*: Truffle related folders. They are compulsory when dealing with truffle projects. Also, at this level, the build folder is generated.
- *node_modules and src*: Folders related to node and the front-end part. The *src* folder contains some *.css* files (bootstrap) in order to improve the clarity of the web page.
- *index.html*:
- *app.js*:

Contracts ChangeLog

This section contains all the modifications applied to the delivered Final Term smart contracts. All the modifications have a motivation assigned.

- *Vickrey - Removed file Util.sol*: the only method that was provided in this file (computation of the nonce and amount hash for the Vickrey auction) has been moved to the file Vickrey auction.
- *Both - Added auctioneer role*: the text of the project explicitly asks for an auctioneer role. The owner decides which account is the auctioneer when he calls the newly added method `createAuction`. If he does not decide an auctioneer, he becomes the one to take the role. As a note, in the Vickrey auction, the role of the auctioneer doesn't have a clear purpose. The way it will be used during the tests is to take care of the phase switches (at his expenses).
- *Both - Moved start time*: Both auctions, in the provided FinalTerm implementation, started in the moment the constructor was called. This was split in two phases. Phase one starts when the constructor is called, and it is the phase in which the auction is on the blockchain, but is not yet active (note, grace period is considered active). The secondPhase starts when the owner calls the method `createAuction`. In this method, the grace period begins and an event is emitted.
- *Dutch - Added auctionPhase*: In order to better implement the previous point, the boolean variable `ended` has been changed to a phase enum in the Dutch Auction. The phases are NEW, ALIVE and ENDED.

- *Dutch - Changed assert to require*: In order to better respect the require-assert definition, the assert in the constructor has been changed to a require.
- *Both - Added some getters*: In order to improve clarity and respect the previous changes in the contracts, some getters have been added.