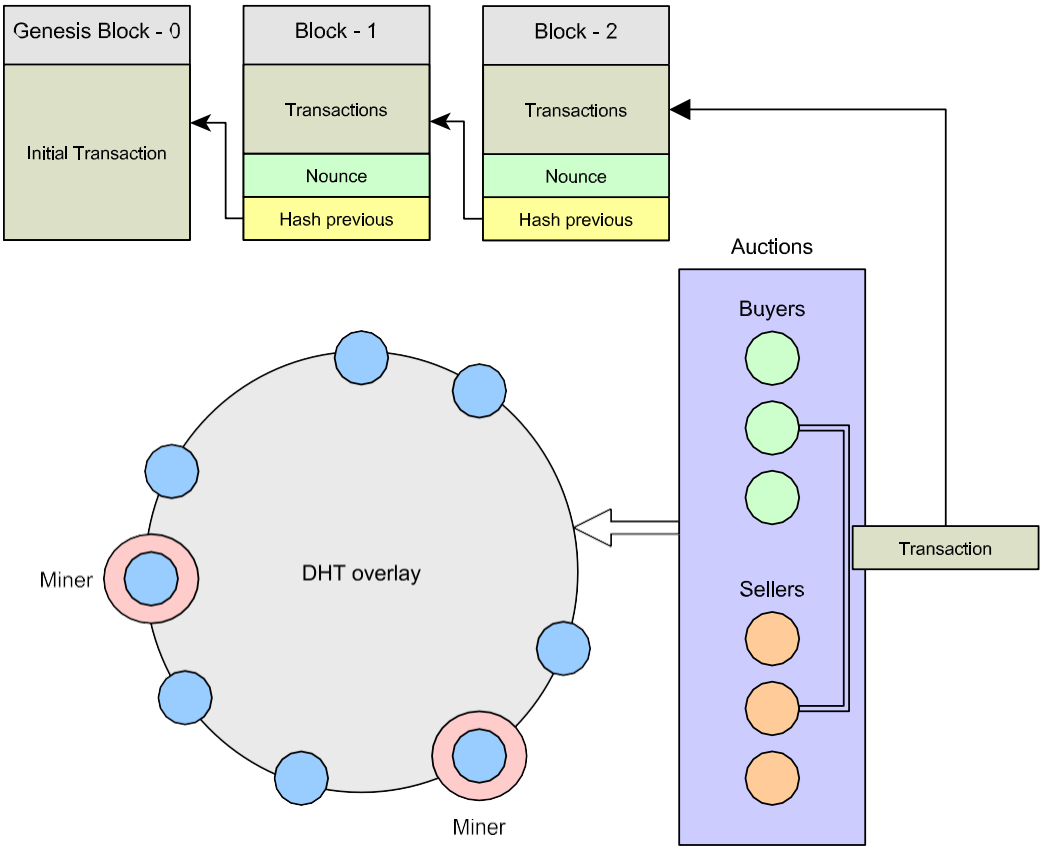
Assignment: Public Ledger for Auctions

System and Data Security 23/24

Feb 13, 2024

This assignment requires the implementation of a public blockchain (non-permissioned) [1, 2], but unlike Bitcoin and Ethereum, the purpose here is to have a decentralized public ledger capable of storing auction transactions. All code must be written in Rust (preferably) or Java.



The work is divided into 3 parts, **distributed ledger**, **secure P2P** and **auction mechanisms**:

- The secure ledger should be modular, and it must support PoW and Delegated Proof-of-Stake (DPoS) [9] (configurable) as the core consensus:
  - using proof-of-work [1] (2pt) and,

  - Proof-of-stake [3,9] (2pt). **The reputation mechanisms from be integrated here.**

- A P2P layer to gossip the necessary data to support the blockchain that must include (4pt):
  - must implement S/Kademlia [5] (3pt).
  - Resistance to Sybil and Eclipse attacks (2pt).
  - Implement trust mechanisms [6,9] (2pt). **To be integrated in proof-of-stake.**
- An auction system capable of supporting sellers and buyers using a single attribute auction following the English auction (2pt):
  - Transactions should be saved in the blockchain (using public key crypto) and most be properly gossiped to all the nodes in the system (2pt)
  - A publisher/subscriber should be built on top of Kademlia to support auctions [8]. (2pt)
- A Fault injection mechanism that allows to shutdown one or more nodes in the system, simultaneously. During presentation this mechanism must be used to show the resiliency of the system (3pt)
- Mandatory report with a maximum of 6 pages (with unlimited pages for references), A4, font size 11, using latex.
- The architecture must be clearly presented, with the design choices being driven by the functional requirements (presented above). It also must present the assumptions made, namely mandated by theoretical and practical limitations.

Limitations - Warnings! You can only use low level libraries to help you, such as Netty for communications, or BouncyCastle for crypto. You cannot reuse existing open-source projects to do this assignment, namely for proof-of-* and P2P. **In doubt, ask.**

The expected outcomes are the following:

- Design and implementation of the system, with security as a first-class citizen. This means that identity, authorization, authentication, access control and trust/authoritative domains must be addressed. Beware of the communications channels and crypto you use; be sure they fit your needs. The codebase must be hosted in a private repository in Github or Bitbucket.
- **Use of LLMs is allowed, but keep in mind that with great power comes great responsibility (Peter Parker™). If you cannot explain a snippet of code in your project, it will be an automatic flunk. If you cannot explain the rationale for your design decisions will also carry an automatic flunk.**

Hint: Use the IntelliJ IDEs for development (https://www.jetbrains.com/idea/). They are free for students. Scholar is your friend. Use it for exploring related work https://scholar.google.com

References

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[2] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32, 2014.

[3] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS)*, 20(4):398–461, 2002.

[4] João Sousa, Eduardo Alchieri, and Alysson Bessani. State machine replication for the masses with bft-smart. 2013.

[5] Petar Maymounkov and David Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *International Workshop on Peer-to-Peer Systems*, pages 53–65. Springer, 2002.

[6] Ingmar Baumgart and Sebastian Mies. S/kademlia: A practicable approach towards secure key- based routing. In *Parallel and Distributed Systems, 2007 International Conference on*, pages 1–8. IEEE, 2007.

[7] Francis N. Nwebonyi, Rolando Martins, and Manuel E. Correia. Reputation based approach forimproved fairness and robustness in p2p protocols. *Peer-to-Peer Networking and Applications*, Dec 2018.

[8] Bittorrent publish/subscribe protocol. http://bittorrent.org/beps/bep_0050.html. Accessed: 2021-02-15.

[9] Yu, Jiangshan, David Kozhaya, Jeremie Decouchant, and Paulo Esteves-Verissimo. "Repucoin: Your reputation is your power." *IEEE Transactions on Computers* 68, no. 8 (2019): 1225-1237.