Public Ledger for Auctions

André Miguel Faria da Silva
Dep. de Ciência de Computadores
Fac. de Ciências da Univ. do Porto
Porto, Portugal
up201906559

João Guilherme Marques Afecto
Dep. de Ciência de Computadores
Fac. de Ciências da Univ. do Porto
Porto, Portugal
up201904774

Marcos Guilherme de Freitas Tauber
Dep. de Ciência de Computadores
Fac. de Ciências da Univ. do Porto
Porto, Portugal
up201810809

I. Architecture

The architecture of this project is the following: DHT (Kademlia), Blockchain with Merkle Trees for transaction protection and a simple English auction system. The kademlia works on its own and can be used though a REPL (Read-Execute-Print Loop). The blockchain has no interface, it is only a library. The auction works on its own and can also be used through a REPL.

A. Kademlia

The Kademlia DHT implementation follows mostly [2], with security considerations in [1].

The implementation uses SHA-1 as a hash function, which is known to be cryptographically insecure. However, we chose to use it anyway since it is the function the original paper is based on, and it can be swapped in later since no strict dependency was placed on it, other than its hash length (as in the original paper).

The proposed methods for protecting against Eclipse and Sybil attacks in [1] were also implemented.

The DHT was also extended to provide a publish/subscribe mechanism based on simple network flood.

B. Blockchain

The blockchain was designed in 4 phases: Blocks with PoW, Blocks with PoS, Chains and Merkle Trees.

Blocks are defined by their index, timestamp, previous block hash, nonce, list of Valid Transactions, the Merkle root hash for those transactions and the difficulty of the PoW needed, meaning total number of zeros in the beginning of the block hash.

There are only 2 ways of getting a complete block, guess a nonce that generates a hash that complies with the difficulty, aka mining or put your own money on the transaction (stake) and signing it, be aware that there is a minimum amount of money that you are allowed to put in order for it to be accepted, and in case its not accepted there are penalties, which were not implemented.

Following, we build the Chain, the chain consists of blocks that are verified before entering the chain. When needing to choose a chain, we always choose the biggest. In case they are equal we do nothing.

To finish this part we constructed Merkle trees. A Merkle tree is a binary tree we chose to implement in the easiest way possible, a vector.

The objective of Merkle trees is that we can condense N transactions into a single hash, and also efficiently verify properties of the transaction list, namely contents and ordering. We followed the implementation in [11].

C. Auction Mechanisms

For the auction itself, we made a simple auction ”House” that can be expanded to work above the P2P network. It works based on a key pair (Public and Private) that is used to identify each node on the auction world. Anyone can create an Auction, and signing it to be sure that is legit. Any auction broadcasted without a valid signature are not considered and therefore discarded.

Anyone can participate in the auction, by simply sending a signed bid to the network, the auction automatically closes after a define period of time without receiving any bids and the winner is the latest bid.

II. Difficulties

We could not finish the work required to integrate all three components, despite having each of them work independently. Also, since the integration was not successful, there are no trust mechanisms being used, and since implementation of trust mechanisms was scheduled to only happen after integration, the section was left undone.

For proof-of-stake, our work was done based on the research outlined in III-E. Also in proof-of-stake, the implementation is not properly secured. In order to properly create stakes, we needed to have a method of freezing the currency staked, so it can only be traded after a certain amount of time passed.

III. Our Thoughts about the assignment

A. Explanation

This section is our take in the assignment and our opinion on a more interesting way to put a project of this dimension. It’s also a summary of our understanding of the state of the art technologies relative to the topics in the assignment.

B. Opposite what now?

"This assignment requires the implementation of a public blockchain (non-permissioned), but opposite to Bitcoin and Ethereum, the purpose here is to have a decentralized public ledger capable of storing auction transactions.", says the assignment description. Three months into this work, we're still trying to find why this work isn't to implement Bitcoin and/or Ethereum *exactly*, and then add some way to hold an auction on top of that. Speaking of which...

C. Auctions in a decentralized world

In the world outside computer networks, auctions are held with the recourse to a trusted third-party, usually an auction house. They also are backed by the law and all of the justice system that comes along with it.

1) The decentralized world: In the decentralized world, there is no government, no laws and no trusted third-parties. For an auction to function, we must be able to guarantee that the seller and the winning buyer trade currency and product atomically, i.e. that one cannot receive theirs, without giving what was agreed upon. Since our only method for establishing the sort of atomicity required by financial transaction of this (and any other) sort is the blockchain, we must be able to encode and enshrine the entirety of the final trade transaction into it. To do so, the items in the trade must also be entirely representable and valued on and by the blockchain. An interesting consequence of this is that only two items can truly be auctioned: cryptocurrencies and non-fungible tokens (NFTs).

So, to create a useful auction system, this work must also include beyond all that was described in the assignment, a mechanism to also mint and trade NFTs.

2) Smart contracts: Having established that an implementation of NFTs is desired, if not outright required in order to provide an useful auction system on a decentralized ledger, a question arises as to why not implement both with blockchain smart contracts. Widely used, these albeit complex tools allow for the ledger to process more than just what was proposed, and underpin NFT technology.

D. Networking layers

In [4], the Ethereum community describes their approach to P2P networking. And while they do use Kademlia, they also say: "The protocol used for the node-bootnode interactions is a modified form of Kademlia which uses a distributed hash table to share lists of nodes. [...] For discovery, where a node simply wants to make its presence known in order to then establish a formal connection with a peer, UDP is sufficient. However, for the rest of the networking stack, UDP is not fit for purpose. The informational exchange between nodes is quite complex and therefore needs a more fully featured protocol that can support resending, error checking etc."

Herein lies an interesting observation. The Kademlia protocol as described in [2] is very effective at what they call discovery, the process of finding nodes to talk to, either in general or in order to find a specific node. However, they also note that in order to support the whole ecosystem, the Kademlia DHT is not enough, and in fact may not be desirable. In order to, for example, implement a publish/subscribe system, in the current iteration of the Ethereum blockchain, they make use of libP2P [5] and it's publish/subscribe implementation, gossipsub [6].

Considering this, one must wonder why we are attempting to "fit a square peg in a round hole", trying to extend the DHT, when considering the state of the art we should be building a whole overlay network instead.

E. Trust

1) The state-of-the-art: Table I includes the top 5 cryptocurrencies today by market capitalization. It also includes their consensus mechanism and the methods by which they ensure trust. This table mostly exists as a result of our research in trying to understand how different implementations deal with the issue of trust. The short answer is: they don't. We will ignore the two stablecoins, since they rely on other coins or on third-parties, and were only included for the sake of completeness.

The Bitcoin [3] methodology is very simple. Miners will want to generate blocks the majority of the network will consider valid, and since nodes will follow the longest chain they can see, i.e. the chain with the most compute power behind it, Bitcoin creates a no-trust environment, where each node will not rank others based on anything. It simply follows wherever the majority of compute power goes.

It the case of Ethereum and BNB, they apply a proof-of-stake algorithm, but the novelty over the likes of [10], it what is at stake. Nodes can trust the validators, because the validators have put forward a set amount of cash they stand to lose if they decide to dissent. Important to note that none of this intends to establish a comparison between different methods. Instead we illustrate the methods that have been wildly successful, since one cannot help but wonder why we are implementing trust mechanisms (and at the networking layer no less), when billions of dollars stand behind conceptually simpler and just as if not more effective measures.

2) Sybil attack hardening on the upper layer: In [1], the authors propose that a crypto puzzle be used the prevent Sybil attacks. However, above we noted that highly valuated cryptocurrencies take a different approach to trust, and their protection against Sybil attacks is also *sui generis*.

Assuming we are willing to accept transactions from any source, we consider that for validation of the next block, a validator must stake their money into the block being valid. In order to acquire more votes and take control of the network consensus, an attacker must therefore control not a majority (or supermajority) of nodes, nor of compute power, but of the total amount of staked currency. Important to note that staked currency is locked for a period of time before and after validation of a given block occurs, to prevent double-use. Placing and removing currency from the stake is a transaction on the blockchain, and therefore through the

Currency	Market Capitalization	Type	Trust Origin
Bitcoin	$521 billion	Proof-of-work	Intrinsic. The network trusts the majority of the compute power.
Ethereum	$217.5 billion	Proof-of-stake	Validators stake their money. The network trusts the majority stake.
Tether	$82.9 billion	2nd Layer	Relies on other chains. Is a stablecoin (locked to USD by a third-party).
BNB	$48.2 billion	Proof-of-stake	Based on Tendermint. [7] Validators stake their money. The network trusts the majority stake.
USD Coin	$29.5 billion	2nd Layer	Relies on other chains. Is a stablecoin (locked to USD by a third-party).

TABLE I
LIST OF CRYPTOCURRENCIES BY MARKET CAP, SHOWING THEIR CONSENSUS MECHANISM. INFORMATION FROM [8] AND [9]

regular methodology used for the chain, a consensus is reached over who are the validators and of what the total stake is.

F. What do we propose?

We believe it to be impolite to point out flwas in others' works, without providing constructive criticism. To that extent, we follow with suggestions over how an assignment like this could be conducted, while also allowing for the opportunity for students to deliver fully functional implementations, regardless of their computing language prowess and/or available time.

1) For a decentralized auction: If the objective is for students to: a) Familiarize themselves with the technologies in study and b) try their hand at using them in new and/or different ways, we propose allowing students to instead of developing entire technology stacks from scratch, they try their hand instead at developing a smart contract. This contract could be for an auction house, or an exchange, or another novelty. While the usefulness of the contract could be contested, attempting to write one at least once has didactic value regardless.

2) For the whole assignment: If the objective is to have students implement these technologies and focus on their security properties, then a better method of presenting this assignment is to split the work into pieces designed to operate together, much like popular projects in this field are implemented. Each piece is assigned to one group, and students must ensure their work conforms to the agreed-upon (and teacher-guided) interface. Multiple groups can work on the same piece, with the constraint that each piece must be made interchangeable with another group's work. In the end, the whole class must be able to come up with a whole system, and it will be far easier to have it be as close as possible to what was assigned.

As a bonus, this method is valuable for students to learn to work as a team in conditions similar to those found in the broader job market, where they must work to follow a specification, in a team that is only but a part of a larger combined effort to develop large, interconnected software packages, with all the difficulties inherent to such a goal.

REFERÊNCIAS

[1] I. Baumgart and S. Mies, "S/Kademlia: A practicable approach towards secure key-based routing," 2007 International Conference on Parallel and Distributed Systems, Hsinchu, Taiwan, 2007, pp. 1-8, doi: 10.1109/ICPADS.2007.4447808.

[2] Maymounkov, P., & Mazieres, D. (2002). Kademlia: A peer-to-peer information system based on the xor metric. In Peer-to-Peer Systems: First InternationalWorkshop, IPTPS 2002 Cambridge, MA, USA, March 7–8, 2002 Revised Papers (pp. 53-65). Berlin, Heidelberg: Springer Berlin Heidelberg.

[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: http://bitcoin.org/bitcoin.pdf

[4] https://ethereum.org/en/developers/docs/networking-layer/

[5] https://github.com/libp2p/specs

[6] https://github.com/libp2p/specs/blob/master/pubsub/gossipsub/gossipsub-v1.0.md

[7] Kwon, J. (2014). Tendermint: Consensus without mining. Draft v. 0.6, fall, 1(11).

[8] https://coinmarketcap.com/, retrieved 2023-05-21

[9] https://www.centre.io/usdc, retrieved 2023-05-21

[10] J. Yu, D. Kozhaya, J. Decouchant and P. Esteves-Verissimo, "RepuCoin: Your Reputation Is Your Power," in IEEE Transactions on Computers, vol. 68, no. 8, pp. 1225-1237, 1 Aug. 2019, doi: 10.1109/TC.2019.2900648.

[11] Merkle, R. C. (1988). "A Digital Signature Based on a Conventional Encryption Function". Advances in Cryptology — CRYPTO '87. Lecture Notes in Computer Science. Vol. 293. pp. 369–378. doi:10.1007/3-540-48184-2_32. ISBN 978-3-540-18796-7.