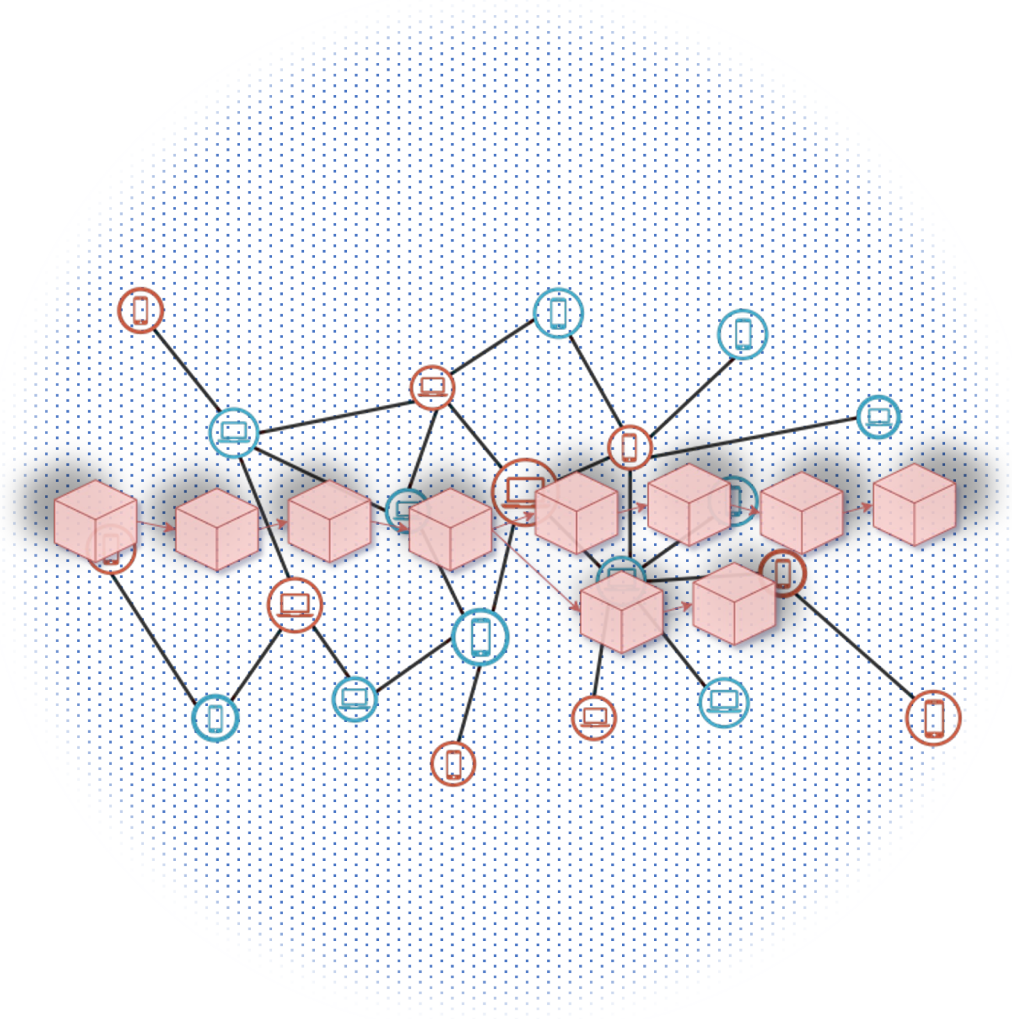
Public Ledger for Auctions

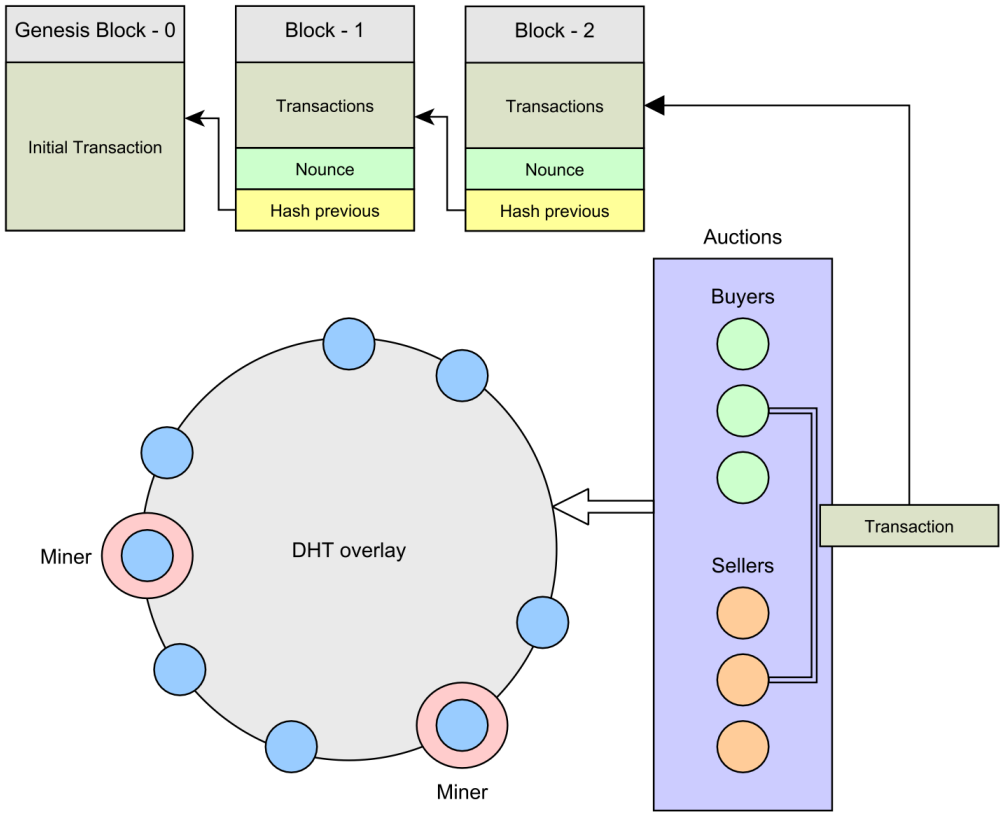
System and Data Security 22/23



Public Ledger for Auctions - Overview

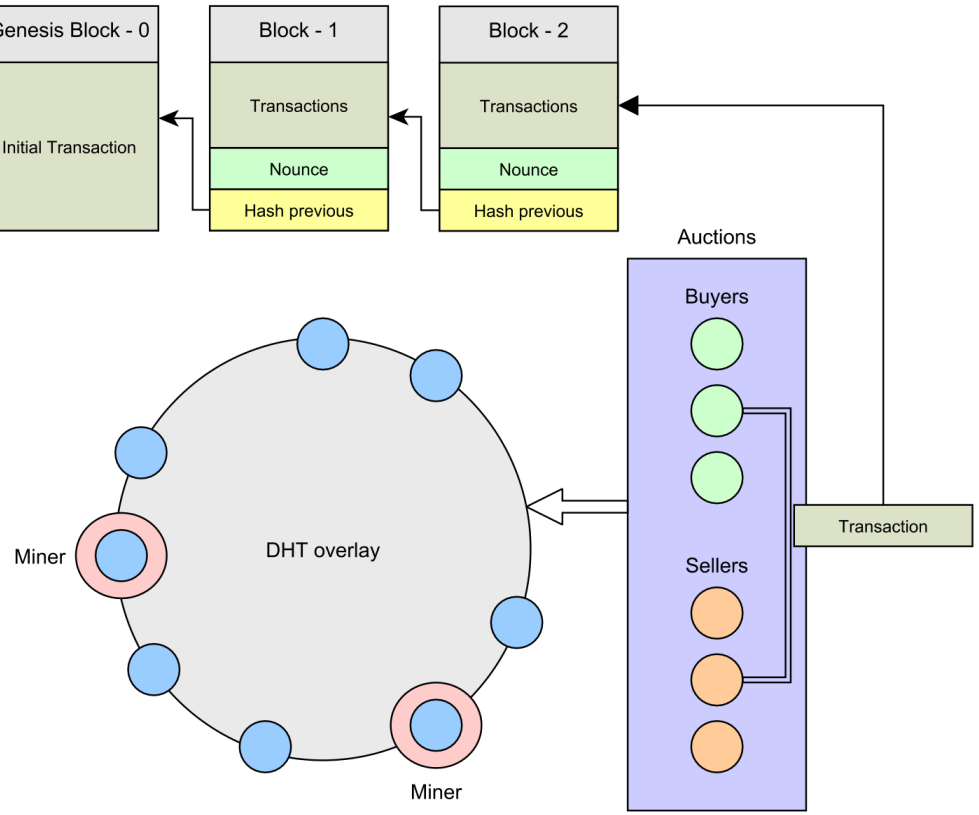
- Design and build a decentralized public ledger for storing auction transactions

- The work is divided into 3 parts
  - Secure P2P overlay
  - Secure and Distributed ledger
  - Auction mechanisms



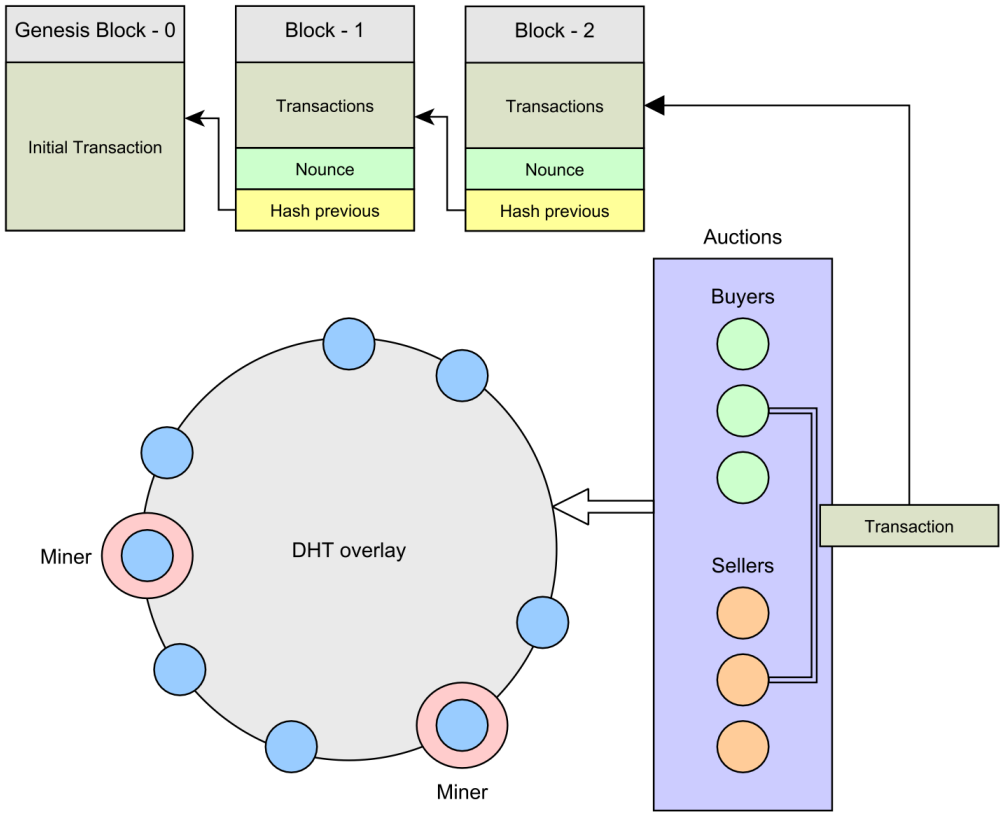
Secure P2P Overlay

- Offers storage using a DHT approach
  - Based on Kademlia DHT
  - Resistance to Sybil and Eclipse attacks
  - Implement trust mechanisms
- Based on gossip communication protocols for exchanging information



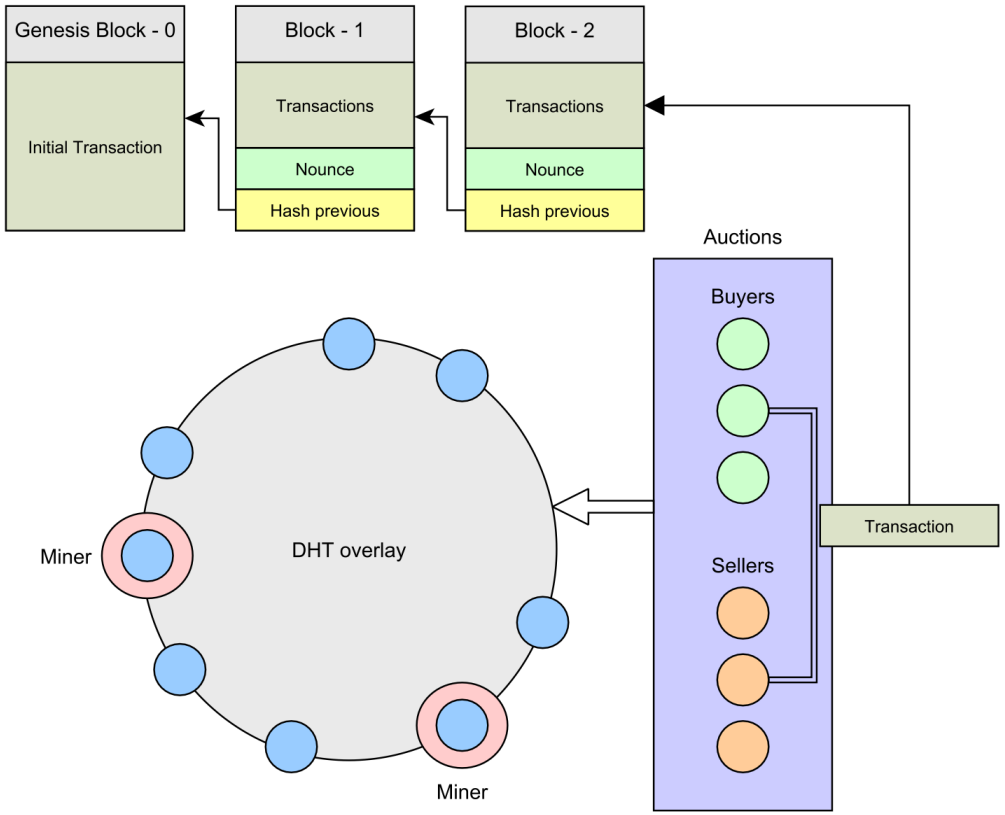
Secure Distributed ledger

- Based on a permissionless approach
  - I.e., public
  - Nodes need not be known ‘a priori’
- Should be modular
  - Support PoW as the main consensus mechanism
- Optional support for
  - Support Proof-of-stake as an option
  - Permissioned approach based on BFT consensus



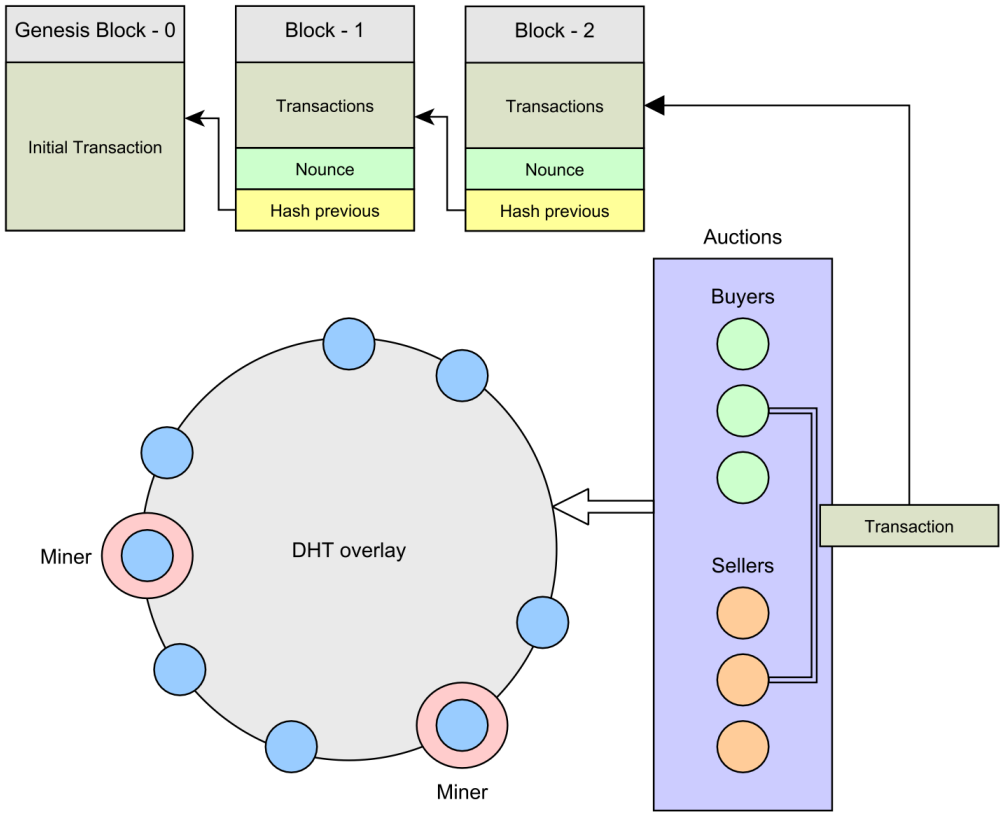
Auction system

- Capable of supporting sellers and buyers
- Using a single attribute auction
  - Following the English auction model
- Transactions (auctions & biddings) should be saved using the ledger
  - Use public key crypto
- Using a publisher/subscriber system to support the auctions
  - Built on top of Kademlia



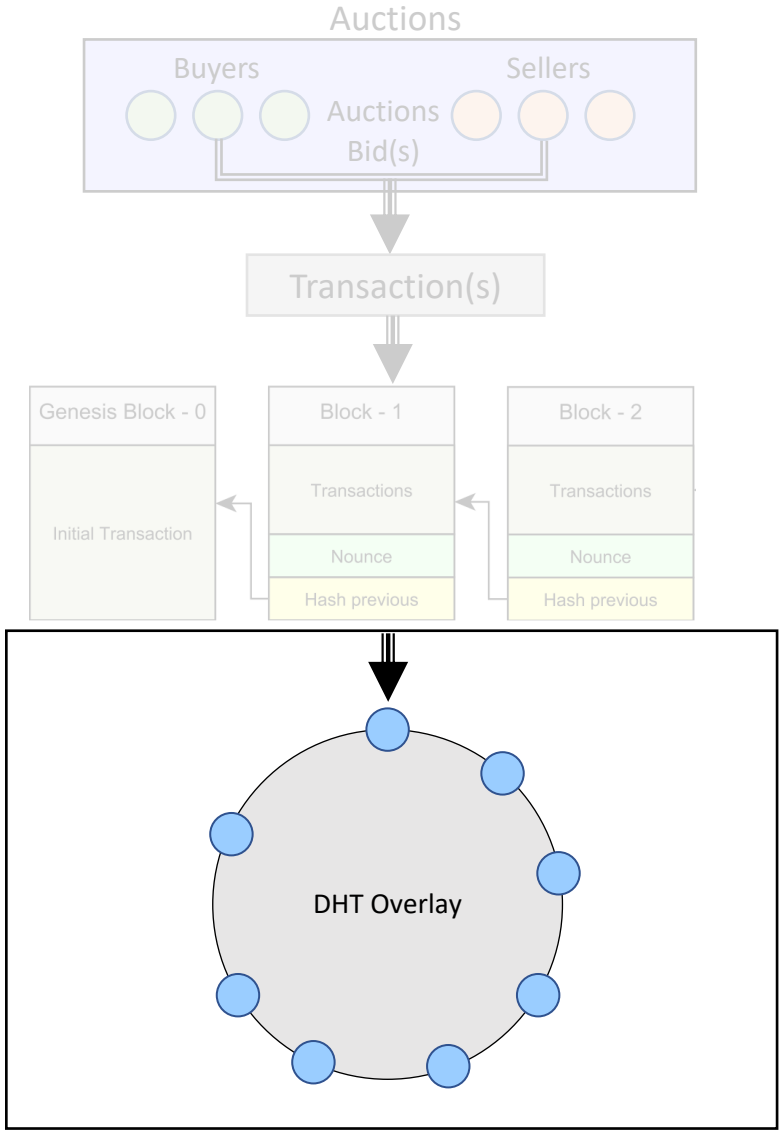
Considerations

- You can only re-use low level libraries
  - Netty for communications
  - BouncyCastle for crypto
  - **Cannot reuse existing** open-source **projects**
- Follow a secure-by-design approach during design and development phases
  - Address identity, authorization, authentication, access control, trust/authoritative domains, secure communication, etc.
  - On all layers
- The codebase must be hosted in a private repository in Github or Bitbucket



Secure P2P Overlay

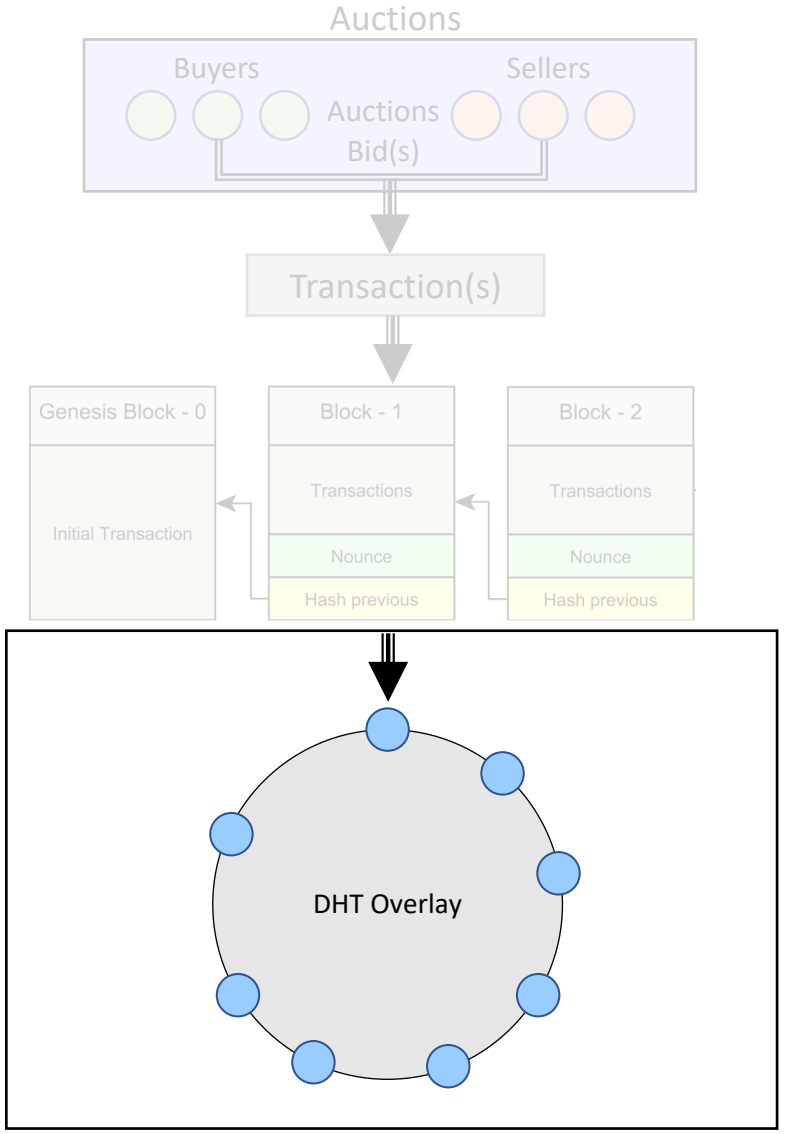
- Used to store data
  - Based on the Kademlia DHT
    - PING
    - STORE
    - FIND_NODE
    - FIND_VALUE
- Nodes can dynamically enter and leave the system
  - Requires managing node membership to know the nodes of the overlay



Secure P2P Overlay

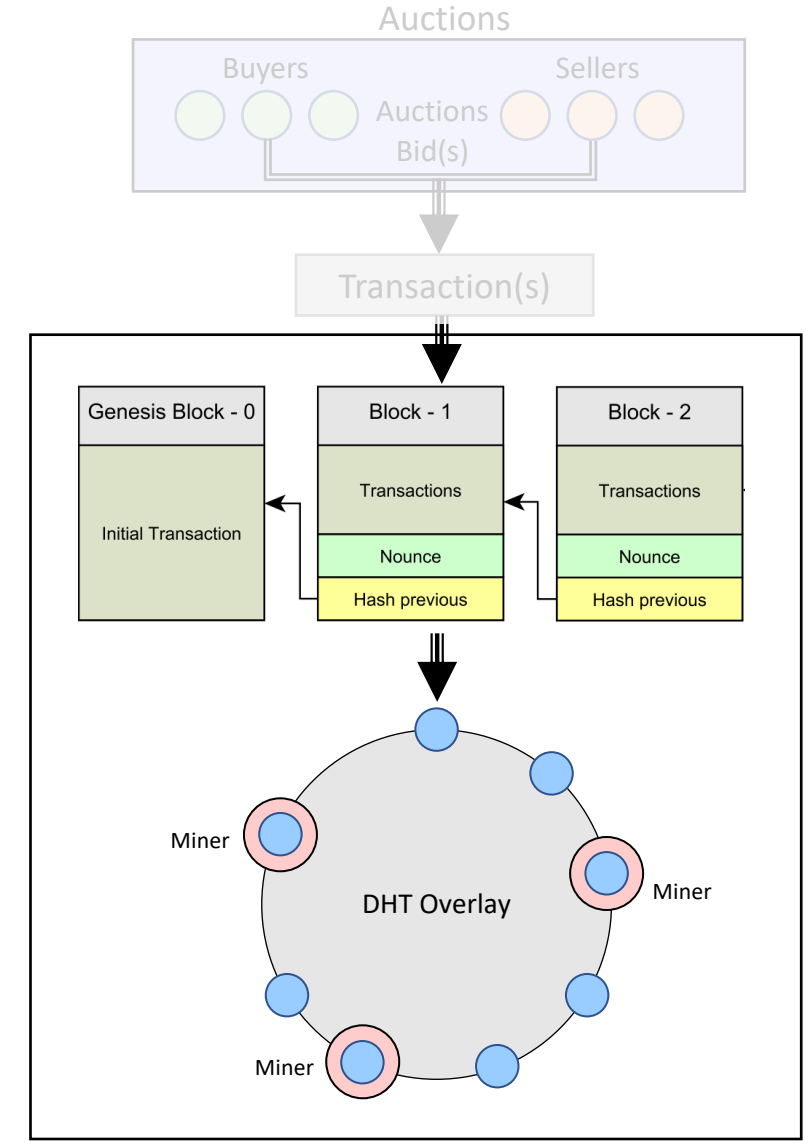
- Security aspects
  - How to prevent Sybil and Eclipse attacks
    - How to prevent nodes from having multiple IDs?
    - How to prevent nodes from impersonating other nodes?
    - How to prevent loss of information when nodes leave?
  - What if a node does not follow protocol (malicious/byzantine behavior)?

- To support the overlying ledger
  - What modification (added information) is required to maintain the ledger?



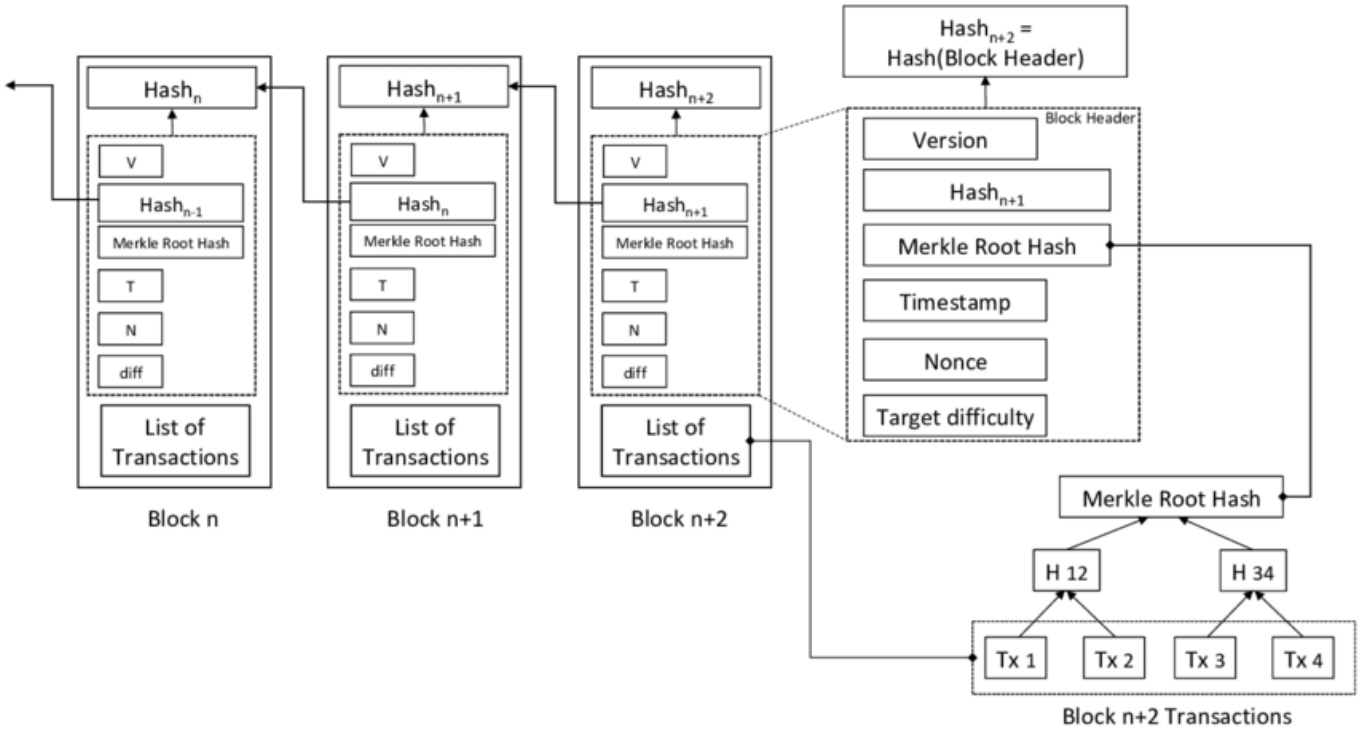
Distributed Ledger

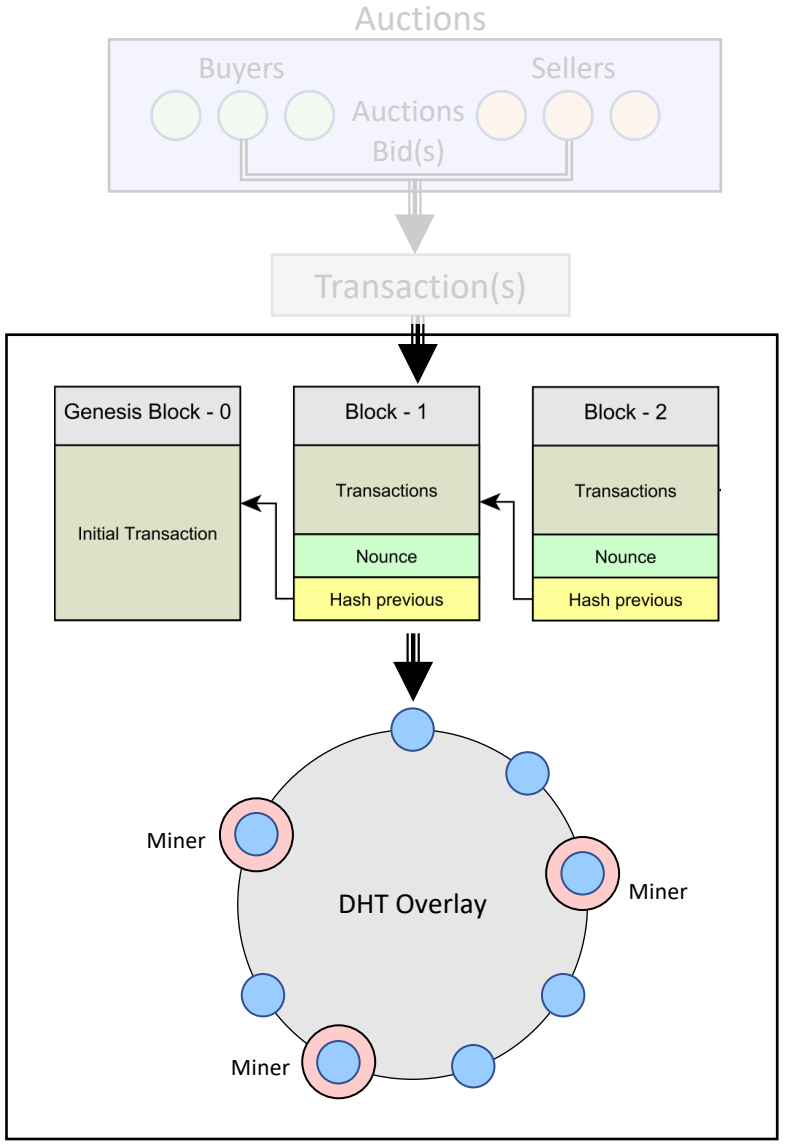
- A distributed ledger can be seen as an **append only immutable list of records**
  - Used to maintain the application's state/information
  - Records are added to the end of the list
- Allows adding new records and consulting previous ones
- Records are grouped into blocks to increase performance
  - Forming of a blockchain



Distributed Ledger

- Uses a PoW scheme for “mining” blocks
  - The first miner to solve the problem wins
  - Determining the order of the block

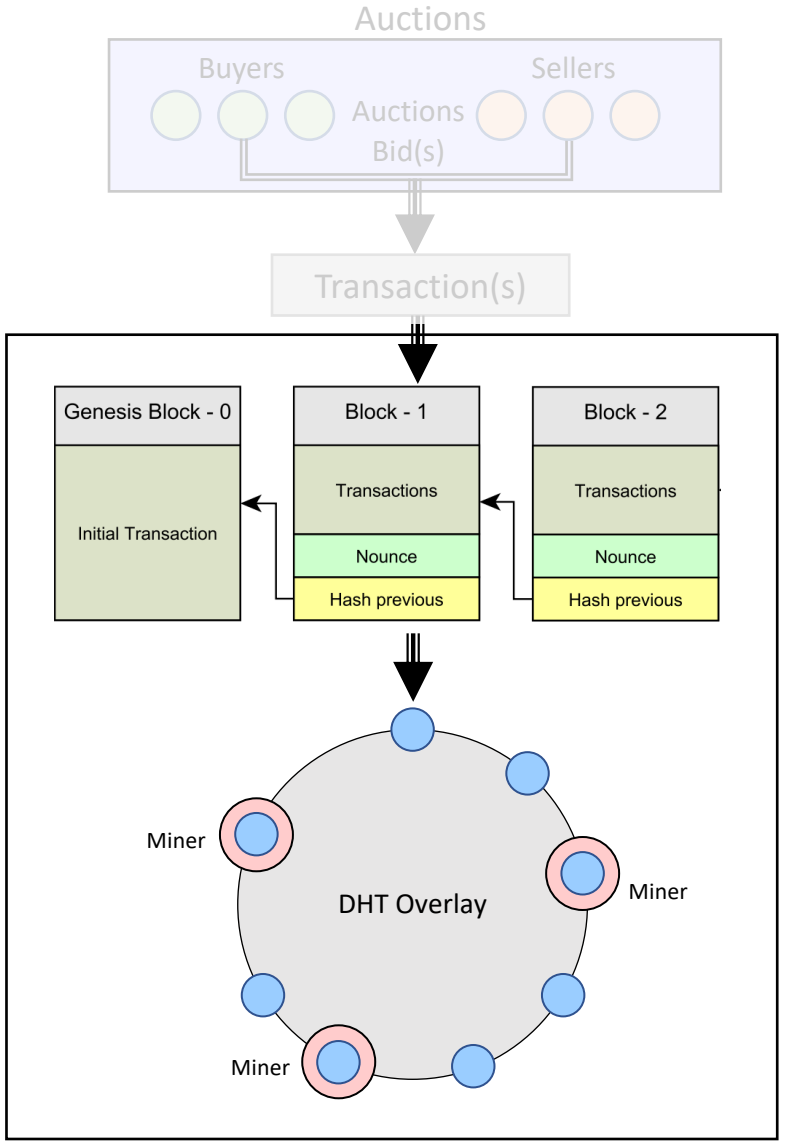




Distributed Ledger

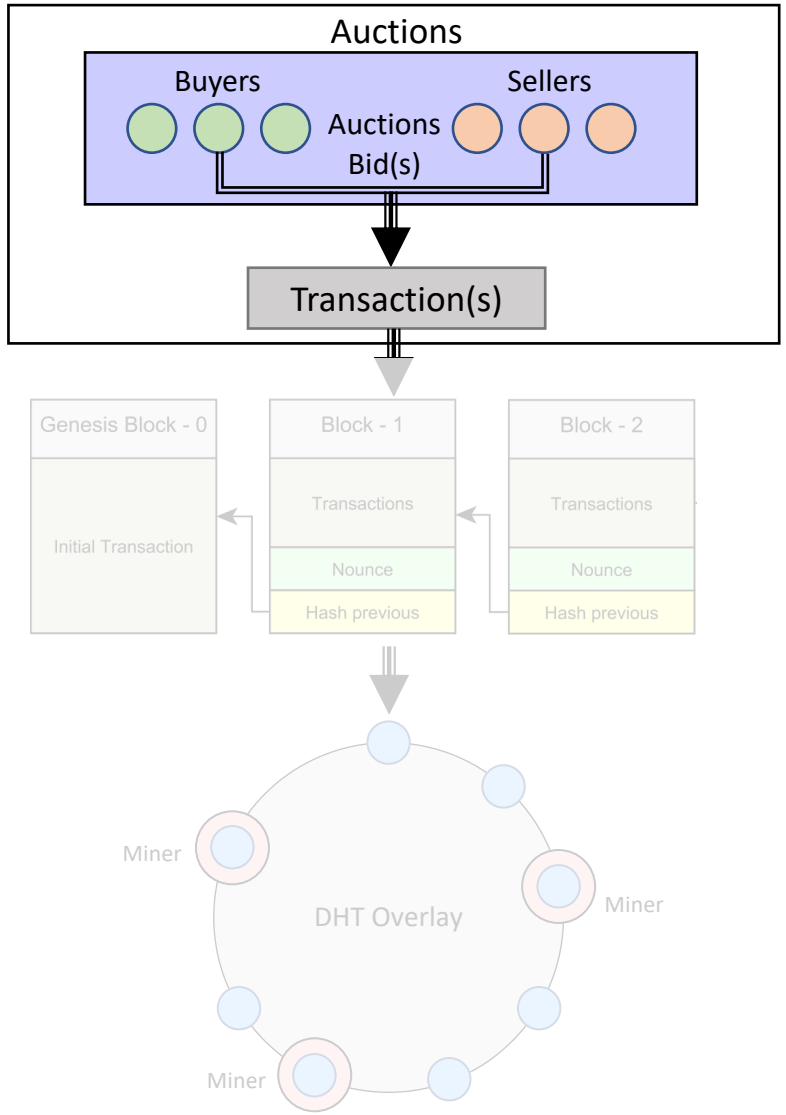
- Storing the ledger and mining blocks are the responsibility of the Blockchain nodes

- How to manages nodes and membership?
  - Do all nodes have the same function?
  - How to differentiate nodes?
- How do we secure the “ledger”?
  - Prevent ledger from being forged
  - Deal with concurrent updates
  - Prevent the same transaction from occurring multiple times
  - Guarantee that written transactions are “real"
    - i.e., were submitted by a client



Auction Application

- Capable of supporting **sellers** and **buyers**
- Based on a **publisher/subscriber approach**
  - Built on top of Kademlia
  - Auctions are published through the networks
  - Buyers subscribe to interested auctions
  - Buyers place bid in zero or more auctions
- Transactions (auctions & biddings) are stored persistently in the ledger



Auction Application

- How do we prevent a transaction from occurring multiple times in the ledger?
- How do we guarantee that all transactions are eventually written in the ledger?
- How do we detect/deal with malicious actors trying to compromise the system?