Blockchain Honor Degree Sem VII

HBCC701 : Blockchain Development

Module - 1 : Ethereum Ecosystem (4 Hours)

Instructor : Mrs. Lifna C S

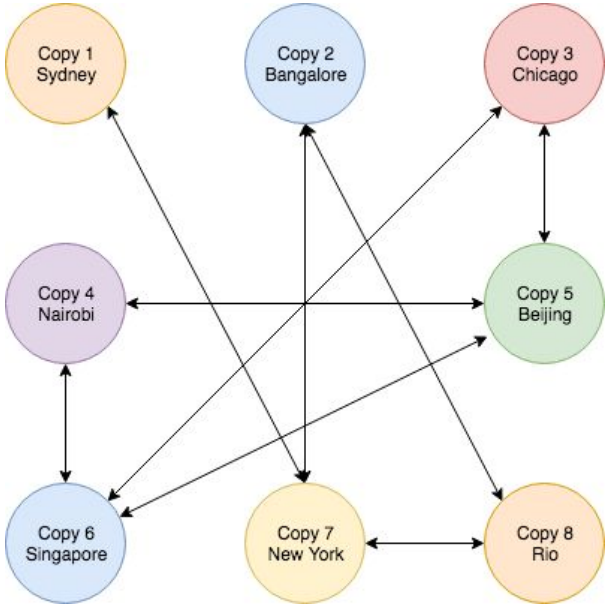
Topics to be covered

- What is Ethereum ?
- History of Ethereum
- Features of Ethereum
- Real-World Applications of Ethereum
- Ethereum Vs Bitcoin
- **Ethereum components: Node - miner and mining node, Ethereum virtual machine, Ether, Gas, accounts, Ethash, Transactions, swarm and whisper**
- **End to end transaction in Ethereum,**
- **Architecture of Ethereum.**
- How does Ethereum work ?
- Benefits of Ethereum
- Drawbacks of Ethereum

Self-learning Topics: Emerging blockchain platforms

What is Ethereum?

- a **public, blockchain based distributed computing platform.**
- as **one big computer** made up of small computers around the world.
- Eg: One can write applications and run them on this global computer.
- The platform **guarantees that your application will always run without any downtime, censorship, fraud or third-party interference.**
- Ethereum blockchain can **also transfer money between 2 parties without a central authority**



What is Ethereum?

- A Blockchain network that introduced a built-in **Turing-complete programming language** that can be **used for creating** various decentralized applications(also called **Dapps**).
- Ethereum network is fueled by its own **cryptocurrency called 'ether'.**
- Allow the implementation of **smart contracts.** Smart contracts can be thought of as 'cryptographic bank lockers' which contain certain values.
- These cryptographic lockers can only be unlocked when certain conditions are met.
- Ethereum is a network that can be applied to various other sectors.
- called **Blockchain 2.0** → it proved that blockchain can be used beyond the financial sector.
- The consensus mechanism used in Ethereum is **Proof of Stakes(PoS)**, which is **more energy efficient** than, Proof of Work(PoW).
- PoS depends on the **amount of stake a node holds**.

History of Ethereum

2013

- Ethereum was **first described in Vitalik Buterin's white paper**
- Goal of **developing decentralized applications.**

2014:

- **EVM was specified in a paper by Gavin Wood**, and the formal development of the software also began.

2015:

- Ethereum **created its genesis block** marking the **official launch of the platform.**

2018:

- Ethereum **took second place** in Bitcoin **in terms of market capitalization.**

2021:

- a major network upgrade named **London included Ethereum improvement proposal 1559**
- **introduced a mechanism** for **reducing transaction fee volatility.**

2022:

- Ethereum has **shifted from PoW(Proof-of-Work) to PoS(Proof-of-State)**
- Also known as **Ethereum Merge**.
- It has **reduced Ethereum's energy consumption by ~ 99.95%.**

Features of Ethereum

1. Smart contracts:
   a. Ethereum allows the creation and deployment of smart contracts.
   b. Smart contracts are created mainly using a programming language called **solidity.**
2. Ethereum Virtual Machine (EVM):
   a. **a runtime environment for compiling and deploying Ethereum-based smart contracts.**
3. Ether:
   a. **cryptocurrency of the Ethereum network**.
   b. **only acceptable form of payment for transaction fees** on the Ethereum network.
4. Decentralized applications (Daaps):
   a. **has its backend code running on a decentralized peer-to-peer network**.
   b. Has a frontend and UI to make calls and query data from its backend.
   c. They **operate on Ethereum** and **perform the same function irrespective of the environment** in which they get executed.
5. Decentralized autonomous organizations (DAOs):
   a. **works in a democratic and decentralized** fashion.
   b. **relies on smart contracts for decision-making** within the organization.

Real-World Applications of Ethereum

1. **Voting**:
   a. Voting systems are **adopting Ethereum.**
   b. The **results of polls are available publicly, ensuring a transparent fair system thus eliminating voting malpractices.**
2. **Agreements**:
   a. With Ethereum smart contracts, agreements and contracts **can be maintained and executed without any alteration.**
   b. Ethereum can be **used for creating smart contracts and for digitally recording transactions based on them.**
3. **Banking systems**:
   a. **Due to the decentralized nature** of the Ethereum blockchain it becomes **challenging for hackers to gain unauthorized access to the network**.
   b. **Makes payments on the Ethereum network secure**
4. **Shipping**:
   a. **Ethereum provides a tracking framework** that helps with the **tracking of cargo and prevents goods from being misplaced.**
5. **Crowdfunding**:
   a. **helps to increase trust and information symmetry**.
   b. It **creates many possibilities for startups by raising funds** to create their own digital cryptocurrency.

Ethereum Vs Bitcoin

Basis	Bitcoin	Ethereum
Definition	Bitcoin (abbreviation: BTC; sign: ₿) is a decentralized digital currency that can be transferred on the peer-to-peer bitcoin network.	Ethereum is a decentralized global software platform powered by blockchain technology. It is most commonly known for its native cryptocurrency, ether (ETH).
History	The word bitcoin was defined in a white paper published on 31 October 2008. The currency began use in 2009.	Ethereum was conceived in 2013 by programmer Vitalik Buterin, and then went live on 30 July 2015.
Purpose	The purpose of bitcoin was to replace national currencies during the financial crisis of 2008.	The purpose of Ethereum was to utilize blockchain technology for maintaining a decentralized payment network and storing computer code.

Ethereum Vs Bitcoin

Basis	Bitcoin	Ethereum
Smart Contracts	Although bitcoin do have smart contracts, they are not as flexible or complete as Ethereum smart contracts. Smart contracts in Bitcoin does not have all the functionality that a programming language would give them.	Ethereum allows us to create smart contracts. Smart contracts are computer codes that is stored on a blockchain and executed when the predetermined terms and conditions are met.
Smart Contract Programming Language	Smart contracts on Bitcoin are written in programming languages like Script, Clarity.	Smart contracts on Ethereum are written in programming languages like Solidity, Vyper, etc.
Transactions	Generally, bitcoin transactions are only for keeping notes.	Ethereum transactions may contain some executable code.

Ethereum Vs Bitcoin

Basis	Bitcoin	Ethereum
Hash Algorithm	Bitcoin runs on the **SHA-256** hash algorithm.	Ethereum runs on the **Keccak-256** hash algorithm.
Consensus Mechanism	The Proof-of-Work (PoW) is the consensus mechanism used by the Bitcoin network.	The Proof-of-Stake is the consensus mechanism used by Ethereum.
Block Time	The block time of bitcoin is 10 minutes.	The block time of Ethereum is 14 to 15 seconds.
Block Limit	The bitcoin blockchain has a block limit of 1 MB.	The Ethereum blockchain does not have a block limit.

Ethereum Vs Bitcoin

Basis	Bitcoin	Ethereum
Popularity	Bitcoin is the most popular digital currency in the market to date.	Ether, native currency of Ethereum is the second-largest cryptocurrency after bitcoin to date.
Energy Consumption	Energy consumption is very high.	Energy consumption is very low as compared to bitcoin
Energy Consumption rate	Energy consumption rate of bitcoin mining system 3.2 Million household.	Energy consumption rate of bitcoin mining system 1.2 Million household.
Structure	Structure of bitcoin is simple and robust.	Structure of Ethereum is complex and feature rich

Ethereum Vs Bitcoin

Basis	Bitcoin	Ethereum
Rewards	Miner got nearly 6.25 BTC on successfully adding new block in network.	Miner got nearly 5 BTC along with same additional rewards on successfully adding new block in network.
Assets	Assets of Bitcoin is BTC.	Assets of Ethereum is Ether.

Components of Ethereum Network

1. Ethereum Node
2. Ethereum Client
3. Ether
4. Gas
5. Ethereum Accounts
6. Nonce
7. Storage Root
8. Ethash
9. Transactions
10. Ethereum Virtual Machine (EVM)

Components of Ethereum Network - (1) Ethereum Node

- a **computer that is running the software client**.
- Nodes **communicate with one another in order to validate transactions and record data** about the status of the blockchain.
- **Responsible for storing, validating, and trading data.**
- Each node keeps its **own copy of the blockchain** and **strives to verify** that it matches the copies of all the other nodes.
- Every node on the network **must process any action that requires a new block to be added** to the blockchain.
- This network of continually communicating nodes **allows us to avoid relying on a single source of truth and all of the challenges it entails.**
- A new block is **added based on whether or not the majority of nodes accept it.**

Components of Ethereum Network - (1) Ethereum Node (Types)

1. **Full Node**:
   - verify and validate each and every transaction that takes place inside the network
   - **maintain the state / a full copy** of the blockchain.
   - When a smart contract transaction occurs,
     - Full nodes also **execute all of the instructions in the smart contract**.
     - It determines whether or not the smart contract execution is **producing the expected results.**
   - It **keeps receiving copies of the entire blockchain** including its transactions which are stored locally and keeps the latest state of transaction with itself.
   - Eg: **Person A performs a transaction to person B**,
     - transaction is added to the blockchain,
     - Full nodes verify whether the transaction complies with all the Ethereum specifications,
     - Maintain the latest state of the blockchain by storing or removing the specification if it does not comply.
   - **Example of a discarder transaction** is when a person transfers X ETH to another person but their account contains less ETH.

Components of Ethereum Network - (1) Ethereum Node (Types)

2. **Archive Nodes:**
   - **complete nodes that have the "archive mode" option enabled.**
   - While a **Full Node only stores the latest state of the transaction,**
   - the Archive nodes **hold all of the blockchain's history data dating back to the genesis block**.
   - **used when blocks prior to the latest 128 blocks are required.**
   - Eg : using functions like **eth_getBalance** of a historic address would require an archive node, as will interacting with smart contracts launched far earlier in the blockchain.
   - Archive Nodes **memory requirement** :
     - **require more than 6 Terabytes of space**, contrary to Full node which only requires a little over 500 Gigabytes of disk space.
     - **are not useful for average people**,
     - they are **effective in the application of block exploring, wallet vending, and chain analytics**.

Components of Ethereum Network - (1) Ethereum Node (Types)

3. **Light Nodes:**

- does not hold the complete current blockchain state
- stores only the block header.
- suitable for low memory and computational devices since maintaining a light node involve the least investment in hardware, running costs, and technical skill.
- Light nodes **rely on full nodes to function.**
- These nodes **do not need to run continuously or read and publish a large amount of data on the blockchain.**
- It provides **an easy way to create a wallet, especially for beginners.**
- Eg : **Solid-State Drives (SSD)** cannot afford to store the gigabytes of data that other nodes take.
- But there are some limitations of light nodes which cannot be denied, there is no guarantee that the light wallet provider will be online when it is needed.

Components of Ethereum Network - (2) Ethereum Client

- **software program** that is used **to implement the Ethereum specification**
- **connect itself with other Ethereum clients** over a peer-to-peer network.
- Different Ethereum clients can communicate with one another if they follow the reference specification and the defined communication protocols.
- These interactions among different clients in the network **take place using various programming languages**
  - like Geth (Go), OpenEthereum (Rust),
  - Nethermind (C#, .NET), Besu (Java),
  - Erigon (Go/Multi).
- The **yellow paper** is the Ethereum protocol that allows anybody to run a client to construct a node.
- **Ethereum sets standard behaviors that all Ethereum clients must adhere to**
- **Ethereum's specs** enabled the blockchain to allow for different, but interoperable, software implementations of an Ethereum client by providing standard documentation and simple language.

Components of Ethereum Network - (2) Ethereum Client (Types)

1. **Full Client:**
   - **save the complete Ethereum blockchain**,
   - which **might take several days to synchronize** and
   - takes a **massive amount of disc space** – more than 1 Terabyte, according to the most recent estimates.
   - **Enable connected nodes to conduct all network functions**, including as
     - mining,
     - transaction
     - block-header validation,
     - smart contract execution.

Components of Ethereum Network - (2) Ethereum Client (Types)

2. **Light Client:**
   - **do not always need to necessary keep all of the data,**
   - when data storage and performance are concerns, developers utilize the “light clients”.
   - Light clients **provide a portion of full client capability.**
   - they can provide quick delivery and free up data storage space.
   - The functionality of a light client is adapted to the purposes of the Ethereum client.
   - **widely used within wallets to maintain private keys and Ethereum addresses.**
   - **manage smart contract interactions and transaction broadcasts**.
   - **useful for web3 instances** within JavaScript objects, Dapp browsers
   - **obtaining the exchange rate data.**

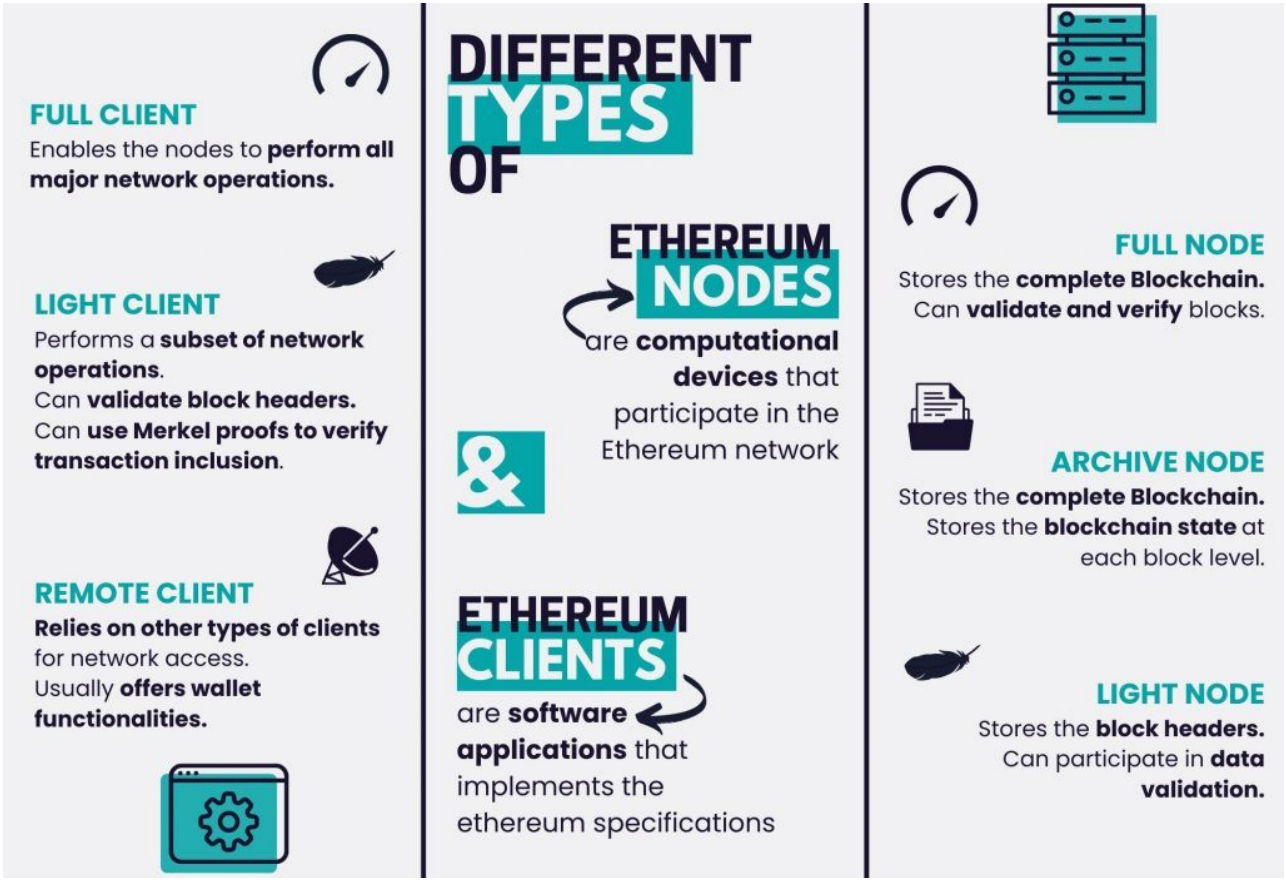
Components of Ethereum Network - (2) Ethereum Client (Types)

3. **Remote Client:**
   - A remote client is **much like a light client**.
   - The primary distinction is that a **remote client does not keep its own copy of the blockchain or validate transactions or block headers.**
   - Remote clients, on the other hand, rely entirely on a full or light client to have access to the Ethereum blockchain network.
   - These clients are **mostly used as wallets for transmitting and receiving transactions.**

Components of Ethereum Network - Node Vs Client

Ethereum Node	Ethereum Client
A machine running Ethereum client software is referred to as an “Ethereum Node”	A client is an Ethereum implementation that validates all transactions in each block, ensuring the network’s security and data accuracy
The three types of Ethereum Nodes are Full, Light, Archive, and Miner Nodes.	The three types of Ethereum Clients are Full, Light, and Remote Clients
The Ethereum node operating system allows us to access the internet	The Ethereum client computer allows a user to access the node operating system

Components of Ethereum Network - Node Vs Client



Courtesy : Chainstack

Components of Ethereum Network -

	Pros	Cons
Light nodes	• Portable • Resource-efficient • User-friendly	• Do not validate the network • Do not propagate blocks • Do not maintain consensus • Less secure
Full nodes	• Validate the network • Propagate blocks • Maintain consensus • More secure	• Resource-heavy • Harder to maintain • Less user-friendly
Pruned	Flexible storage	Need to revalidate old blocks
Archive	Carry full history	Resource and storage heavy
Mining	• Easily trackable involvement • Can pool with others to increase reward rate	• High and wasteful energy consumption • High equipment cost and barrier to entry
Staking	• Low barrier to entry • Low energy consumption	• Reward system based on luck • Low transparency in staking pools
Masternodes	• Balanced network benefits and rewards • Lower maintenance costs	• High initial investment • Difficult setup process

Courtesy : Chainstack

Components of Ethereum Network - (3) Ether

- type of **cryptocurrency used in the Ethereum network**
- It is a **peer-to-peer currency**
- It **tracks and promotes each transaction** in the network.
- It is the **second-largest cryptocurrency in the world.**
- Other cryptocurrencies can be used to get ether tokens, but vice versa is not true.
    - It means that **ether tokens can't be interchanged by other cryptocurrencies** to render computing power for Ethereum transactions.
- paid as a commission for any execution that affects the state in Ethereum.
- **used** in the Ethereum algorithm **as an incentive for miners to blocks to the blockchain using PoW**
- only currency that can be **used to pay transaction costs**, which go to miners as well.
- Aside from paying for transactions, **ether is often used to purchase gas**, which is used to pay for the computation of any transaction on the Ethereum network.

Components of Ethereum Network - (4) Gas

- **An internal currency** of the Ethereum network.
- We need gas to run applications on the Ethereum network, much as we need gas to run a vehicle.
- To complete every transaction on the Ethereum network, a consumer must first make a payment—send out ethers—and **the intermediate monetary value is known as gas.**
- Gas is a unit of measurement on the Ethereum network **for the computing power used to execute a smart contract or a transaction.**
- The price of gas is **very low compared to Ether.**
- **The execution and resource utilization costs are predetermined in Ethereum** in terms of Gas units, called **gwei.**

Components of Ethereum Network - Ether Denominations

Value (in wei)	Exponent	Common Name	SI Name
1	1	wei	wei
1,000	10^3	babbage	kilowei or femtoether
1,000,000	10^6	lovelace	megawei or picoether
1,000,000,000	10^9	shannon	gigawei or nanoether
1,000,000,000,000	10^{12}	szabo	microether or micro
1,000,000,000,000,000	10^{15}	finney	milliether or milli
1,000,000,000,000,000,000	*10^{18}*	*ether*	*ether*
1,000,000,000,000,000,000,000	10^{21}	grand	kiloether
1,000,000,000,000,000,000,000,000	10^{24}		megaether

Courtesy : Zastrin

Components of Ethereum Network - (5) Ethereum Accounts

- similar to a bank account,
- but for ethers or ETH, where Ethereum can be held, transferred to other accounts,
- can also be **used to execute smart contracts**.
- An entity that is composed of **an Ethereum address along with a private key.**
  - The first 20 bytes of the SHA3 hashed public key is the Ethereum address.
- Ethereum has two types of accounts:
  - **Externally owned account (EOA):**
  - **Contract Account**:

Components of Ethereum Network - (5) Ethereum Accounts (Types)

Ethereum has two types of accounts:

1. **Externally owned account (EOA):**
   - **controlled by private keys**.
   - Each EOA **has a public-private key pair**.
   - The **users can send messages by creating and signing transactions**.

Advantages of EOA

1. Transactions from an external account to a contract account **can trigger code that can execute many different actions,** such as transferring tokens or even creating a new contract.
2. Externally Owned Accounts cannot list incoming transactions.

Components of Ethereum Network - (5) Ethereum Accounts (Types)

2. **Contract Account**:
   - **controlled by contract codes**.
   - These **codes are stored with the account**.
   - Each contract account **has an ether balance associated with it**.
   - The contract code of these accounts **gets activated every time a transaction from an EOA or a message from another contract is received by it**.
   - When the contract code activates, **it allows to read/write the message to the local storage, send messages and create contracts.**
   - **Types of Contract Accounts :**
     i. Simple Account: The account is *created and owned by a single account holder*.
     ii. Multisig (multisignature) Account: A Multisig Wallet *contains several owner Accounts*, one of which is also the creator Account.

Components of Ethereum Network - (5) Ethereum Accounts (Types)

Advantages of CA:

1. A contract account can list incoming transactions.
2. Contract accounts can be set up as Multisig Accounts.
3. A Multisig Account can be structured such that it has a daily limit that you specify, and only if the daily limit is exceeded will multiple signatures be required.

Disadvantages of CA:

1. Creating contract accounts costs gas because they use the valuable computational and storage resource of the network.
2. Contract accounts can't initiate new transactions on their own. Instead, contract accounts can only fire transactions in response to other transactions they have received either from an externally owned account or from another contract account.

Components of Ethereum Network - (5) Ethereum Account (EOA Vs CA)

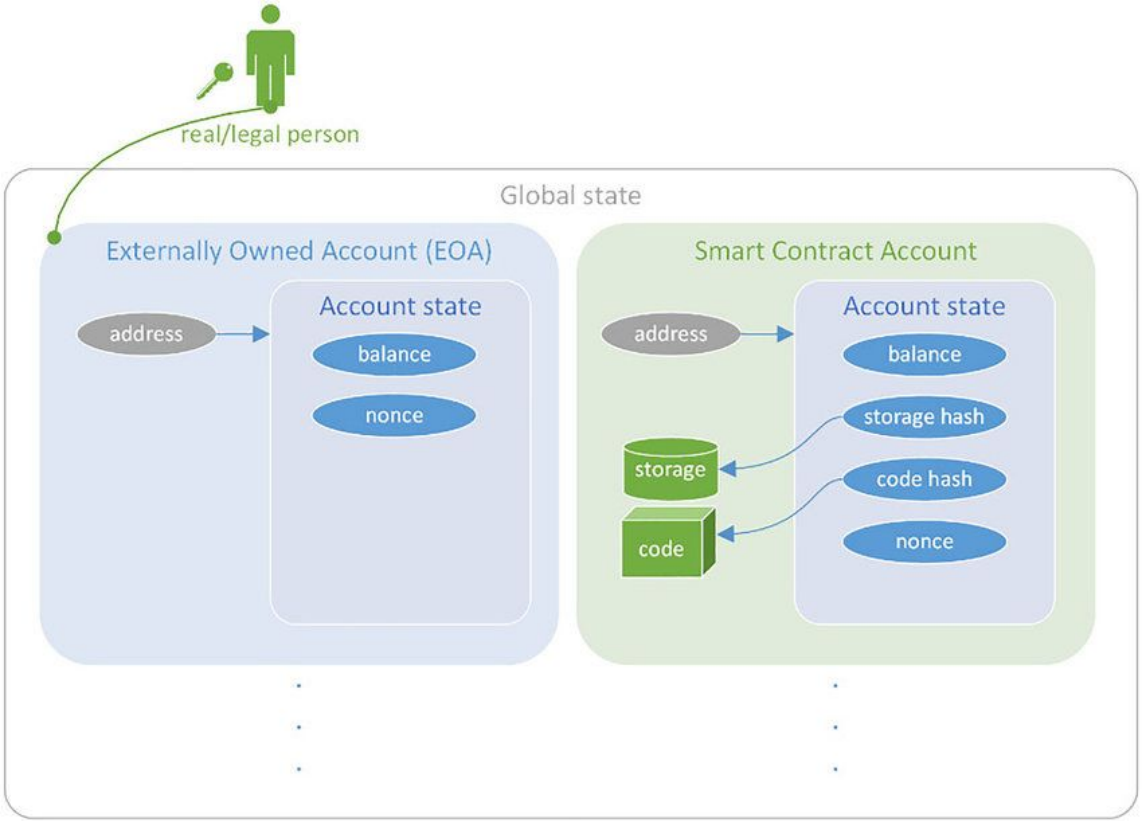
Externally Owned Accounts	Contract Account
Controlled by third party	Controlled by Contract Code
Private Key is needed to access EOAs	No key is needed to access Contract Accounts
EOAs are created automatically on creating wallet	CA require EOAs to be activated
EOA doesn’t have a code associated with it	CA have their own associated code
No execution fee is associated with EOAs	Execution fee is associated with CAs
Code hash = empty string	Code hash represents the code associated with tha account

Components of Ethereum Network - (5) Ethereum Account (Field)

Different Fields in Ethereum Accounts

1. **Nonce:**
   - The nonce in an Ethereum account indicated the number of transactions that have been sent from that account.
   - This ensures that each transaction is made only once by taking count every time it takes place.
2. **Ether Balance:** the amount of ether present in an ether repository of the current ether account.
3. **Contract Code**:
   - This is non-mandatory to fill, in case it is present since not all accounts have a contract code.
   - But note, that they cannot be altered once executed.
4. **Storage:** This field remains not filled unless mentioned.
5. **Code Hash**:
   - hash that refers to the code present in that Ethereum account
   - since no code is associated with externally owned Ethereum accounts, ⇒ **code hash = empty string.**

Components of Ethereum Network - (5) Ethereum Account (Field)



Courtesy : ResearchGate

Components of Ethereum Network - (5) Ethereum Accounts & Key Pairs

An Ethereum account is a **private-public key pair** that may be **linked to a blockchain address**.

Private Key

- It is a “owned” or “externally owned” account if the **private key is known and controlled by someone**.
- **Contract accounts** do not have a private key connected with them, although EOAs have.
- Control and access to one’s assets and contracts are granted through the EOA private key.
- The **user keeps the private key safe.**

Public Key

- Account’s **public key is open**.
- This key serves as the account’s identity.
- A one-way cryptographic function is **used to produce the public key from the private key.**

For example, if you create an account on Ethereum,

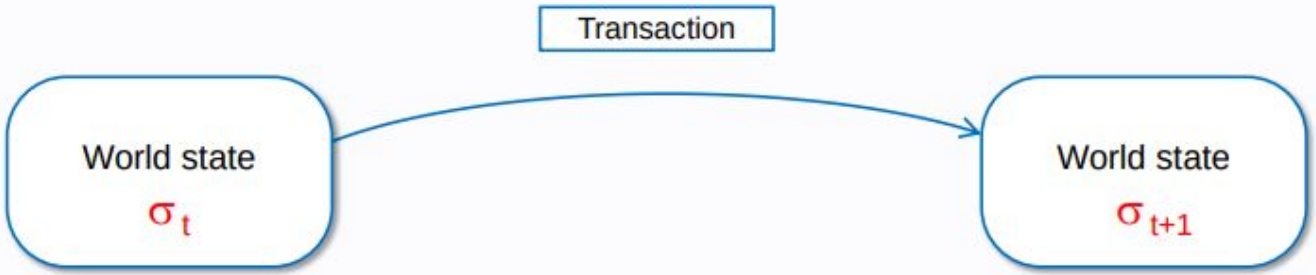
- **Retain the private key**
- **Share the public key.** As transactions between accounts are completed using public keys.

Components of Ethereum Network - Nonce, Storage Root, Ethash

6. **Nonce**
   - **For EOAs,** nonce means the number of transactions via this account.
   - **For CA,** nonce means the number of contracts generated via this account.
7. **Storage Root**
   - It is the **main root node of a Merkle tree**.
   - **Hash of all details of the account** is stored here.
   - The **root of the Merkle tree** is the verification of all transactions.
8. **Ethash**
   - **PoW algorithm for Ethereum 1.0**
   - most **recent version of Dagger-Hashimoto**

Components of Ethereum Network - Transactions

A transaction-based state machine



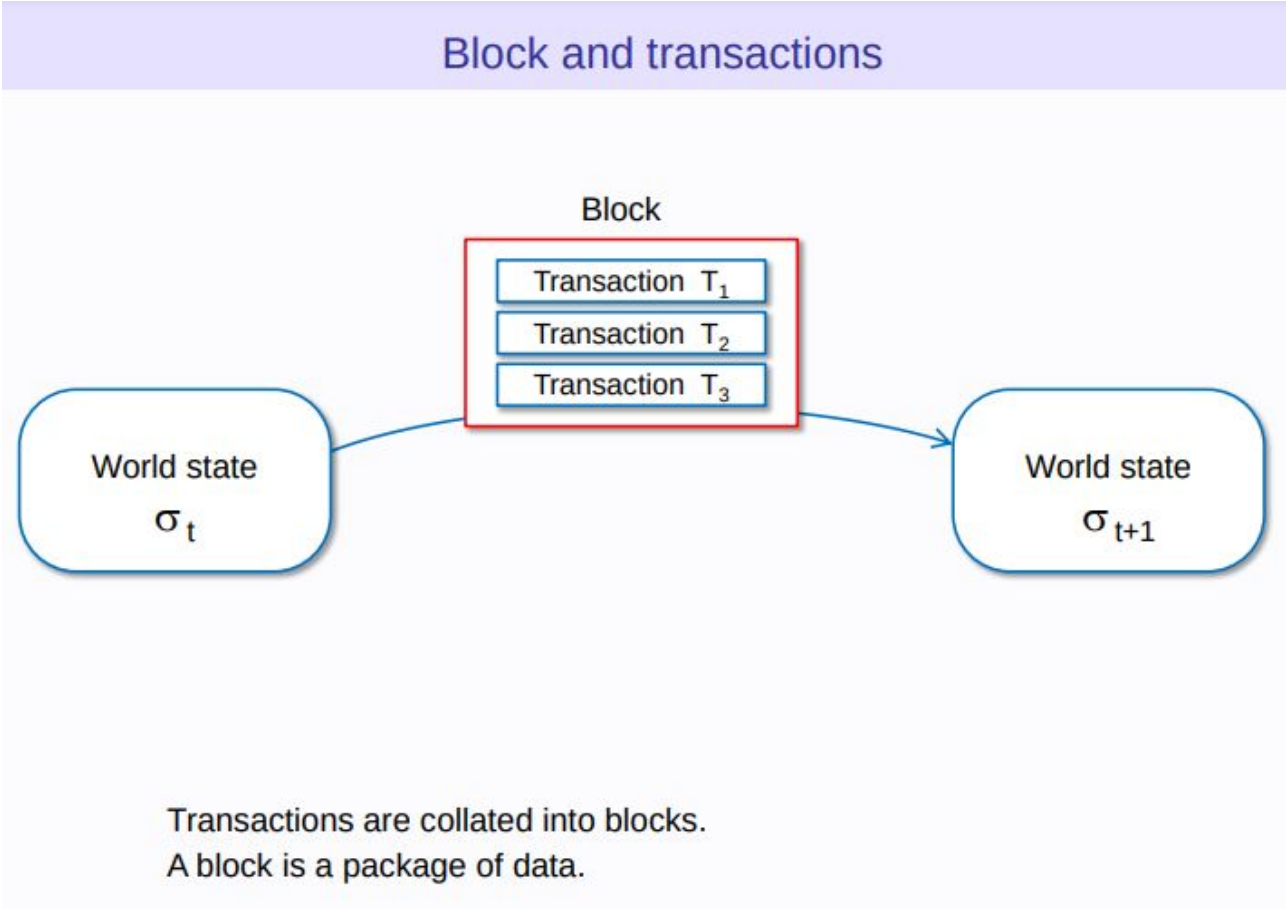
Ethereum can be viewed as a transaction-based state machine.

Components of Ethereum Network - Transactions

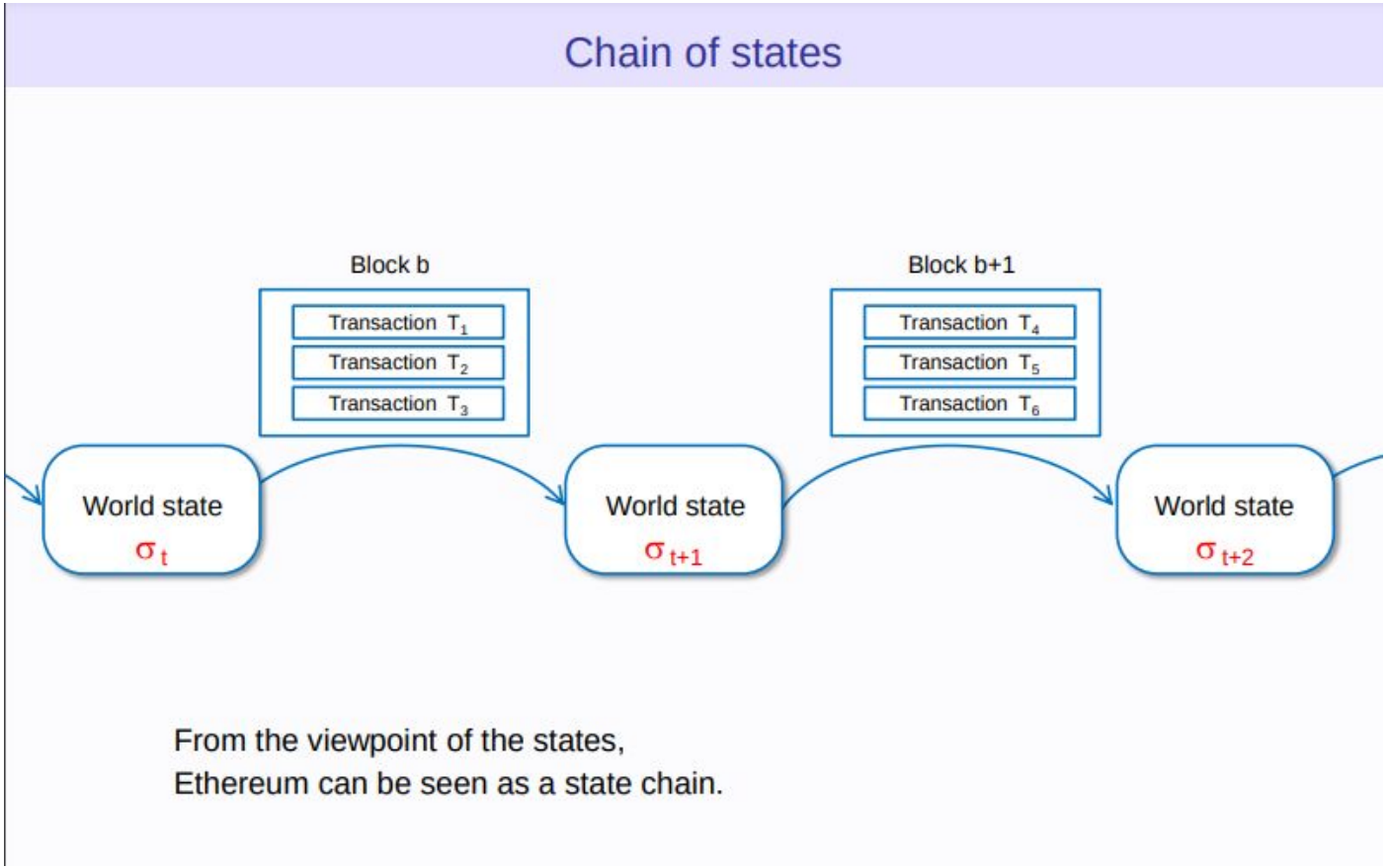
A transaction-based state machine

A transaction represents a valid arc between two states.

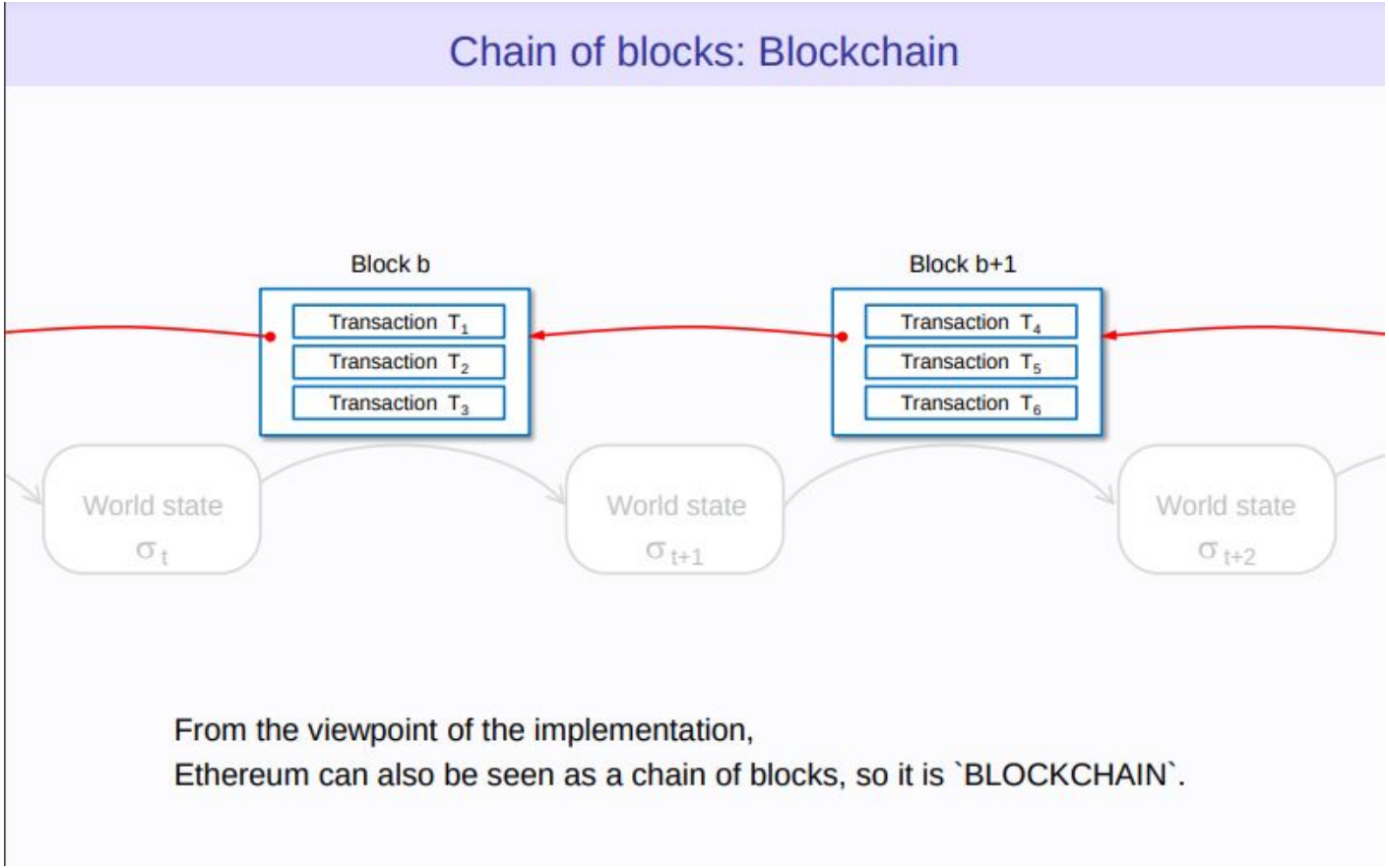
Components of Ethereum Network - Transactions



Components of Ethereum Network - Transactions

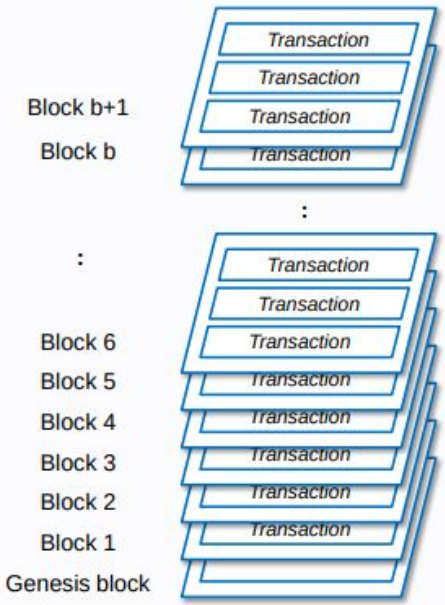


Components of Ethereum Network - Transactions



Components of Ethereum Network - Transactions

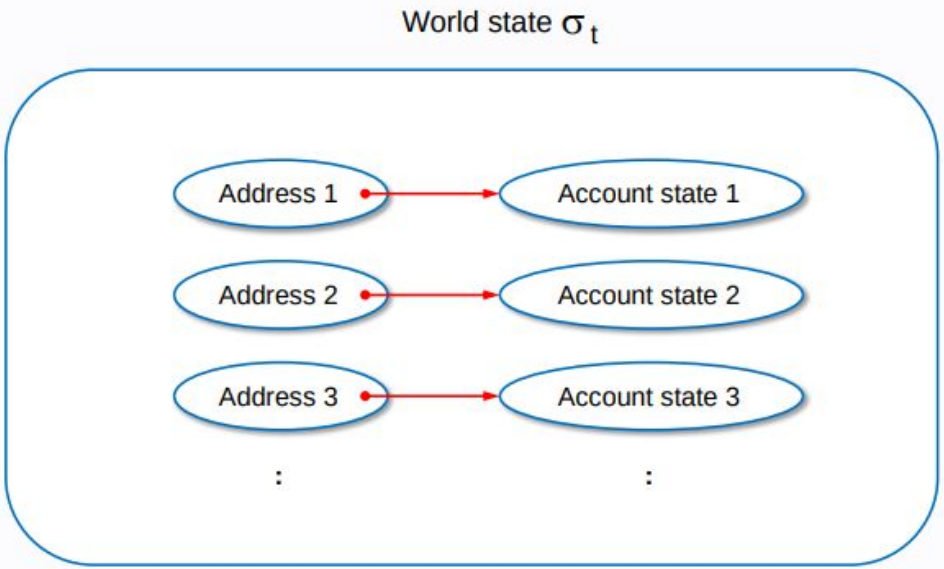
Stack of transactions : Ledger



From the viewpoint of the ledger,
Ethereum can also be seen as a stack of transactions.

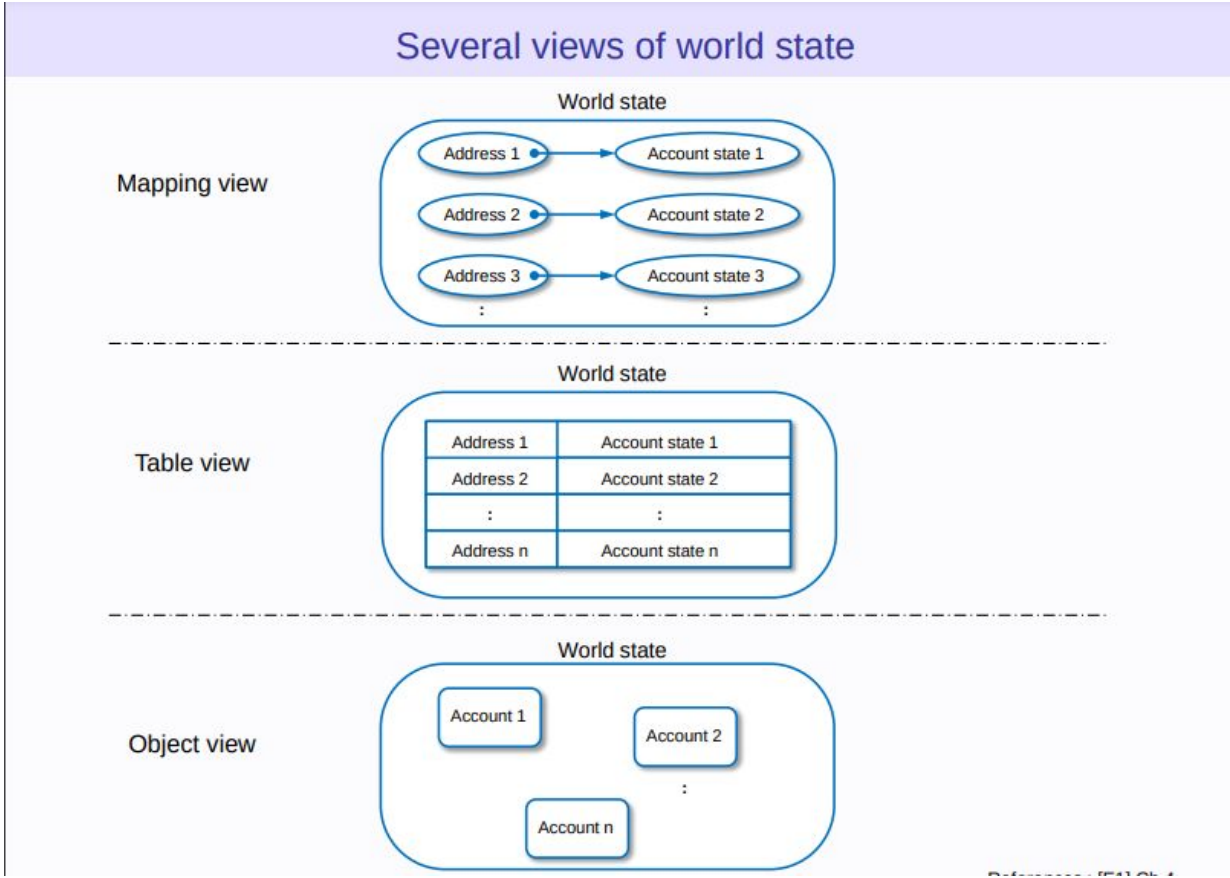
Components of Ethereum Network - Transactions

World state

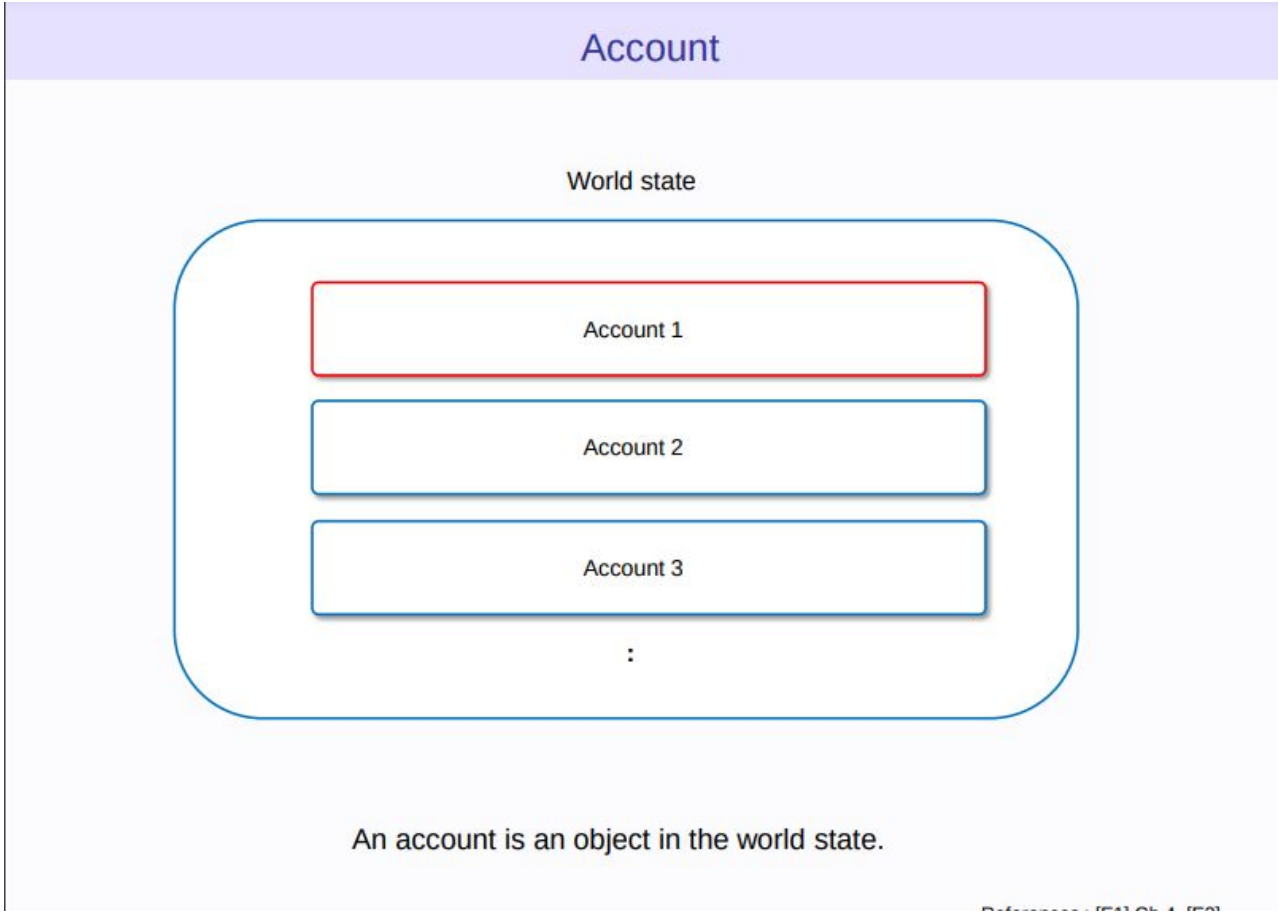


The world state is a mapping between address and account state.

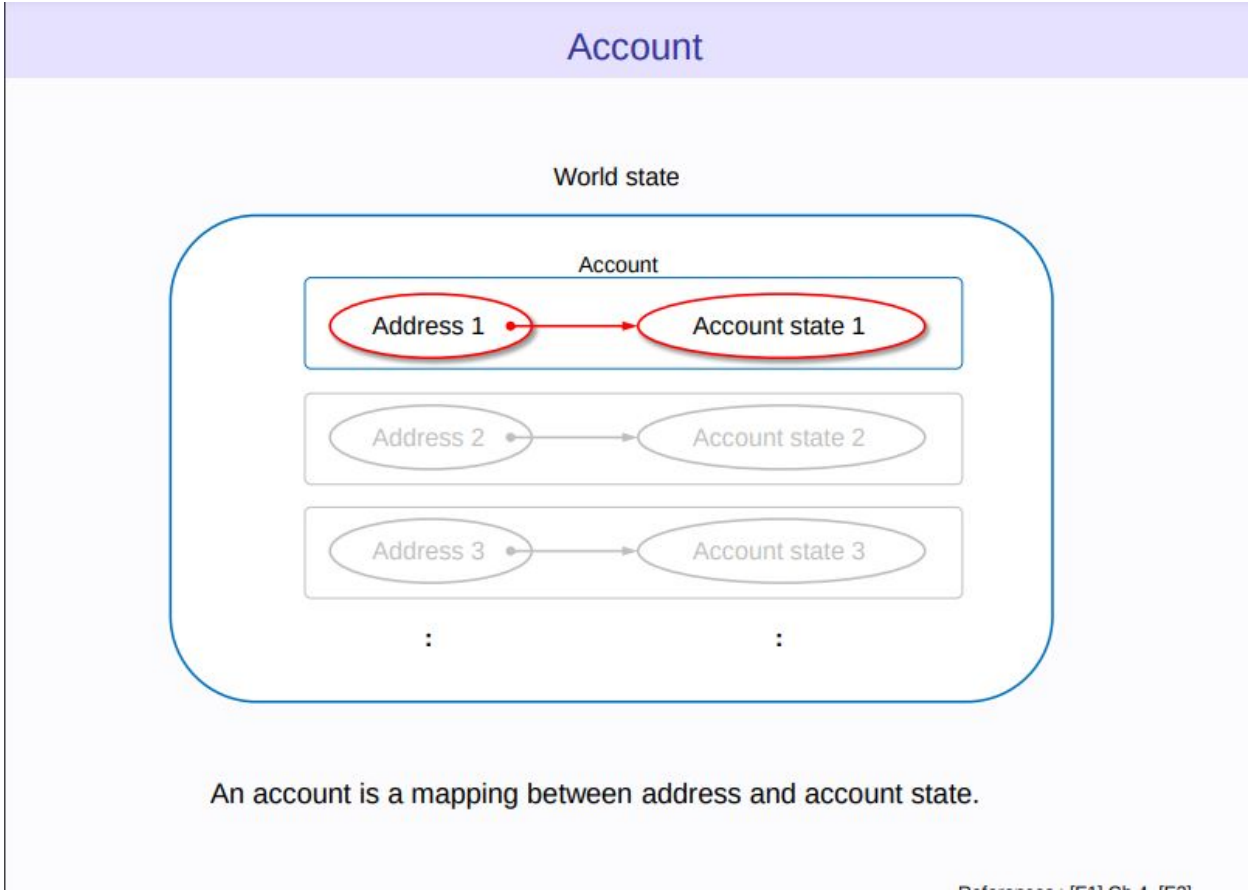
Components of Ethereum Network - Transactions



Components of Ethereum Network - Transactions

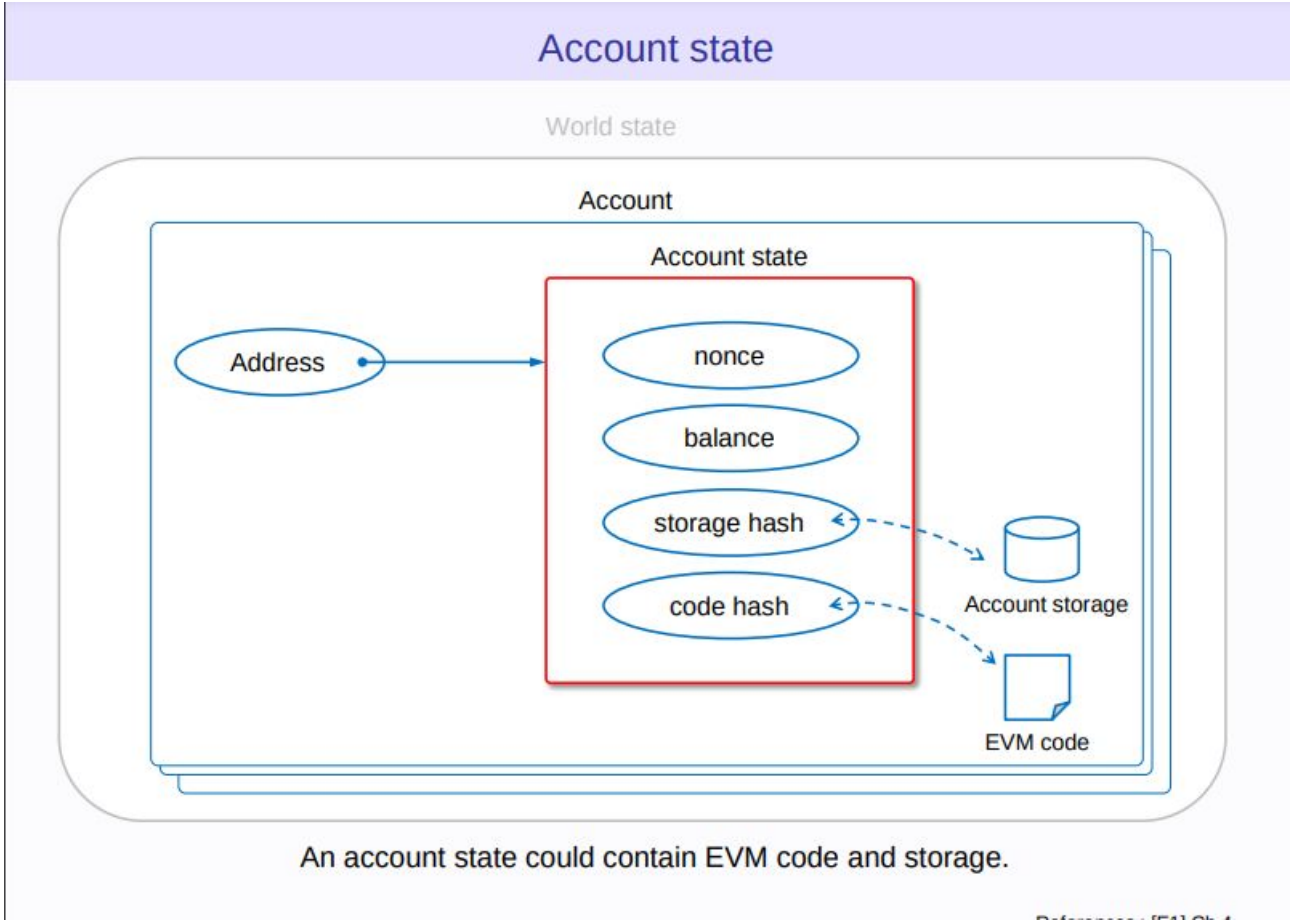


Components of Ethereum Network - Transactions

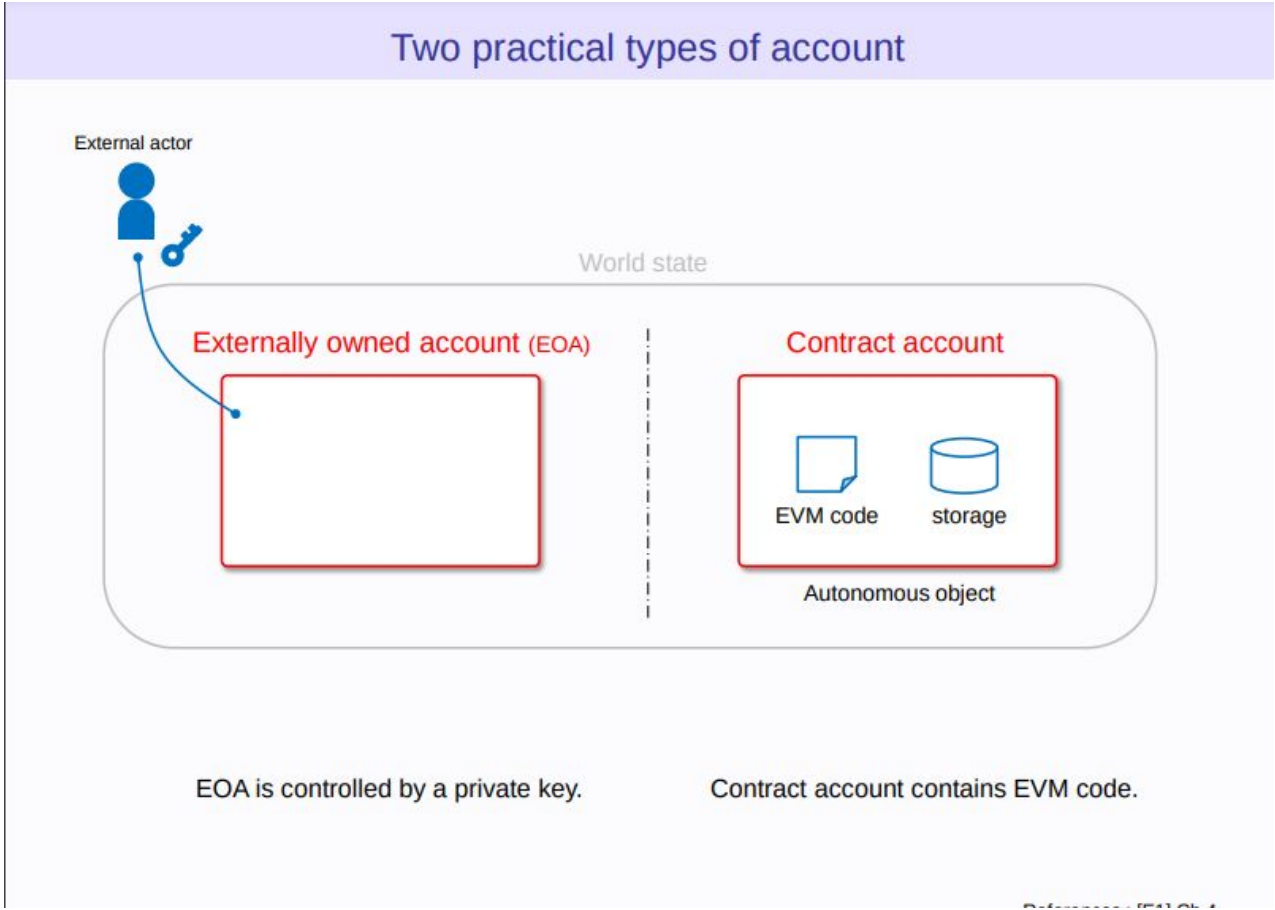


Courtesy : Takenobu T

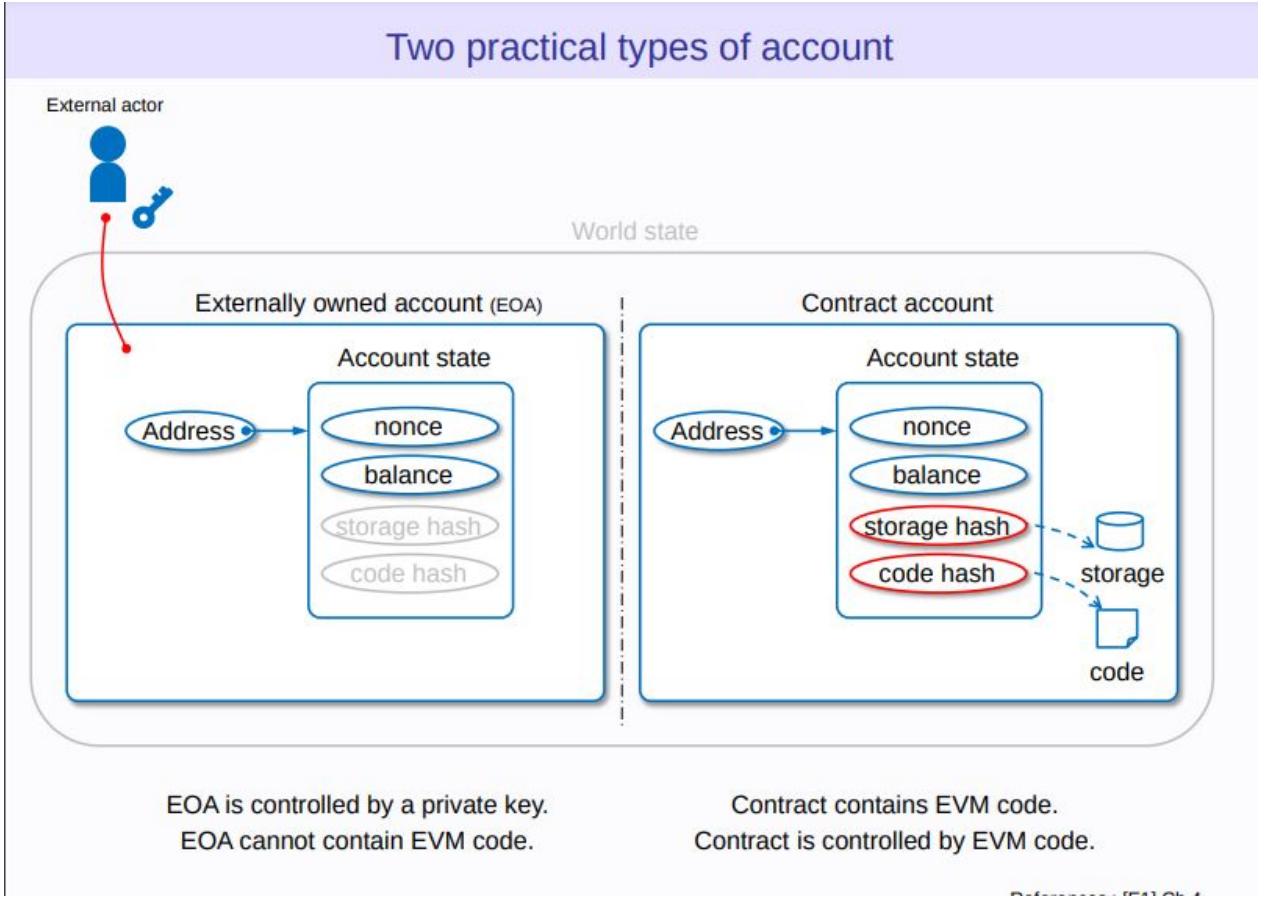
Components of Ethereum Network - Transactions



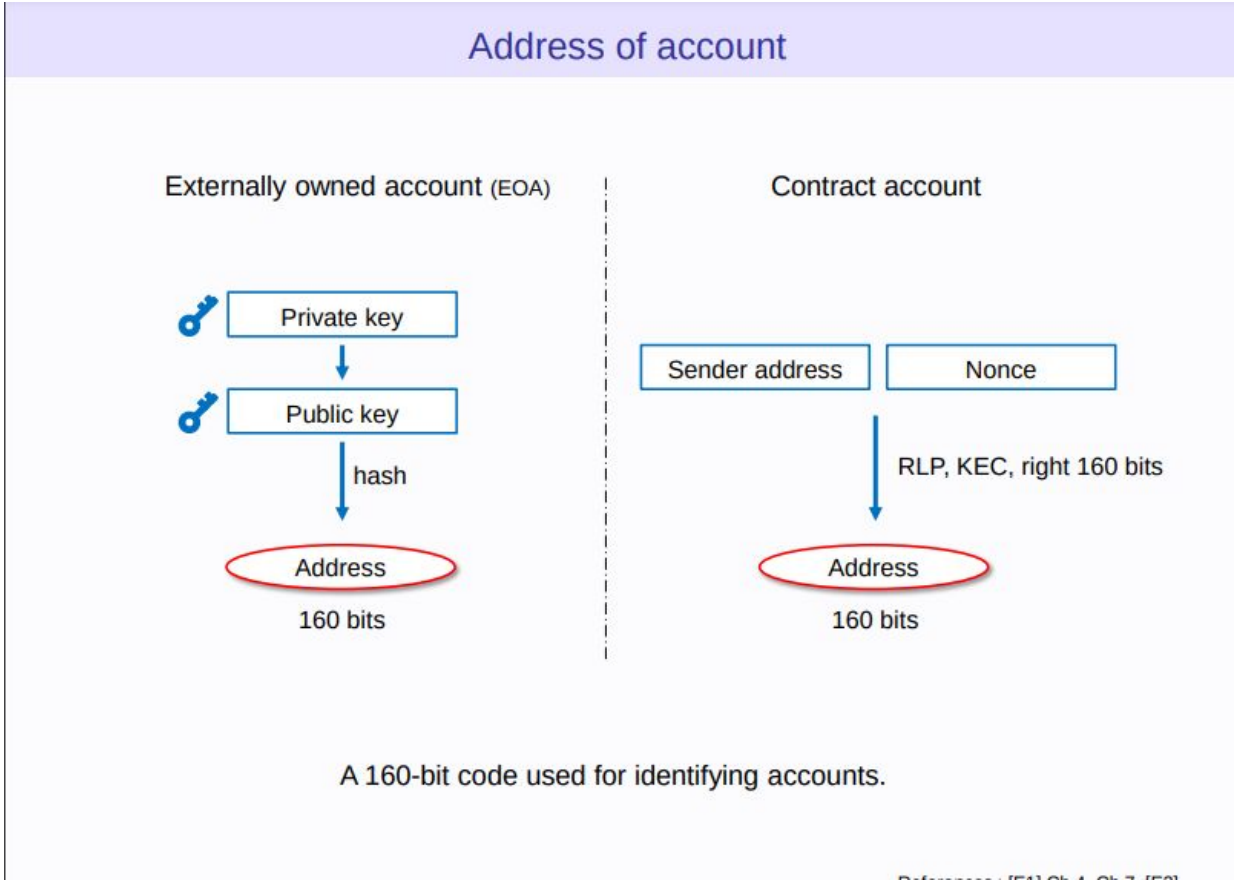
Components of Ethereum Network - Transactions



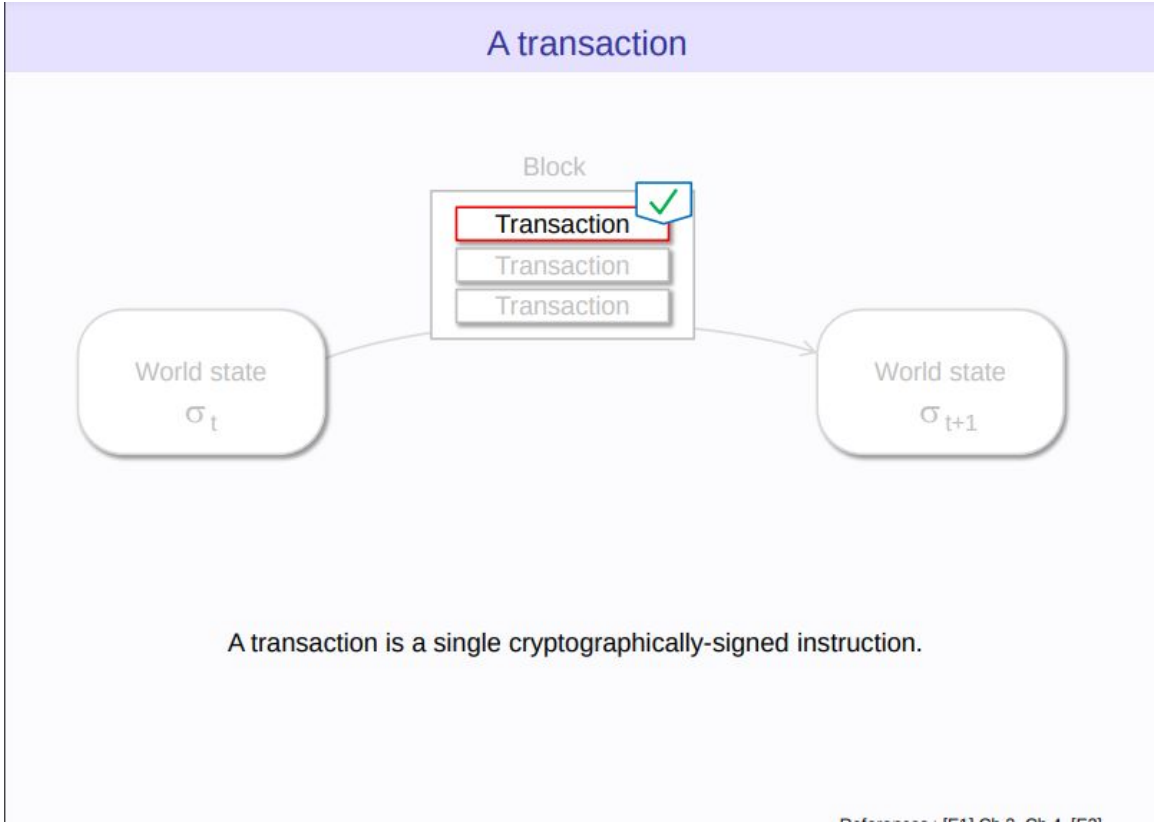
Components of Ethereum Network - Transactions



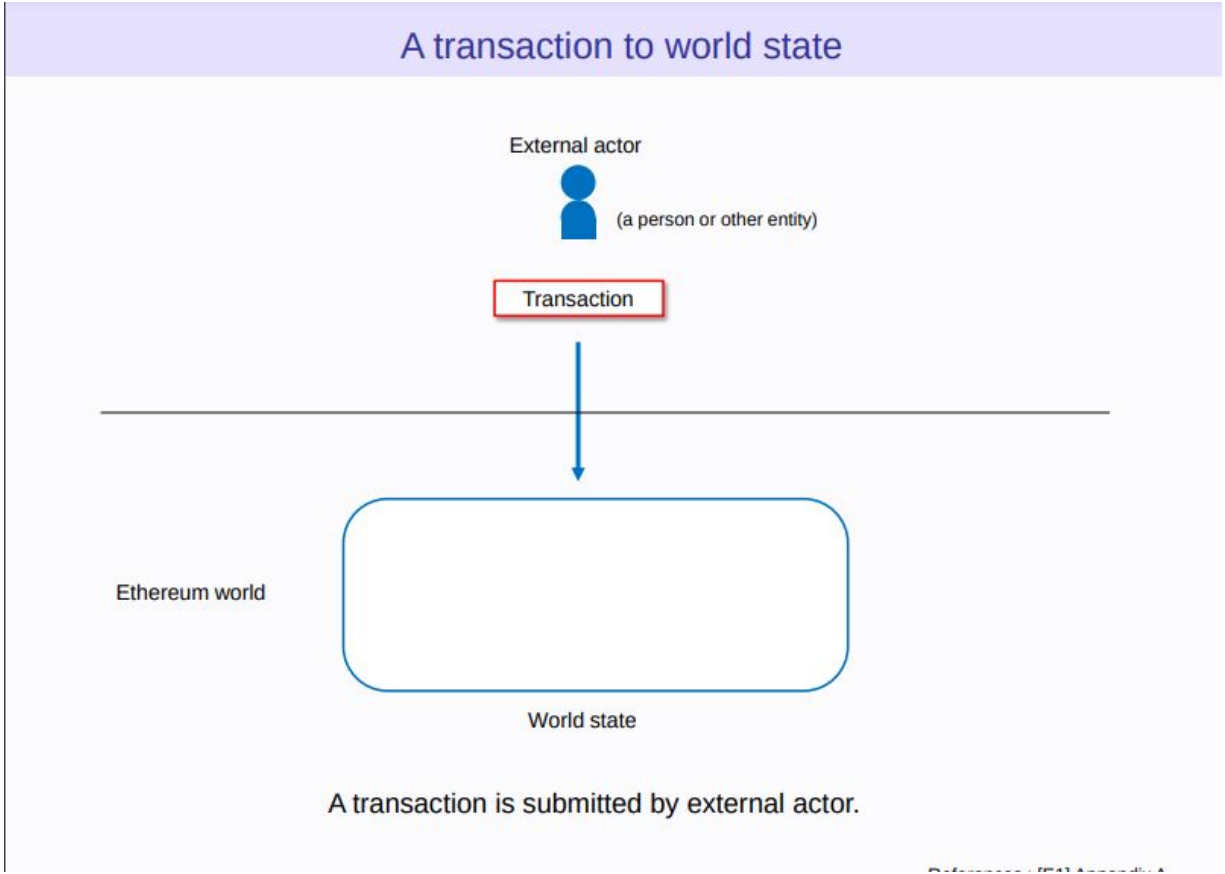
Components of Ethereum Network - Transactions



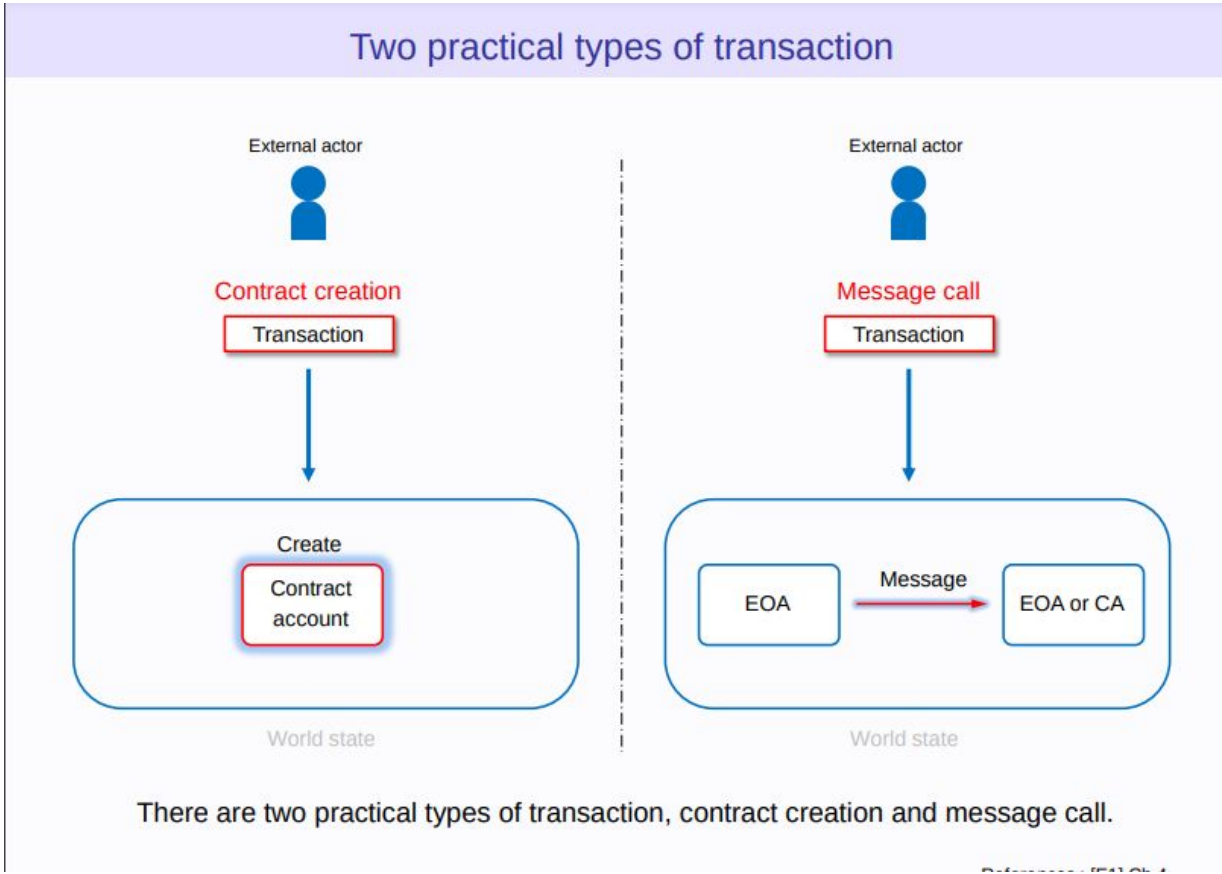
Components of Ethereum Network - Transactions



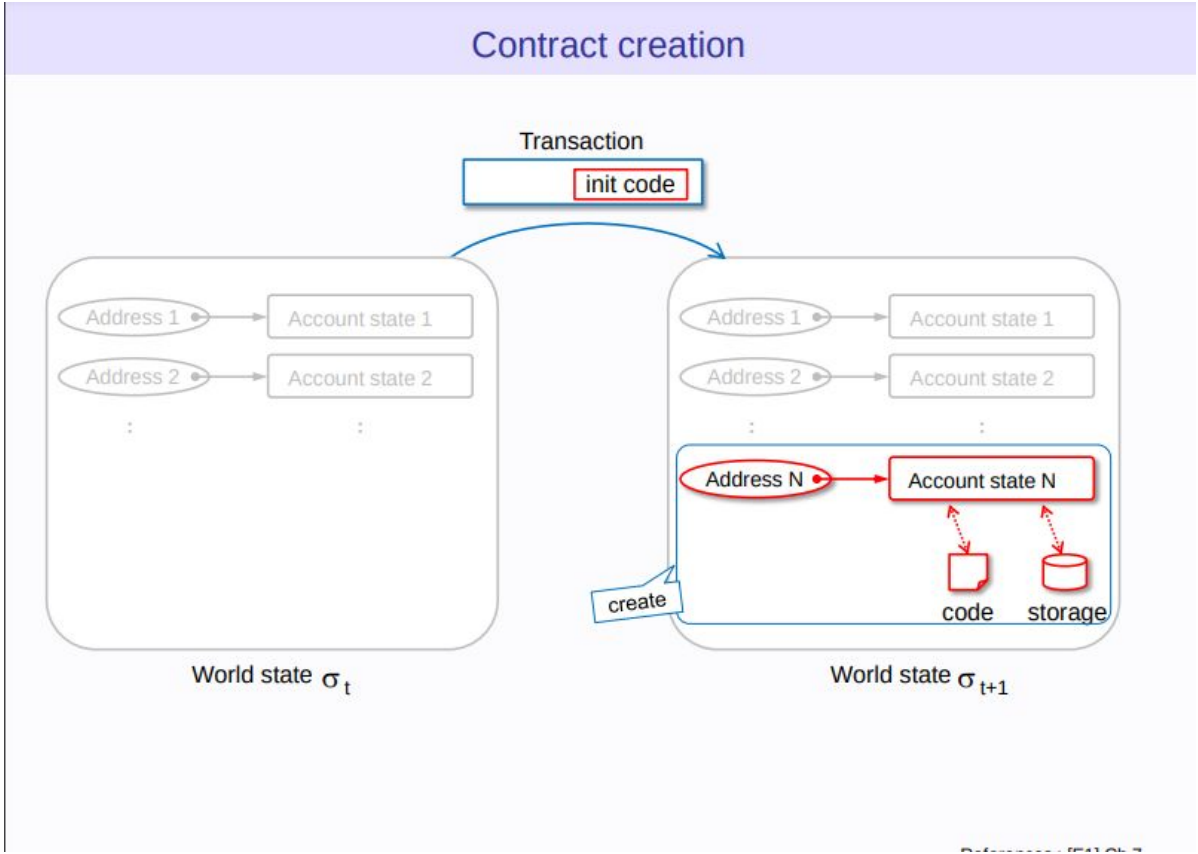
Components of Ethereum Network - Transactions



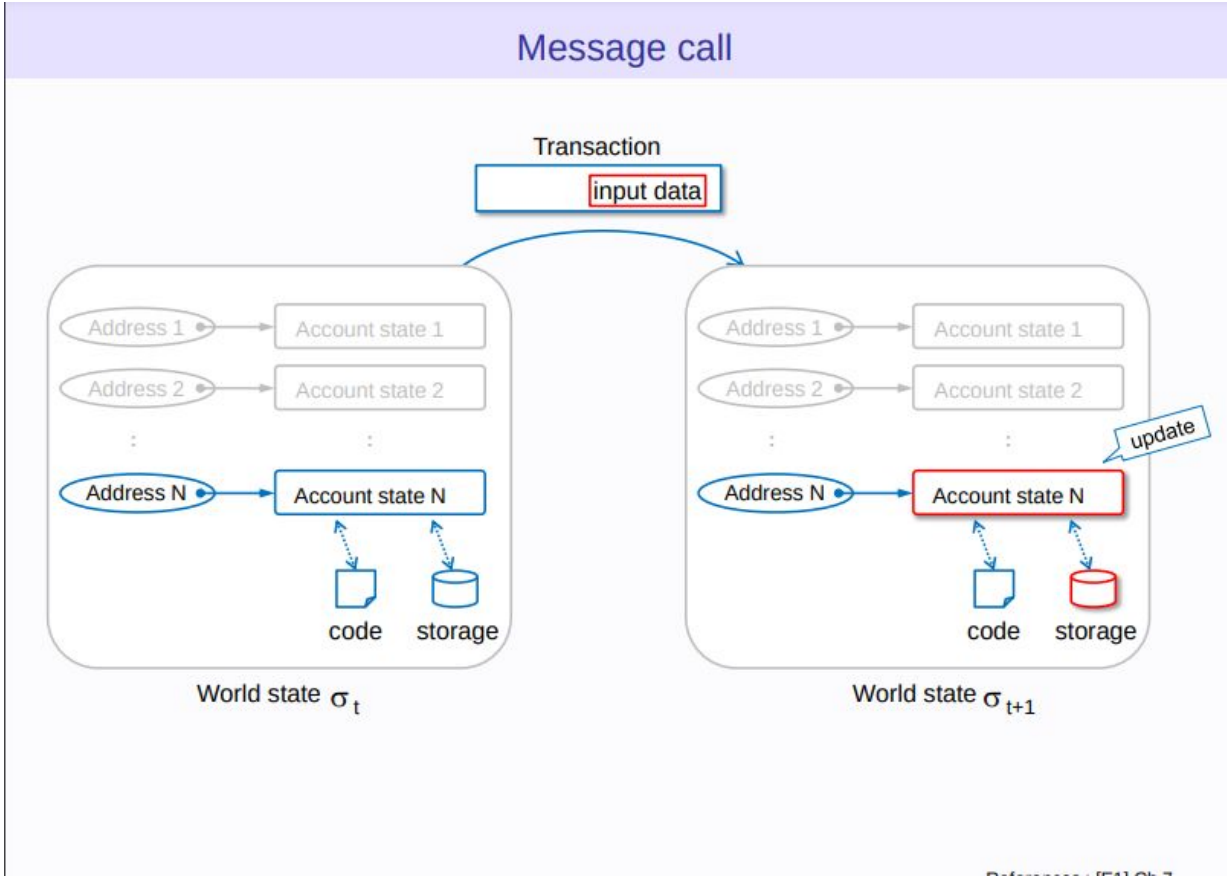
Components of Ethereum Network - Transactions



Components of Ethereum Network - Transactions

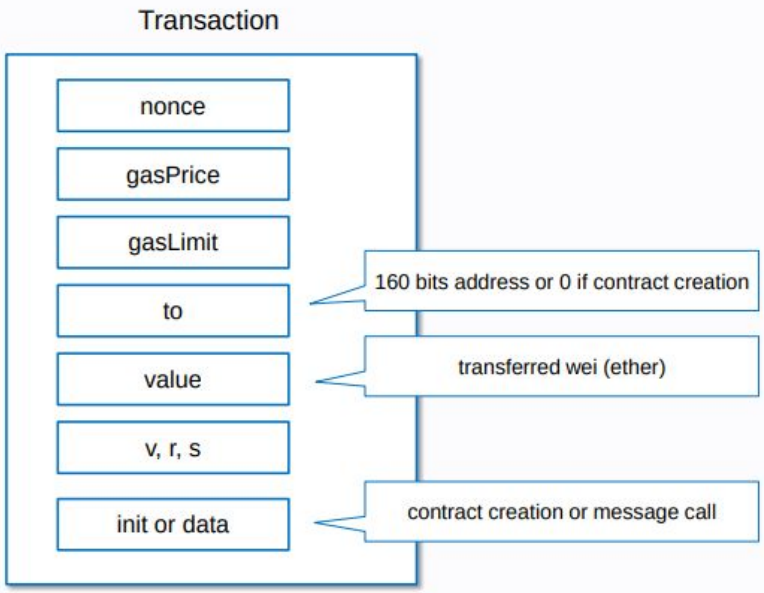


Components of Ethereum Network - Transactions



Components of Ethereum Network - Transactions

Field of a transaction



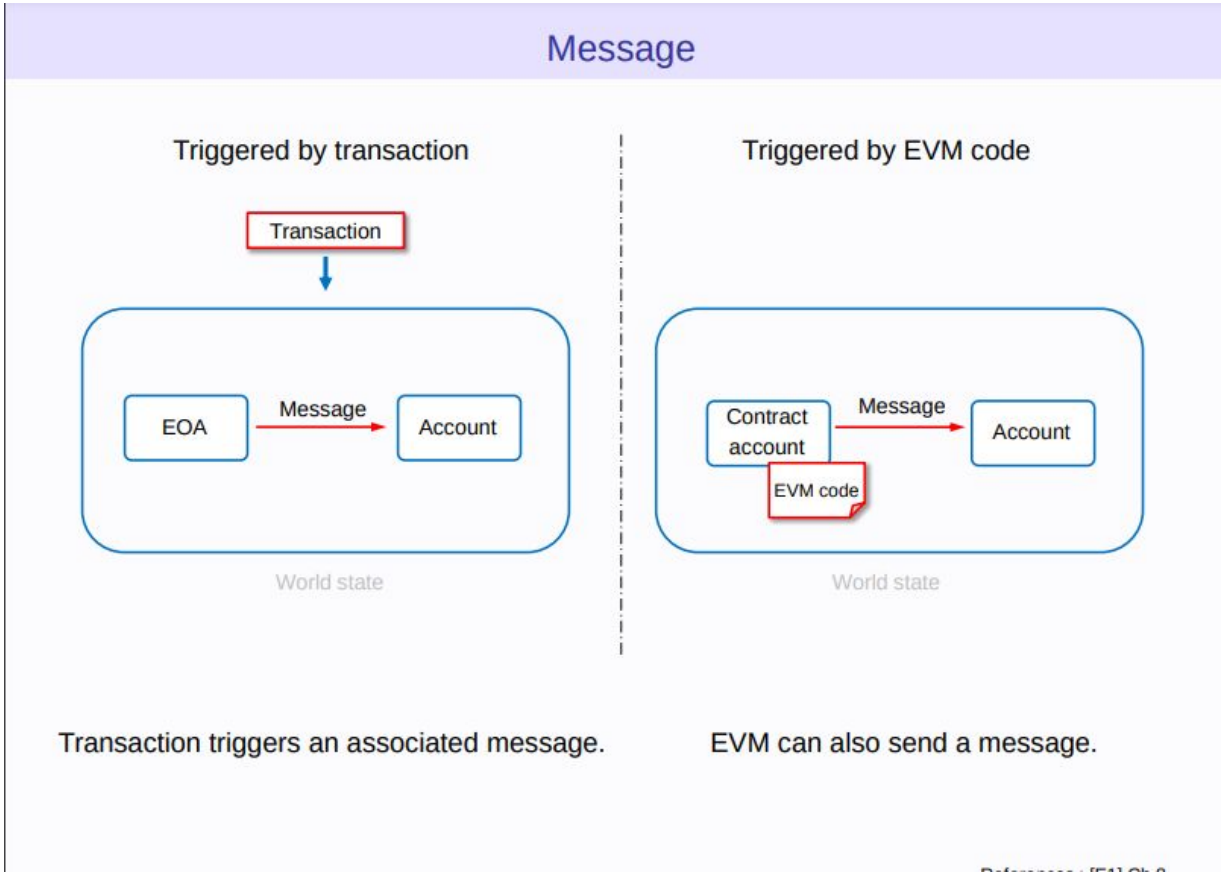
Components of Ethereum Network - Transactions

Message

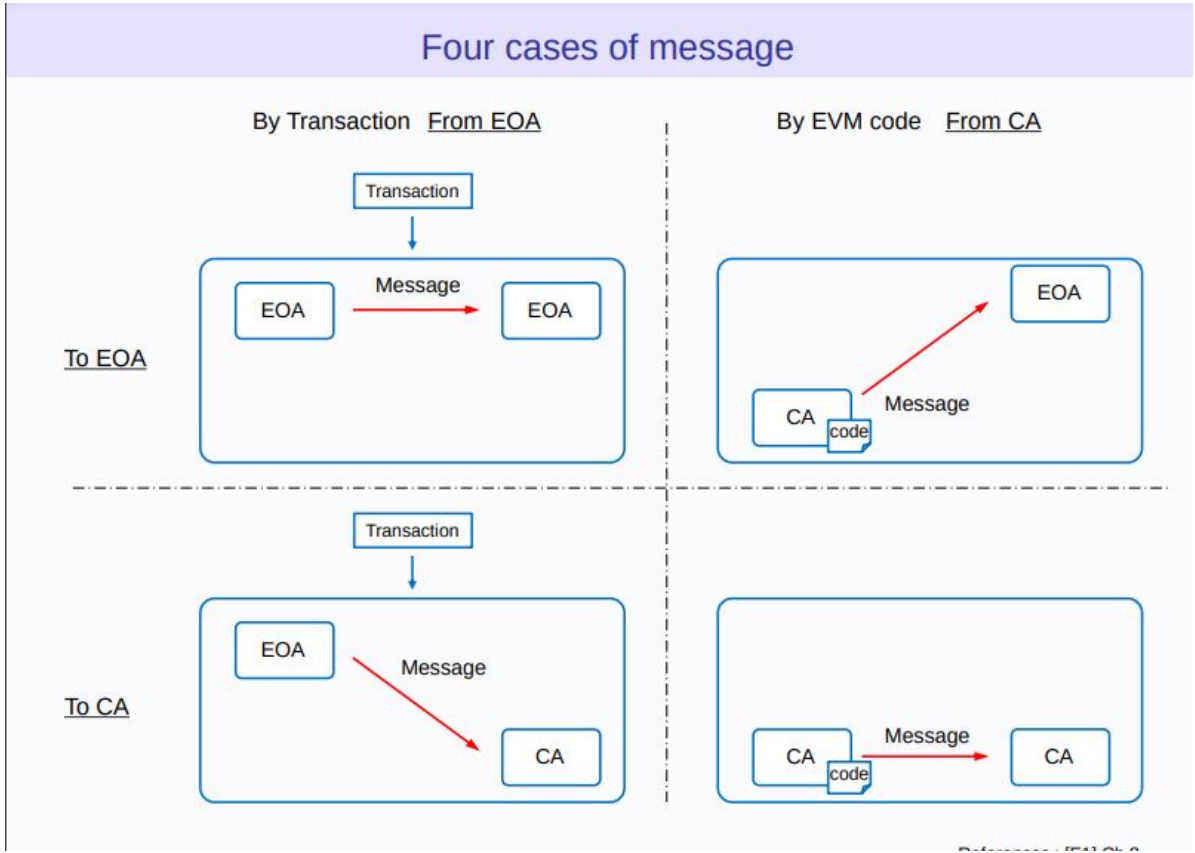
Message is passed between two Accounts.

Message is Data (as a set of bytes) and Value (specified as Ether) .

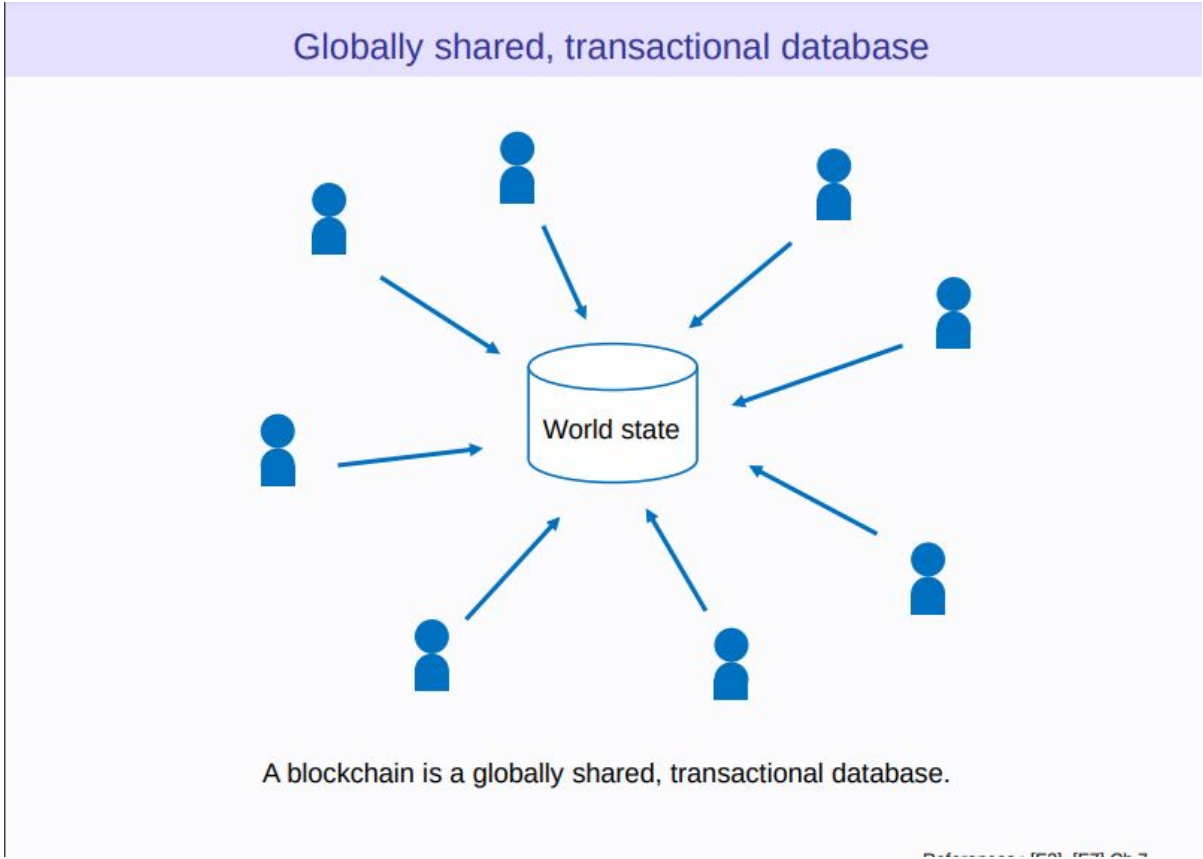
Components of Ethereum Network - Transactions



Components of Ethereum Network - Transactions

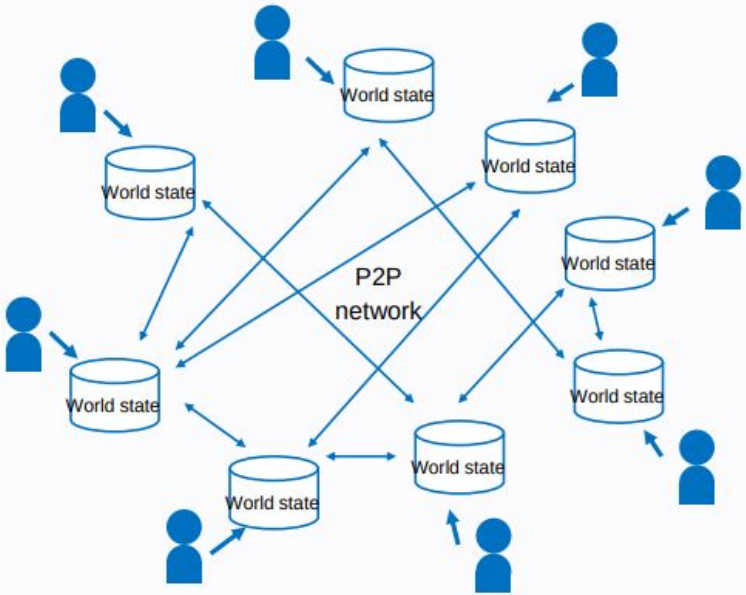


Components of Ethereum Network - Transactions



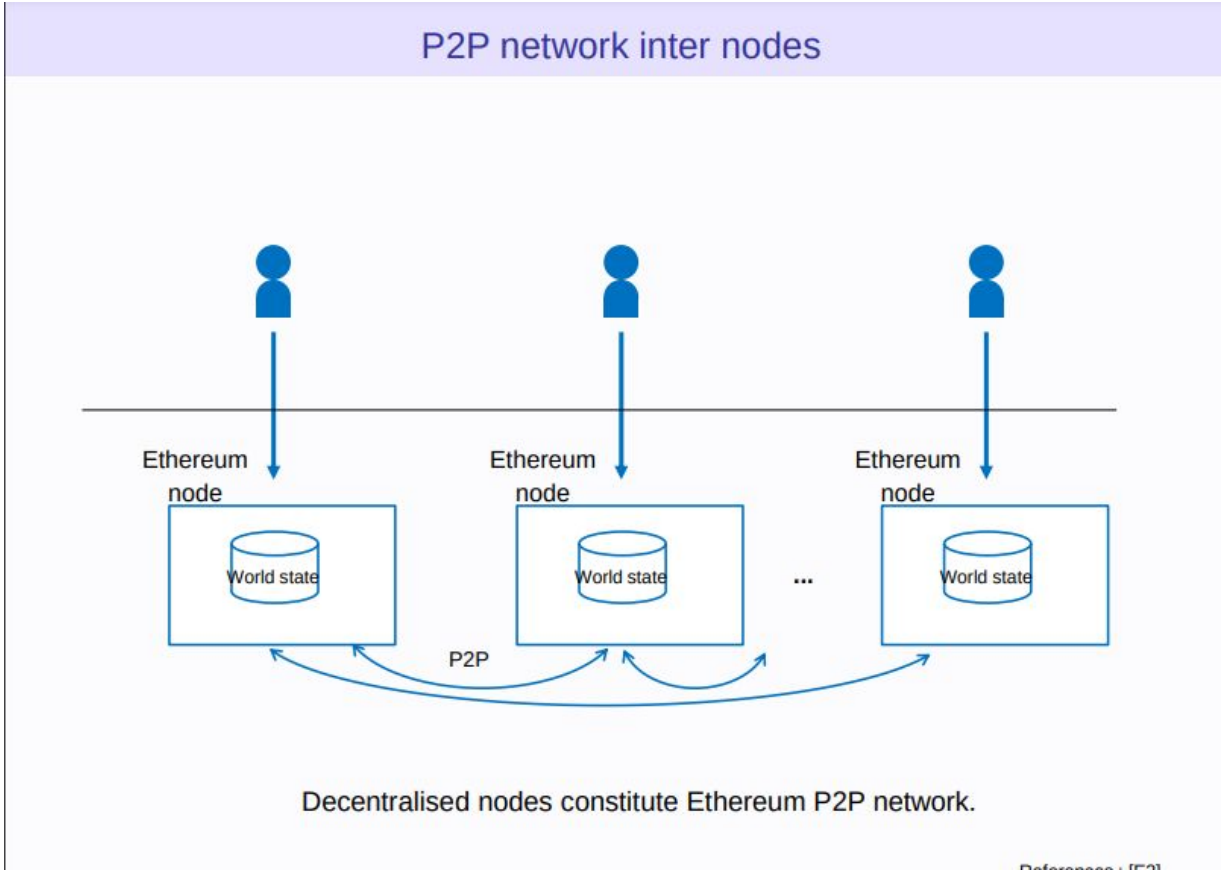
Components of Ethereum Network - Transactions

Decentralised database

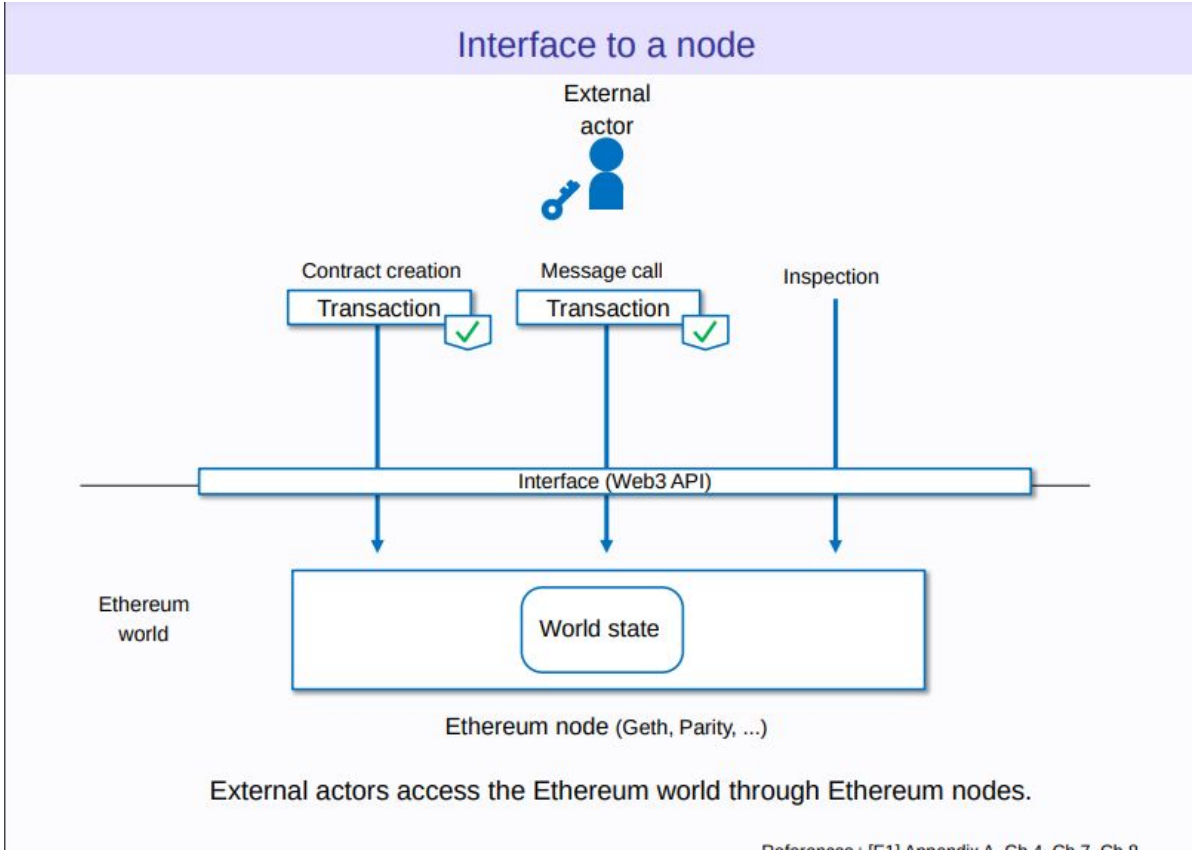


A blockchain is a globally shared, decentralised, transactional database.

Components of Ethereum Network - Transactions



Components of Ethereum Network - Transactions



Components of Ethereum Network - Transactions

Atomicity of transaction

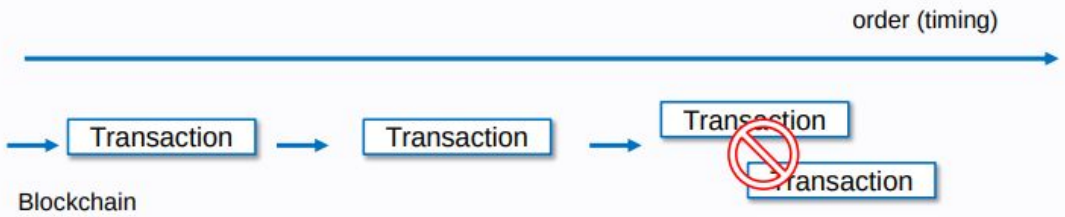
A transaction is an atomic operation. Can't divide or interrupt.

Transaction or Transaction

That is, All (complete done) or Nothing (zero effect).

Components of Ethereum Network - Transactions

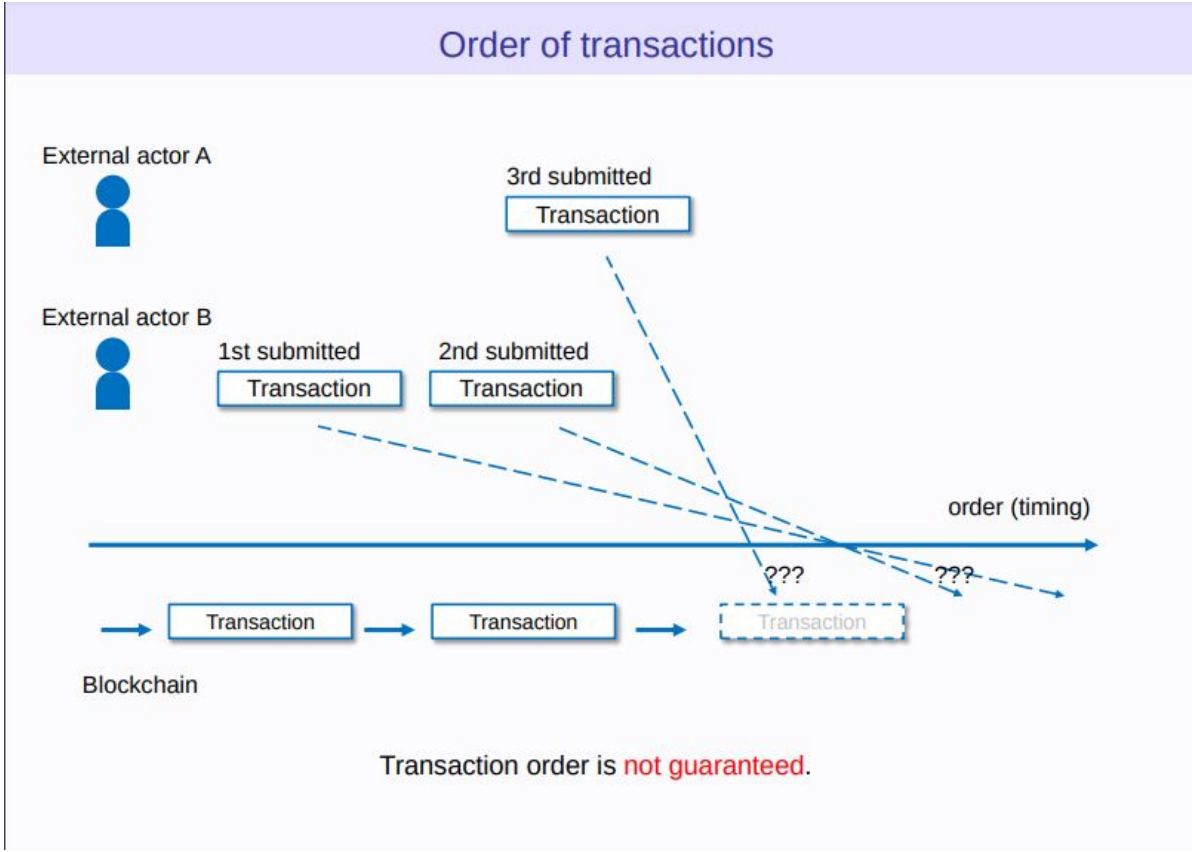
Order of transactions



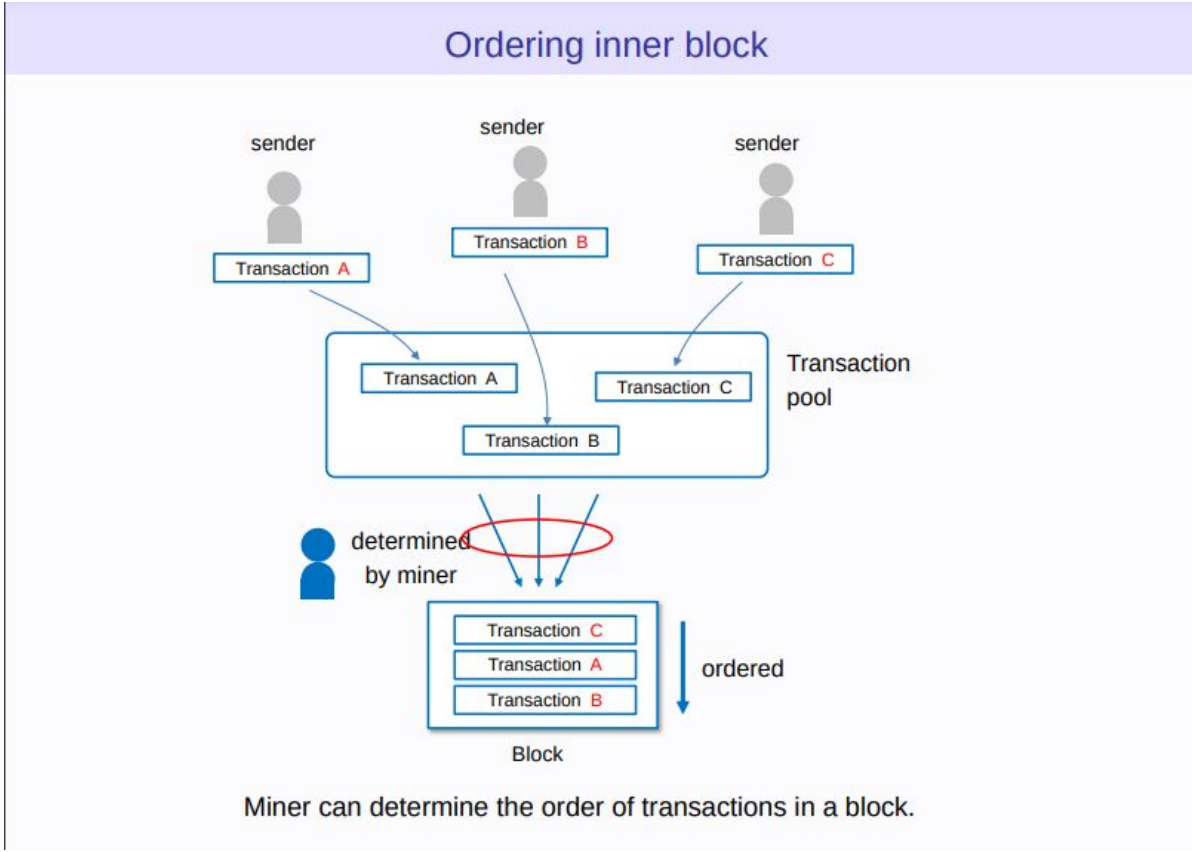
Transactions cannot be overlapped.

Transactions must be executed sequentially.

Components of Ethereum Network - Transactions

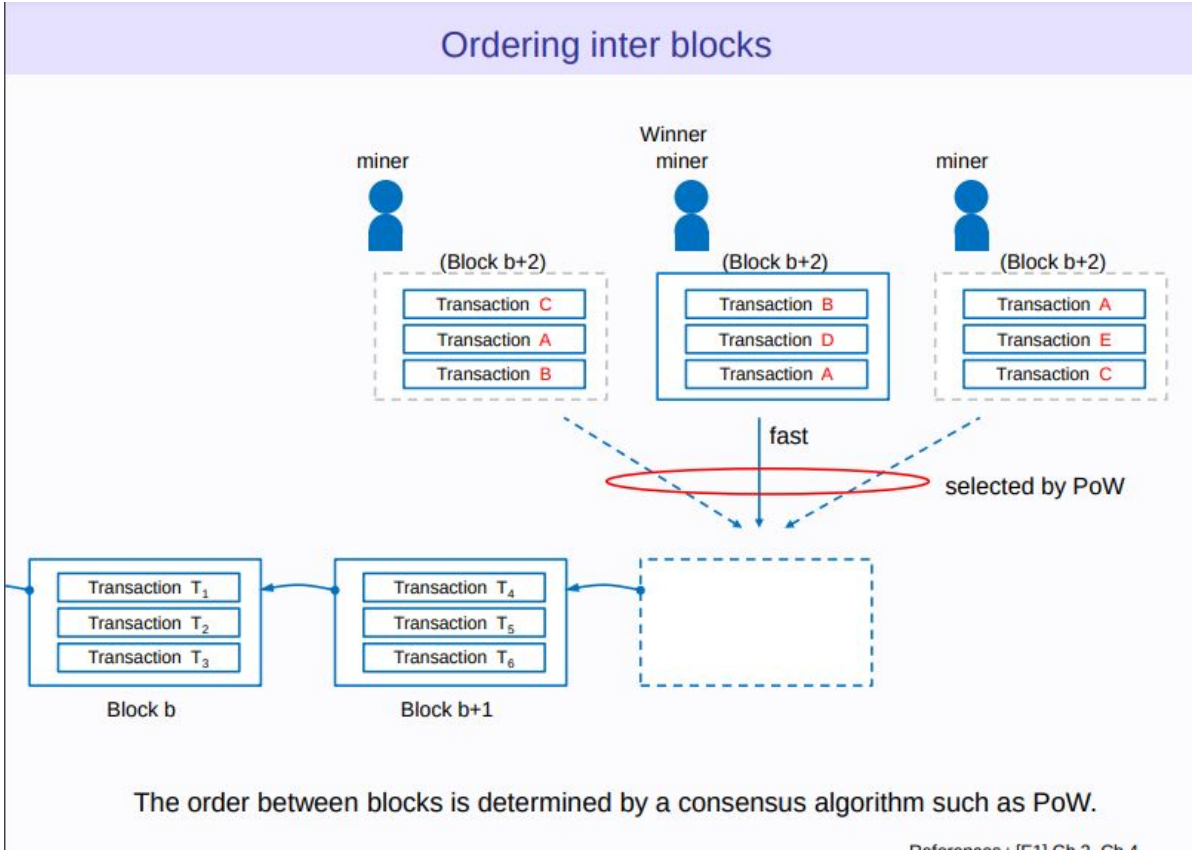


Components of Ethereum Network - Transactions

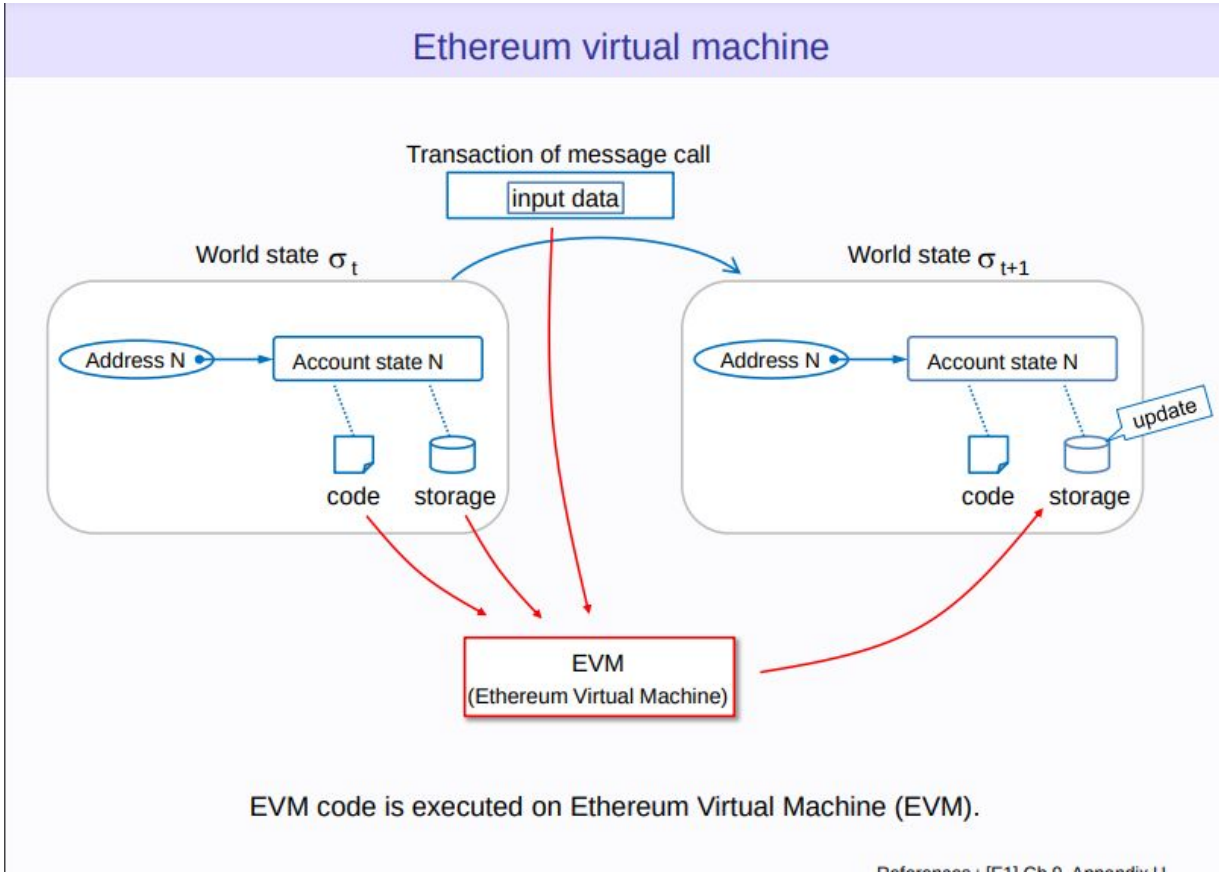


Courtesy : Takenobu T

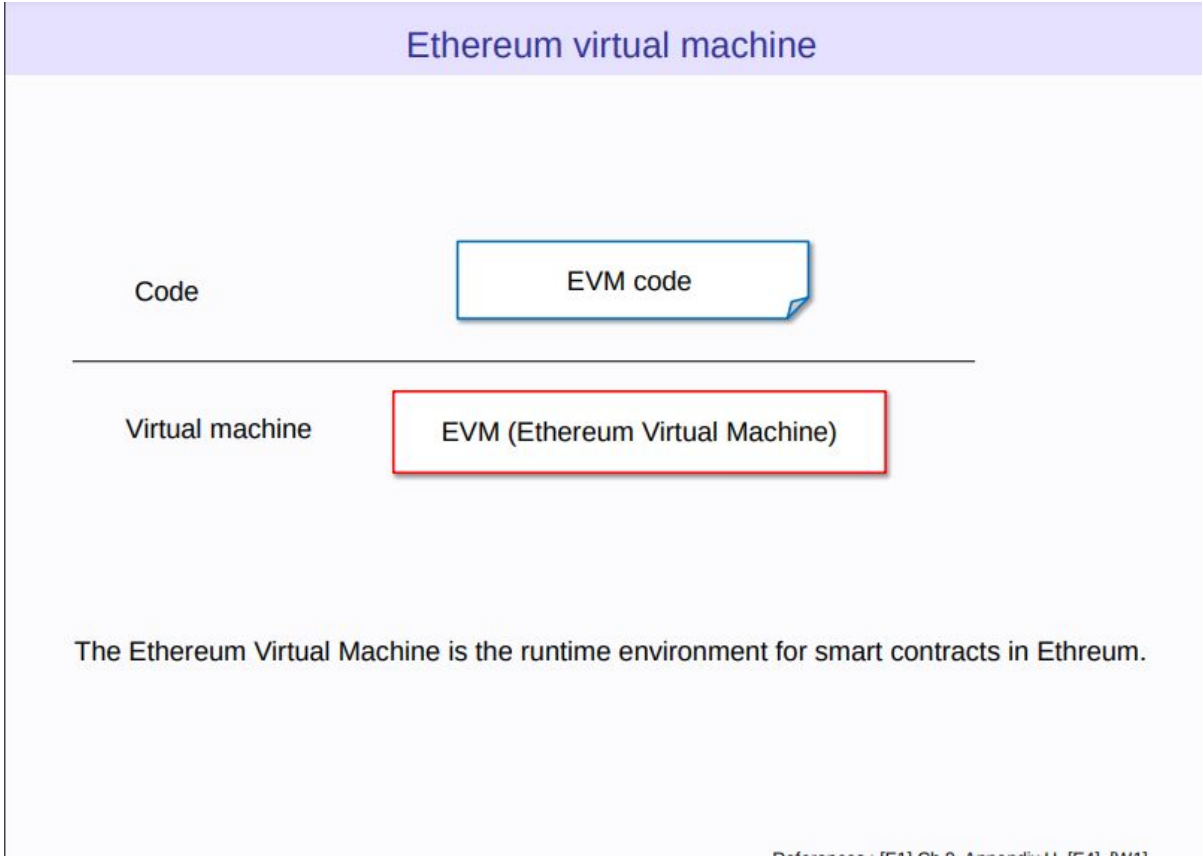
Components of Ethereum Network - Transactions



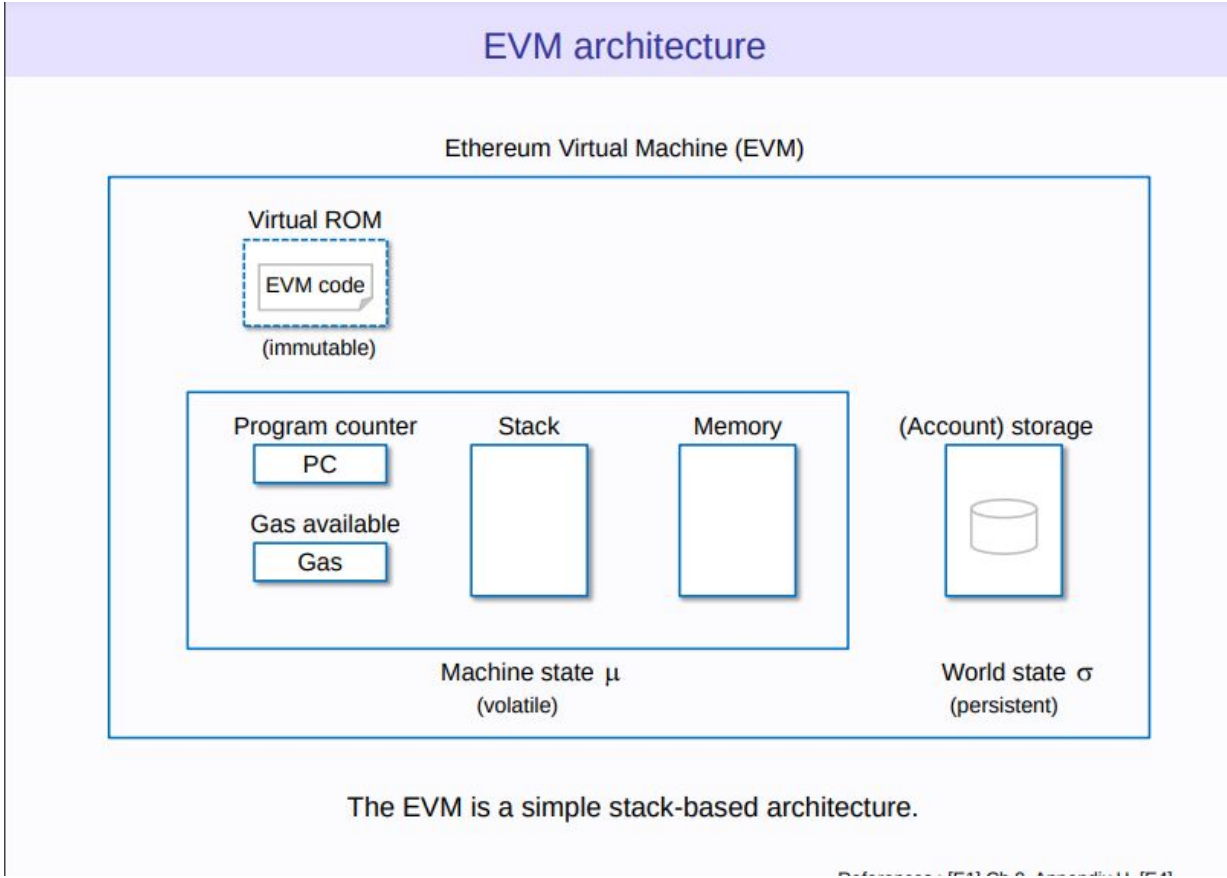
Components of Ethereum Network - Transactions & EVM



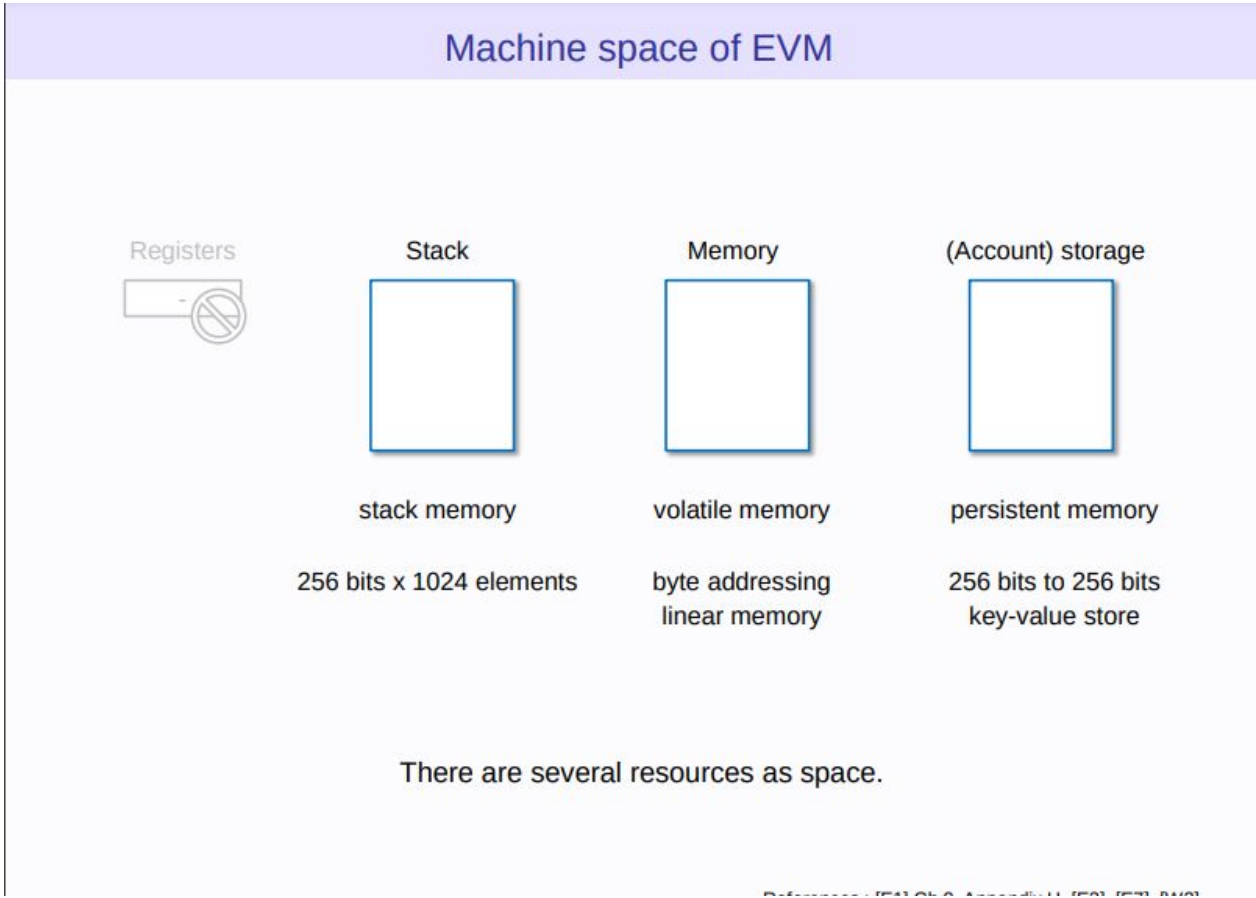
Components of Ethereum Network - EVM



Components of Ethereum Network - EVM Architecture

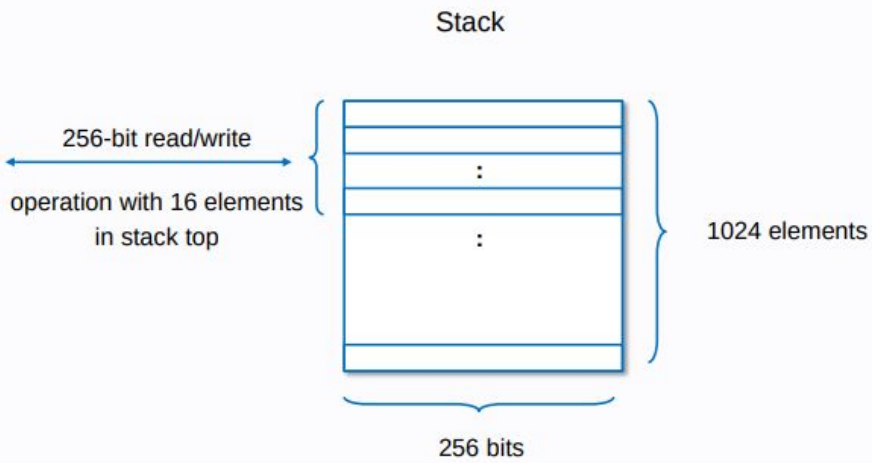


Components of Ethereum Network - EVM Architecture



Components of Ethereum Network - EVM Architecture

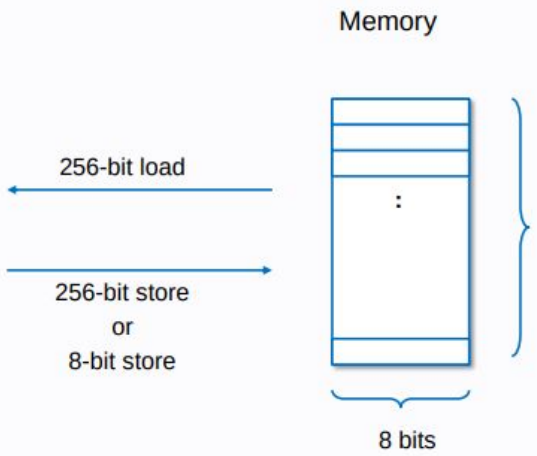
Stack



All operation are performed on the stack.
Access with many instructions such as PUSH/POP/COPY/SWAP, ...

Components of Ethereum Network - EVM Architecture

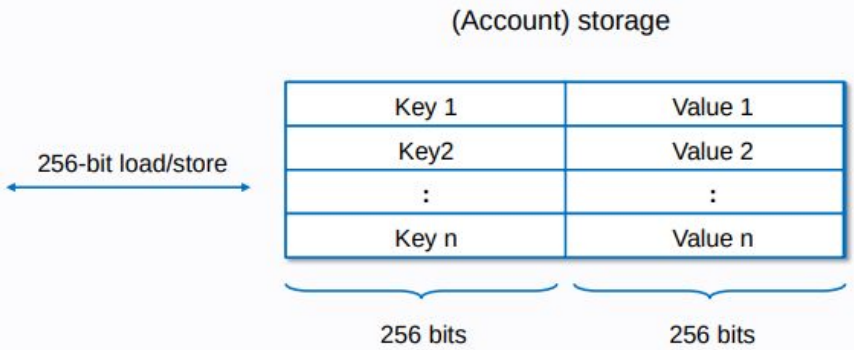
Memory



Memory is linear and can be addressed at byte level.
Access with MSTORE/MSTORE8/MLOAD instructions.
All locations in memory are well-defined initially as zero.

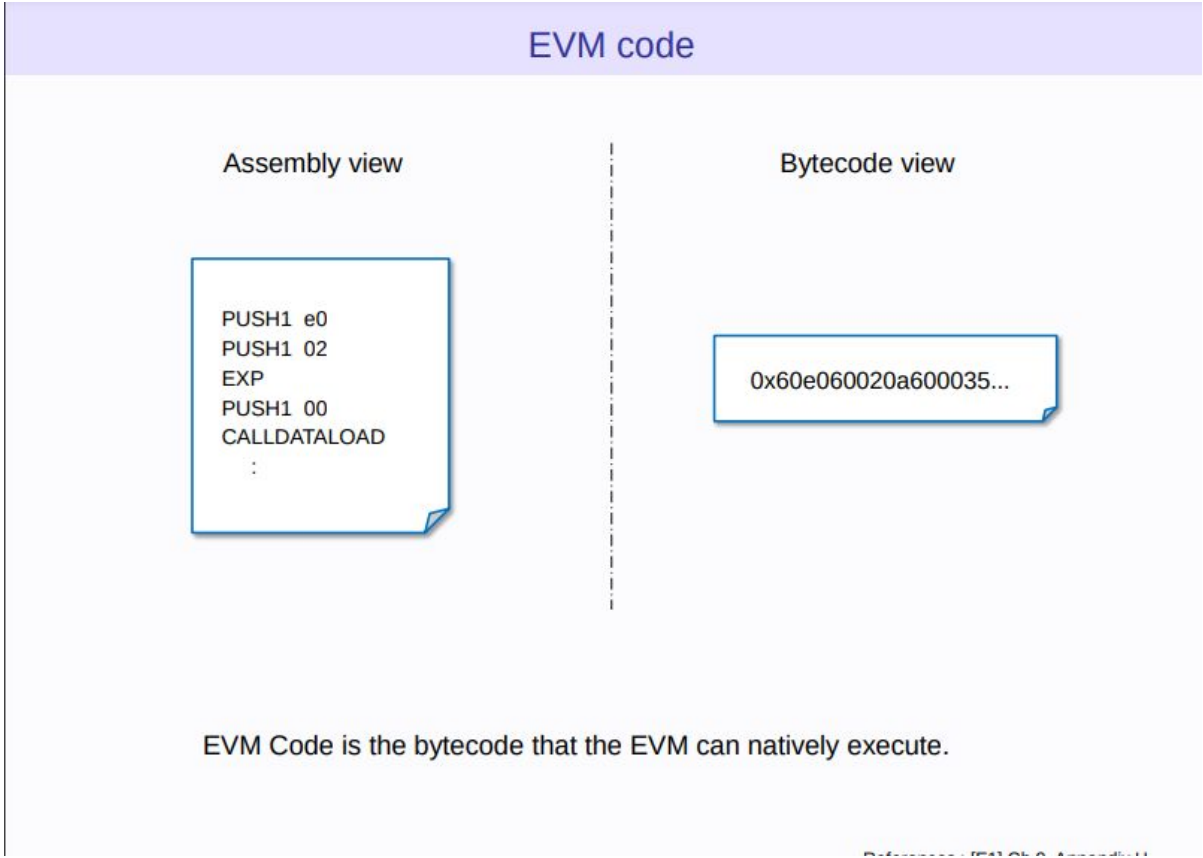
Components of Ethereum Network - EVM Architecture

Account storage

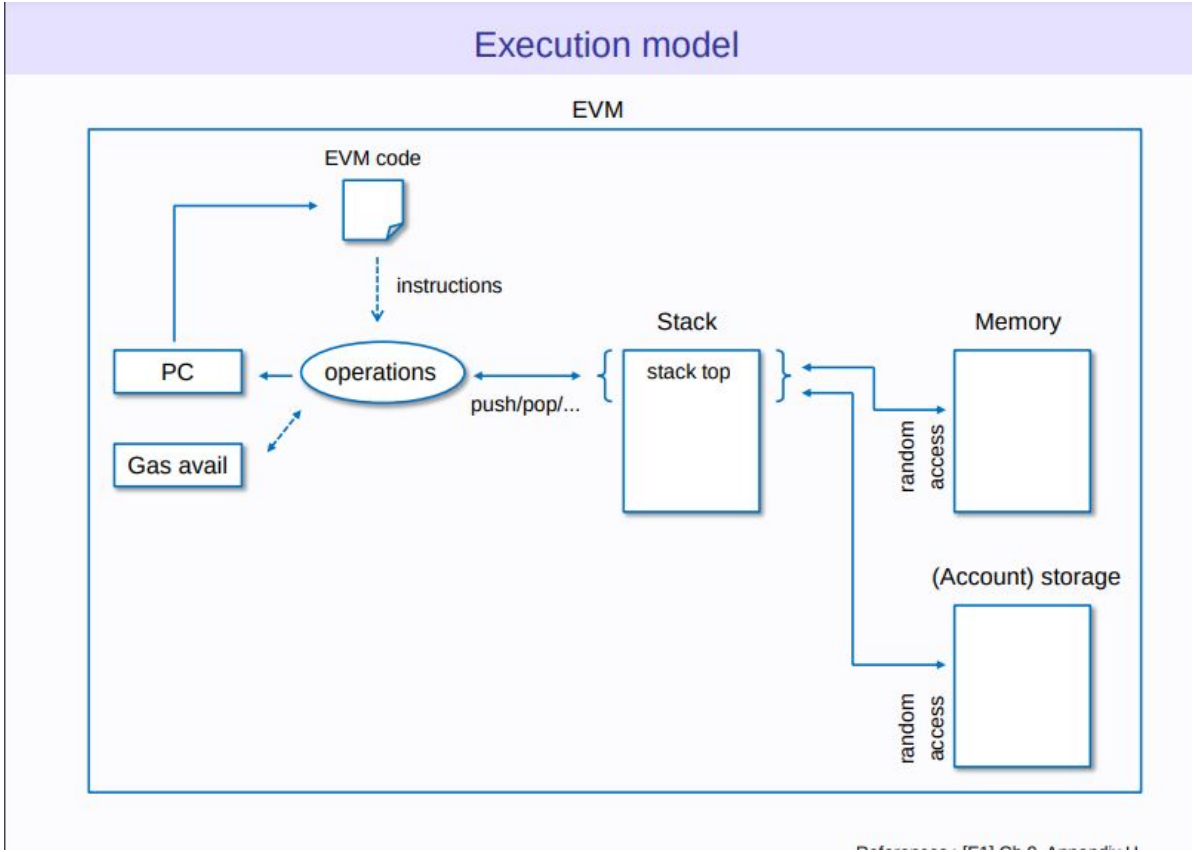


Storage is a key-value store that maps 256-bit words to 256-bit words.
Access with SSTORE/SLOAD instructions.
All locations in storage are well-defined initially as zero.

Components of Ethereum Network - EVM Architecture

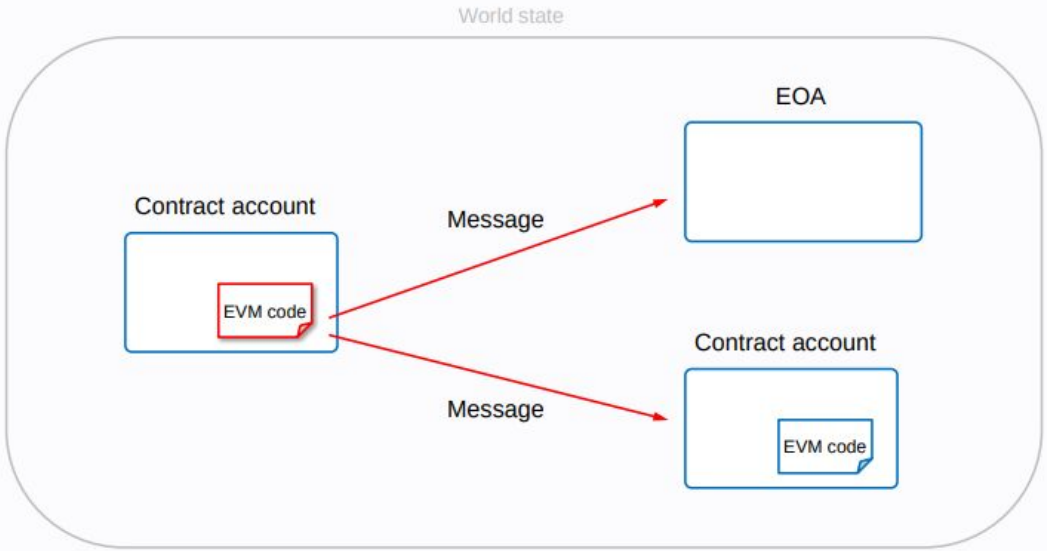


Components of Ethereum Network - EVM Architecture



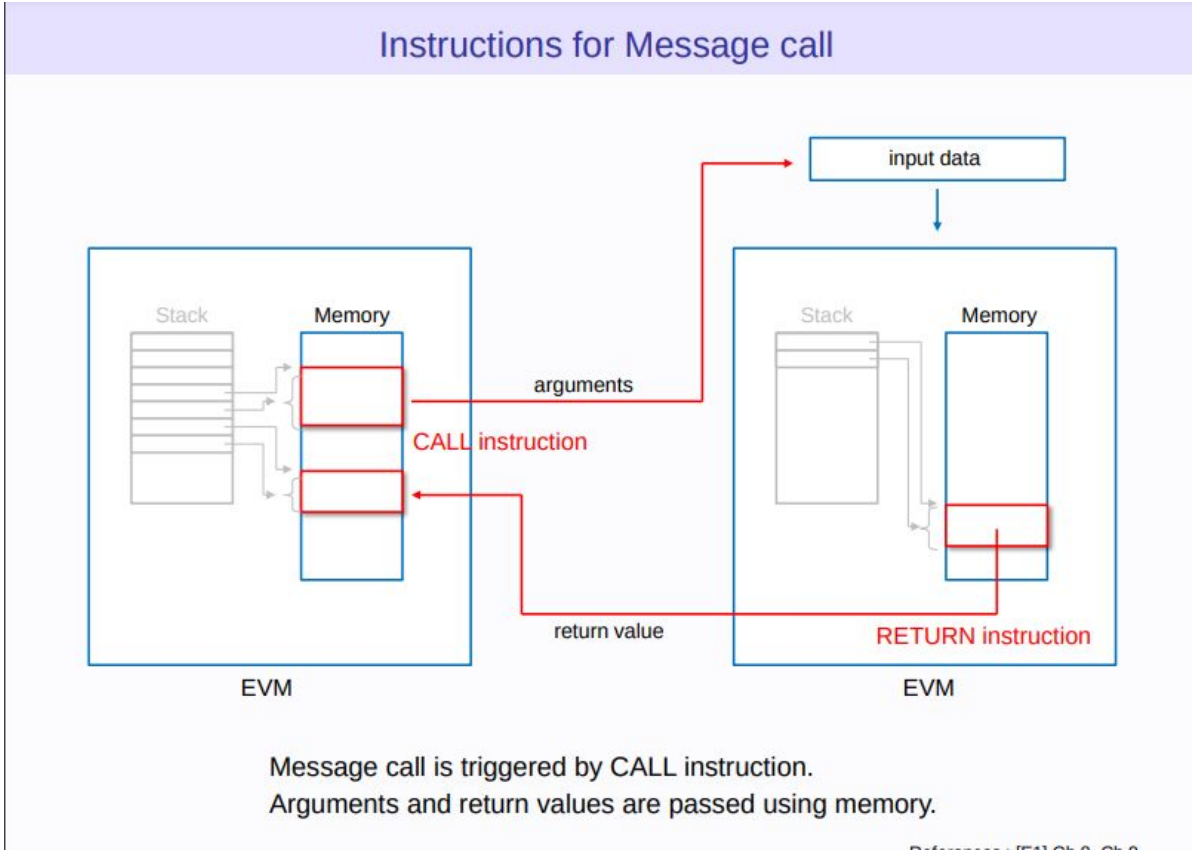
Components of Ethereum Network - Transactions

Message call

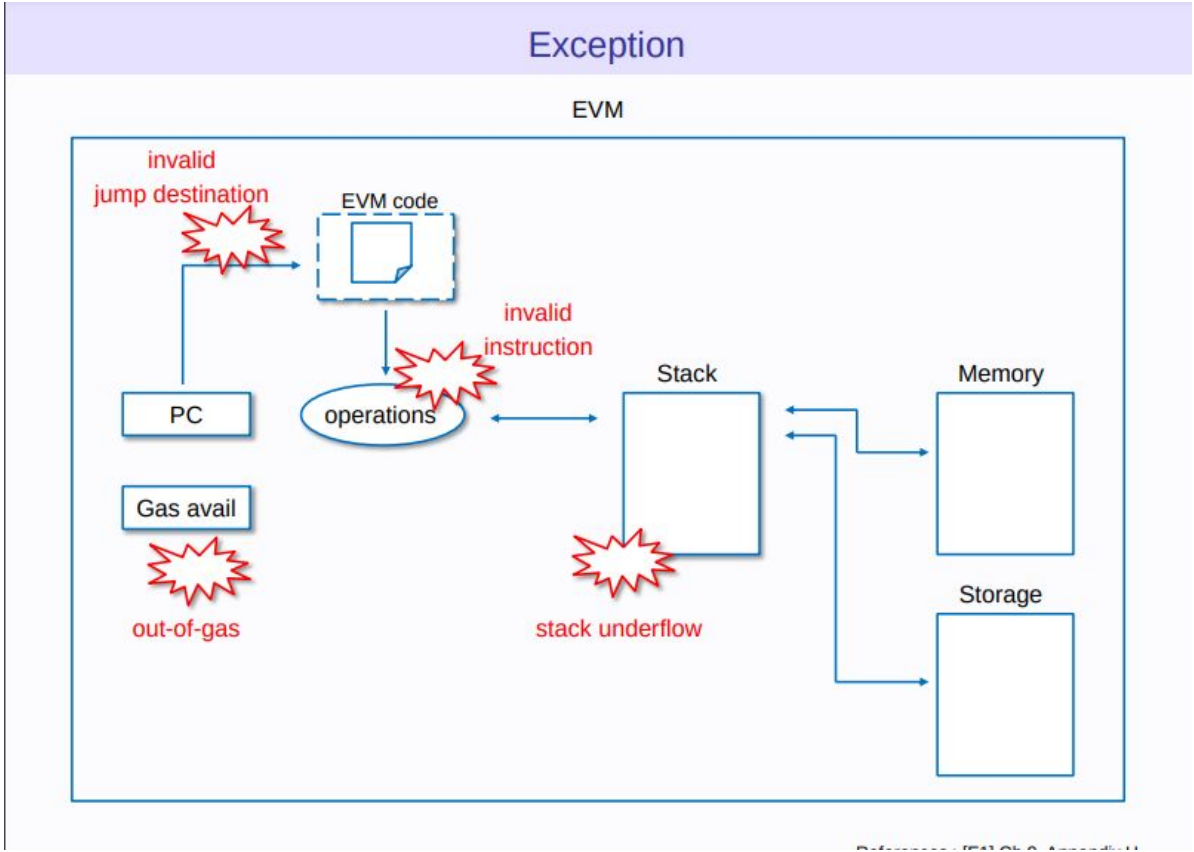


EVM can send a message to other account.
The depth of message call is limited to less than 1024 levels.

Components of Ethereum Network - Transactions

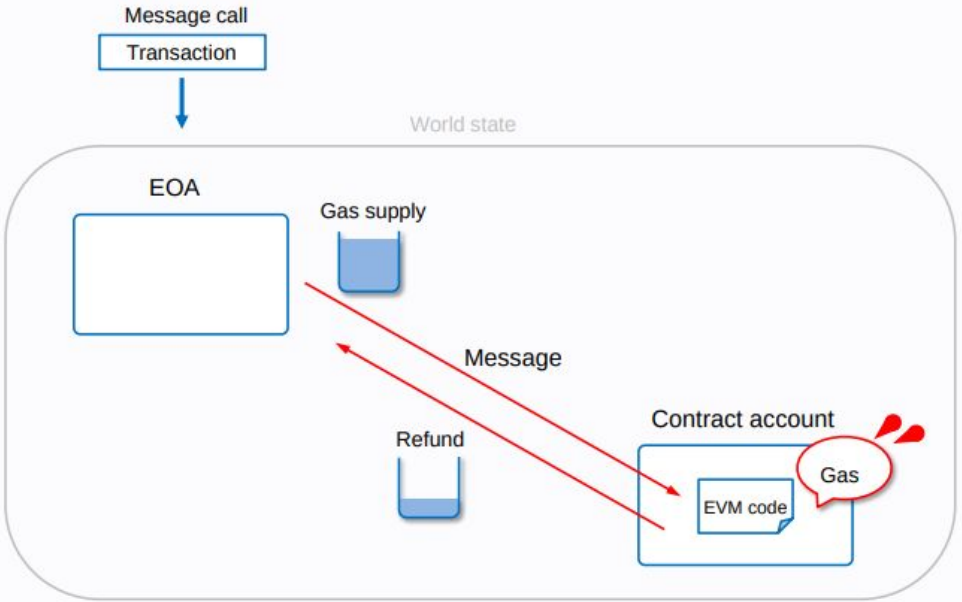


Components of Ethereum Network - Transactions



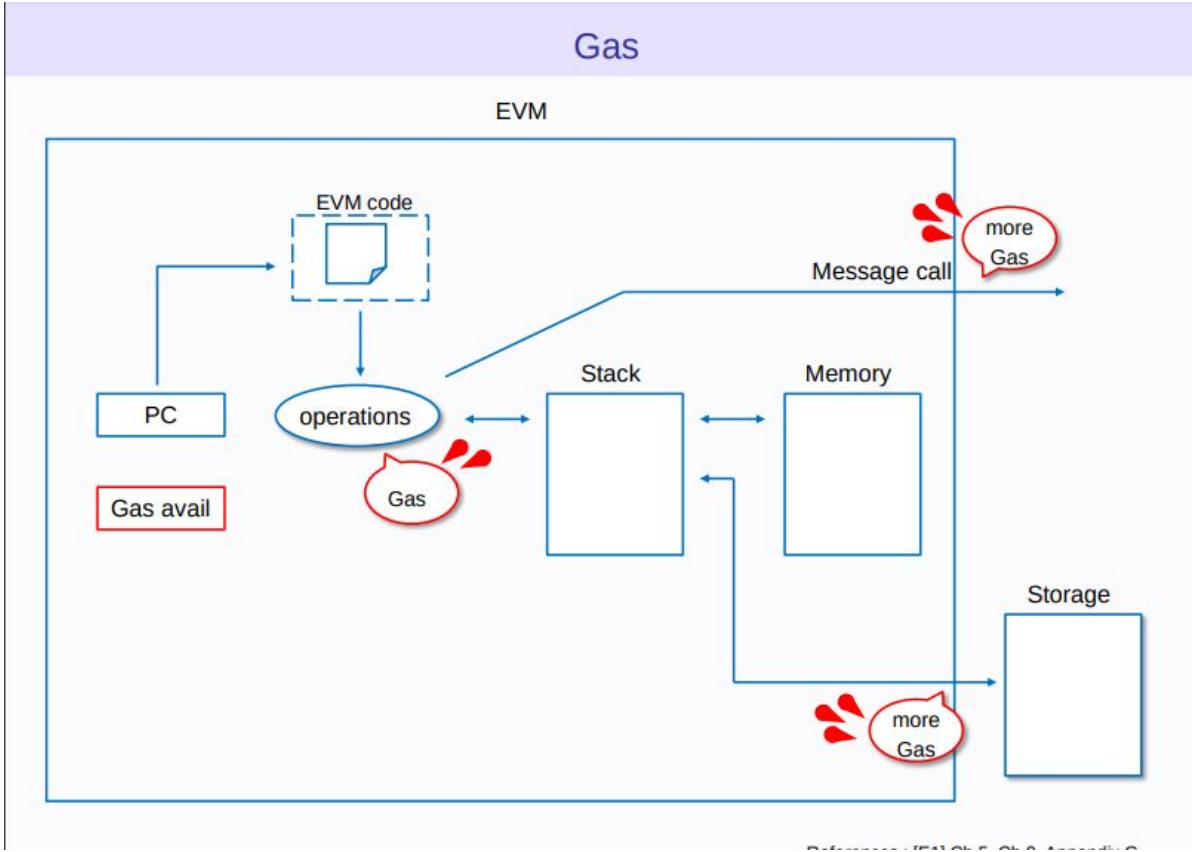
Components of Ethereum Network - Transactions

Gas and fee



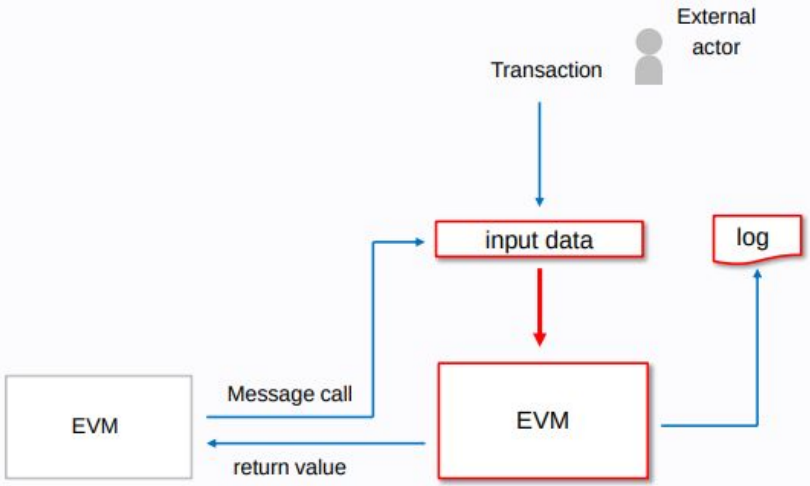
All programmable computation in Ethereum is subject to fees (denominated in gas).

Components of Ethereum Network - Transactions



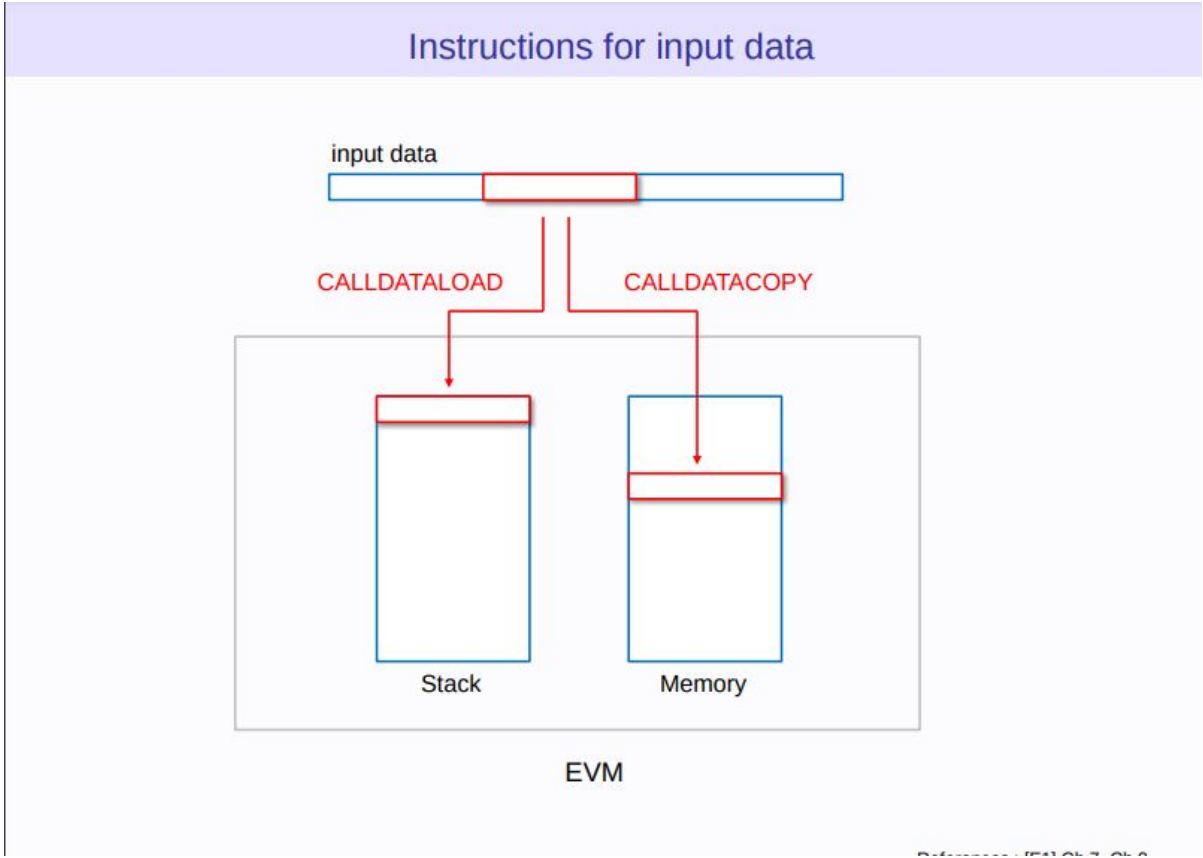
Components of Ethereum Network - Transactions

Input and Output of EVM

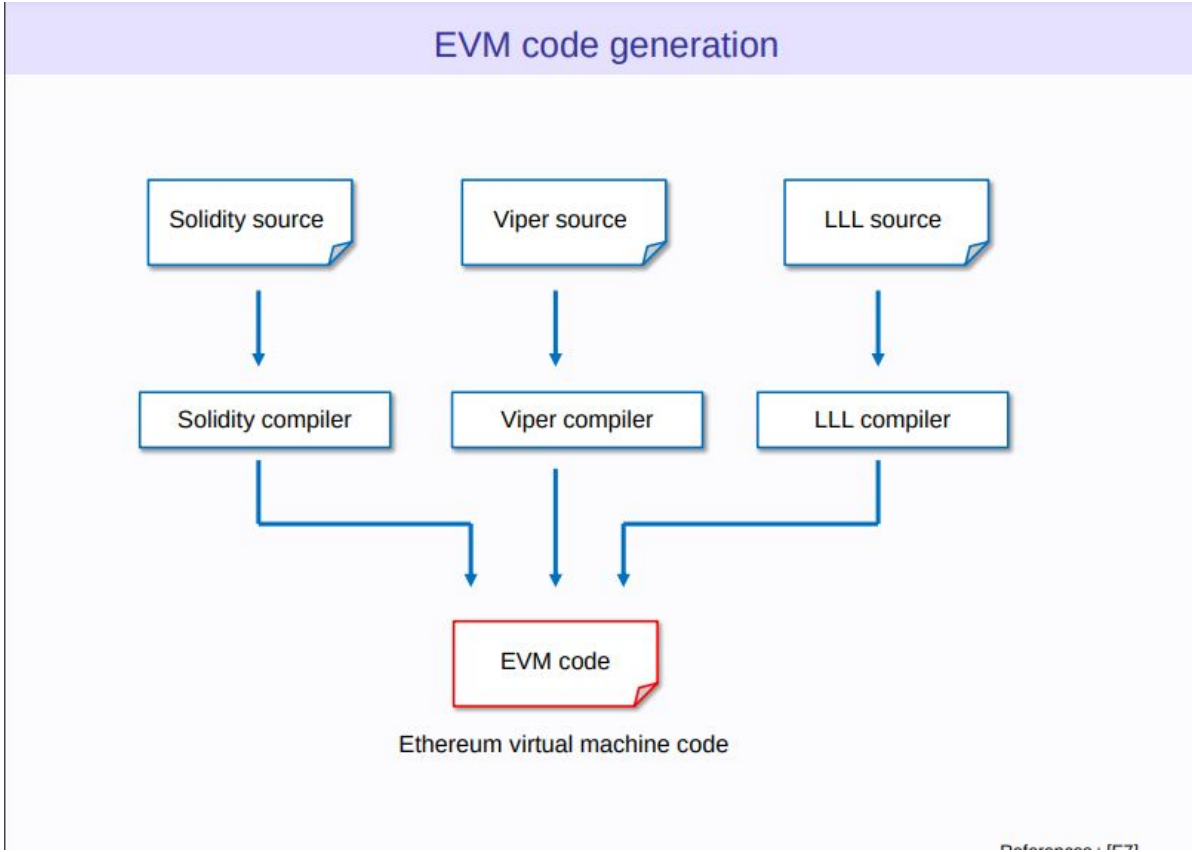


EVM can input external data from a message call.
EVM can output log. EVM can also return values to Caller EVM.

Components of Ethereum Network - Transactions

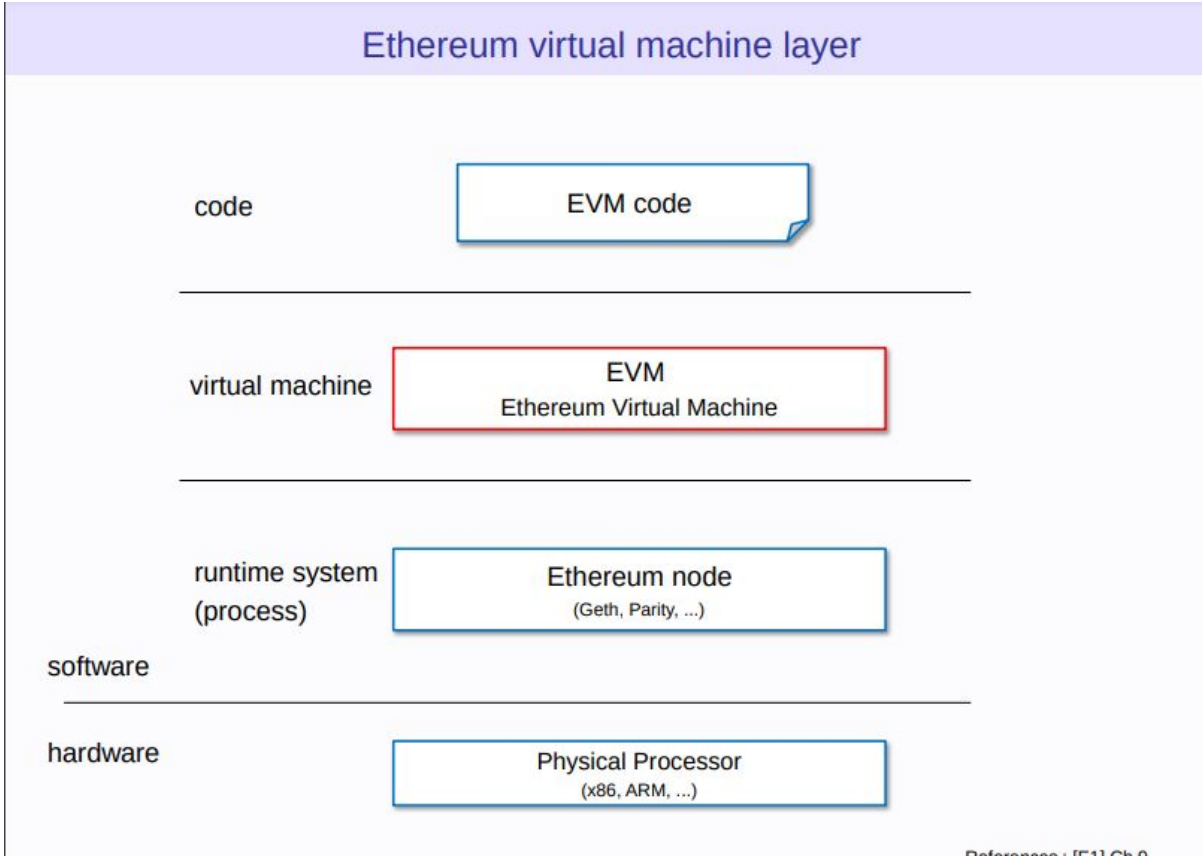


Components of Ethereum Network - EVM



Courtesy : Takenobu T

Components of Ethereum Network - EVM



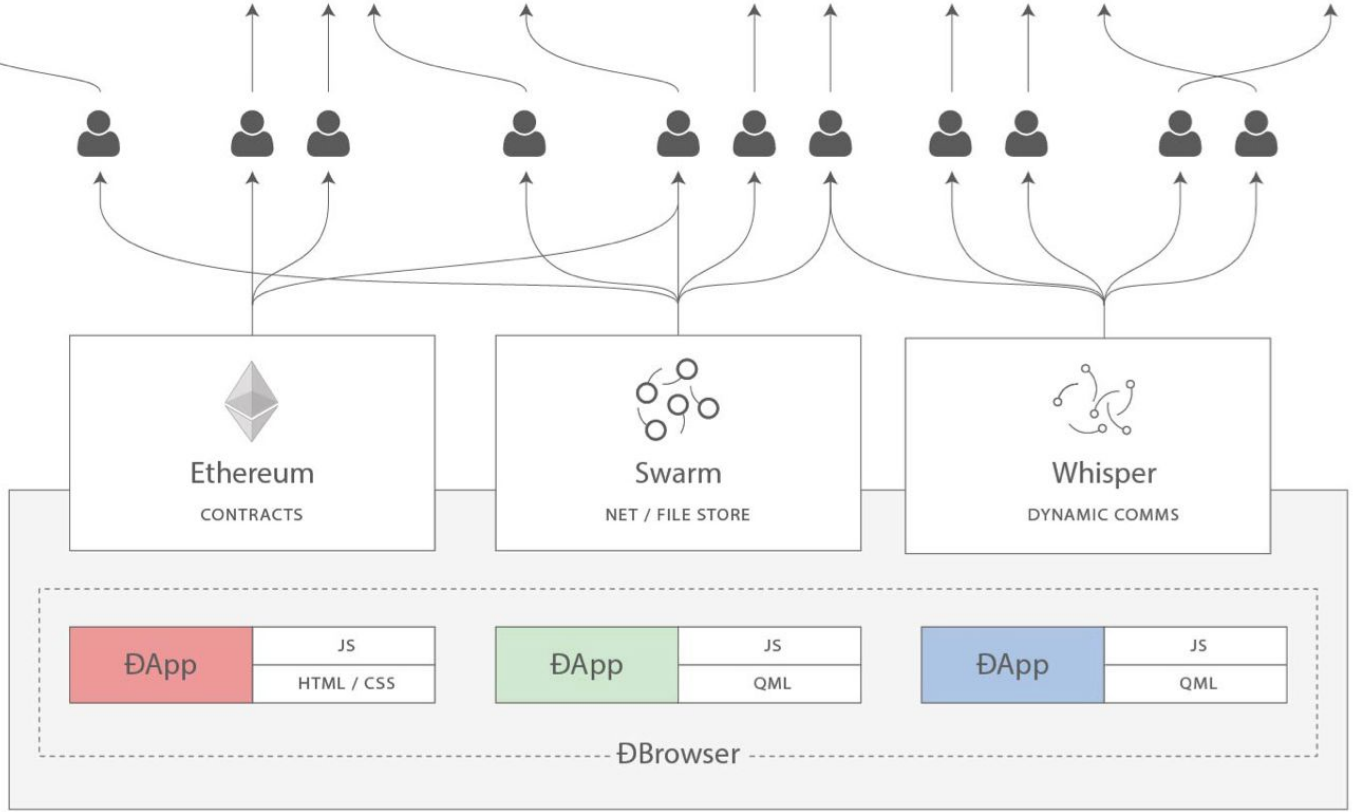
Courtesy : Takenobu T

Components of Ethereum Network - Swarm & Whisper

- Swarm and Whisper are complementary technologies contributing to the vision of Ethereum as a "world computer".
- When imagining Ethereum as a metaphor for a shared computer, it should be noted that computation alone is not enough.
- For a computer to be fully useful, it also needs storage to "remember" things and bandwidth to "communicate" them. This could be summarised as such:
  - **Contracts**: decentralized logic
  - **Swarm**: decentralized storage
  - **Whisper**: decentralized messaging

Courtesy : Ethereum Stackexchange

Components of Ethereum Network - Swarm & Whisper



Interaction including Ethereum contracts, Swarm storage, Whisper comms

Courtesy : Ethereum Stackexchange

Components of Ethereum Network - Swarm & Whisper

Swarm

- designed as an *accounting protocol* that benefits from the automatic execution of so-called "smart contracts" running on the Ethereum Virtual Machine (EVM).
- This accounting protocol is *independent of the physical storage mechanism*.
- That is, it is **not intrinsically tied to a specific storage system**.
  - It could be IPFS, BitTorrent, or some future technology not yet invented.
- it is part of the vision of a fully decentralized web.

Courtesy : Ethereum Stackexchange
Oreilly - Whisper
Etehreum Blog

Components of Ethereum Network - Swarm & Whisper

Whisper

- provides **decentralized peer-to-peer messaging capabilities** to the Ethereum network.
- It is an **identity based messaging system**
- It is a **communication protocol that DApps use to communicate with each other.**
- The data and routing of messages are **encrypted within Whisper communications**.
- uses the DEVp2p wire protocol for exchanging messages between nodes on the network.
- designed to be **used for smaller data transfers** and in scenarios where **real-time communication is not required.**
- **designed to provide a communication layer that cannot be traced**
- provides dark communication between parties.

Note : Blockchain can be used for communication, but that is expensive, and a consensus is not really required for messages exchanged between nodes.

- used as a **protocol that allows censor-resistant communication**.

Courtesy : Ethereum Stackexchange
Oreilly - Whisper
Etehreum Blog

Components of Ethereum Network - Swarm & Whisper

Whisper

- At a considerable cost of bandwidth and latency, whisper ⇒ deliver a 100% dark operation.
- that is **zero leakage of metadata during peer-to-peer communication**
- Normal communication protocol's ⇒ to maximize the bandwidth and minimize latency.
- Goal: **to nullify leakage of metadata and achieve true darkness, where no third party can eavesdrop while two peers are communicating**.
    - For this, whisper is **willing to give up on both bandwidth and latency constraints.**
- Whisper messages are ephemeral (short lived) and have an associated time to live (TTL)

Courtesy : Ethereum Stackexchange
Oreilly - Whisper
Etehreum Blog

Components of Ethereum Network - Swarm & Whisper

Whisper

- Allows nodes in the network communicate with each other.
- Supports broadcasting, user-to-user, encrypted messages, and so on.
- It's **not designed to transfer bulk data**.
- **deliver secure messaging between peers without writing any information to the blockchain**.
- It was **part of the DevP2P wire protocol** but is **now deprecated**.

Courtesy : Ethereum Stackexchange
Oreilly - Whisper
Etehreum Blog

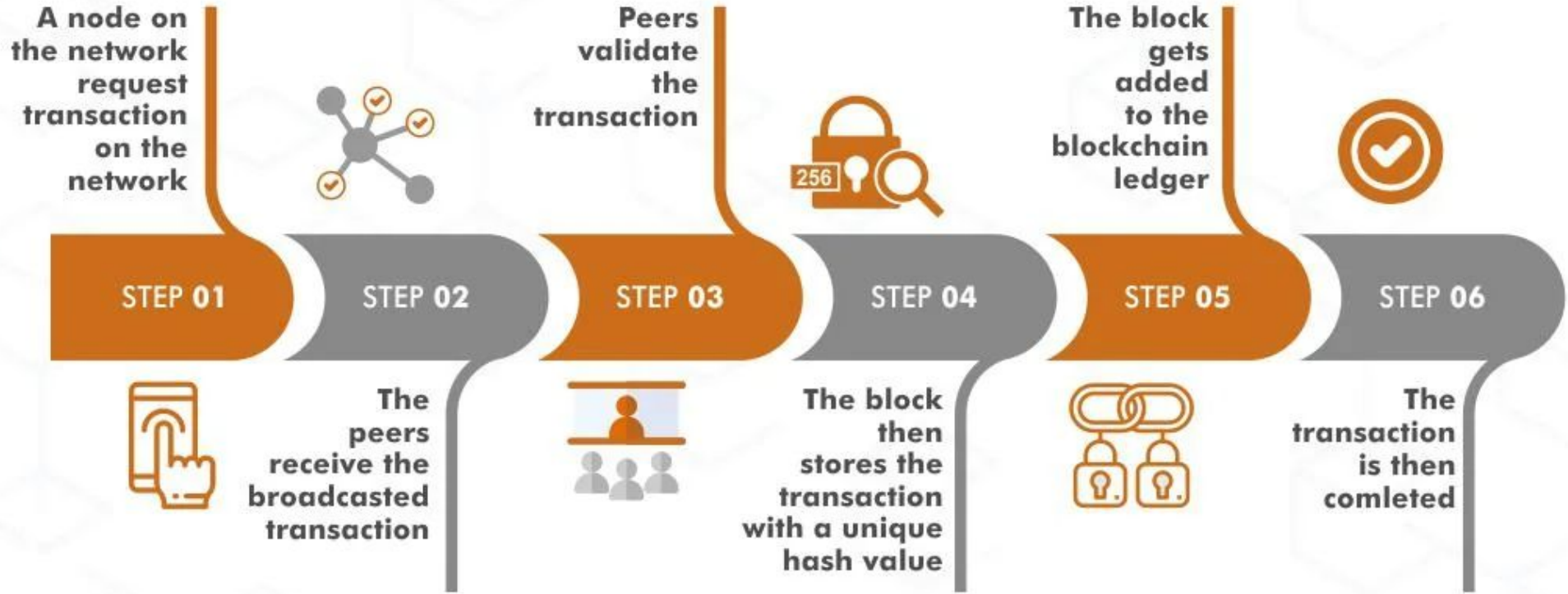
How Does Ethereum Work?

Ethereum **implements an execution environment** called **Ethereum Virtual Machine (EVM).**

1. When a **transaction triggers a smart contract** all the nodes of the network will execute every instruction.
2. **All the nodes** will **run EVM** for **block verification**, where the nodes will go through the transactions listed in the block and runs the code as triggered by the transaction in the EVM.
3. **All the nodes** on the network must **perform the same calculations for keeping their ledgers in sync.**
4. Every transaction must include:
   a. **Gas limit.**
   b. **Transaction Fee** that the sender is willing to pay for the transaction.
5. If **total amount of gas needed to process the transaction < = the gas limit** then the *transaction will be processed*
6. if the **total amount of the gas > the gas limit** then the *transaction will not be processed* *and* the fees are still lost.

Thus it is **safe to send transactions with the gas limit above the estimate to increase the chances of getting it processed**

Transactions in Blockchain - Life Cycle



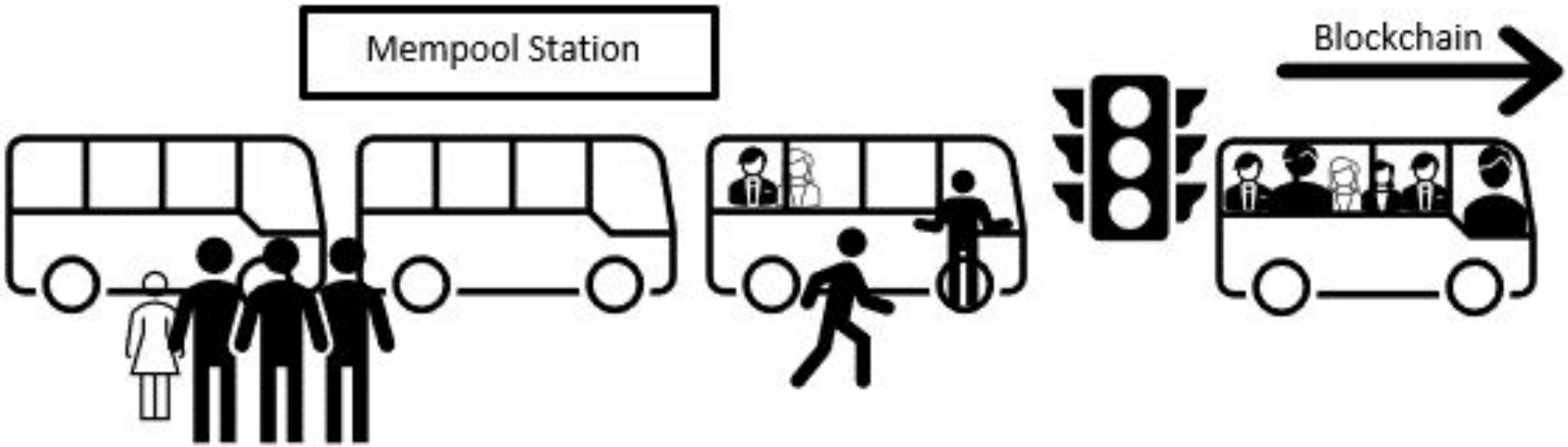
Courtesy : CBCAmerica

Transactions in Blockchain - Life Cycle

1. Someone requests a transaction. The transaction could involve cryptocurrency, contracts, records, or other information.
2. **Transaction is broadcast to all P2P** participation computers in the specific blockchain network. These are called **Nodes**. All transactions are published to the **Mempool** or memory pool, where they are considered 'pending'. **Gas fees** are paid by users as part of the transaction to compensate for the computing energy required to process and validate transactions on the blockchain.
3. **Miners** verify the transaction. Every computer in the network checks the transaction against some validation rules that are set by the creators of the specific blockchain network.
4. **Validated transactions** are stored into a block and are sealed with a lock referred to as the **Hash**.
5. **New block is added to the existing Blockchain**. This block becomes part of the blockchain when other computers in the network validate if the lock on the block is correct.
6. The transaction is complete. Now the transaction is part of the blockchain and cannot be altered in any way.

Courtesy : Ledger.com

Transactions in Blockchain - The Bus Station Analogy

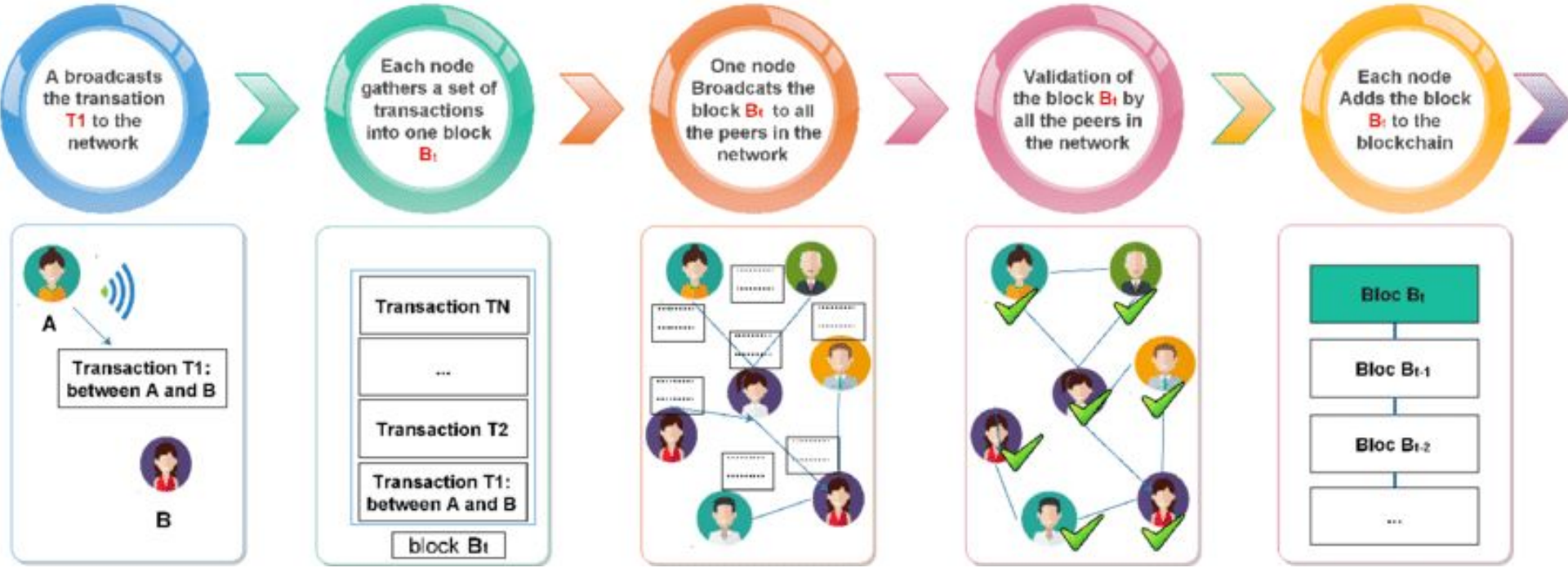


Courtesy : TxStreet

Transactions in Blockchain - Live Demo

Courtesy : TxStreet

Transactions in Blockchain - Example



Courtesy : Research Gate

Transactions in Blockchain - Example

- Alice wants to send two coins to Bob.
- Each transaction has **three main parts**:
  - **The input:** Alice's private coin address, she wants to spend.
  - **The output:** Bob's public key or coin address.
  - **Amounts:** the amount of coins Alice wants to spend.

1. **Alice signs a message with the transaction details using her private key.** The message contains the input, output, and amount to be sent.
2. The **transaction is then broadcast to the network** saying the amount of coins in her account should go down by two. The amount in Bob's account should increase by two.
3. **Each computer in the network will receive the message** and apply the requested transaction to its copy of the ledger, updating the account balances.
4. **Add the transactions into the MemPool.**
5. **Miner** select few transactions from MemPool and tries to **solve Cryptographic Puzzle.**
6. On solving the Puzzle, the **Block is broadcasted to the network for validation**.
7. After adding Block into the Blockchain, **Transactions added in the Block are later removed from MemPool**

Courtesy : Research Gate

End-End Transaction in Ethereum

Sam wants to send $$ to Mark

Transaction reaches the network

The nodes compete and validate the transaction

A new block with transaction is generated

Both Accounts are updated

The Block is replicated and stored on all nodes

Courtesy : Packt Pub

Information in a Submitted Transaction in Ethereum

- from – the address of the sender, that will be signing the transaction. This will be an externally-owned account as contract accounts cannot send transactions.
- recipient – the receiving address (if an externally-owned account, the transaction will transfer value. If a contract account, the transaction will execute the contract code)
- signature – the identifier of the sender. This is generated when the sender's private key signs the transaction and confirms the sender has authorized this transaction
- nonce - a sequentially incrementing counter which indicates the transaction number from the account
- value – amount of ETH to transfer from sender to recipient (denominated in WEI, where 1ETH equals 1e+18wei)
- input data – optional field to include arbitrary data
- gasLimit – the maximum amount of gas units that can be consumed by the transaction. The EVM specifies the units of gas required by each computational step
- maxPriorityFeePerGas - the maximum price of the consumed gas to be included as a tip to the validator
- maxFeePerGas - the maximum fee per unit of gas willing to be paid for the transaction (inclusive of baseFeePerGas and maxPriorityFeePerGas)

Courtesy : Packt Pub

Benefits of Ethereum

1. **Availability:**
   - As the Ethereum network is decentralized so **there is no downtime**.
   - Even if one node goes down other computing nodes are available.
2. **Privacy:** Users don't need to enter their personal credentials while using the network for exchanges, thus **allowing them to remain anonymous**.
3. **Security**: Ethereum is **designed to be unhackable,** as the hackers have to get control of the majority of the network nodes to exploit the network.
4. **Less ambiguity**: The **smart contracts that are used as a basis for trade and agreement** on Ethereum ensure stronger contracts than traditional contracts which require follow-through and interpretation.
5. **Rapid deployment:** On Ethereum decentralized networks, enterprises can easily deploy and manage private blockchain networks instead of coding blockchain implementation from scratch.
6. **Network size**: Ethereum network **can work with hundreds of nodes and millions of users.**
7. **Data coordination:** Ethereum **decentralized architecture better allocates information** so that the **network participants don't have to rely on a central entity** to manage the system and mediate transactions.

Drawbacks of Ethereum

- **Complicated programming language:** Learning solidity from programming smart contracts on Ethereum can be challenging and one of the main concerns is the scarcity of beginner-friendly classes.
- **Volatile cryptocurrency:** Ethereum investing can be risky as the price of Ether is very volatile, resulting in significant gains as well as a significant loss.
- **Low transaction rate**:
  a. **Bitcoin** has an average transaction rate of **7TPS**
  b. **Ethereum** has an average speed of **15 TPS** which is almost double that of bitcoin but it is still not enough.

Online Materials

1. Mastering Ethereum : Building Smart Contracts and DApps - Andreas M. Antonopoulos ,Dr. Gavin Wood
2. https://github.com/ethereumbook/ethereumbook
3. https://antonopoulos.com/
4. http://gavwood.com/
5. https://www.coding-bootcamps.com/blog/ethereum-architecture-and-components.html
6. https://subscription.packtpub.com/book/data/9781839213199/11/ch11lvl1sec77/components-of-the-ethereum-ecosystem
7. https://cointelegraph.com/learn/architectural-components-of-the-ethereum-blockchain-what-are-they
8. https://iq.opengenus.org/components-of-ethereum/
9. https://www.geeksforgeeks.org/components-of-ethereum-network/
10. https://cointelegraph.com/learn/ethereum-upgrades-a-beginners-guide-to-eth-2-0
11. https://www.simplilearn.com/tutorials/blockchain-tutorial/what-is-ethereum