



BLOCKCHAIN TECHNOLOGY

Ethereum 101




Introduction

- Vitalik Buterin conceptualized Ethereum in 2013.
- The critical idea was
 - the development of a Turing-complete language that allows the development of arbitrary programs (smart contracts) for blockchain and decentralized applications.
- This concept contrasts with Bitcoin
 - the scripting language is limited in nature and allows necessary operations only




The yellow paper

- The Ethereum yellow paper
 - available at <https://ethereum.github.io/yellowpaper/paper.pdf>
 - serves as a formal definition of the Ethereum protocol
 - Anyone can implement an Ethereum client based on specifications defined in the paper
- 




The Ethereum network

- a peer-to-peer network
 - nodes participate in order to
 - maintain the blockchain
 - contribute to the consensus mechanism
 - Networks can be divided into three types
 - Mainnet
 - Testnet
 - Private net
- 




The Ethereum network

- Mainnet
 - the current live network of Ethereum
 - chain ID is 1
 - is used to identify the network
 - Testnet
 - also called Ropsten
 - Allows experimentation and research
 - is used to test smart contracts and DApps before being deployed mainnet.
 - The main testnet is called Ropsten
 - contains all features of other smaller and special purpose testnets
 - Kovan and Rinkeby
- 

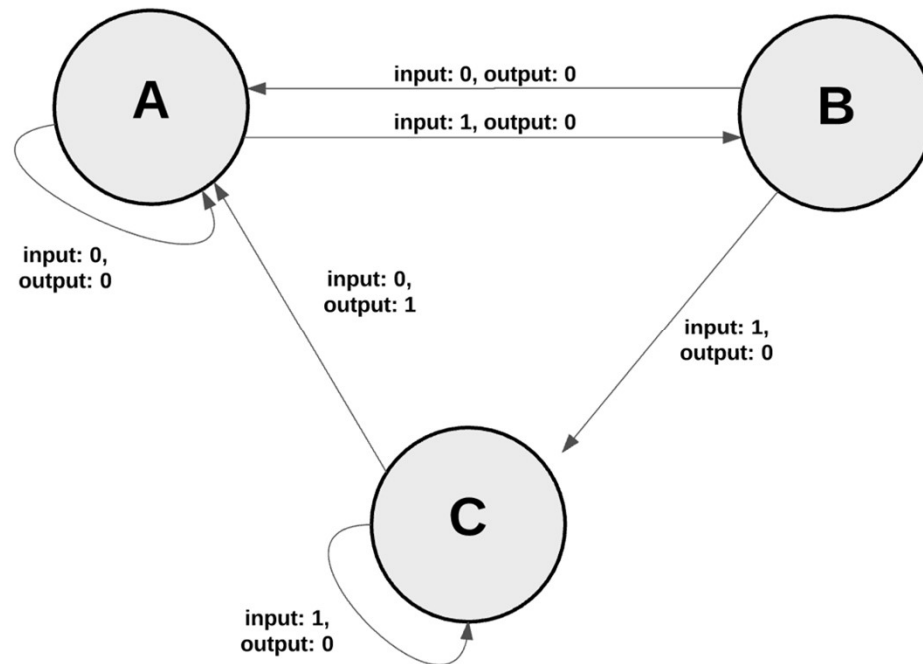


The Ethereum network

- Private net
 - the private network
 - can be created by generating a new genesis block
 - a private group of entities start their blockchain and use it as a permissioned blockchain.
- 

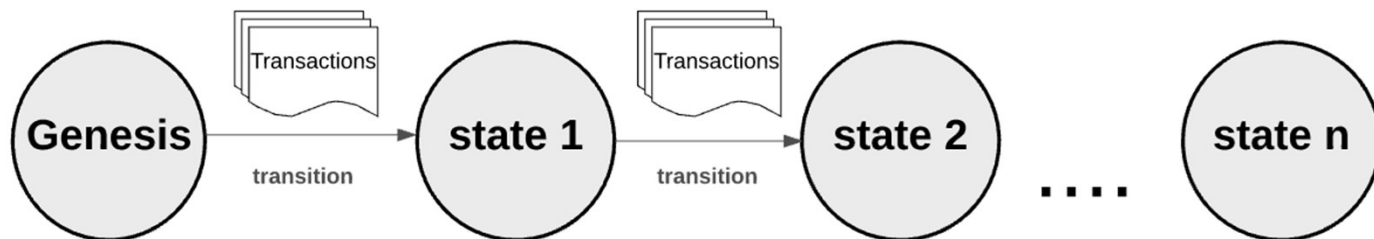
The Ethereum blockchain paradigm

- a **transaction-based state machine.**
 - will read a series of inputs
 - transition to a new state based on those inputs



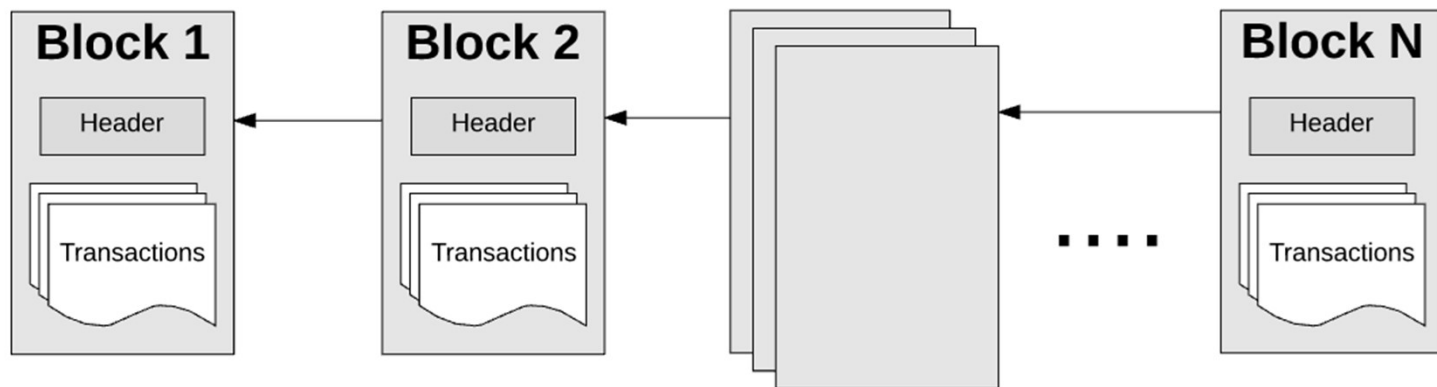
The Ethereum blockchain paradigm

- a **transaction-based state machine.**
 - we begin with a “genesis state.”
 - analogous to a blank slate
 - before any transactions have happened on the network.
 - When transactions are executed
 - the genesis state transitions into some final state
 - this final state represents the current state of Ethereum
 - At any point in time



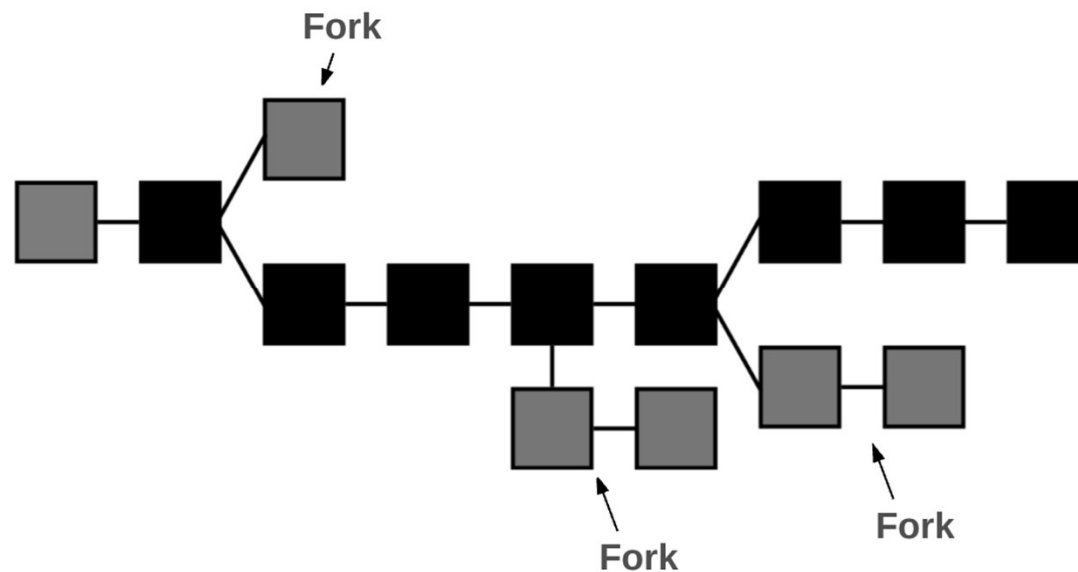
The Ethereum blockchain paradigm

- a **transaction-based state machine**.
 - state of Ethereum has millions of transactions.
 - These transactions are grouped into “blocks.”
 - contains a series of transactions
 - each block is chained together with its previous block



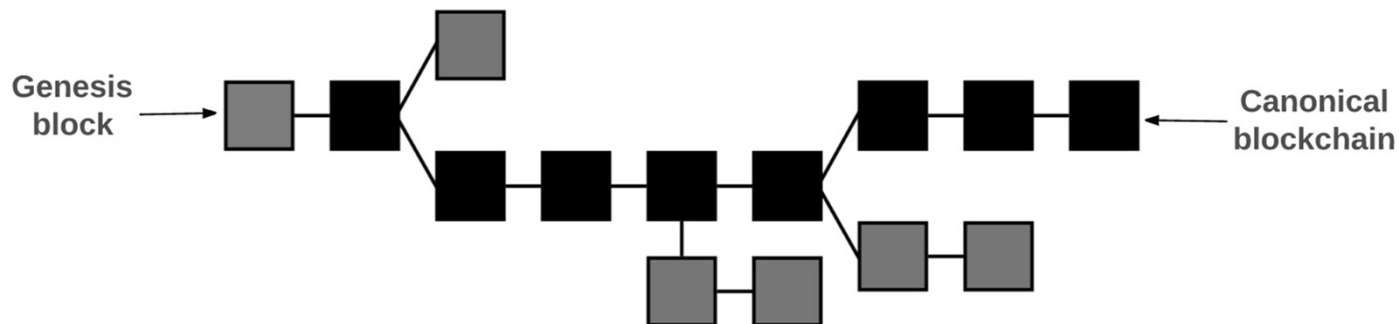
The Ethereum blockchain paradigm

- a “fork” occurs whenever multiple paths are generated



The Ethereum blockchain paradigm

- a “fork” occurs whenever multiple paths are generated
 - GHOST protocol (Greedy Heaviest Observed Subtree)
 - pick the path with the most computation done upon it
 - use the block number of the most recent block
 - The higher the block number
 - the longer the path
 - the greater the mining effort






Ethereum main components

- keys and addresses
- accounts
- state
- gas and fees
- transactions
- blocks
- transaction execution
- mining
- proof of work




Keys and addresses

- Keys and addresses are used mainly to represent ownership and transfer of Ether.
 - Keys are used in pairs of private and public type.
 - The private key is generated randomly and is kept secret
 - The public key is derived from the private key
 - Addresses are derived from the public keys
 - are a 20-bytes code used to identify accounts
- 




Keys and addresses

- The process of key generation and address derivation
 1. a private key is randomly chosen
 - 256 bits positive integer under the rules defined by elliptic curve secp256k1 specification
 2. The public key is then derived from this private key
 - using ECDSA function
 3. An address is derived from the public key
 - the right most 160 bits of the Keccak hash of the public key.
- 



Ethereum main components

- Accounts
 - The global “shared-state” of Ethereum is comprised of many small objects (“accounts”)
 - can interact with one another
 - through a message-passing framework
 - each has
 - a state associated with
 - a 20-byte address
- 




Ethereum main components

- Accounts

- There are two types of accounts:
 - Externally owned accounts (EOA)
 - are controlled by private keys
 - have no code associated with them
 - Contract accounts (CA)
 - are controlled by their contract code
 - have code associated with them.



Ethereum main components

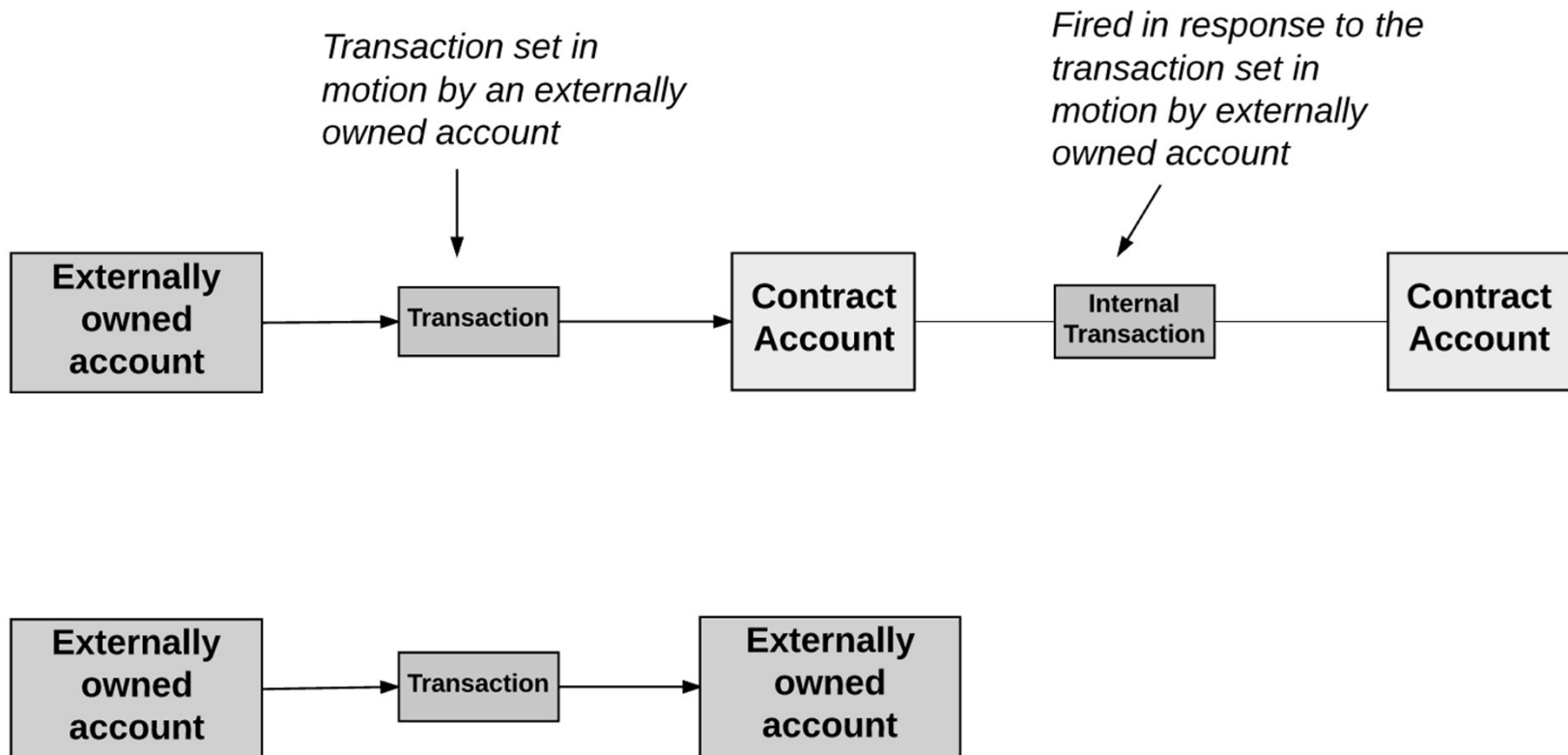
- Externally owned accounts
 - can send messages to
 - other EOAs
 - or to other CAs
 - by creating and signing a transaction using its private key
 - A message between two EOAs
 - is simply a value transfer
 - A message from an EOA to a CA
 - activates the CA's code
 - allowing it to perform various actions
 - transfer tokens, write to internal storage, mint new tokens, perform some calculation, create new contracts, etc.
- 



Ethereum main components

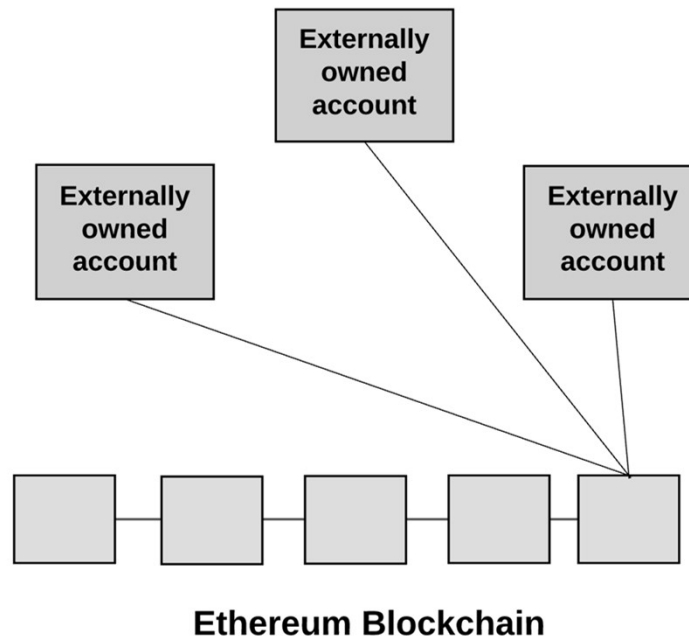
- Contract accounts
 - can't initiate new transactions on their own.
 - can only fire transactions in response to other transactions they have received
 - from an EOA
 - or from another CA

Ethereum main components



Ethereum main components

- Any action that occurs on the Ethereum blockchain
 - is always set in motion by transactions fired from EOAs.



Ethereum main components

- Account state
 - consists of four components (regardless of account type):
 - Nonce
 - represents the number of transactions sent for EOAs
 - represents the number of contracts created by the account for CAs.
 - Balance
 - The number of Wei owned by this address
 - There are $1e+18$ Wei per Ether
 - StorageRoot
 - A hash of the root node of a Merkle Patricia tree
 - we'll explain Merkle trees later on
 - CodeHash
 - The hash of the EVM (Ethereum Virtual Machine—more on this later) code of this account.
 - For CAs
 - this is the code that gets hashed and stored as the codeHash
 - For EOAs
 - the codeHash field is the hash of the empty string.




Ethereum main components

- World state
 - global state consists of a mapping between account addresses and the account states
 - is stored in a data structure known as a Merkle Patricia tree.
 - the combination of a:
 - Patricia Trie: a data structure in which “keys” represent the path to reach a node
 - Merkle Tree: A hash tree in which each node’s hash is computed from its child nodes hashes.



Ethereum main components

- Gas and payment
 - Every computation that occurs as a result of a transaction incurs a fee
 - paid in a denomination called “gas.”
 - Gas
 - is the unit used to measure the fees
 - Gas price is the amount of Ether you are willing to spend on every unit of gas
 - is measured in “gwei.”
 - “Wei” is the smallest unit of Ether
 - 10^{18} Wei represents 1 Ether
 - One gwei is 1,000,000,000 Wei.
- 

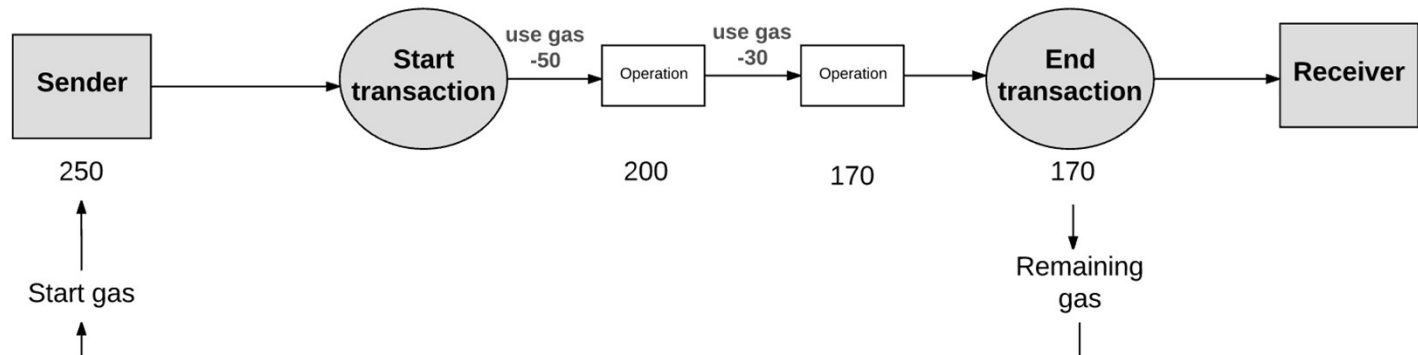
Ethereum main components

- Gas and payment
 - for every transaction
 - sender sets a gas limit and gas price
 - The product represents the maximum amount of Wei that the sender is willing to pay for transaction.

| | | | | |
|---------------------|---|----------------------|---|------------------------------------|
| Gas Limit 50,000 | x | Gas Price 20 gwei | = | Max transaction fee 0.001 Ether |
|---------------------|---|----------------------|---|------------------------------------|

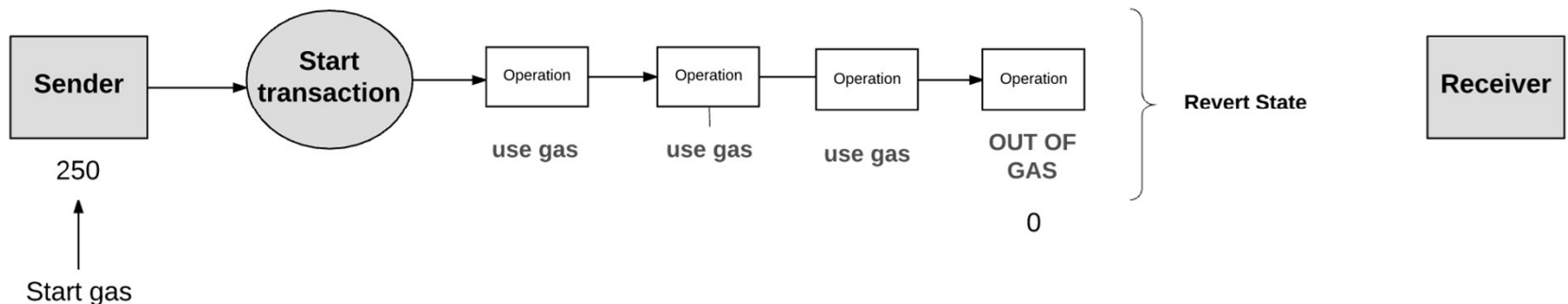
Ethereum main components

- Gas and payment
 - sender is refunded for any unused gas at the end of the transaction



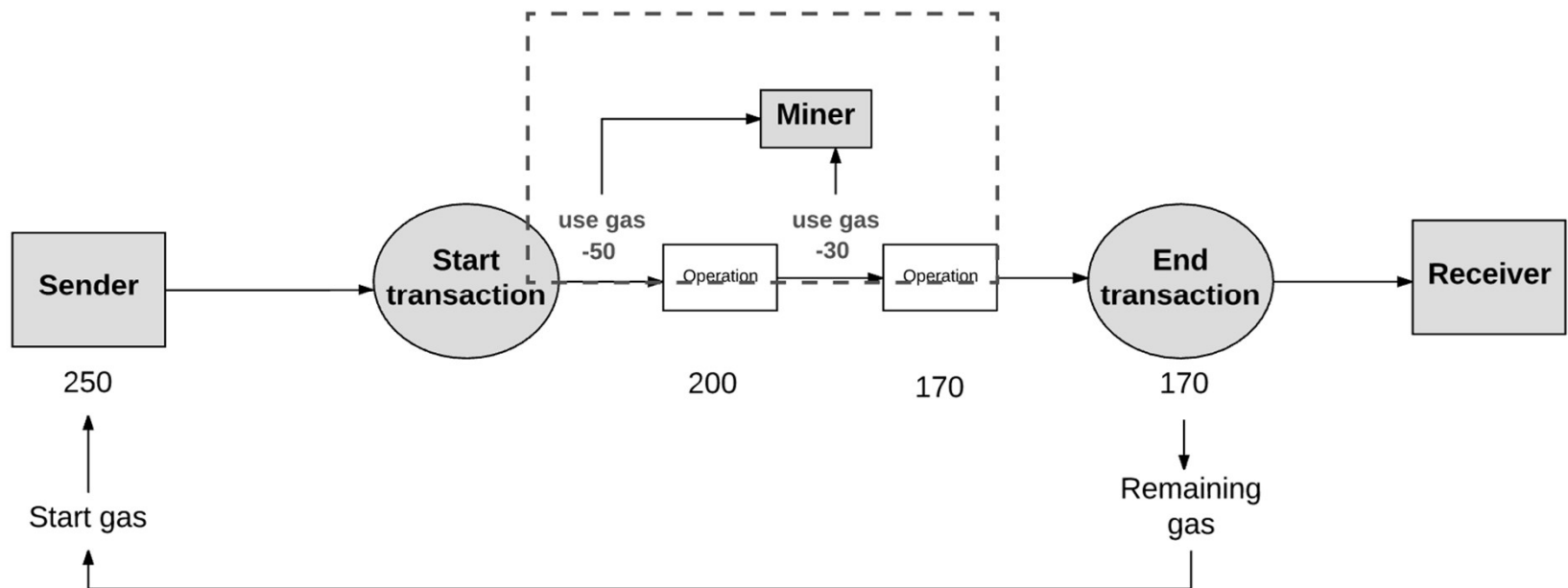
Ethereum main components

- Gas and payment
 - If sender does not provide the necessary gas
 - transaction
 - runs “out of gas”
 - is considered invalid.
 - transaction processing aborts
 - any state changes that occurred are reversed
 - a record of the transaction failing gets recorded
 - none of the gas is refunded to the sender
 - because the machine already expended effort to run the calculations



Ethereum main components

- Gas and payment
 - All the money spent on gas by the sender is sent to the miner's address






Ethereum main components

- Gas and payment
 - There are fees for storage, too
 - proportional to the smallest multiple of 32 bytes used
 - if a transaction has a step that clears an entry in the storage
 - the fee for executing that operation is ignored
 - a refund is given for freeing up storage space





Ethereum main components

- Gas and payment
 - What's the purpose of fees?
 - are incentives for miners
 - prevent users from overwhelming the network
 - protect the network from deliberate attacks
 - E.g., infinite loop in transaction
- 



Ethereum main components

- Transaction and messages
 - a transaction is a cryptographically signed piece of instruction that
 - is generated by an externally owned account
 - serialized
 - and then submitted to the blockchain
 - There are two types of transactions
 - message calls
 - contract creations

Ethereum main components

- Transaction and messages
 - All transactions contain the following components
 - nonce: a count of the number of transactions sent by the sender.
 - gasPrice: number of Wei that the sender is willing to pay per unit of gas.
 - gasLimit: max amount of gas that the sender is willing to pay for transaction
 - to: the address of the recipient.
 - empty value for contract-creating transaction.
 - value: the amount of Wei to be transferred from the sender to the recipient.
 - serves as the starting balance for contract-creating transaction
 - v, r, s: used to generate the signature that identifies the sender of the transaction.
 - Init (only exists for contract-creating transactions):
 - a code fragment that is used to initialize the new contract account
 - is run only once, and then is discarded
 - returns the body of the account code
 - the piece of code that is permanently associated with the contract account.
 - data (optional field that only exists for message calls):
 - the input data (i.e. parameters) of the message call.

Ethereum main components

- Blocks
 - block header consists of:
 - parentHash: a hash of the parent block's header
 - ommersHash:
 - a hash of the current block's list of ommers
 - blocks whose parent is equal to the current block's parent's parent
 - block times are much lower (~15 seconds) than bitcoin
 - enables faster transaction processing
 - downside is that more competing block solutions are found by miners.
 - also referred to as “orphaned blocks” (i.e. mined blocks do not make it into the main chain).
 - ommers is to help reward miners for including orphaned blocks.
 - beneficiary
 - the account address that receives the fees for mining this block
 - stateRoot: the hash of the root node of the state trie
 - transactionsRoot: the hash of the root node of the trie that contains all transactions listed in this block



Ethereum main components

- **Blocks**

- block header consists of:

- receiptsRoot: the hash of the root node of the trie that contains the receipts of all transactions listed in this block
 - Transaction Receipts record the transaction outcome
 - E.g., new contract's contractAddress
 - logsBloom: a Bloom filter (data structure) that consists of log information
 - difficulty: the difficulty level of this block
 - number: the count of current block
 - gasLimit: the current gas limit per block



Ethereum main components

- Blocks

- block header consists of:

- gasUsed: the sum of the total gas used by transactions in this block
 - timestamp: the unix timestamp of this block's inception
 - extraData: extra data related to this block
 - mixHash: a hash that, when combined with the nonce, proves that this block has carried out enough computation
 - nonce: a hash that, when combined with the mixHash, proves that this block has carried out enough computation




Ethereum main components

- Transaction Execution
 - initial set of requirements for transactions
 - must be a properly formatted RLP (Recursive Length Prefix)
 - a data format used to encode nested arrays of binary data
 - Valid transaction signature.
 - Valid transaction nonce
 - gas limit must be equal to or greater than the intrinsic gas
 - The intrinsic gas includes:
 1. a predefined cost of 21,000 gas for executing the transaction
 2. a gas fee for data sent with the transaction
 3. an additional 32,000 gas for contract-creating transactions
 4. gas cost of each operation performed by transaction




Ethereum main components

- Transaction Execution
 - If the transaction meets all requirements for validity
 - transaction starts executing
 - various computations required by the transaction are processed
 - the sender is refunded with unused gas
 - the Ether for the gas is given to the miner
 - we're left with the new state and a set of the logs created by the transaction
- 



Ethereum main components

- Ethereum Virtual Machine (EVM)
 - The part of the protocol that actually handles processing the transactions
 - is a Turing complete virtual machine
 - The only limitation is that the EVM is intrinsically bound by gas
 - the total amount of computation is limited by the amount of gas provided.
 - has a stack-based architecture
 - each stack item is 256-bit
 - the stack has a maximum size of 1024
 - has volatile memory
 - also has non-volatile storage
- 



Ethereum main components

- Ethereum Virtual Machine (EVM)
 - also has its own language: “EVM bytecode.”
 - smart contracts typically written in a higher-level language such as Solidity

