



Blockchain Sem VII

HBCC701 : Blockchain Development

Module - 4 : Public Blockchain (8 Hours)

Instructors : Dr. Nupur Giri & Mrs. Lifna C S





Topics to be covered



- **Introduction to Public Blockchain,**
- **Ethereum and its Components -**
 - **Ethereum Virtual Machine (EVM),**
 - **Transaction,**
 - **Accounts,**
- **Architecture and Workflow,**
- **Mining in Ethereum,**
- **Comparison between Bitcoin and Ethereum**
- **Types of test-networks used in Ethereum,**
- **Transferring Ethers using Metamask, Mist Wallet,**
- **Ethereum frameworks,**
- **Case study of Ganache for Ethereum blockchain,**
- **Exploring etherscan.io and ether block structure**



Introduction to Public Blockchain

- does not have restrictions.
- Anyone with an internet connection can get access to the network and start validating blocks and sending transactions.
- Offers incentive for users who validate the blocks.
- The network tends to use Proof of Work or Proof of Stake consensus algorithms for validating the transactions.
- It is a “Public” network in a true sense.
- It was the model that Satoshi Nakamoto suggested back in 2009.
- Later, enterprise companies, tweaked the nature of the decentralized ledger and introduced the private blockchains.
- Anyone can download the protocol anytime, without any permission from anyone.
- it's completely decentralized, no single organization controls the ecosystem.
- Whereas, a private blockchain can be changed and altered by the owning organization.
- A public blockchain surpassed the necessity of a third party.
- A self-governed, purely decentralized and autonomous digital public ledger.



Characteristics:

- It is an **open network and permissionless**
 - where nodes can join and leave without the permission of anyone.
 - Every node has access to read and write on the ledger
 - Anyone can download and add nodes to the system
 - the ledger is shared and transparent.
- Includes a **protocol of incentive mechanism**
 - to ensure the correct operation of the blockchain system.
- It is **secure to the 51% rule.**
- It **offers anonymity**, which means no one can track your transactions back to you
- **No regulation** hence no limit to how one can use the platform for betterment.
- The technology is **fully decentralized in nature**
- **Everyone can change current business models** through the reduction in the use of middlemen.
- It is **not necessary to maintain servers or system administrators.**
- Hence there is **considerable cost reduction for the businesses.**





Introduction to Public Blockchain

Advantages:

- **Transparency:**
 - Public blockchains are transparent and open for everyone to access.
 - This makes the ledger accessible to all,
 - eliminating chances of corruption and ensuring transparency.
- **Security:**
 - Public blockchains are designed to operate with maximum security.
 - The decentralized nature of the network makes it difficult for hackers to compromise the system.
- **Empowerment:**
 - Allows all participants to validate transactions without any central authority overlooking their actions.
- **Immutability:**
 - meaning no one can tamper with the system, ensuring that transactions are secure.



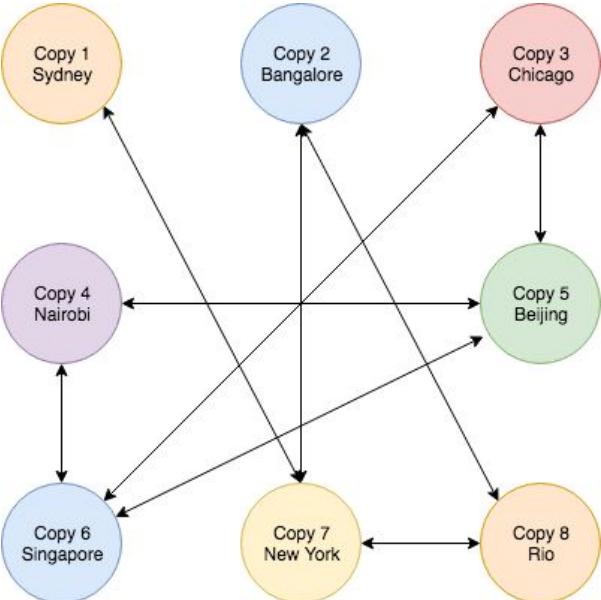
Disadvantages:

- **Power Consumption:**
 - require a lot of computational power due to their decentralized nature.
 - This increases energy consumption and can be detrimental to the environment.
- **Scalability:**
 - With more users on the blockchain, the network becomes burdened with more transactions, leading to scalability issues.
- **Conspiracy:**
 - Due to decentralized nature, no one knows who validates the transactions,
 - increasing the risk of potential conspiracy.
- **Transactions:**
 - Can be slow due to the time it takes to process all transactions on the network.
- **Acceptance:**
 - Due to the openness and transparency, it can be difficult for governments to accept them as they are not controlled by authorities



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**.





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.



- **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.



Components of Ethereum Network

1. Ethereum Node
2. Ethereum Client
3. Ether
4. Gas
5. Ethereum Accounts
6. Nonce
7. Storage Root
8. Ehash
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 discarmer transaction is when a person transfers X ETH to another person but their account contains less ETH.



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.



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 necessarily 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.**



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

FULL CLIENT



Enables the nodes to **perform all major network operations**.

LIGHT CLIENT



Performs a **subset of network operations**.
Can **validate block headers**.
Can **use Merkle proofs to verify transaction inclusion**.

REMOTE CLIENT



Relyes on **other types of clients** for network access.
Usually **offers wallet functionalities**.

DIFFERENT TYPES OF ETHEREUM NODES

are **computational devices** that participate in the Ethereum network

ETHEREUM CLIENTS

are **software applications** that implements the ethereum specifications

FULL NODE



Stores the **complete Blockchain**. Can **validate and verify** blocks.

ARCHIVE NODE



Stores the **complete Blockchain**. Stores the **blockchain state** at each block level.

LIGHT NODE



Stores the **block headers**. Can participate in **data validation**.



Components of Ethereum Network -

	Pros	Cons
Light nodes	<ul style="list-style-type: none">• Portable• Resource-efficient• User-friendly	<ul style="list-style-type: none">• Do not validate the network• Do not propagate blocks• Do not maintain consensus• Less secure
Full nodes	<ul style="list-style-type: none">• Validate the network• Propagate blocks• Maintain consensus• More secure	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• Easily trackable involvement• Can pool with others to increase reward rate	<ul style="list-style-type: none">• High and wasteful energy consumption• High equipment cost and barrier to entry
Staking	<ul style="list-style-type: none">• Low barrier to entry• Low energy consumption	<ul style="list-style-type: none">• Reward system based on luck• Low transparency in staking pools
Masternodes	<ul style="list-style-type: none">• Balanced network benefits and rewards• Lower maintenance costs	<ul style="list-style-type: none">• High initial investment• Difficult setup process





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



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:**



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.**



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.



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 the account

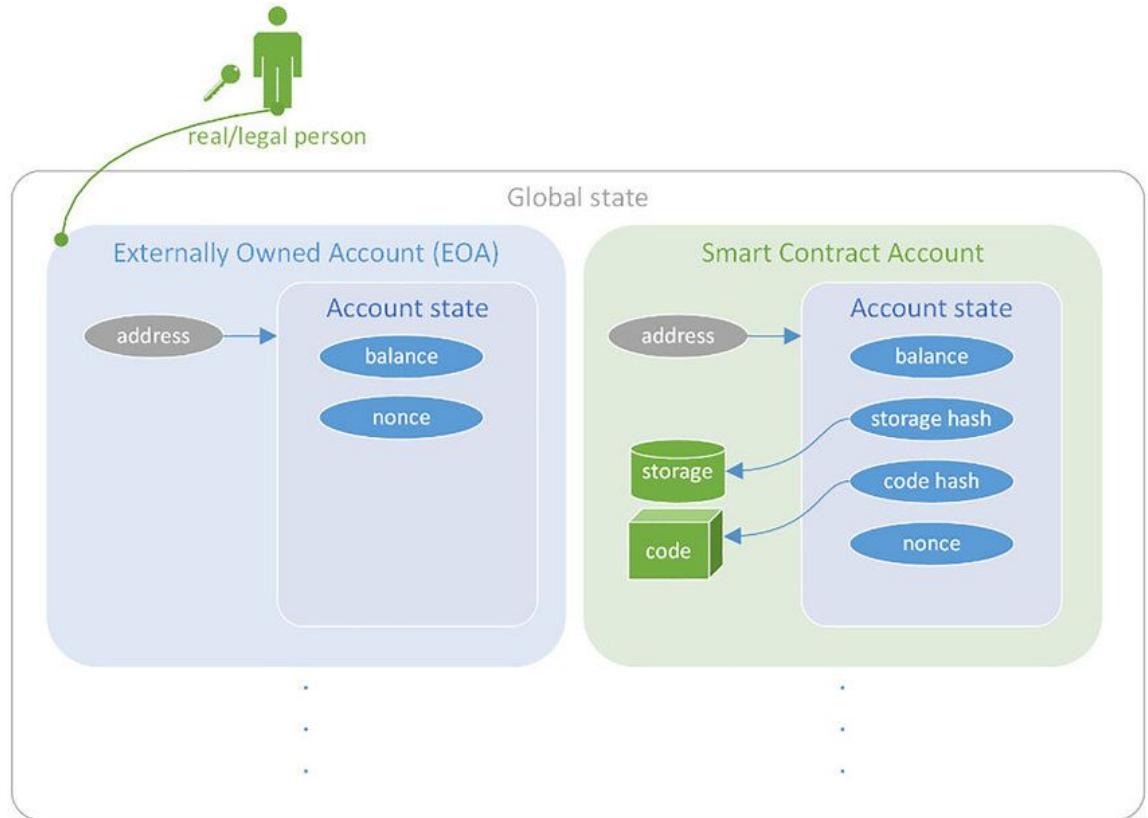


Different Fields in Ethereum Accounts

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)



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.



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. Ehash

- **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

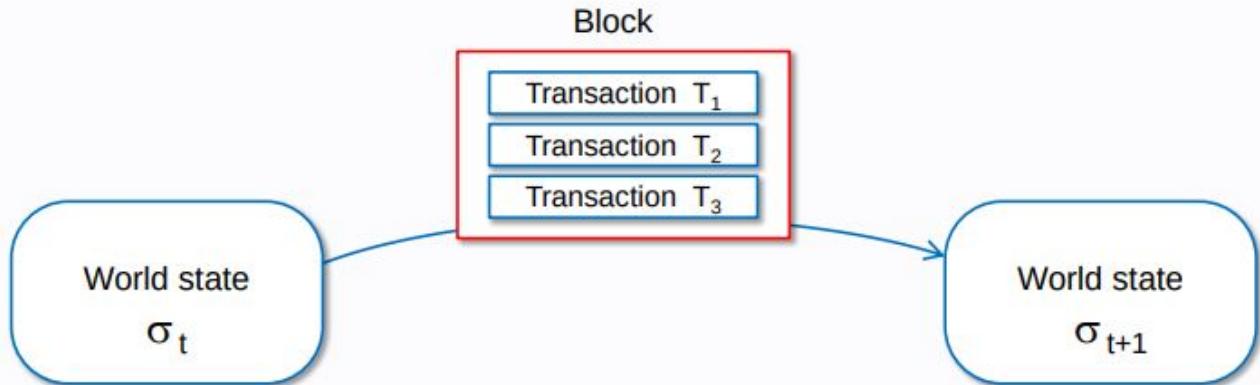


A transaction represents a valid arc between two states.



Components of Ethereum Network - Transactions

Block and transactions

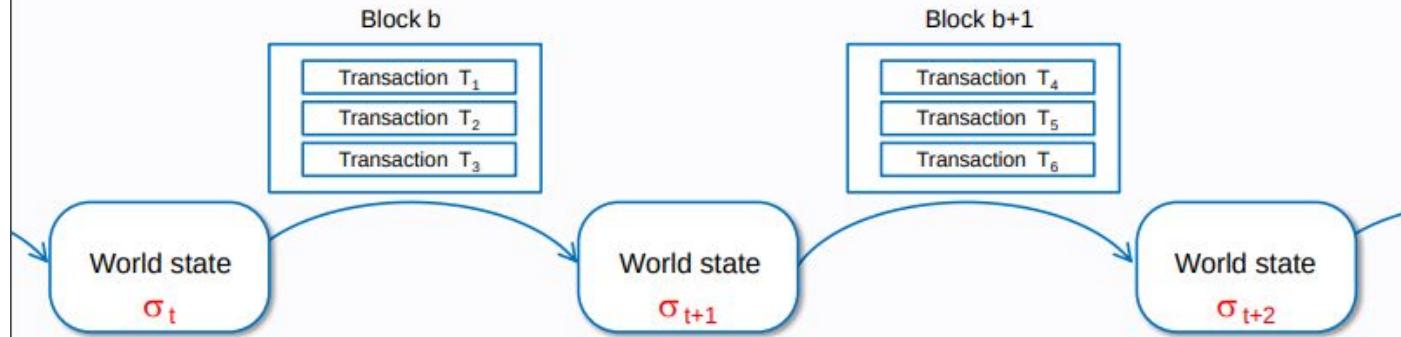


Transactions are collated into blocks.
A block is a package of data.



Components of Ethereum Network - Transactions

Chain of states

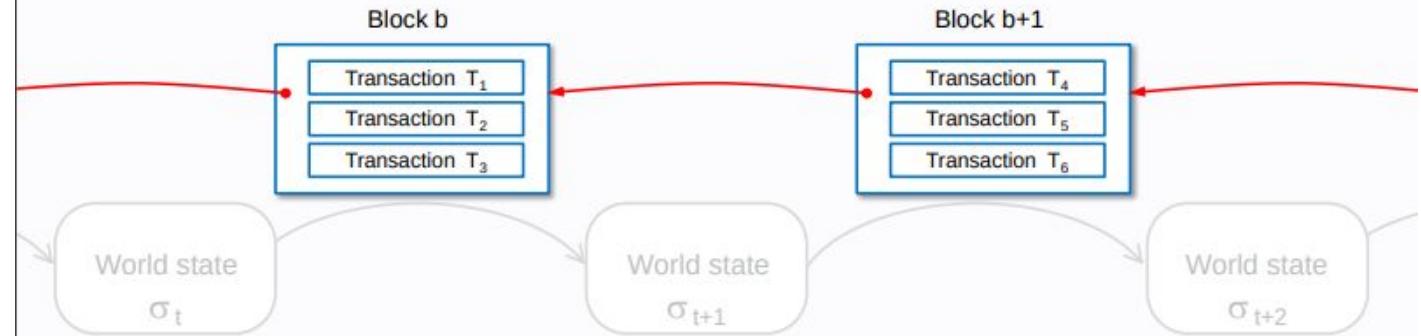


From the viewpoint of the states,
Ethereum can be seen as a state chain.



Components of Ethereum Network - Transactions

Chain of blocks: Blockchain

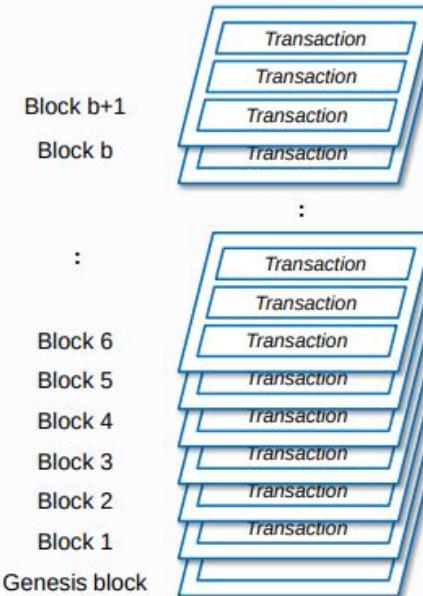


From the viewpoint of the implementation,
Ethereum can also be seen as a chain of blocks, so it is `BLOCKCHAIN`.



Components of Ethereum Network - Transactions

Stack of transactions : Ledger



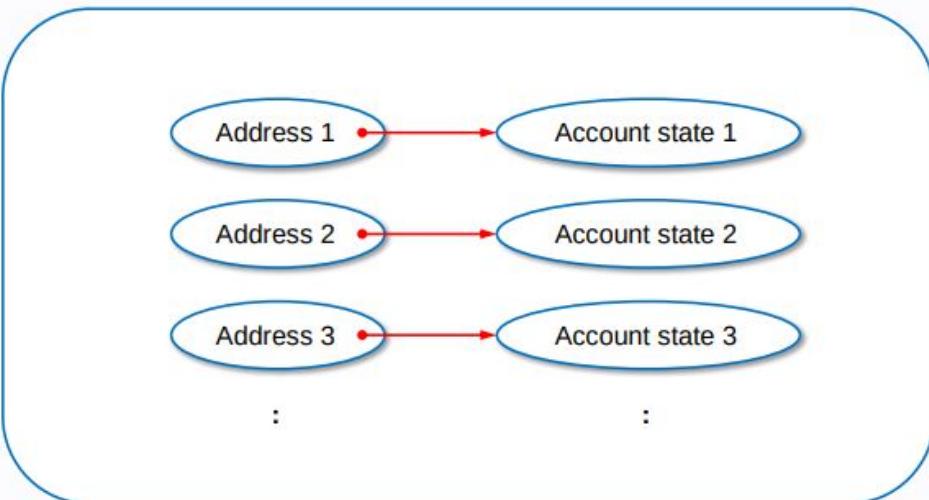
From the viewpoint of the ledger,
Ethereum can also be seen as a stack of transactions.



Components of Ethereum Network - Transactions

World state

World state σ_t



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



Components of Ethereum Network - Transactions

Several views of world state

Mapping view

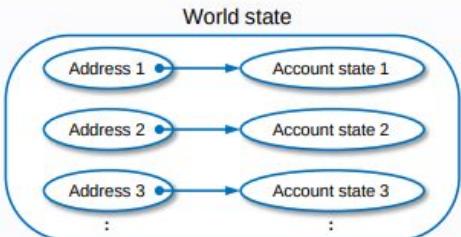
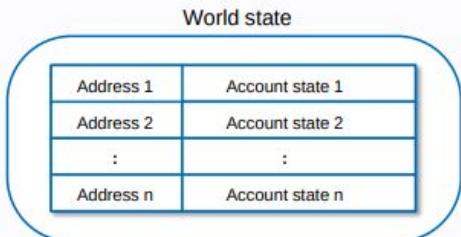
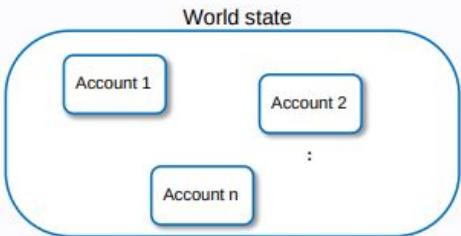


Table view



Object view



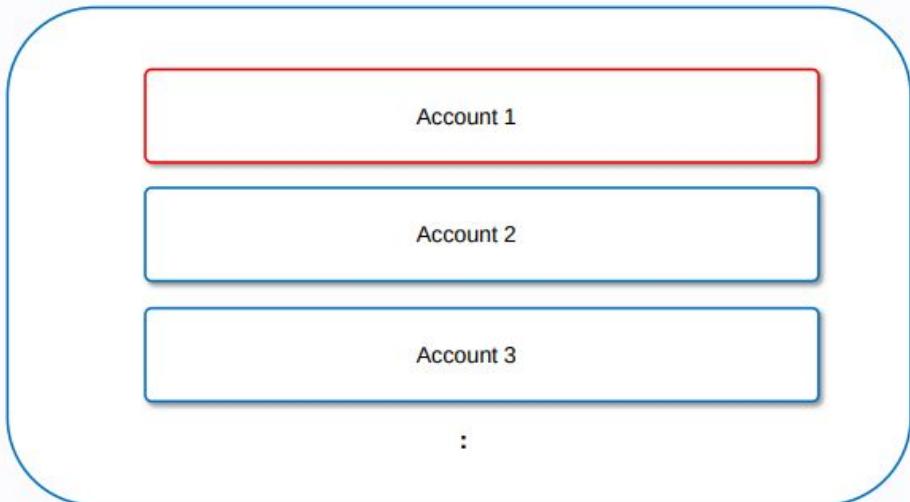
QUESTION



Components of Ethereum Network - Transactions

Account

World state



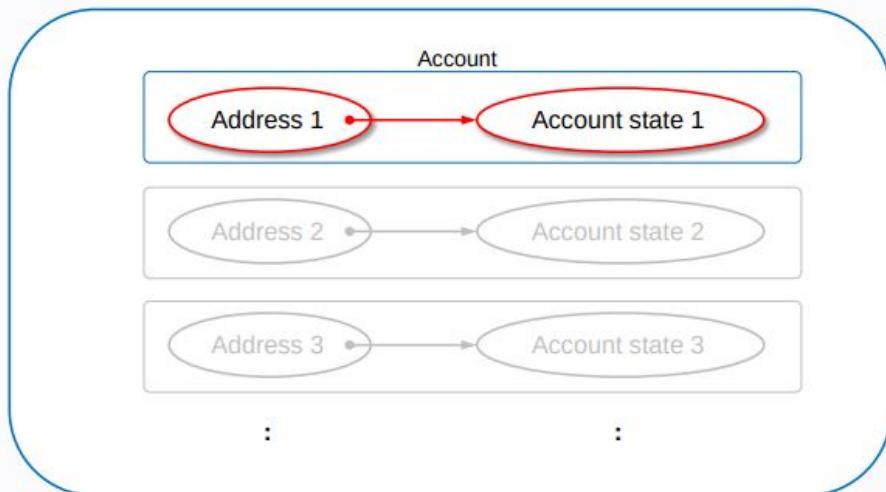
An account is an object in the world state.



Components of Ethereum Network - Transactions

Account

World state



An account is a mapping between address and account state.

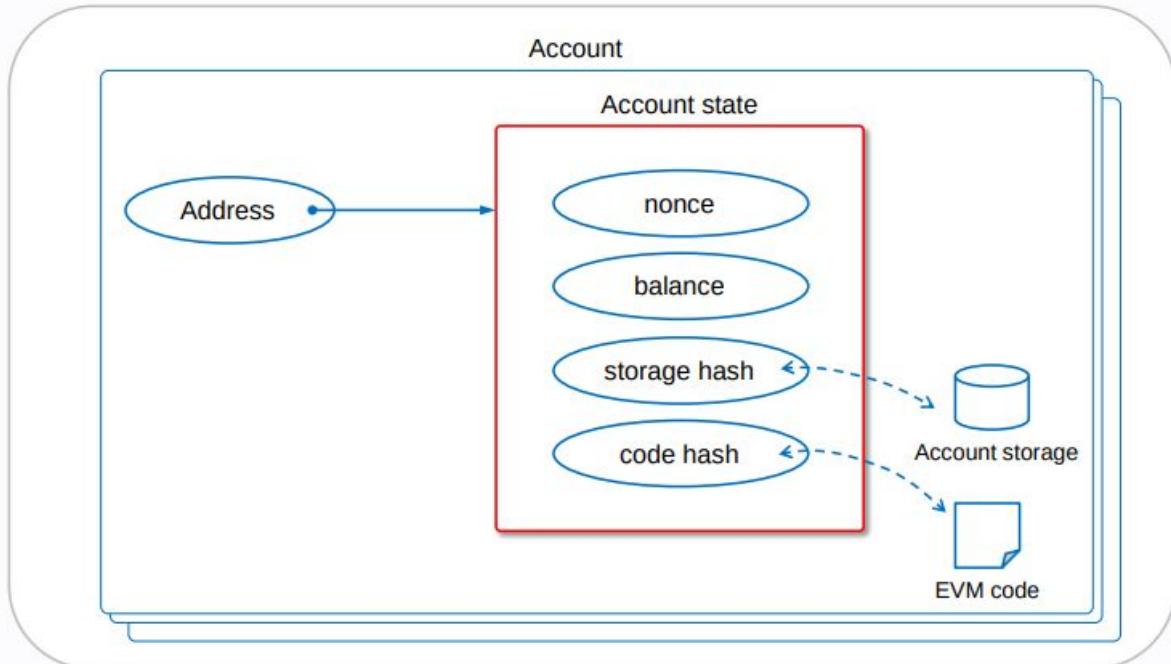
PRESENTED BY



Components of Ethereum Network - Transactions

Account state

World state



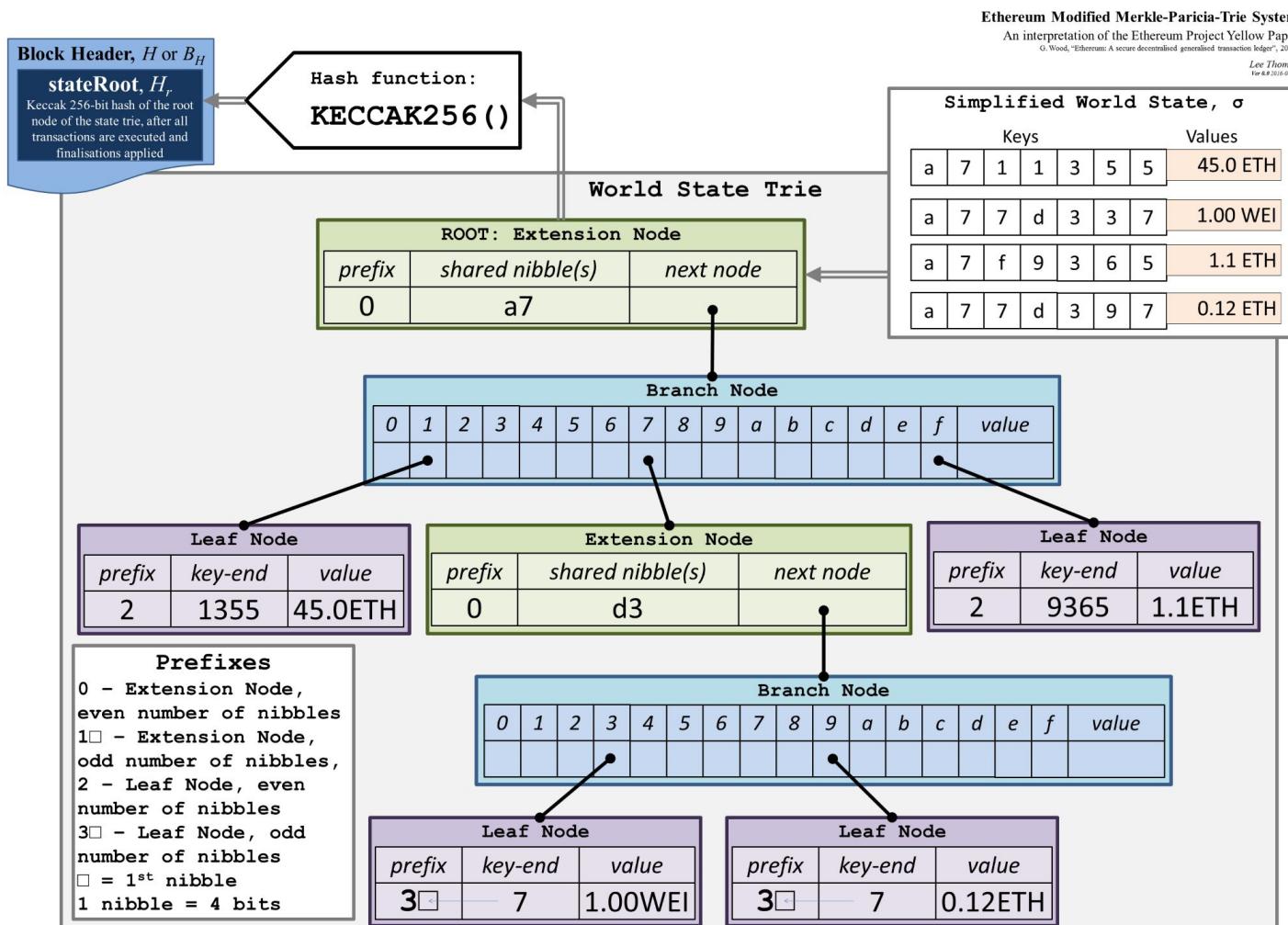
An account state could contain EVM code and storage.

QUESTION



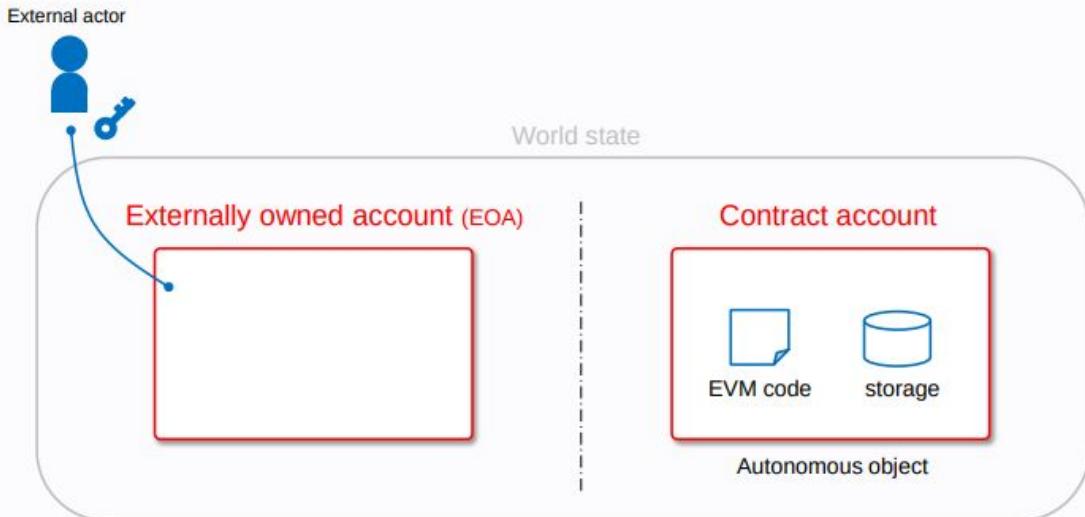
- **codeHash** – This hash refers to the *code* of an account on the Ethereum virtual machine (EVM). Contract accounts have code fragments programmed in that can perform different operations. This EVM code gets executed if the account gets a message call. It cannot be changed, unlike the other account fields. All such code fragments are contained in the state database under their corresponding hashes for later retrieval. This hash value is known as a **codeHash**. For externally owned accounts, the **codeHash** field is the hash of an empty string.
- **storageRoot** – Sometimes known as a storage hash. A 256-bit hash of the root node of a Merkle Patricia trie that encodes the storage contents of the account (a mapping between 256-bit integer values), encoded into the trie as a mapping from the Keccak 256-bit hash of the 256-bit integer keys to the RLP-encoded 256-bit integer values. This trie encodes the hash of the storage contents of this account, and is empty by default.





Components of Ethereum Network - Transactions

Two practical types of account



EOA is controlled by a private key.

Contract account contains EVM code.



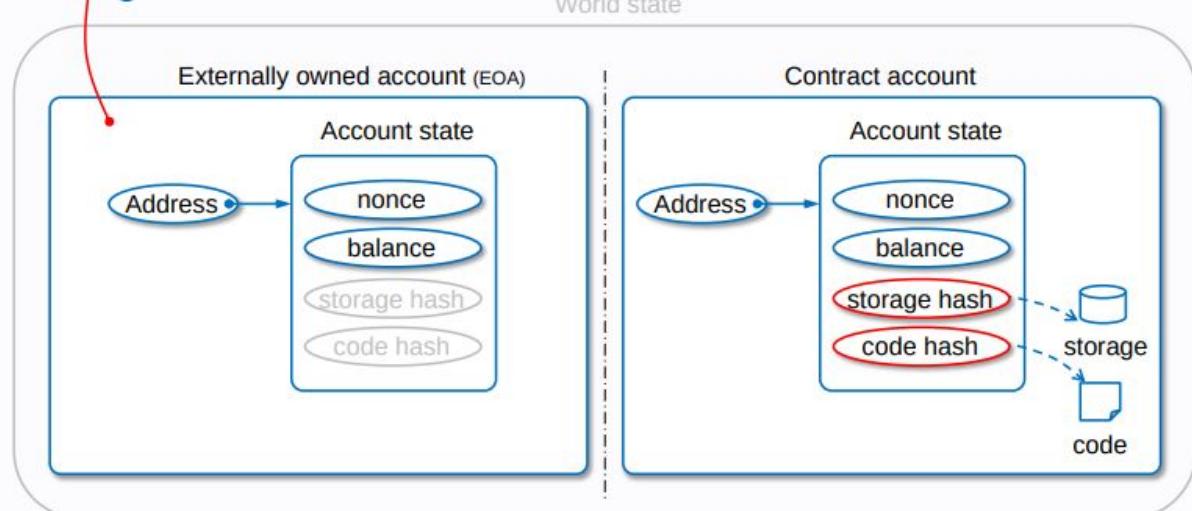
Components of Ethereum Network - Transactions

Two practical types of account

External actor



World state



EOA is controlled by a private key.
EOA cannot contain EVM code.

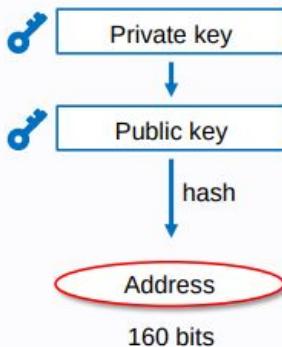
Contract contains EVM code.
Contract is controlled by EVM code.

Components of Ethereum Network - Transactions

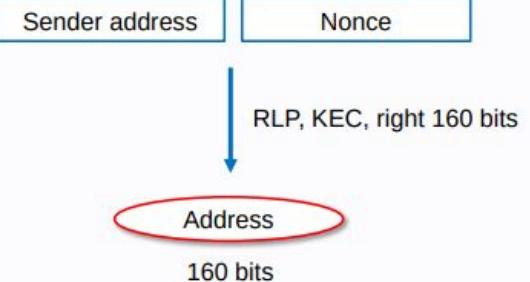


Address of account

Externally owned account (EOA)



Contract account

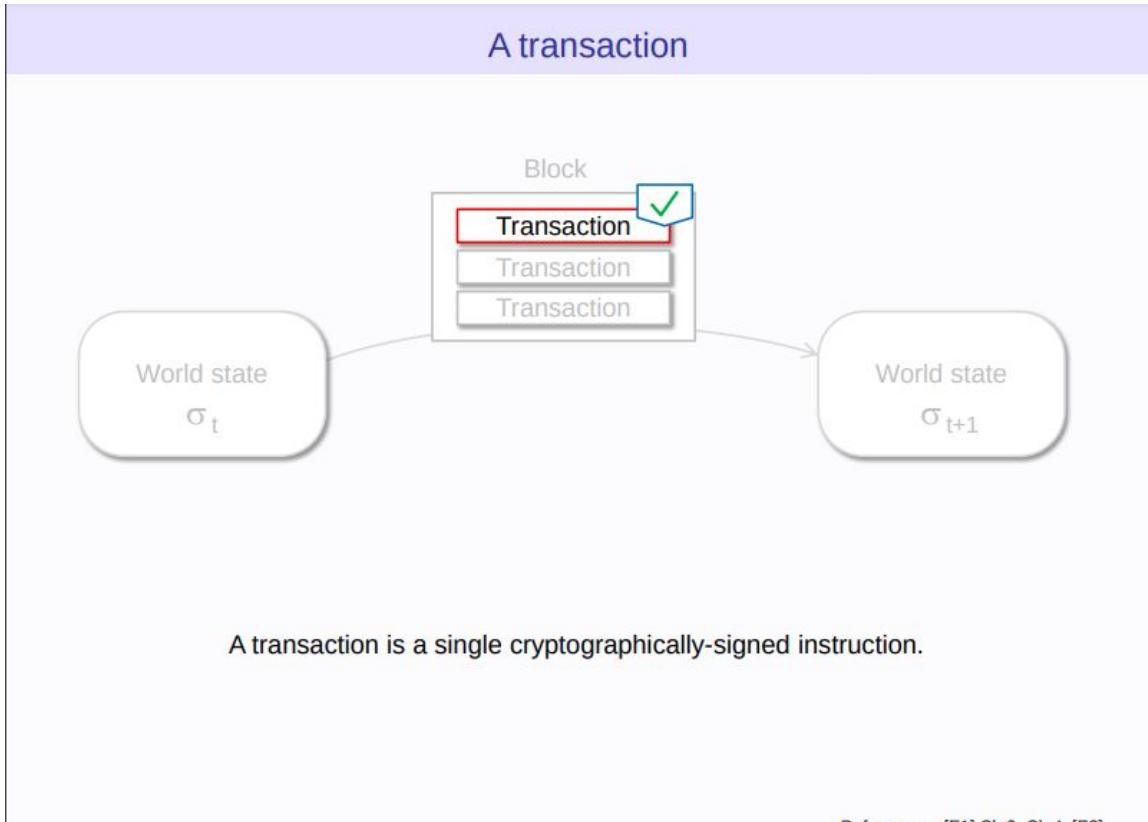


A 160-bit code used for identifying accounts.

RENDERED BY TAKENOBU T



Components of Ethereum Network - Transactions

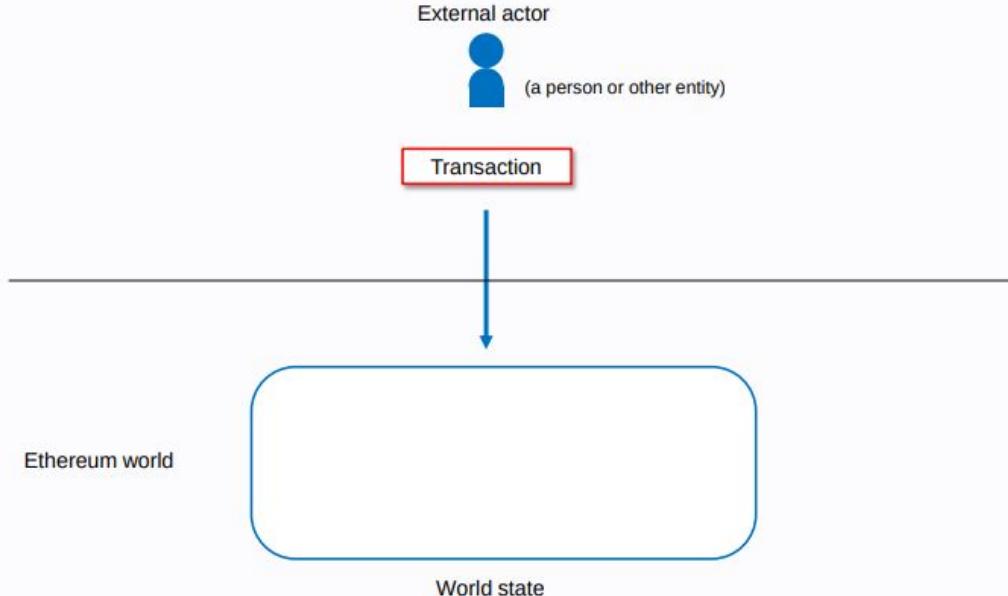


BLOCKCHAIN TECHNOLOGY



Components of Ethereum Network - Transactions

A transaction to world state



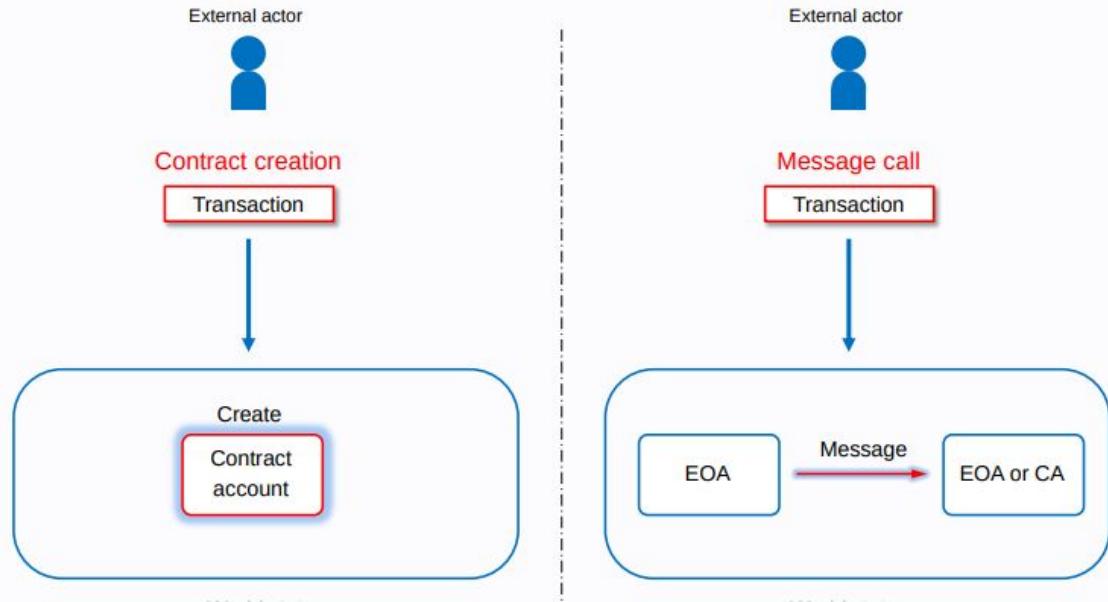
A transaction is submitted by external actor.

https://www.ethereum.org/whitepaper.pdf



Components of Ethereum Network - Transactions

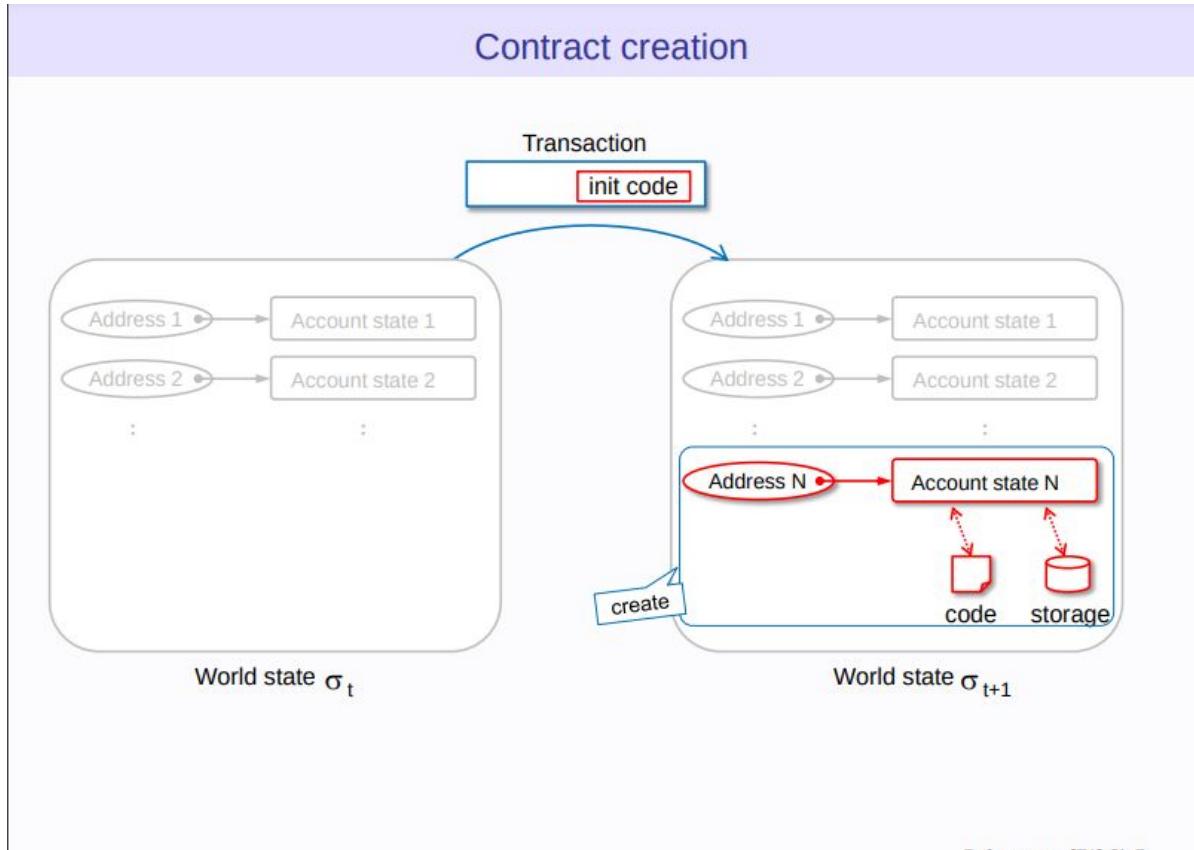
Two practical types of transaction



There are two practical types of transaction, contract creation and message call.



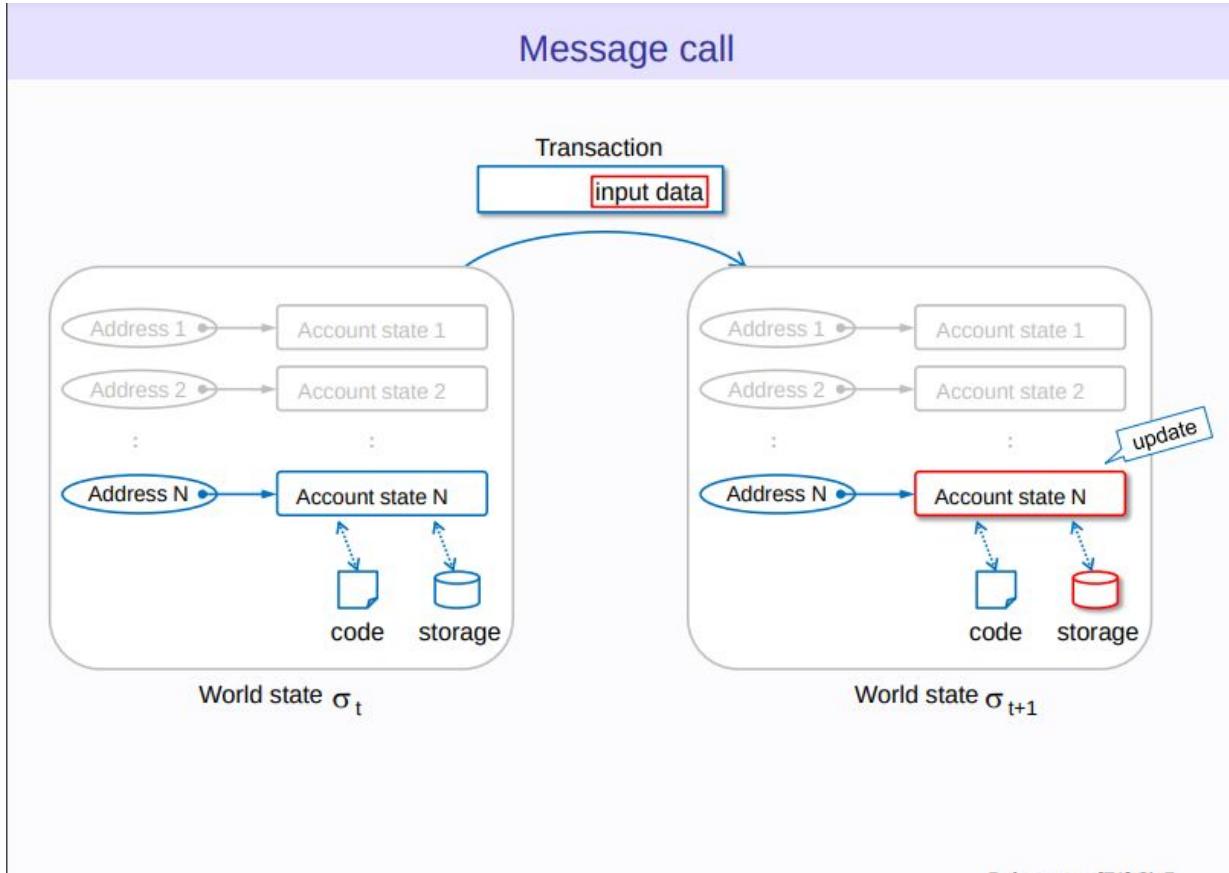
Components of Ethereum Network - Transactions



QUESTION



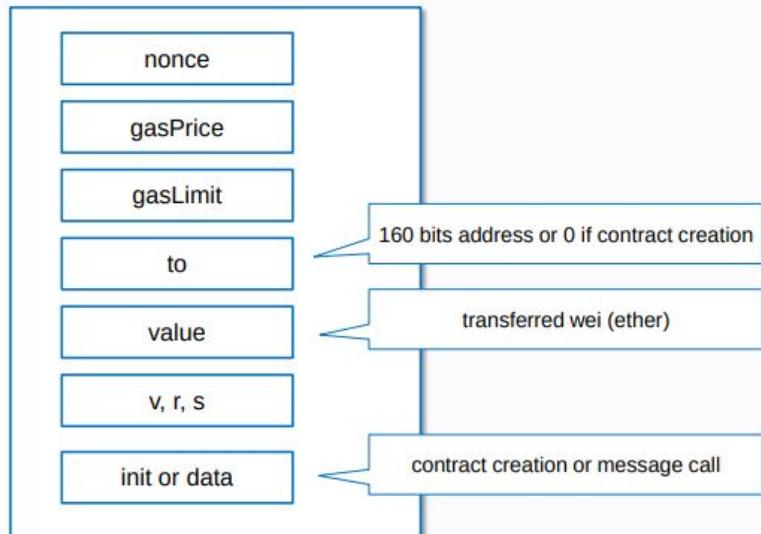
Components of Ethereum Network - Transactions



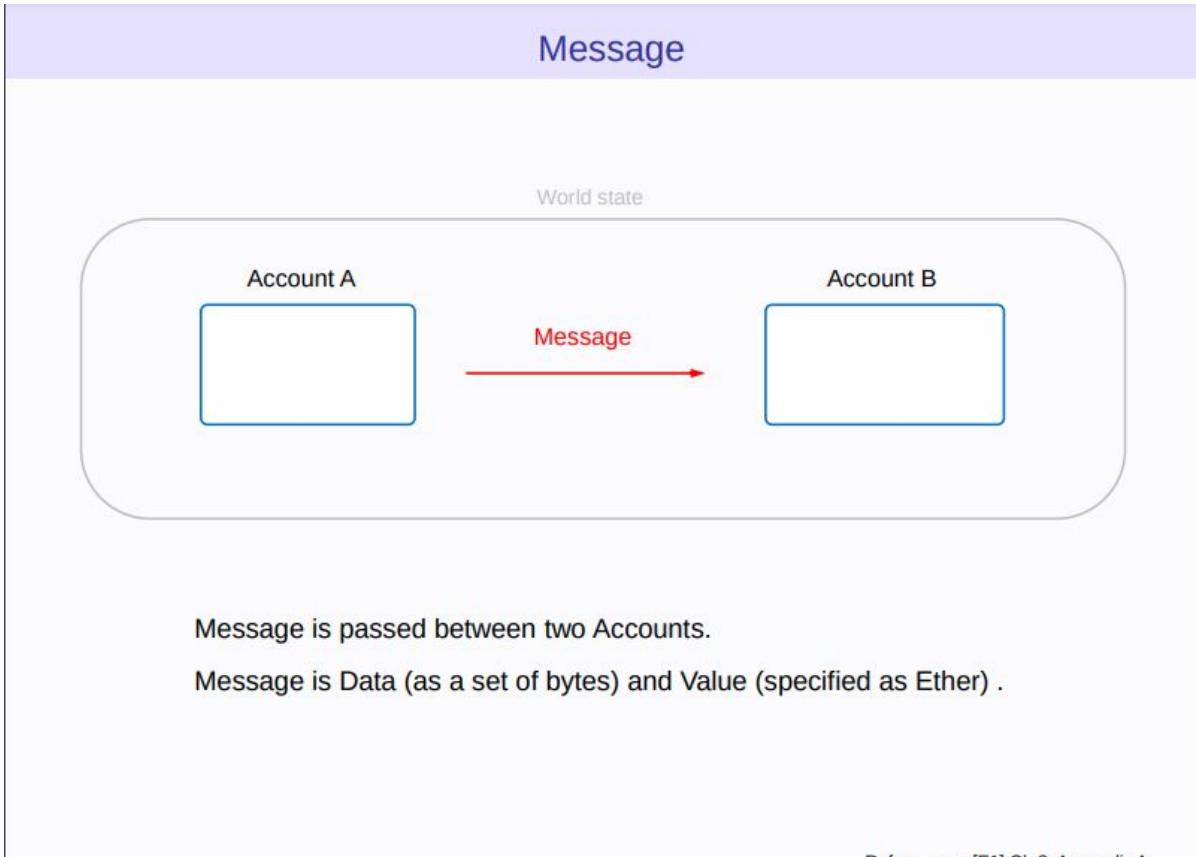
Components of Ethereum Network - Transactions

Field of a transaction

Transaction



Components of Ethereum Network - Transactions



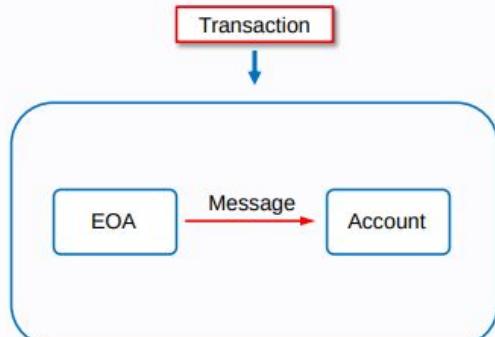
Reference : EIP-1559 Appendix A



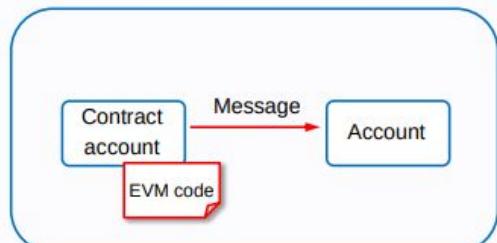
Components of Ethereum Network - Transactions

Message

Triggered by transaction



Triggered by EVM code



Transaction triggers an associated message.

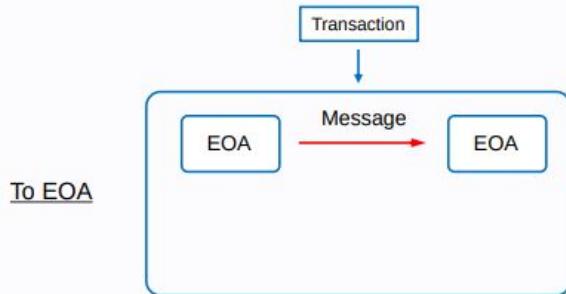
EVM can also send a message.



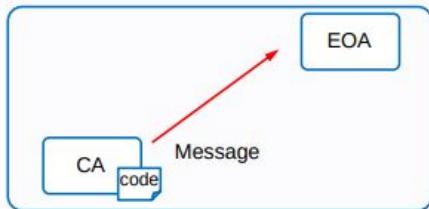
Components of Ethereum Network - Transactions

Four cases of message

By Transaction From EOA



By EVM code From CA



To EOA

To CA

Transaction

Message

EOA

CA

EOA

Message

Message

CA

Transaction

Message

EOA

CA

Message

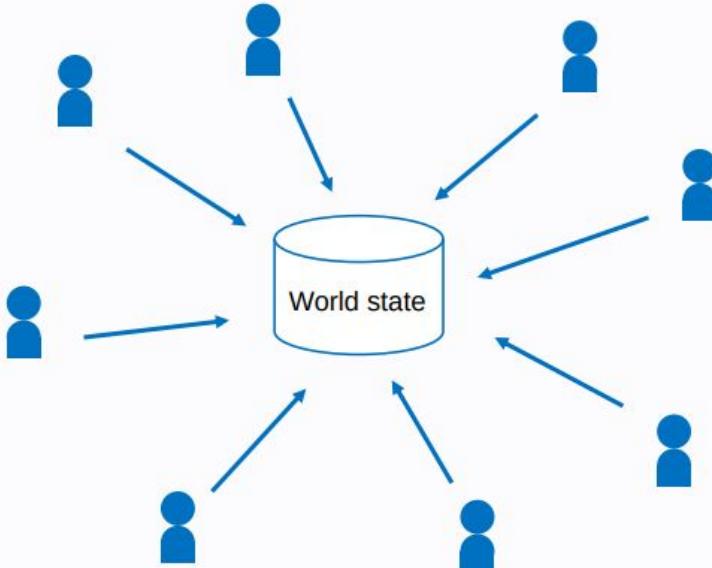
CA

QUESTION PAPER



Components of Ethereum Network - Transactions

Globally shared, transactional database



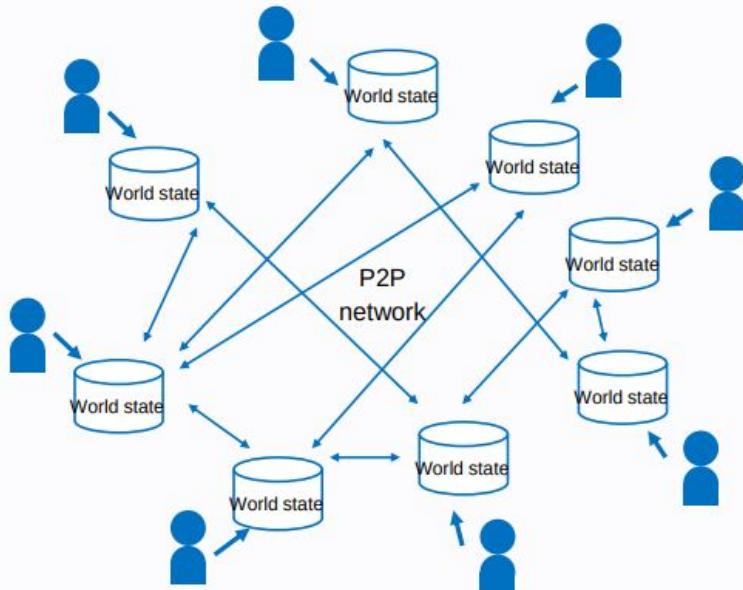
A blockchain is a globally shared, transactional database.

QUESTION



Components of Ethereum Network - Transactions

Decentralised database



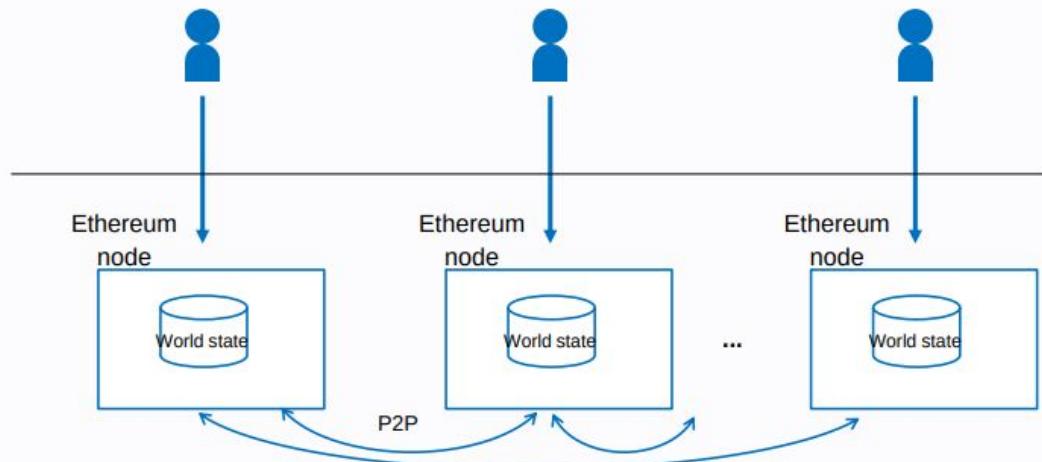
A blockchain is a globally shared, **decentralised**, transactional database.

Blockchain Development



Components of Ethereum Network - Transactions

P2P network inter nodes

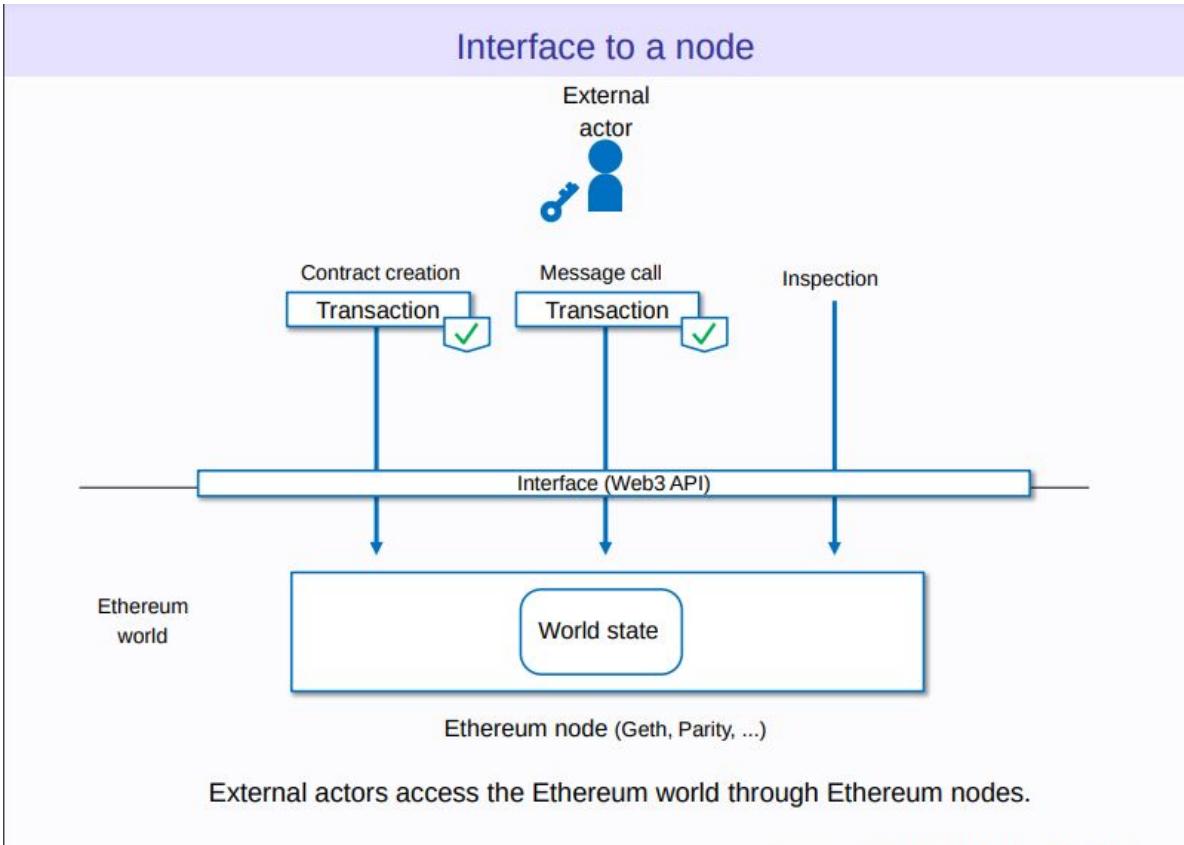


Decentralised nodes constitute Ethereum P2P network.

References : [REDACTED]



Components of Ethereum Network - Transactions



References : EIP-1 Appendix A, Ch 4, Ch 7, Ch 8



Components of Ethereum Network - Transactions

Atomicity of transaction



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

Transaction

or

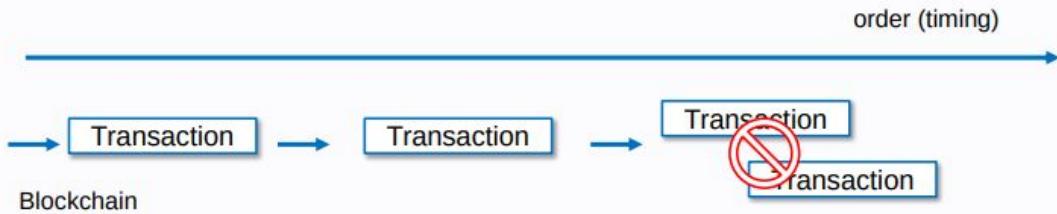
Transaction

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



Components of Ethereum Network - Transactions

Order of transactions

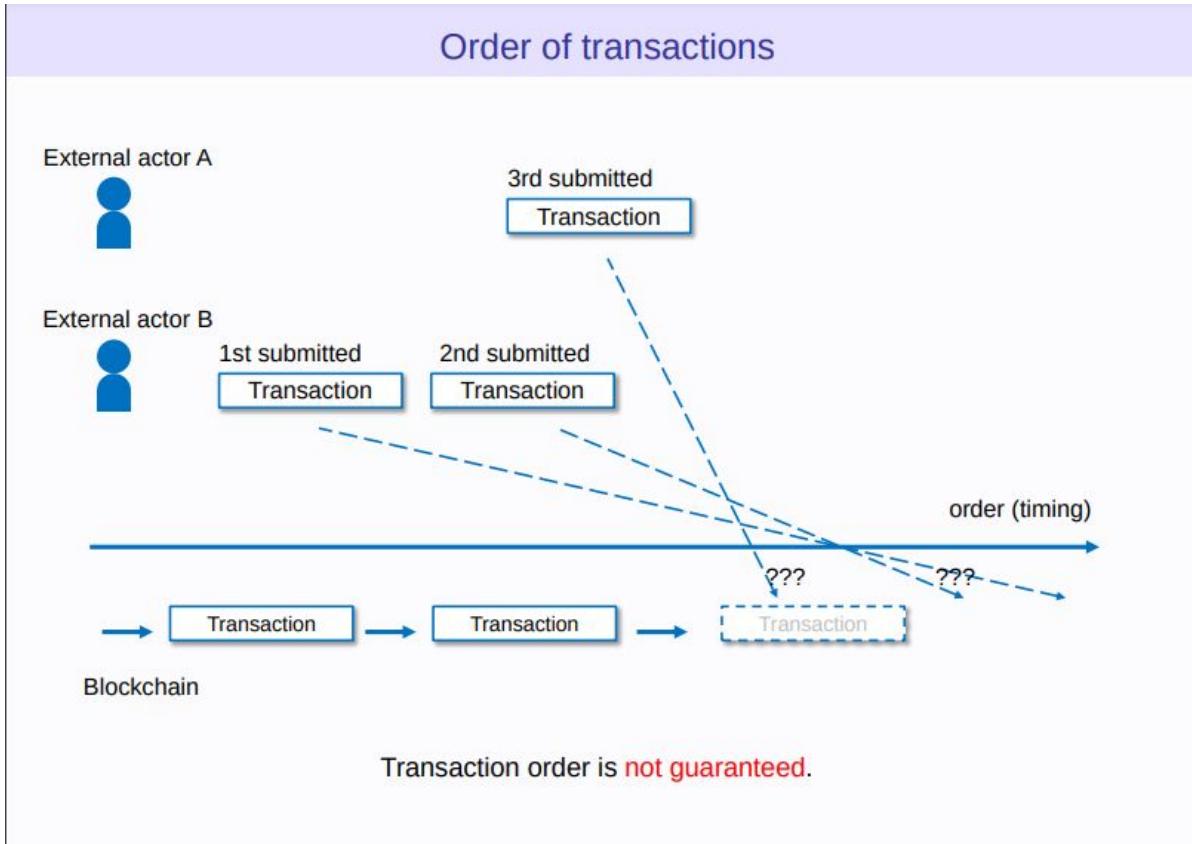


Transactions **cannot be overlapped**.

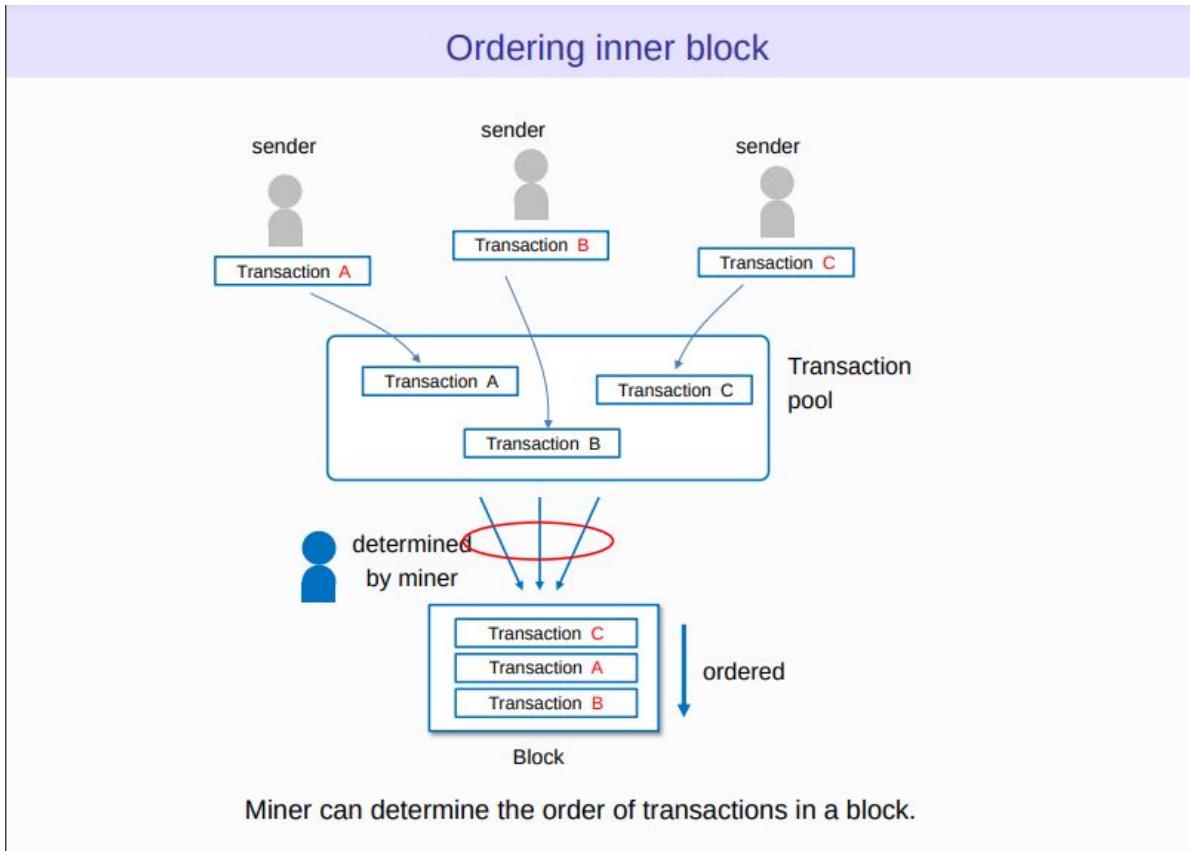
Transactions must be executed sequentially.



Components of Ethereum Network - Transactions

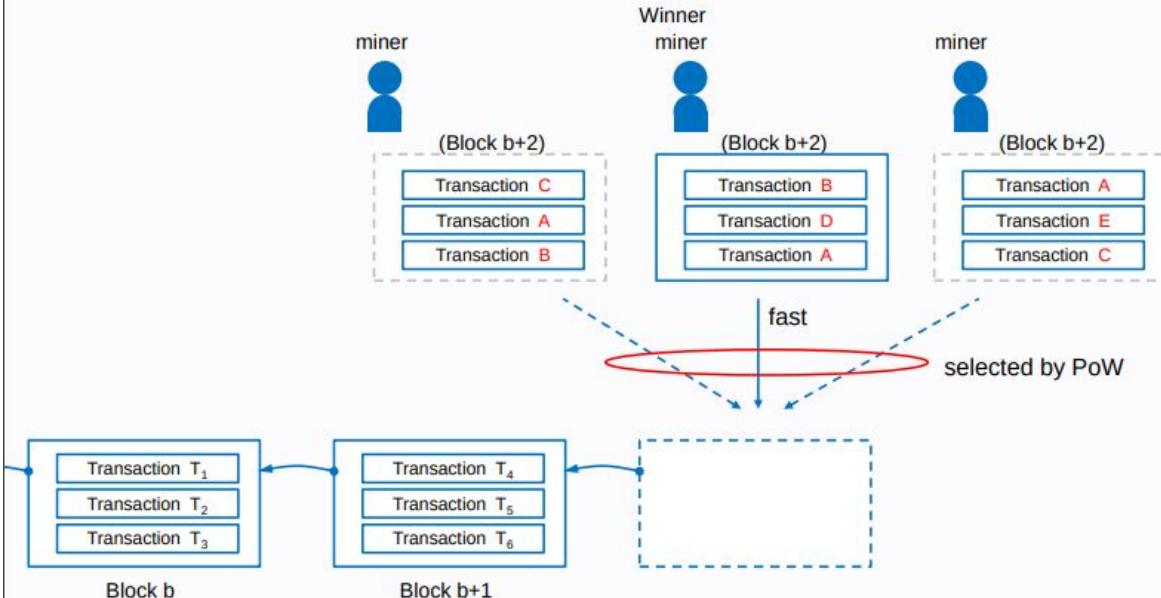


Components of Ethereum Network - Transactions



Components of Ethereum Network - Transactions

Ordering inter blocks

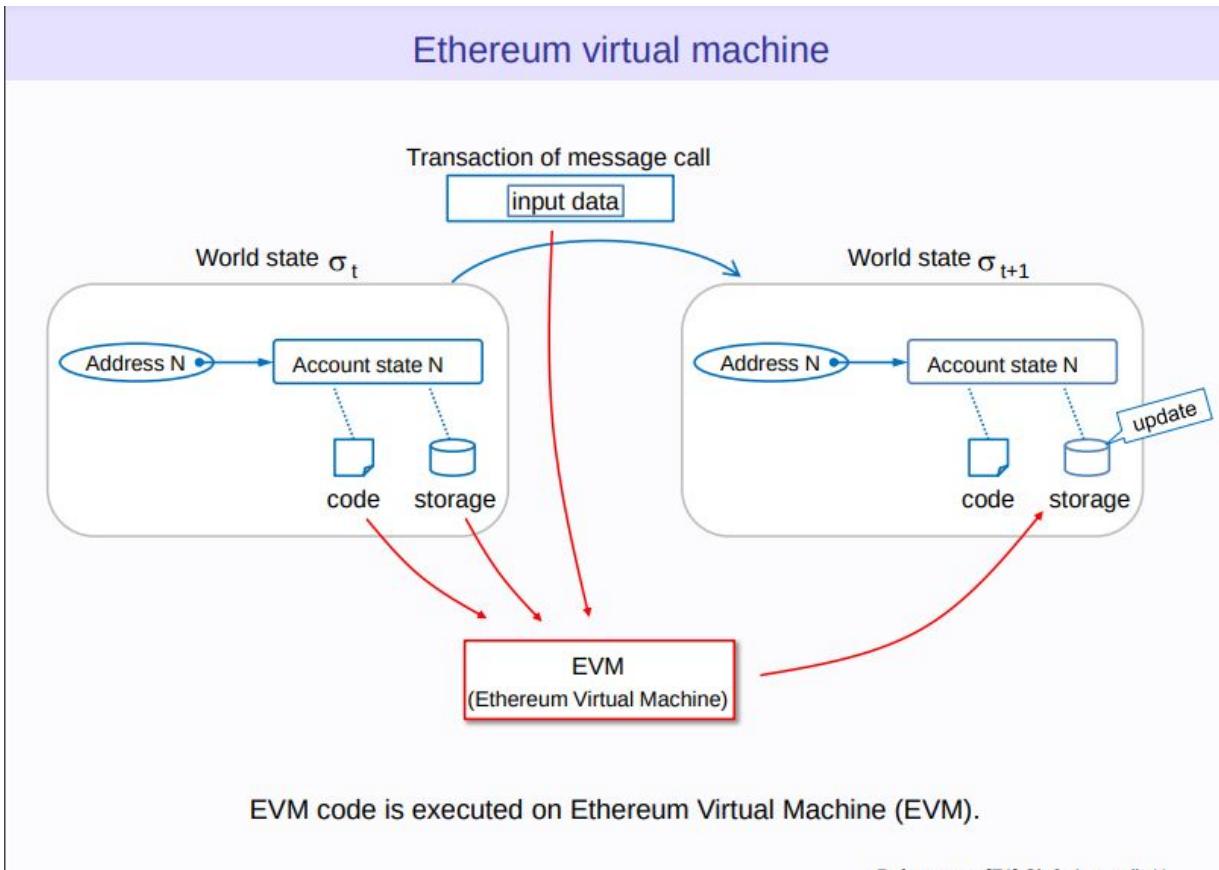


The order between blocks is determined by a consensus algorithm such as PoW.

Reference : EPFL Ch 3, Ch 4



Components of Ethereum Network - Transactions & EVM





Components of Ethereum Network - EVM

Ethereum virtual machine

Code

EVM code

Virtual machine

EVM (Ethereum Virtual Machine)

The Ethereum Virtual Machine is the runtime environment for smart contracts in Ethereum.

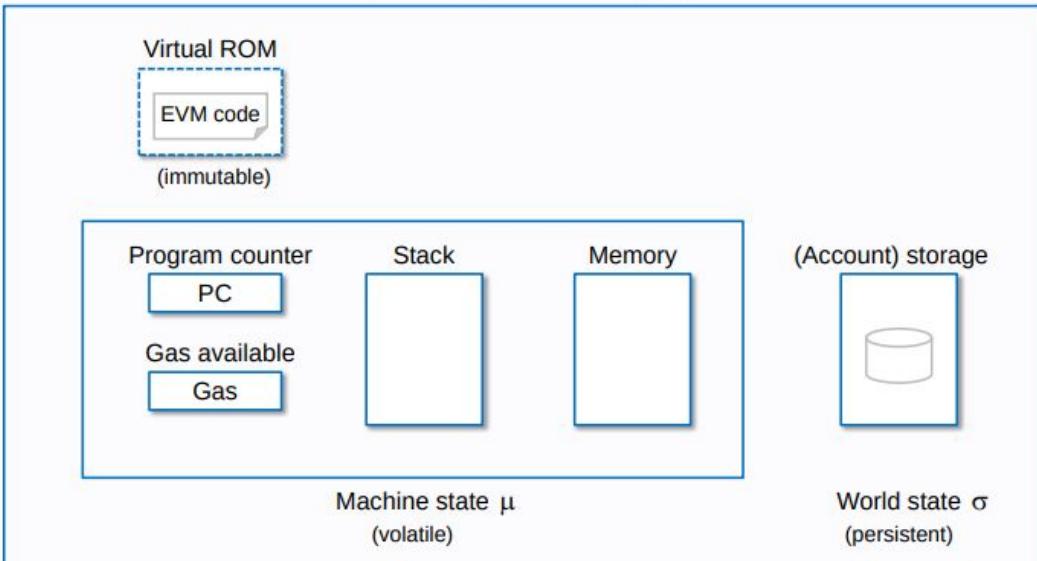
DOCUMENTATION & SUPPORT



Components of Ethereum Network - EVM Architecture

EVM architecture

Ethereum Virtual Machine (EVM)



The EVM is a simple stack-based architecture.

PRESENTED BY: [REDACTED]



Components of Ethereum Network - EVM Architecture

Machine space of EVM



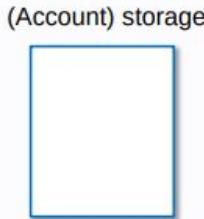
stack memory

256 bits x 1024 elements



volatile memory

byte addressing
linear memory



persistent memory

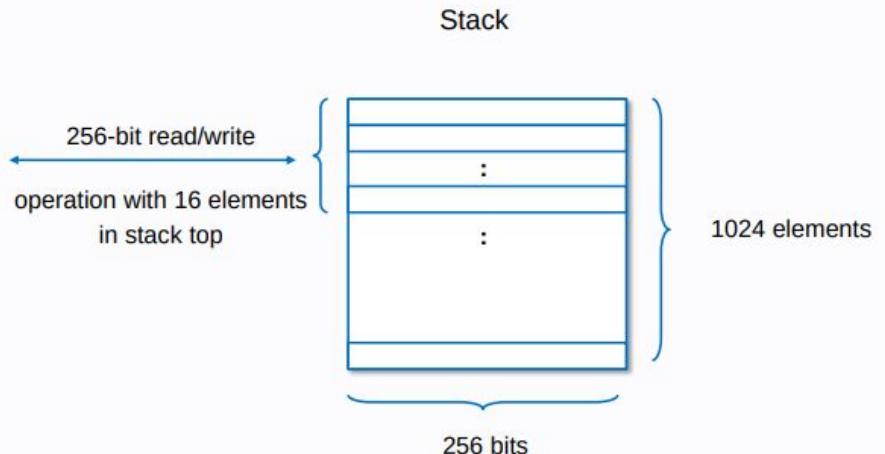
256 bits to 256 bits
key-value store

There are several resources as space.



Components of Ethereum Network - EVM Architecture

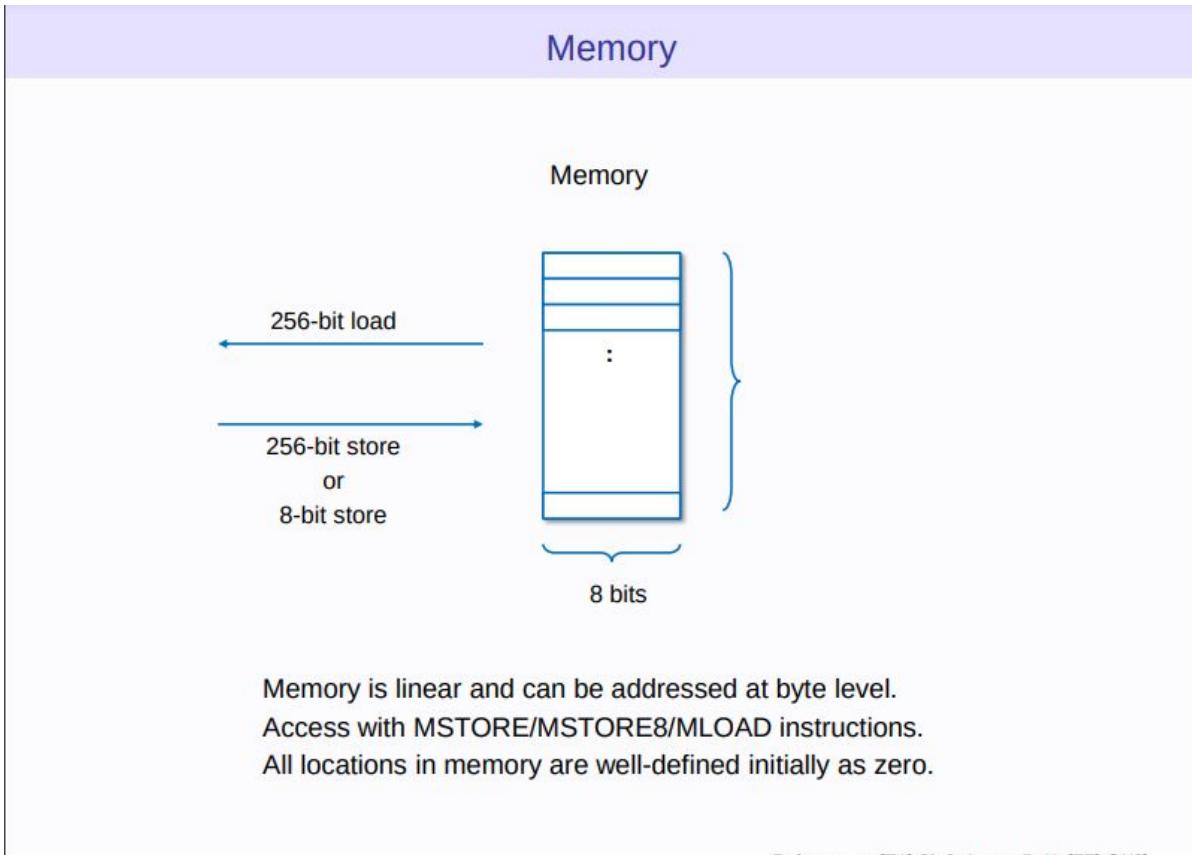
Stack



All operation are performed on the stack.

Access with many instructions such as PUSH/POP/COPY/SWAP, ...

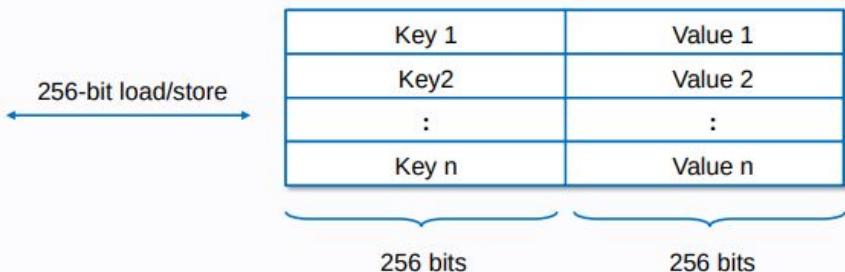




Components of Ethereum Network - EVM Architecture

Account storage

(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.

QUESTION AND ANSWER SESSION





Components of Ethereum Network - EVM Architecture

EVM code

Assembly view

```
PUSH1 e0
PUSH1 02
EXP
PUSH1 00
CALLDATALOAD
:
```

Bytecode view

```
0x60e060020a600035...
```

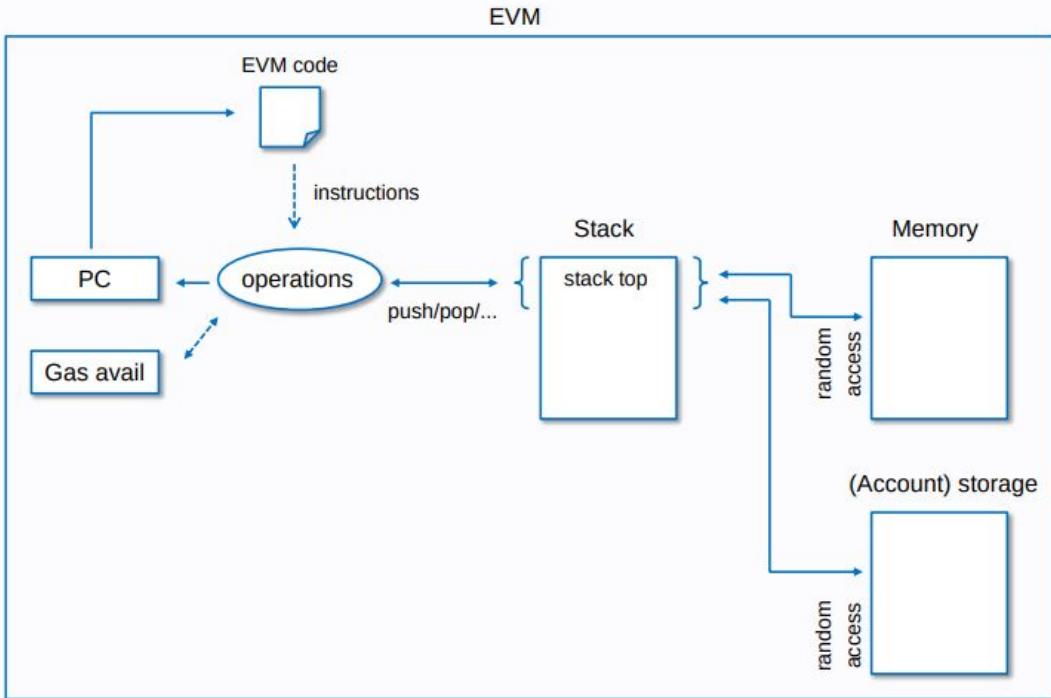
EVM Code is the bytecode that the EVM can natively execute.

QUESTION ANSWER



Components of Ethereum Network - EVM Architecture

Execution model

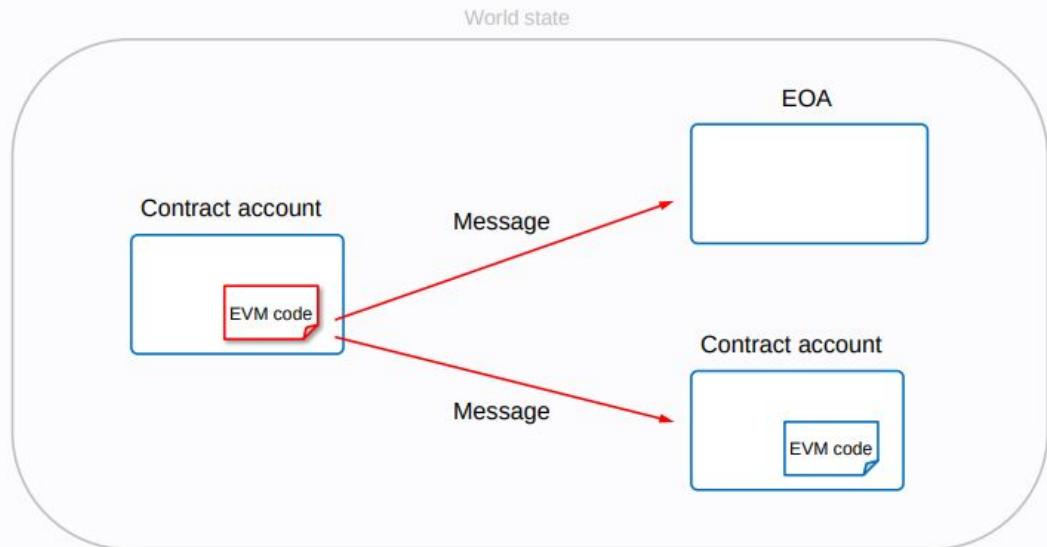


DATA STRUCTURE & ALGORITHMS



Components of Ethereum Network - Transactions

Message call



EVM can send a message to other account.

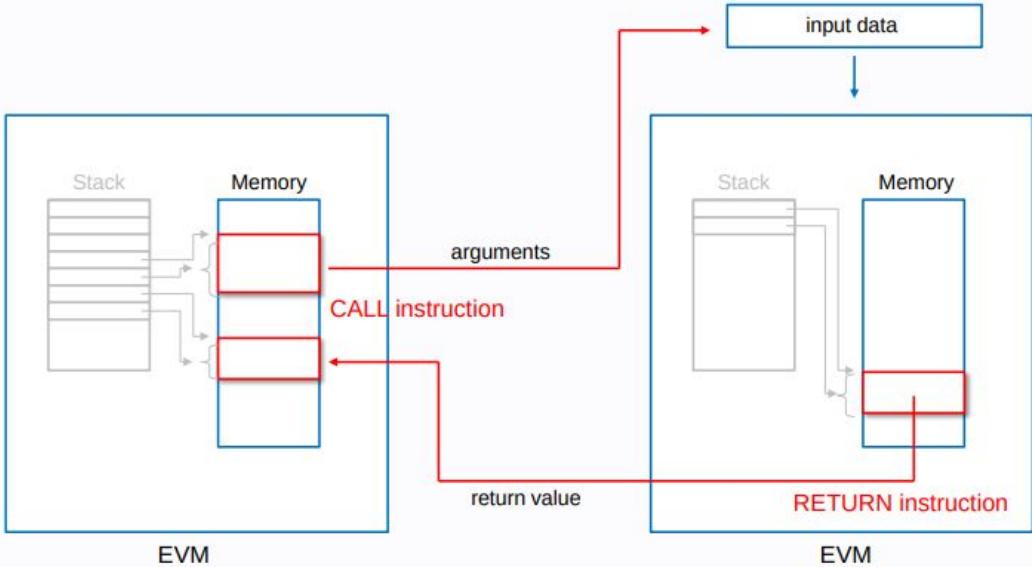
The depth of message call is limited to less than 1024 levels.

QUESTION



Components of Ethereum Network - Transactions

Instructions for Message call



Message call is triggered by CALL instruction.
Arguments and return values are passed using memory.

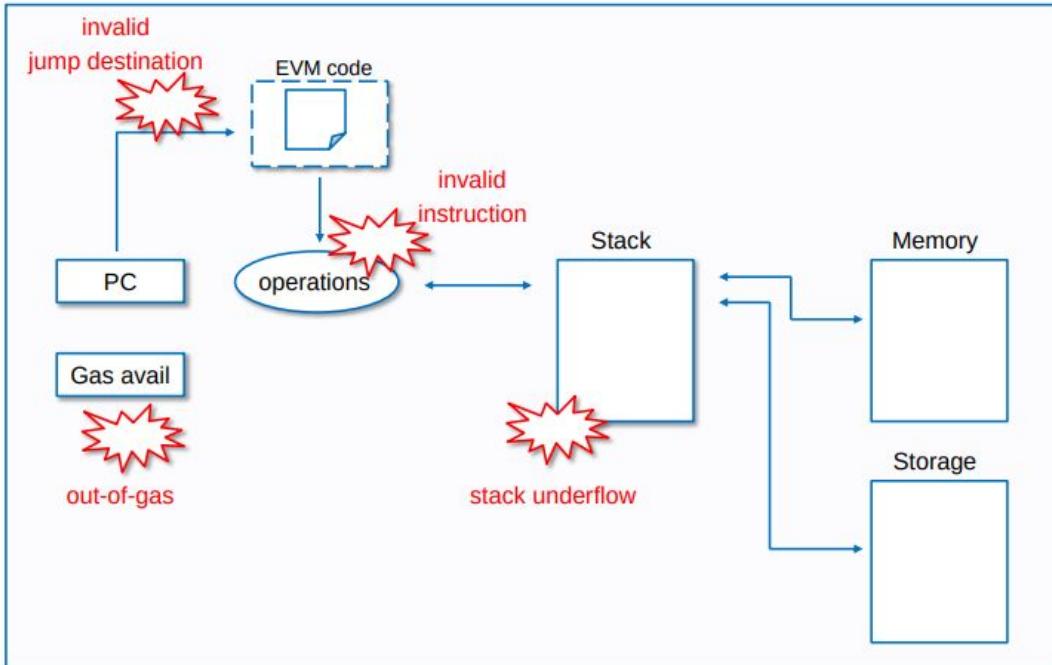
QUESTION PAPER



Components of Ethereum Network - Transactions

Exception

EVM

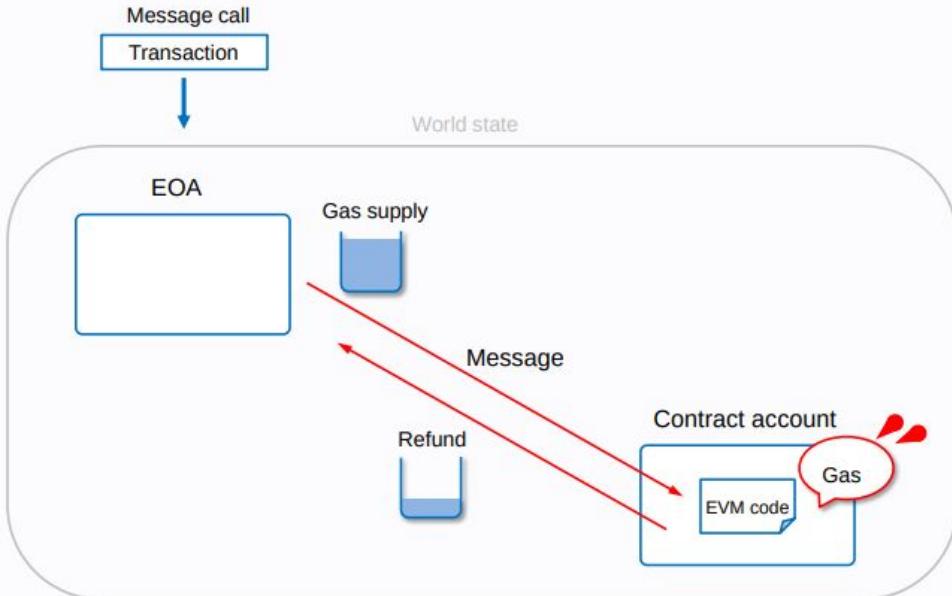


Source: https://ethresear.ch



Components of Ethereum Network - Transactions

Gas and fee

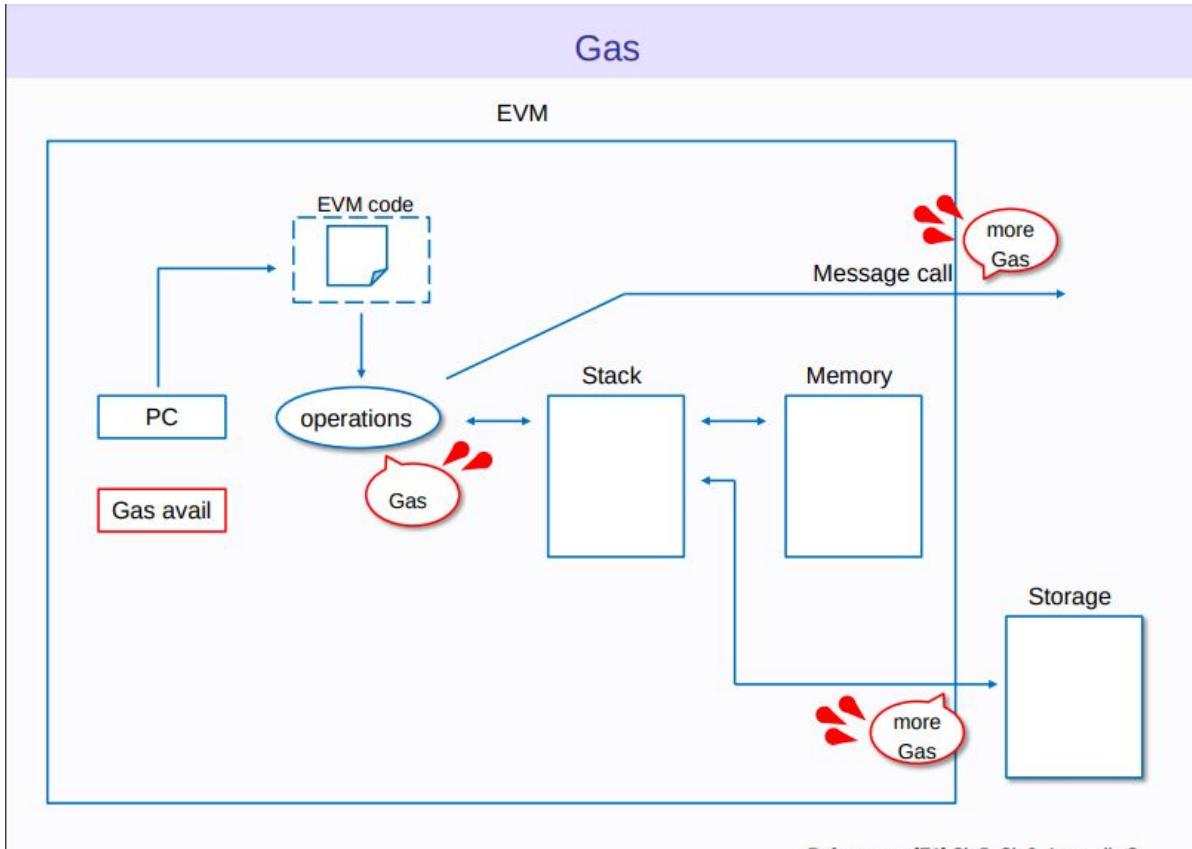


All programmable computation in Ethereum is subject to fees (denominated in gas).

PRINCIPLES OF COMPUTATION



Components of Ethereum Network - Transactions

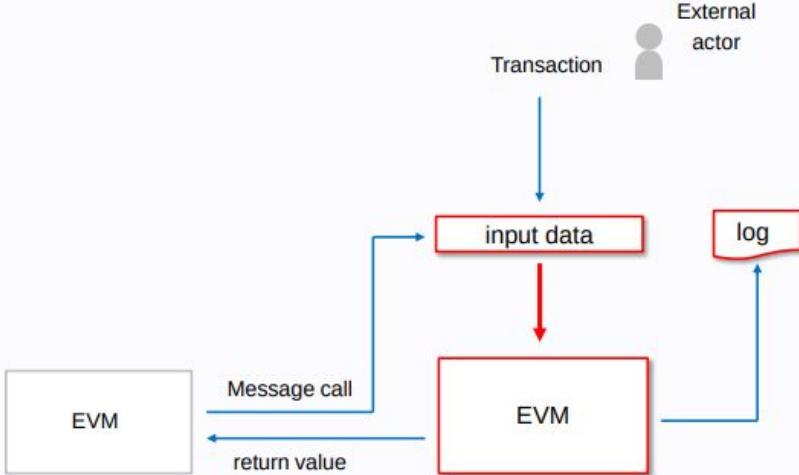


RENDERED IMAGE OF A SLIDE



Components of Ethereum Network - Transactions

Input and Output of EVM



EVM can input external data from a message call.

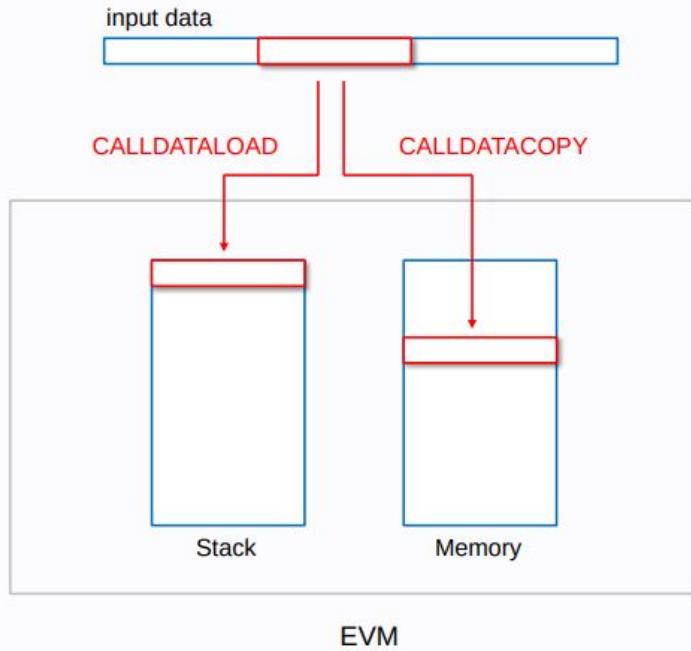
EVM can output log. EVM can also return values to Caller EVM.

QUESTION



Components of Ethereum Network - Transactions

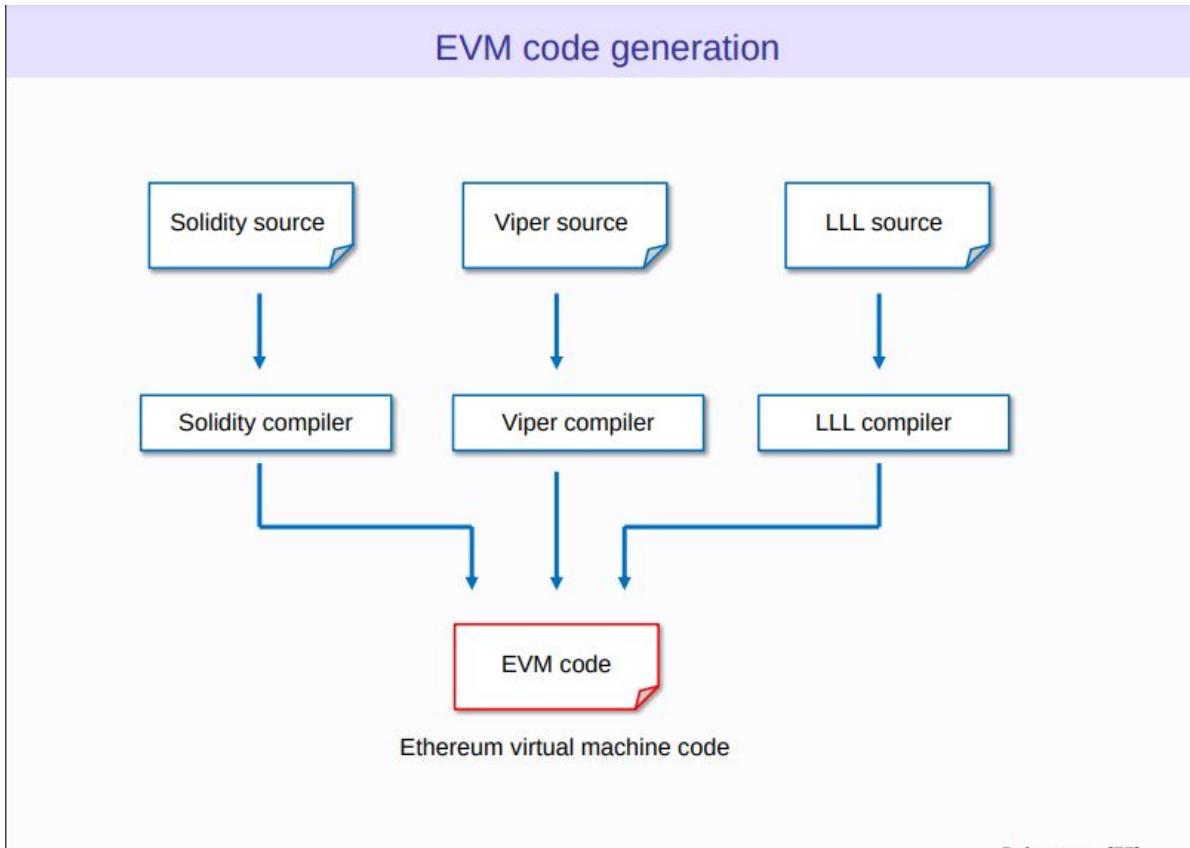
Instructions for input data



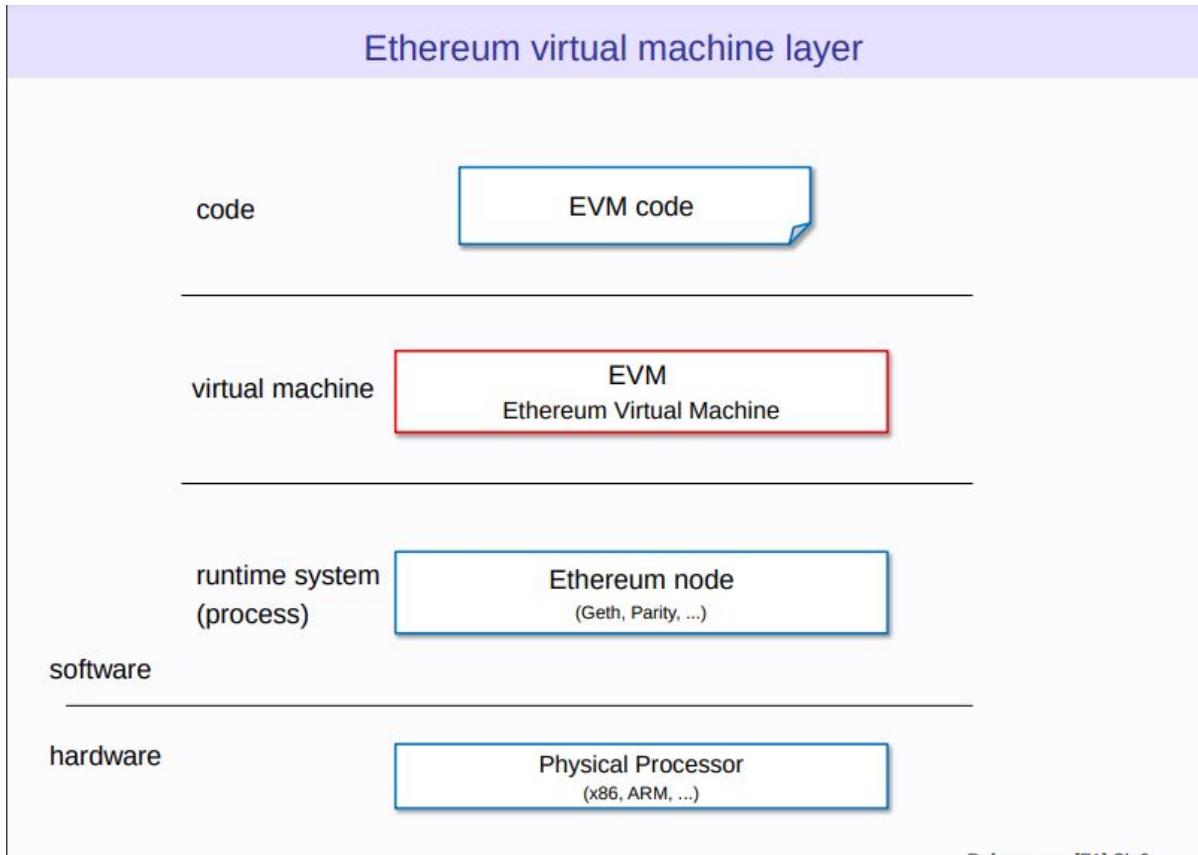
DATASTRUCTURE & ALGORITHMS



Components of Ethereum Network - EVM



Components of Ethereum Network - EVM



References : EP11 Ch 0



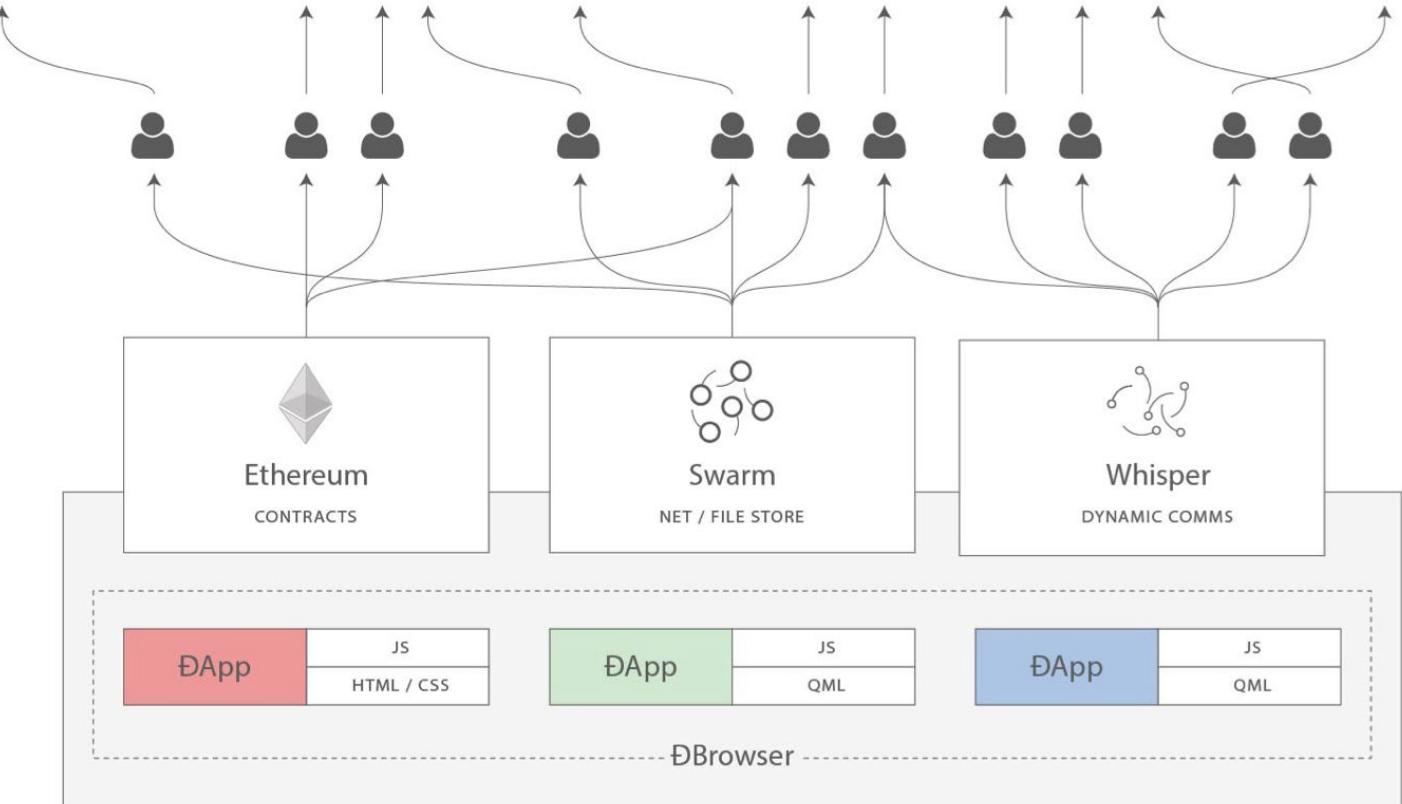
Components of Ethereum Network - Swarm & Whisper



- 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



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](#)



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](#)



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](#)



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?



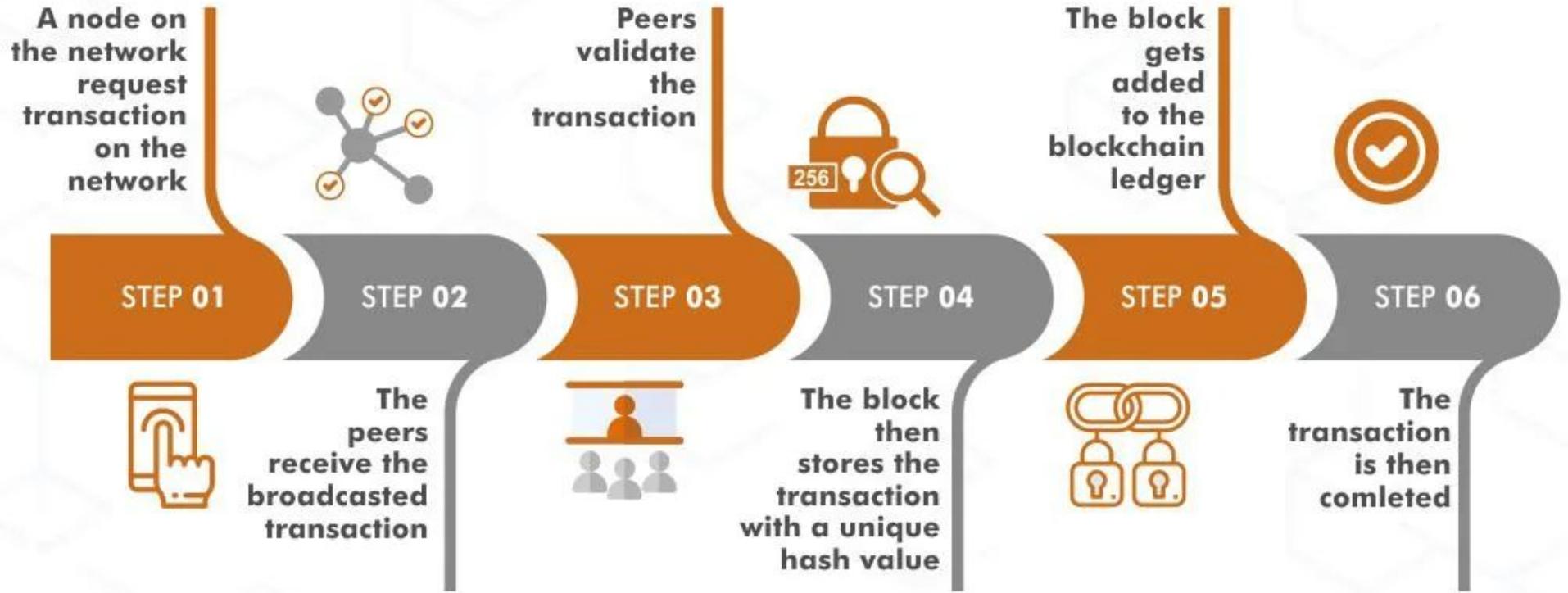
Ethereum implements an execution environment called **Ethereum Virtual Machine (EVM)**.

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 \leq 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

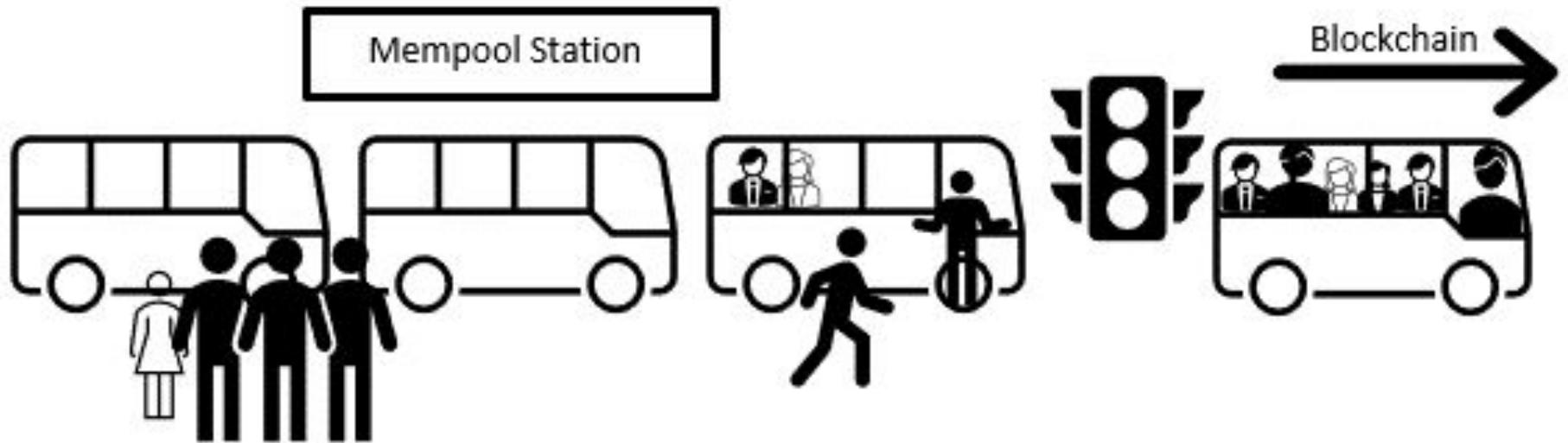


Transactions in Blockchain - Life Cycle

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.



Transactions in Blockchain - The Bus Station Analogy

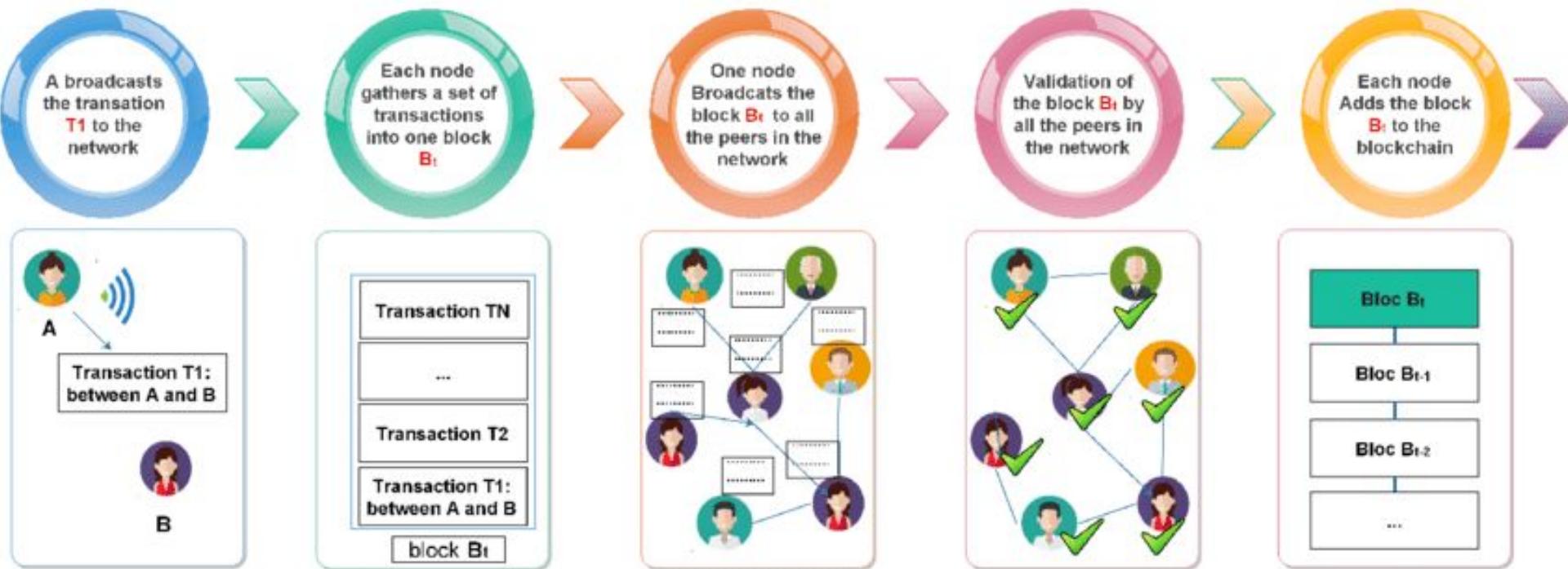




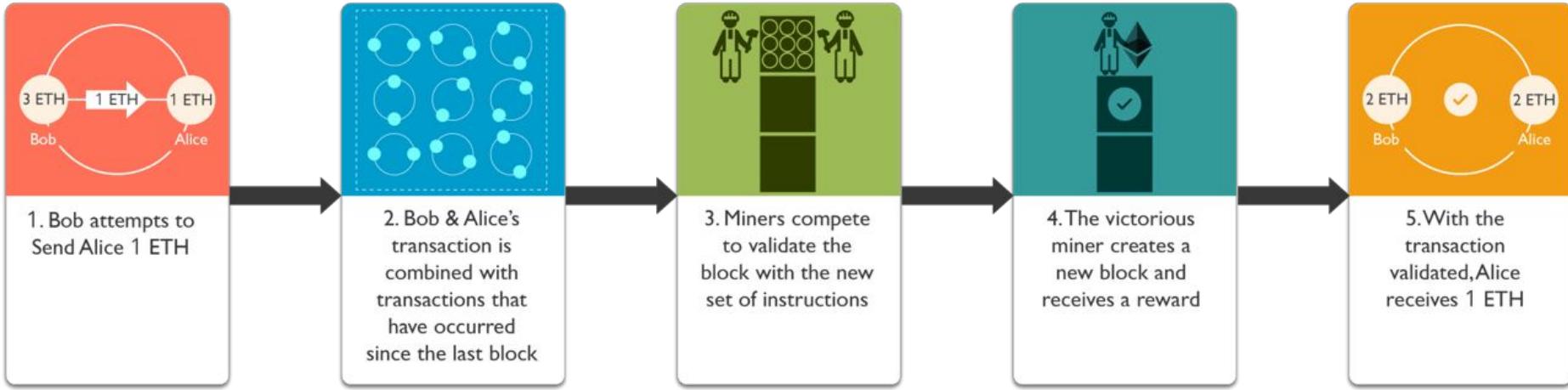
Transactions in Blockchain - Live Demo



Transactions in Blockchain - Example



Transactions in Blockchain - Example



The proof-of-work algorithm used is called **Ethash**, which is a hashing algorithm inspired by the **Dagger-Hashimoto Algorithm**.

Now that we've seen the working architecture of ethereum and discussed it's essential elements, let's see a real-world problem and the ethereum approach to solve the same.





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**



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*



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 $1e+18$ wei)
- `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`)



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.



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.



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.



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.





Types of test-networks used in Ethereum

Ethereum networks

- groups of connected computers that communicate using the Ethereum protocol.
- There is **only one Ethereum Mainnet**,
- But **independent networks**
 - conforms to the **same protocol rules** can be created
 - for testing and development purposes.
 - These independent "networks" that conform to the protocol **without interacting with each other**.

Note :

- One can even start one locally on your own computer for testing your smart contracts and web3 apps.
- Eg : Geth & Ganache Private Networks created during the Lab Experiments
- Your **Ethereum account will work across the different networks**,
- But your account balance and transaction history won't carry over from the main Ethereum network.

For testing purposes :

- it's useful to know which networks are available
- how to get testnet ETH to play around with.

For security considerations : it's **not recommended to reuse mainnet accounts on testnets or vice versa.**



Types of test-networks used in Ethereum

- **PUBLIC NETWORKS**

- Public networks are accessible to anyone in the world with an internet connection.
- Anyone can read or create transactions on a public blockchain and validate the transactions being executed.
- The consensus among peers decides on the inclusion of transactions and the state of the network.

- **Ethereum Mainnet**

- Mainnet is the primary public Ethereum production blockchain,
- where actual-value transactions occur on the distributed ledger.
- When people and exchanges discuss ETH prices, they're talking about Mainnet ETH.



Types of test-networks used in Ethereum

- **Ethereum Testnets**

- These are networks used by protocol developers or smart contract developers to test both protocol upgrades as well as potential smart contracts in a production-like environment before deployment to Mainnet.
- Test any contract code you write on a testnet before deploying to Mainnet.
- Among dapps that integrate with existing smart contracts, most projects have copies deployed to testnets.
- Most testnets started by using a permissioned proof-of-authority consensus mechanism.
- This means a small number of nodes are chosen to validate transactions and create new blocks – staking their identity in the process.
- Alternatively, some testnets feature an open proof-of-stake consensus mechanism where everyone can test running a validator, just like Ethereum Mainnet.
- **ETH on testnets is supposed to have no real value;**
- Most people get testnet ETH for free from faucets.
- Most faucets are webapps where you can input an address which you request ETH to be sent to.



Types of test-networks used in Ethereum

- Which Testnet should I use?
 - The two public testnets that client developers are **Sepolia** and **Goerli**.
 - **Sepolia**
 - recommended default testnet for application development.
 - Features :
 - Closed validator set, controlled by client & testing teams
 - New testnet, less applications deployed than other testnets
 - Fast to sync and running a node requires minimal disk space
 - **Goerli (long-term support)**
 - the Goerli testnet is deprecated and will be replaced by Holesovice in 2023.
 - Features :
 - Open validator set, stakers can test network upgrades
 - Large state, useful for testing complex smart contract interactions
 - Longer to sync and requires more storage to run a node





Types of test-networks used in Ethereum



- **PRIVATE NETWORKS**

- An Ethereum network is a private network if its nodes are not connected to a public network (i.e. Mainnet or a testnet).
- Private only means reserved or isolated, rather than protected or secure.

- **Development networks**

- To develop an Ethereum application, run it on a private network to see how it works before deploying.
- Create a local blockchain instance to test your dapp.
- This allows for much faster iteration than a public testnet.

- **Consortium networks**

- The consensus process is controlled by a pre-defined set of nodes that are trusted.
- For example :
 - a private network of known academic institutions that each govern a single node,
 - and blocks are validated by a threshold of signatories within the network.

Note :

If a **public Ethereum network** ⇒ **public internet**, a **consortium network** ⇒ **private intranet**.





Metamask

- A software (hot storage) crypto wallet that is mainly used to interact with the Ethereum blockchain.
- One can access MetaMask through a browser extension or mobile app, which is then used to interact with decentralized applications (Dapps).

How to Set Up a MetaMask Wallet

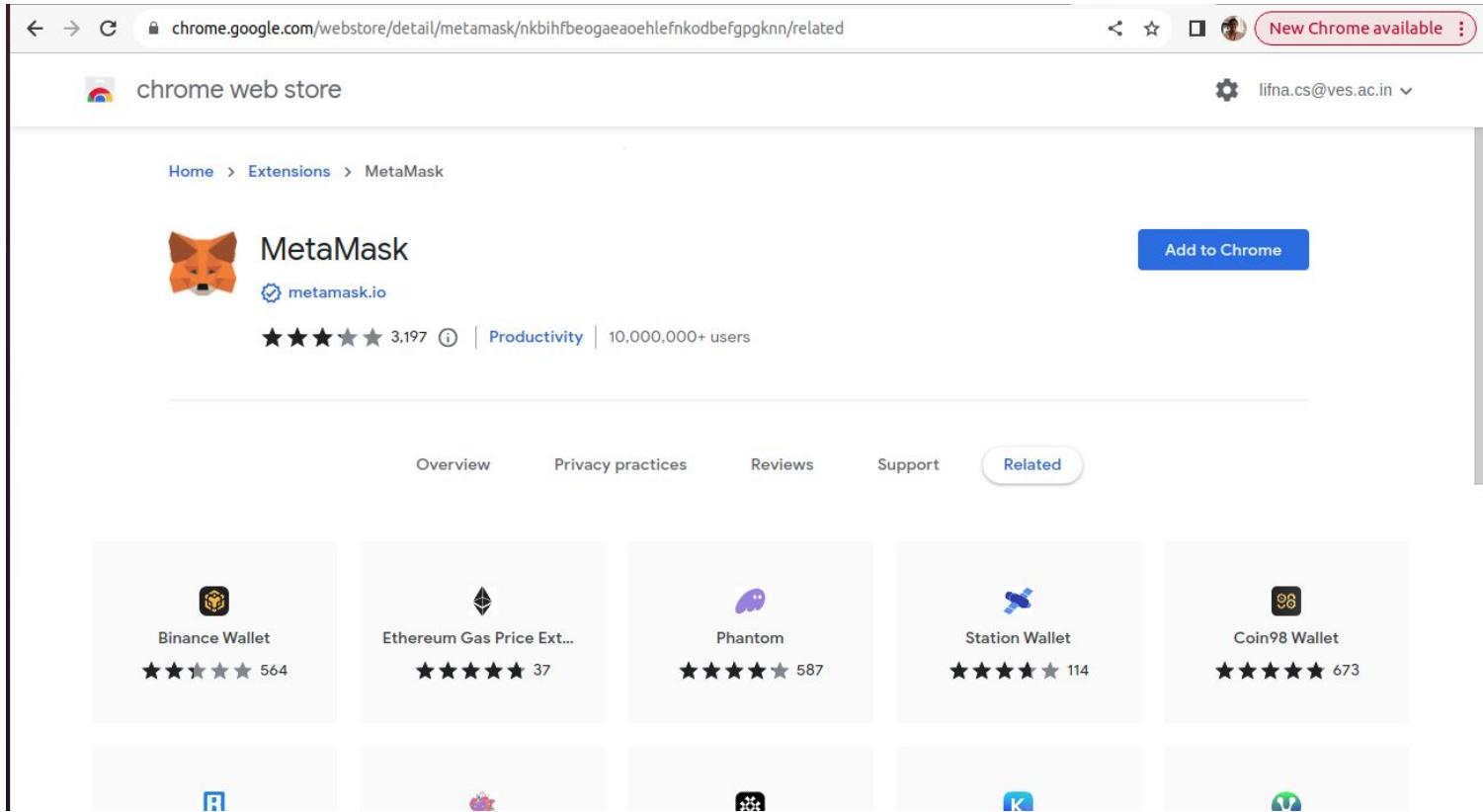
1. Add the chrome extension or download the app
2. Create a wallet
 - Once you choose Get Started, you will be asked to either Import Wallet or Create a Wallet.
3. Create a password
4. Secure your secret phrase
 - make sure you write it down on a piece of paper and store it in a secure location.
 - For more security, write it down on multiple pieces of paper and store it in multiple secure locations.
5. Confirm your secret recovery phrase
 - After writing down your secret phrase you will be asked to enter it again to ensure you wrote it down correctly.
 - Upon entering your phrase, you will be logged into your MetaMask wallet.





Setting up a Metamask Wallet

1. Add the Metamask extension



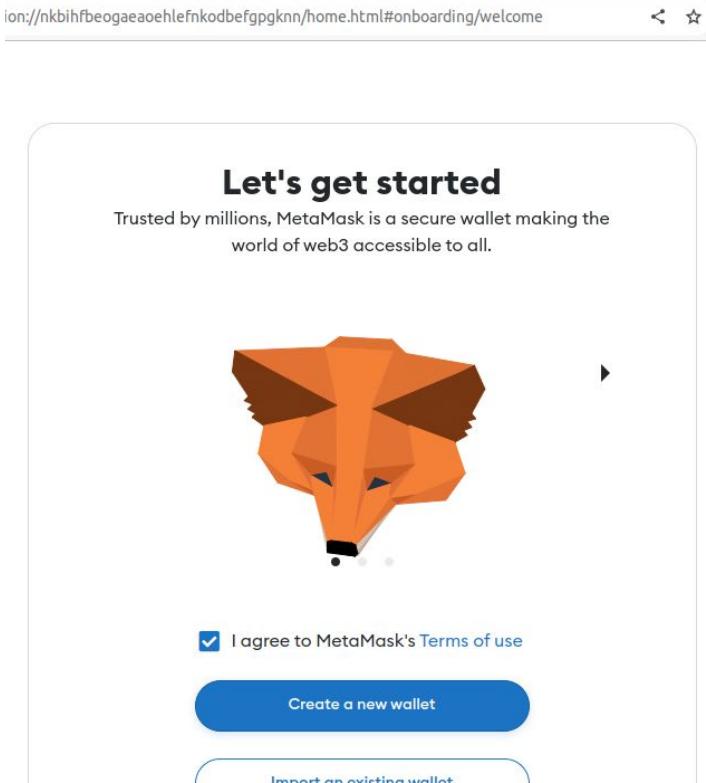
The screenshot shows the MetaMask extension page on the Chrome Web Store. The URL in the address bar is `chrome.google.com/webstore/detail/metamask/nkbihfbeogaeaoehlfnkodbefgpgknn/related`. The page title is "chrome web store". On the right, there is a gear icon and the email "lifna.cs@ves.ac.in". A red banner at the top right says "New Chrome available". The main content area shows the "MetaMask" extension by metamask.io. It has a 5-star rating of 3,197 reviews and is categorized as "Productivity". It has over 10 million users. A large blue "Add to Chrome" button is prominently displayed. Below the extension details, there are tabs for "Overview", "Privacy practices", "Reviews", "Support", and "Related". Under the "Related" tab, five other wallet extensions are listed: Binance Wallet, Ethereum Gas Price Ext..., Phantom, Station Wallet, and Coin98 Wallet, each with its own rating and review count.

Extension	Rating	Reviews
Binance Wallet	★★★★★	564
Ethereum Gas Price Ext...	★★★★★	37
Phantom	★★★★★	587
Station Wallet	★★★★★	114
Coin98 Wallet	★★★★★	673

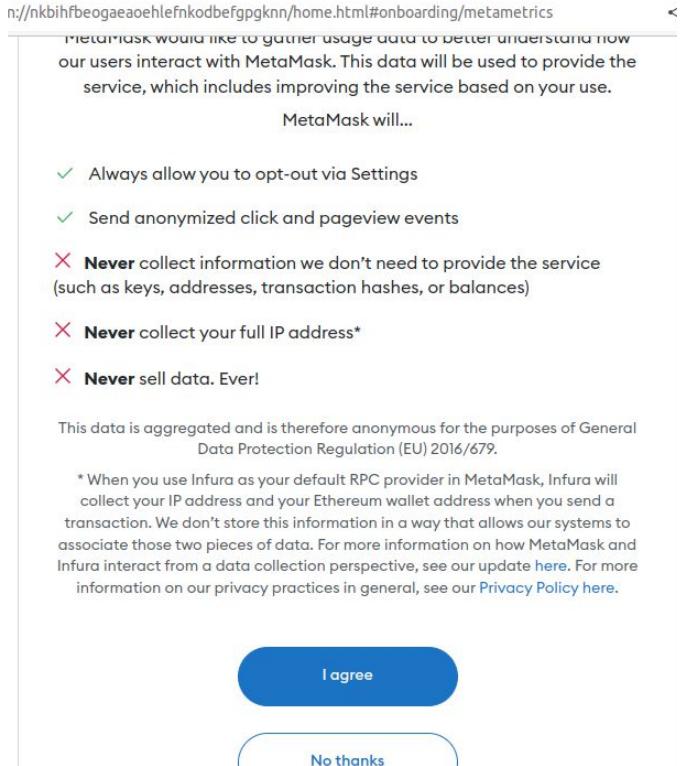


Setting up a Metamask Wallet

2. Create a wallet



The screenshot shows the initial landing page of the MetaMask wallet setup. It features a large orange fox head icon with the text "Let's get started" above it. Below the icon, a subtext reads: "Trusted by millions, MetaMask is a secure wallet making the world of web3 accessible to all." At the bottom left is a checkbox labeled "I agree to MetaMask's Terms of use". To its right is a blue button labeled "Create a new wallet". At the very bottom, there is a faint link "Import an existing wallet".



The screenshot shows the "metametrics" consent screen. It displays a message from MetaMask asking for permission to collect usage data to improve the service. The message includes a list of what data is collected and what is not. At the bottom, there are two buttons: "I agree" (blue) and "No thanks" (grey).

n://nkbihfbeogaeaohlefnkodbefgpgknn/home.html#onboarding/welcome

n://nkbihfbeogaeaohlefnkodbefgpgknn/home.html#onboarding/metametrics

MetaMask would like to gather usage data to better understand how our users interact with MetaMask. This data will be used to provide the service, which includes improving the service based on your use.

MetaMask will...

- ✓ Always allow you to opt-out via Settings
- ✓ Send anonymized click and pageview events
- ✗ Never collect information we don't need to provide the service (such as keys, addresses, transaction hashes, or balances)
- ✗ Never collect your full IP address*
- ✗ Never sell data. Ever!

This data is aggregated and is therefore anonymous for the purposes of General Data Protection Regulation (EU) 2016/679.

* When you use Infura as your default RPC provider in MetaMask, Infura will collect your IP address and your Ethereum wallet address when you send a transaction. We don't store this information in a way that allows our systems to associate those two pieces of data. For more information on how MetaMask and Infura interact from a data collection perspective, see our update [here](#). For more information on our privacy practices in general, see our [Privacy Policy here](#).





Setting up a Metamask Wallet

3. Create a password

//nkbihfbeogaeaoehlefknkodbefgpgknn/home.html#onboarding/create-password

The screenshot shows the first step of the Metamask wallet creation process. At the top, there is a navigation bar with three steps: 1. Create password, 2. Secure wallet, and 3. Confirm secret recovery phrase. Below the navigation bar, the title "Create password" is displayed in bold. A sub-instruction states: "This password will unlock your MetaMask wallet only on this device. MetaMask can not recover this password." There are two input fields: "New password (8 characters min)" containing "*****" and "Confirm password" also containing "*****". Both fields have a "Show" link next to them. Below the fields, a note says: "A strong password can improve the security of your wallet should your device be stolen or compromised." A checked checkbox at the bottom left indicates agreement: "I understand that MetaMask cannot recover this password for me. [Learn more](#)". A large blue button at the bottom right says "Create a new wallet".

Create password

This password will unlock your MetaMask wallet only on this device. MetaMask can not recover this password.

New password (8 characters min) [Show](#)

.....

Password strength: Average

A strong password can improve the security of your wallet should your device be stolen or compromised.

Confirm password ✓

.....

I understand that MetaMask cannot recover this password for me. [Learn more](#)

Create a new wallet





Setting up a Metamask Wallet

4. Secure your secret phrase

bihhfbeogaaoehlefnkodbefgpgknn/home.html#onboarding/secure-your-wallet

1 Create password 2 Secure wallet 3 Confirm secret recovery phrase

Secure your wallet

Before getting started, watch this short video to learn about your Secret Recovery Phrase and how to keep your wallet safe.

0:00 / 1:35

[Remind me later \(not recommended\)](#) [Secure my wallet \(recommended\)](#)

What is a Secret Recovery Phrase?

Metamask Recovery Phrases are 12 words long and can be used to recover your wallet if you lose access to it.

bihhfbeogaaoehlefnkodbefgpgknn/home.html#onboarding/review-recovery-ph...

1 Create password 2 Secure wallet 3 Confirm secret recovery phrase

Write down your Secret Recovery Phrase

Write down this 12-word Secret Recovery Phrase and save it in a place that you trust and only you can access.

Tips:

- Save in a password manager
- Store in a safe deposit box
- Write down and store in multiple secret places

Make sure nobody is looking

[Reveal Secret Recovery Phrase](#)

5. Confirm your secret recovery phrase

ieogaeaoehlefnkodbefgpgknn/home.html#onboarding/confirm-recovery-ph...

English

1 Create password 2 Secure wallet 3 Confirm secret recovery phrase

Confirm Secret Recovery Phrase

Confirm Secret Recovery Phrase

1. pool	2. pretty	3.
4.	5. immense	6. arrive
7. husband	8.	9. speed
10. club	11. shove	12. annual

[Confirm](#)

Courtesy : [Metamask](#)





Setting up a Metamask Wallet

5. Confirm your secret recovery phrase

leogaeaaoehlefknkodbefgpgknn/home.html#onboarding/confirm-recovery-ph...

English

- 1 Create password
- 2 Secure wallet
- 3 Confirm secret recovery phrase

Confirm Secret Recovery Phrase

Confirm Secret Recovery Phrase

1. pool	2. pretty	3.
4.	5. immense	6. arrive
7. husband	8.	9. speed
10. club	11. shove	12. annual

Confirm

rome-extension://nkbihfbeogaeaoehlefknkodbefgpgknn/home.html#onboarding/completion



English

Wallet creation successful

You've successfully protected your wallet. Keep your Secret Recovery Phrase safe and secret -- it's your responsibility!

Remember:

- MetaMask can't recover your Secret Recovery Phrase.
- MetaMask will never ask you for your Secret Recovery Phrase.
- **Never share your Secret Recovery Phrase** with anyone or risk your funds being stolen
- [Learn more](#)

[Advanced configuration](#)

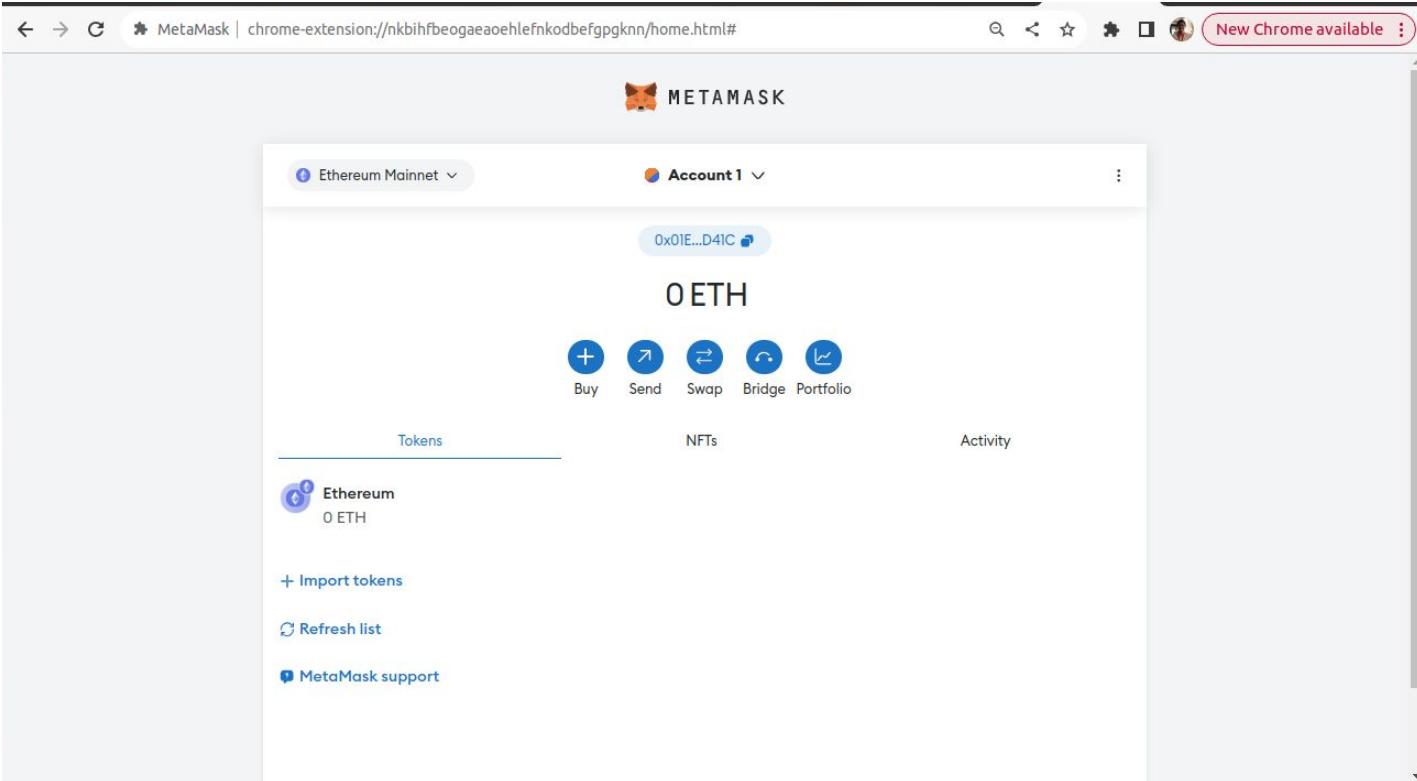
Got it!





Transferring Ethers using Metamask

Default Account is created on the Ethereum Mainnet with 0 ETH



The screenshot shows the MetaMask extension running in a browser window. The title bar indicates it's a Chrome extension. The main interface displays the Ethereum Mainnet account, labeled "Account 1" with address "0x01E...D41C". Below the address, the balance is shown as "0 ETH". A row of icons provides quick access to "Buy", "Send", "Swap", "Bridge", and "Portfolio". The "Tokens" tab is selected, showing a single entry for "Ethereum" with "0 ETH". There are also links for "Import tokens", "Refresh list", and "MetaMask support". The browser's toolbar at the top includes back, forward, search, and other standard navigation buttons.





Transferring Ethers using Metamask

Go to sepoliafaucet.com and create an Alchemy Account

The screenshot shows a web browser window with the URL sepoliafaucet.com. The page has a purple gradient background. At the top, it says "Powered by alchemy". Below that are tabs for "Sepolia", "Goerli", and "Mumbai". On the right, there's a button for "Alchemy Login". The main heading is "SEPOLIA FAUCET" with the subtext "Fast and reliable. 0.5 Sepolia ETH/day.". A large white form box contains a text input field for "Enter Your Wallet Address (0x...) or ETH Mainnet ENS Domain", a "Send Me ETH" button, and a note: "Please [signup](#) or [login](#) with Alchemy to request ETH. It's free!". It also includes a reCAPTCHA checkbox labeled "I'm not a robot" and links for "Privacy" and "Terms". Below the form is a table titled "Your Transactions" with columns "Time" and "-".





Transferring Ethers using Metamask

Paste the Public Key of the Metamask Account to claim 0.5 ETH/day

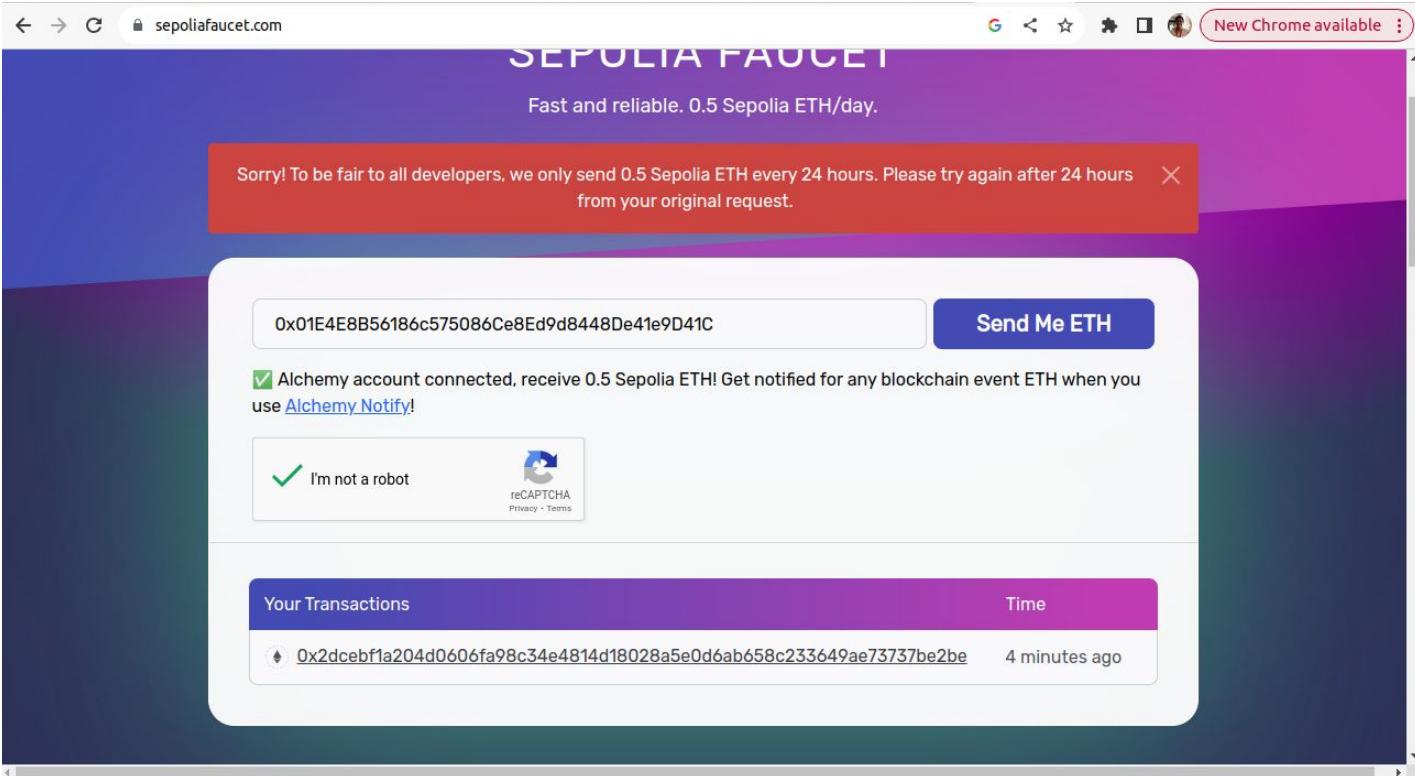
The screenshot shows a web browser window for the Sepolia Faucet at sepoliafaucet.com. The page title is "SEPOLIA FAUCET" and the subtitle is "Fast and reliable. 0.5 Sepolia ETH/day.". A text input field contains the public key: "0x01E4E8B56186c575086Ce8Ed9d8448De41e9D41C". To the right is a blue button labeled "Send Me ETH". Below the input field, there is a checked checkbox with the text "Alchemy account connected, receive 0.5 Sepolia ETH! Get notified for any blockchain event ETH when you use [Alchemy Notify!](#)". There is also an "I'm not a robot" reCAPTCHA field. At the bottom, there is a table titled "Your Transactions" with one row showing a dash "-". The table has columns "Your Transactions" and "Time".





Transferring Ethers using Metamask

Transaction details of the fund transfer is displayed



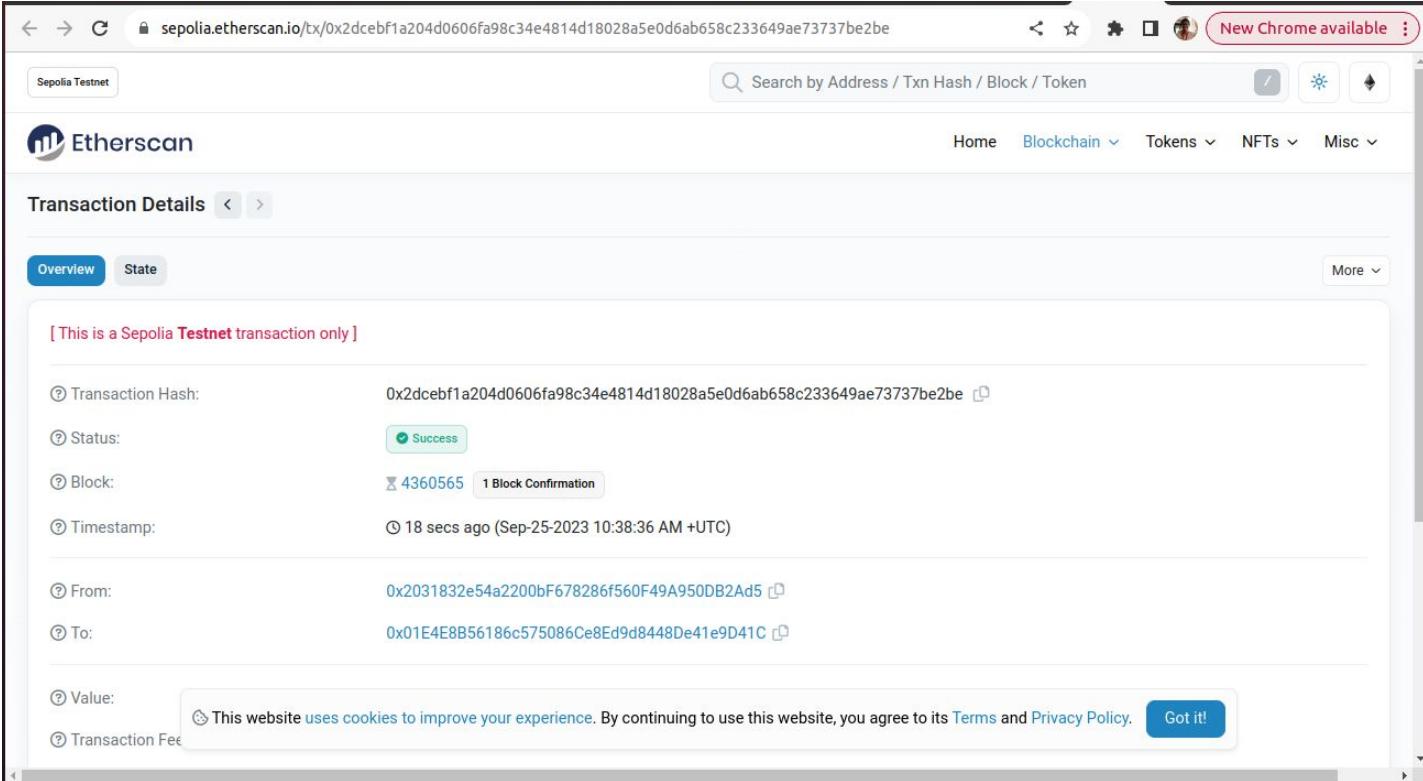
The screenshot shows a web browser window for the Sepolia Faucet at sepoliafaucet.com. The page has a purple header with the text "SEPOLIA FAUCET" and a subtext "Fast and reliable. 0.5 Sepolia ETH/day.". A red banner message states: "Sorry! To be fair to all developers, we only send 0.5 Sepolia ETH every 24 hours. Please try again after 24 hours from your original request." Below the banner is a form field containing the Ethereum address "0x01E4E8B56186c575086Ce8Ed9d8448De41e9D41C" and a blue button labeled "Send Me ETH". There is also a checked checkbox for Alchemy account connection and a reCAPTCHA field. At the bottom, a table titled "Your Transactions" shows one entry: "0xdceb1a204d0606fa98c34e4814d18028a5e0d6ab658c233649ae73737be2be" with a timestamp of "4 minutes ago".





Transferring Ethers using Metamask

On Etherscan.io, the details of the transaction is displayed (@Sepolia Testnet)

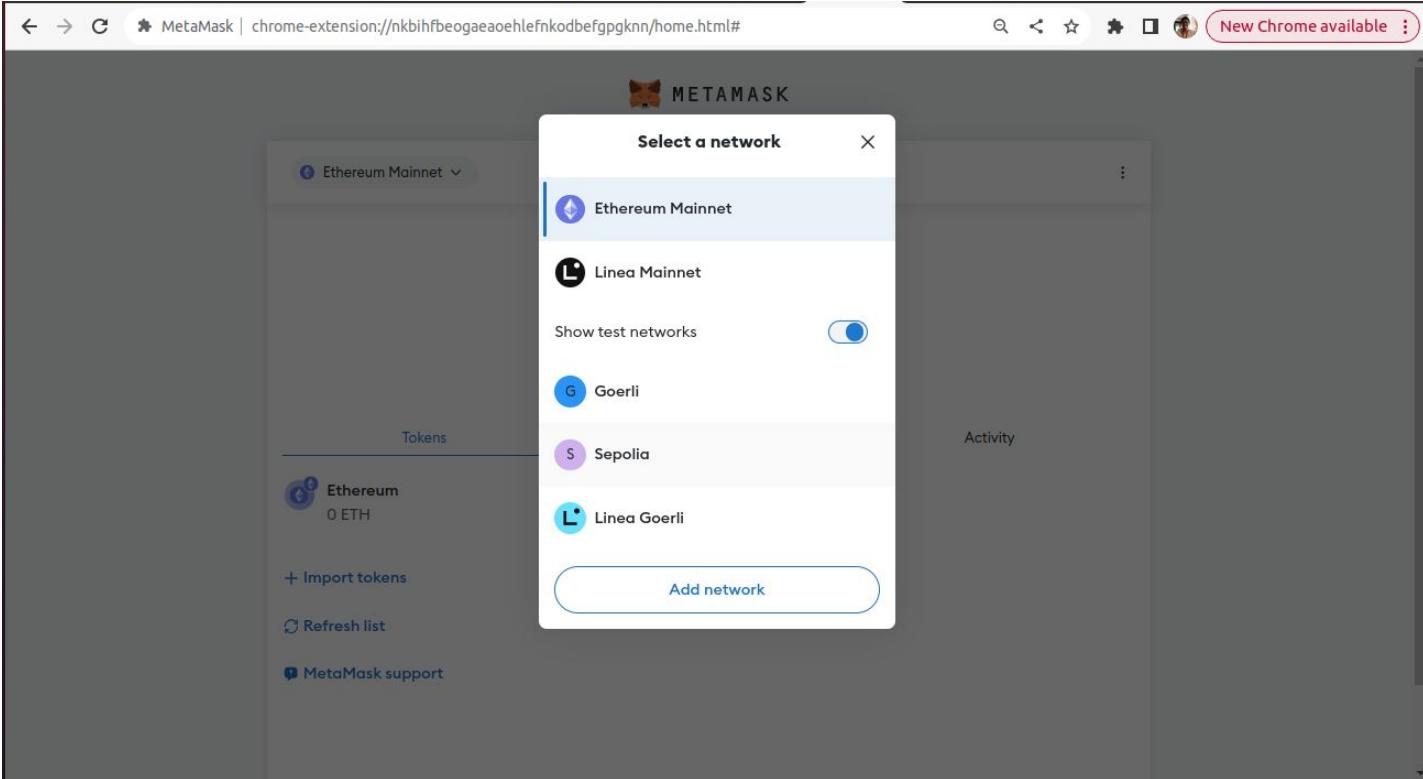


The screenshot shows a transaction details page on Etherscan.io for the Sepolia Testnet. The transaction hash is 0x2dcebf1a204d0606fa98c34e4814d18028a5e0d6ab658c233649ae73737be2be. The status is Success, and it has 1 Block Confirmation. The transaction occurred 18 secs ago (Sep-25-2023 10:38:36 AM +UTC). The From address is 0x2031832e54a2200bF678286f560F49A950DB2Ad5 and the To address is 0x01E4E8B56186c575086Ce8Ed9d8448De41e9D41C. A cookie consent banner at the bottom states: "This website uses cookies to improve your experience. By continuing to use this website, you agree to its Terms and Privacy Policy." with a "Got it!" button.



Transferring Ethers using Metamask

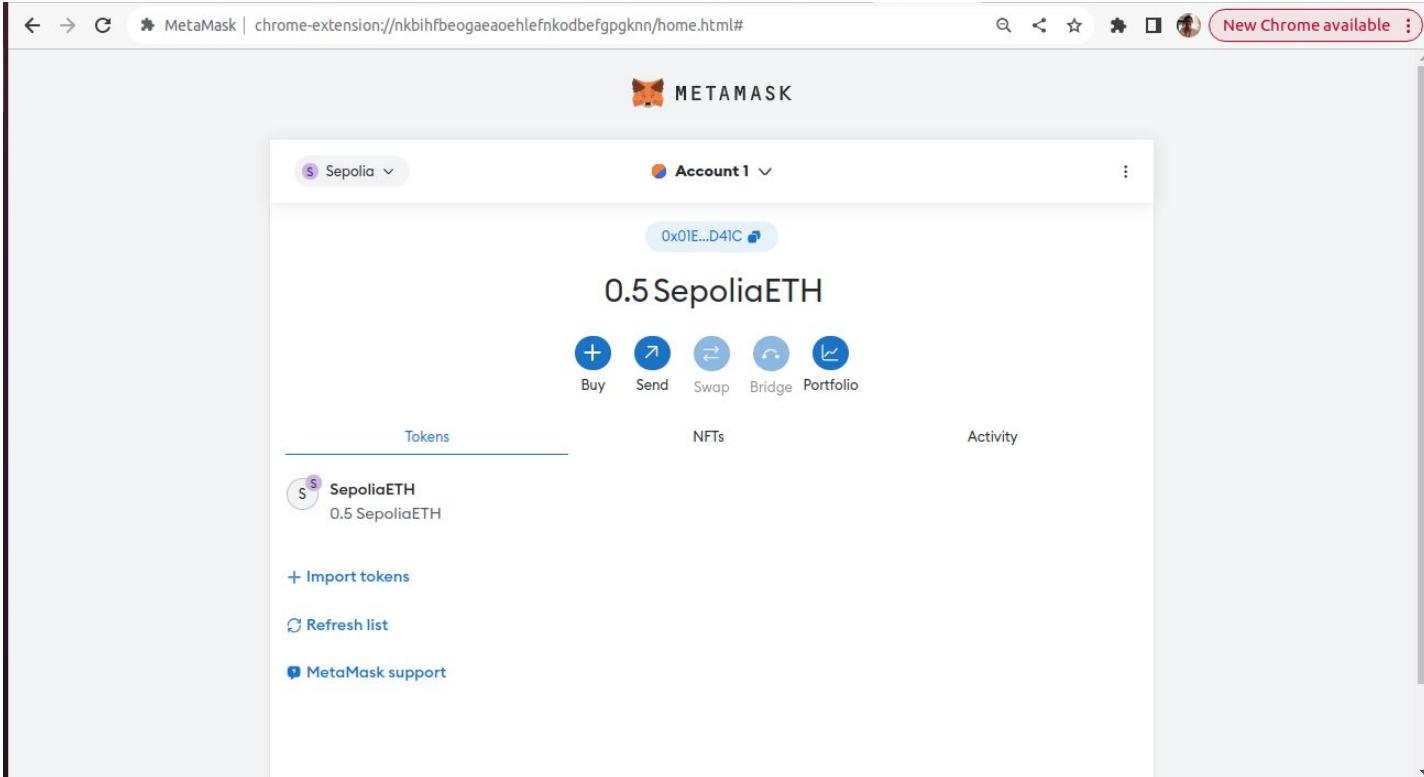
On the Metamask, switch to Sepolia Testnet





Transferring Ethers using Metamask

On the Metamask, under the Sepolia Testnet, the Account got a credit of 0.5ETH is updated

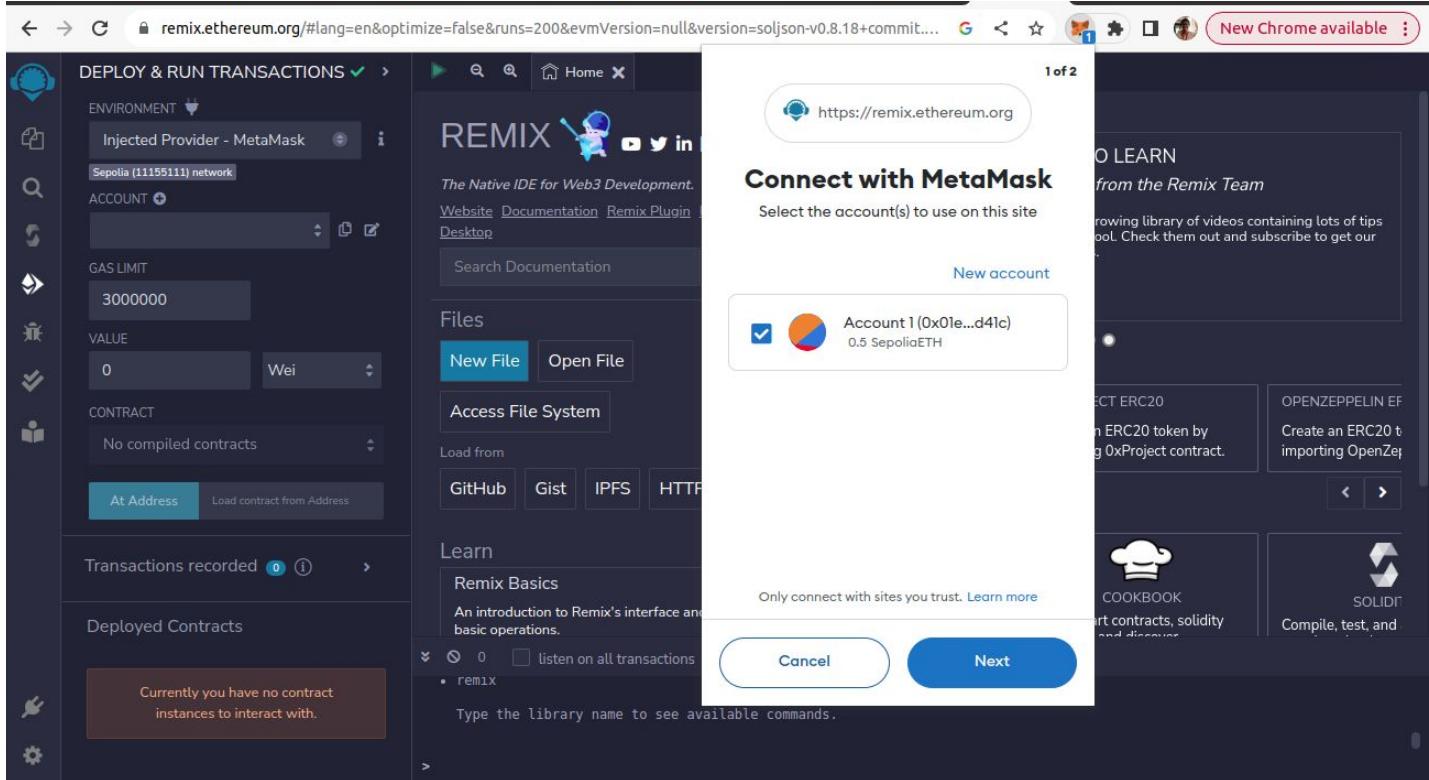


The screenshot shows the MetaMask extension running in a browser window. The title bar indicates it's on the Sepolia Testnet. The main interface displays the account balance as "0.5 SepoliaETH" with the address "0x01E...D41C". Below the balance are five buttons: Buy, Send, Swap, Bridge, and Portfolio. The "Tokens" tab is selected, showing a single entry for "SepoliaETH" with a balance of "0.5 SepoliaETH". There are also links for "Import tokens", "Refresh list", and "MetaMask support".



Transferring Ethers using Metamask

On the Remix IDE, select the Environment as Injected Provider - Metamask

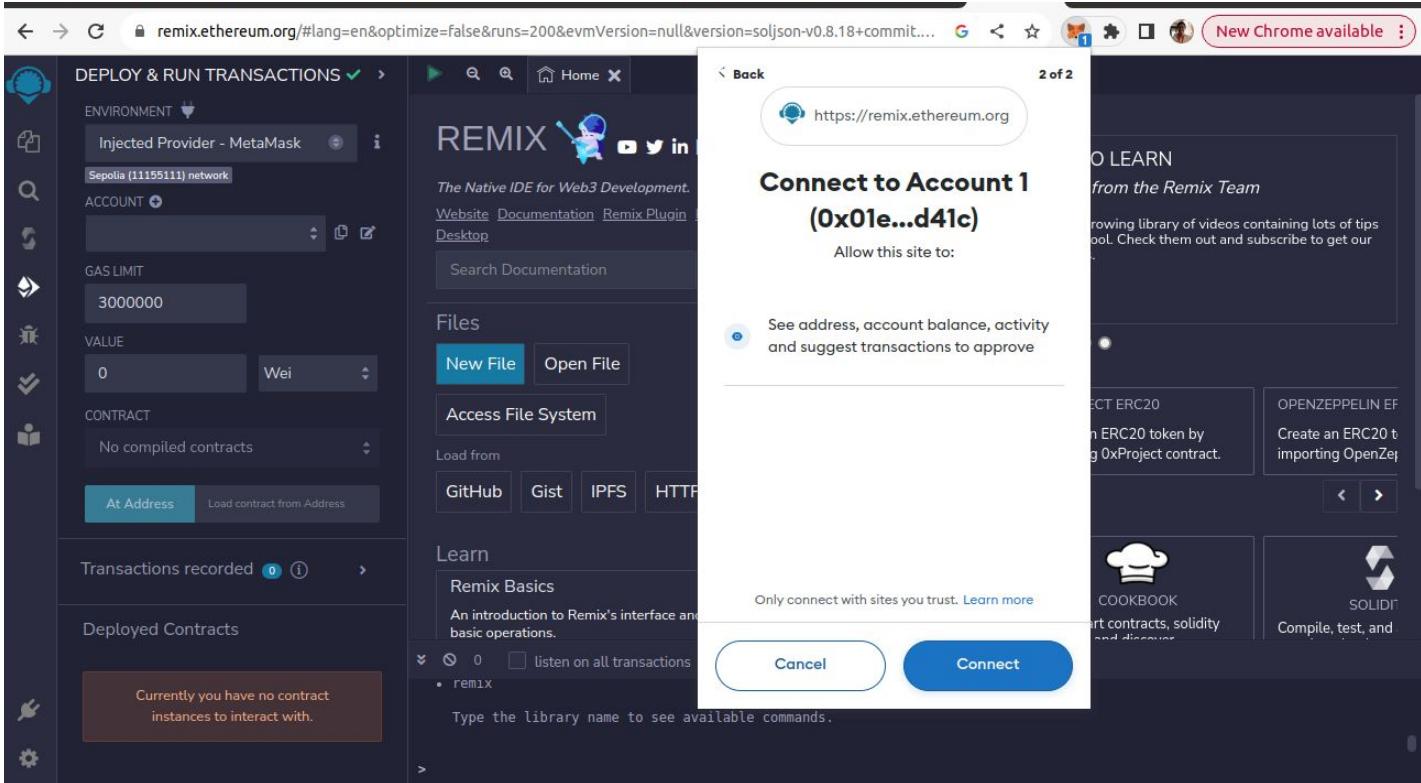


The screenshot shows the Remix IDE interface. On the left, there's a sidebar with various icons and sections like "DEPLOY & RUN TRANSACTIONS", "ENVIRONMENT" (set to "Injected Provider - Metamask"), "ACCOUNT" (dropdown menu), "GAS LIMIT" (set to 3000000), "VALUE" (set to 0 Wei), "CONTRACT" (dropdown menu), and "Transactions recorded". Below these are "Deployed Contracts" and a note: "Currently you have no contract instances to interact with." The main central area has tabs for "Files" (with "New File" and "Open File" buttons) and "Access File System" (with GitHub, Gist, IPFS, and HTTP options). At the bottom, there's a "Learn" section with "Remix Basics" and a note: "An introduction to Remix's interface and basic operations." To the right of the main area, a modal window titled "Connect with MetaMask" is open. It says "Select the account(s) to use on this site" and shows a list with "Account 1 (0x01e...d41c) 0.5 SepoliaETH" checked. There are "New account" and "Cancel" buttons at the bottom of the modal. The background behind the modal shows the Remix documentation page with sections like "O LEARN from the Remix Team", "CREATE ERC20", "OPENZEPPELIN EF", "COOKBOOK", and "SOLIDITY".



Transferring Ethers using Metamask

Select the Account on the Metamask with which the Remix IDE needs to be connected

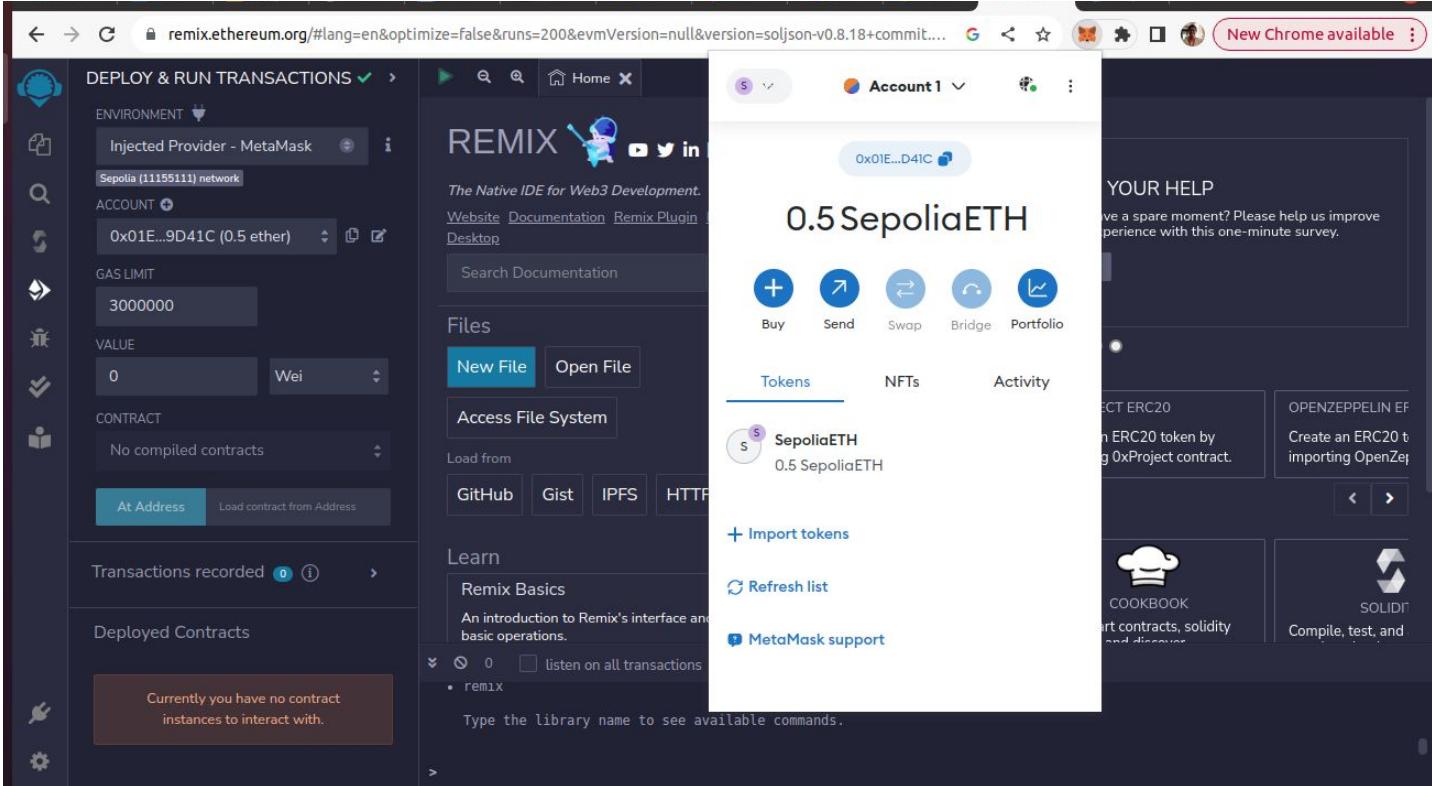


The screenshot shows a web browser window with the URL remix.ethereum.org/. On the left, the Remix IDE interface is visible, showing options for deploying and running transactions. In the center, a modal dialog from Metamask titled "Connect to Account 1" is displayed. The dialog shows the account address **(0x01e...d41c)** and asks for permission to "Allow this site to: See address, account balance, activity and suggest transactions to approve". At the bottom of the dialog are "Cancel" and "Connect" buttons. The background of the browser shows the Remix IDE's documentation and learn sections.



Transferring Ethers using Metamask

On the Remix IDE the Account details are displayed



The screenshot shows the Remix IDE interface. On the left, the sidebar displays "DEPLOY & RUN TRANSACTIONS" with fields for "ENVIRONMENT" (Injected Provider - Metamask, Sepolia), "ACCOUNT" (0x01E...D41C (0.5 ether)), "GAS LIMIT" (3000000), and "VALUE" (0 Wei). Below these are sections for "CONTRACT" (No compiled contracts) and "Transactions recorded" (0). The main area is titled "REMX" and includes "Files" (New File, Open File, Access File System), "Learn" (Remix Basics, MetaMask support), and a "Tokens" section showing "SepoliaETH" (0.5 SepoliaETH). It also features "Buy", "Send", "Swap", "Bridge", and "Portfolio" buttons. A sidebar on the right provides "YOUR HELP" and links to "CREATE ERC20", "OPENZEPPELIN EF", "COOKBOOK", and "SOLIDITY".





Mist Wallet -- deprecated

- The Ethereum mist was launched in 2017.
 - The mist is a blockchain-based browser that serves as a platform for operating blockchain networks and decentralized applications.
 - So, the mist is a search engine with several dApps connected to it, and Ethereum is one of these.
 - Mist is the official [Ethereum](#) wallet and electron application created by the same developers of the Ethereum ecosystem.
 - This [digital wallet](#) comes in two wallet solutions: the simple wallet and the multi-signature wallet which provides extra security.
 - Mist is a hybrid desktop application with a web interface which allows developers to quickly make changes.
- [MyEtherWallet](#) is a Mist application that allows users to access one [dApp](#), which is the official wallet dApp of Ethereum.
- Mist provides users a way to view the Ethereum blockchain and interact with specific components on the [blockchain](#) such as [Ether](#), DAO, and smart contracts





Mist Wallet -- deprecated



github.com/ethereum/mist

Product Solutions Open Source Pricing

Search or jump to... Sign in Sign up

This repository has been archived by the owner on Sep 5, 2020. It is now read-only.

ethereum / mist Public archive Notifications Fork 2.3k Star 7.4k

Code Issues 779 Pull requests 5 Actions Projects 3 Wiki Security Insights

develop 21 branches 42 tags Go to file Code

evertonfraga Update ISSUE_TEMPLATE.md 1 81228a3 on Mar 27, 2019 2,049 commits

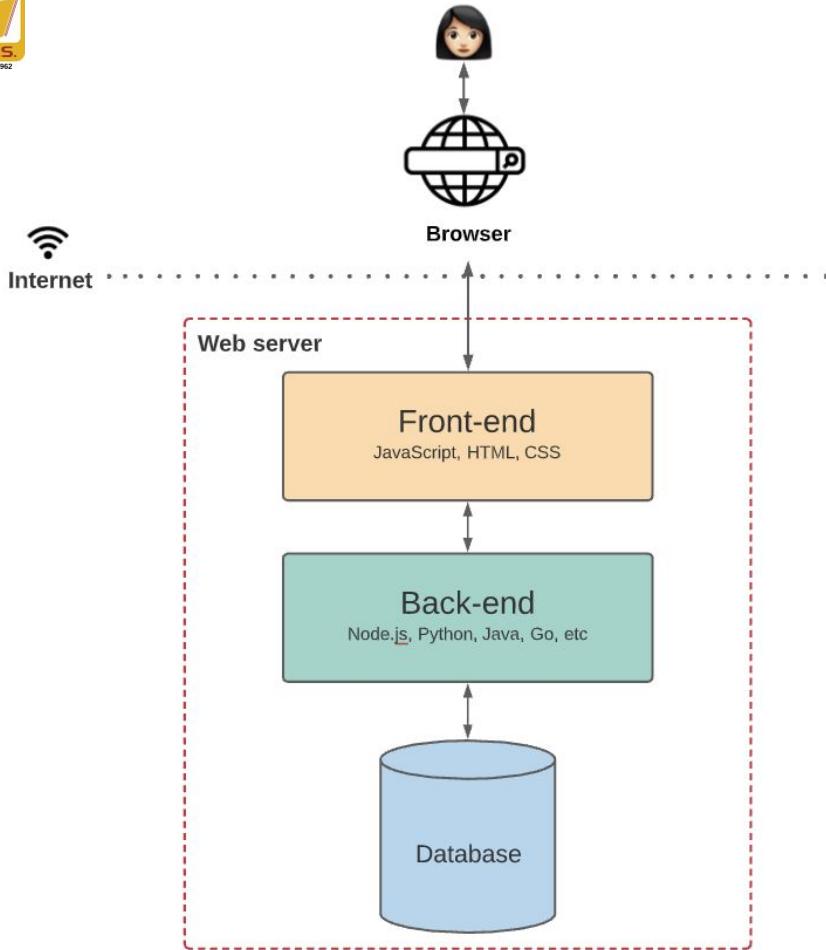
.circleci CircleCI: MacOS builds (#4144) 5 years ago

About [DEPRECATED] Mist. Browse and use Dapps on the Ethereum network. ethereum.org

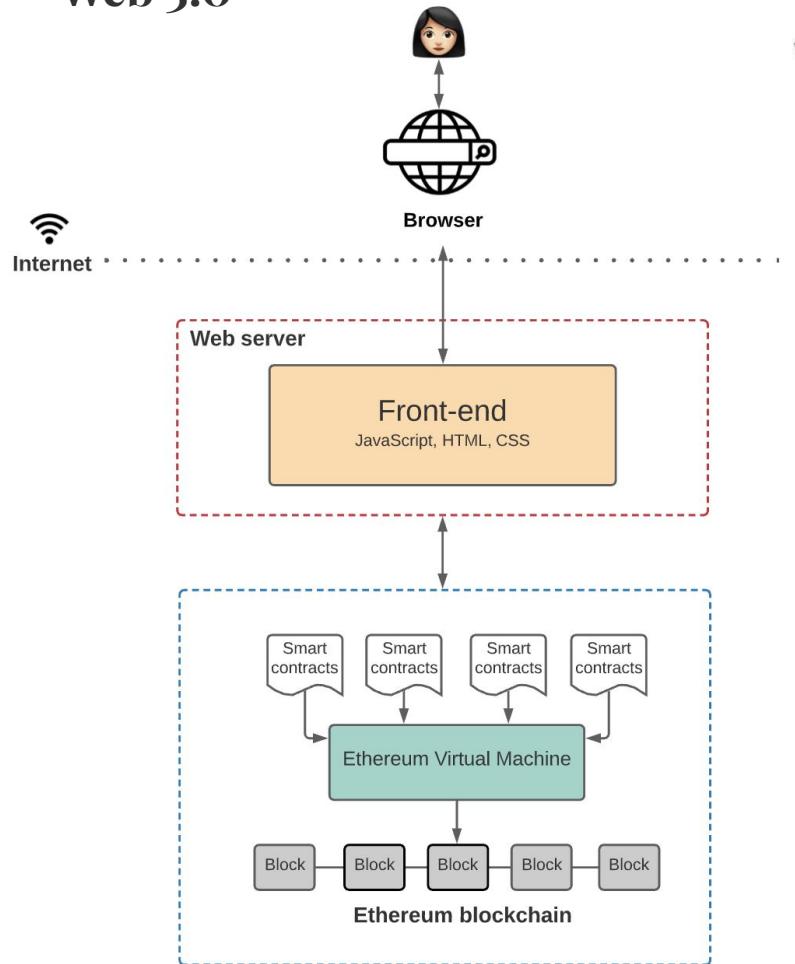
Courtesy : [Ethereum - Mist](#) [InsideBitcoins - Mist Wallet](#)



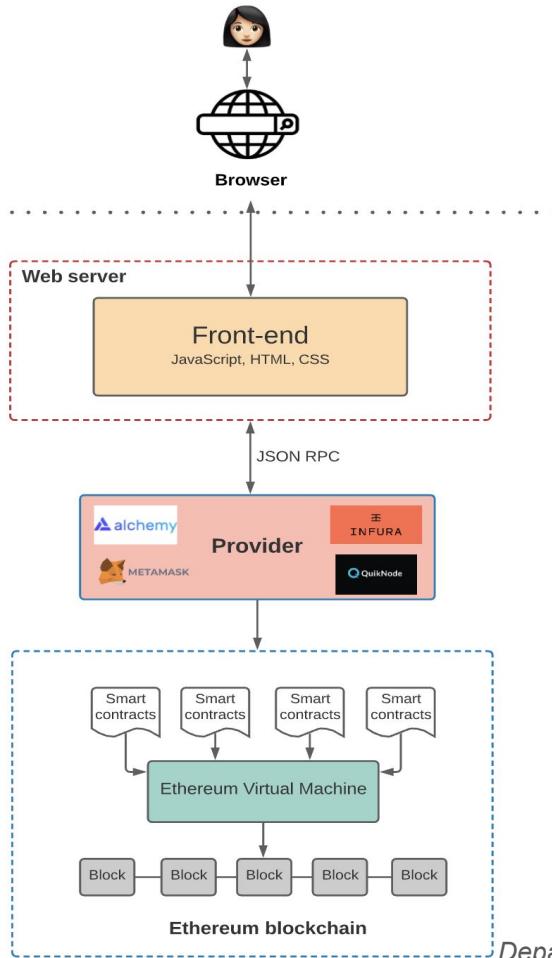
Web 2.0



Web 3.0



Frontend Code Communicate

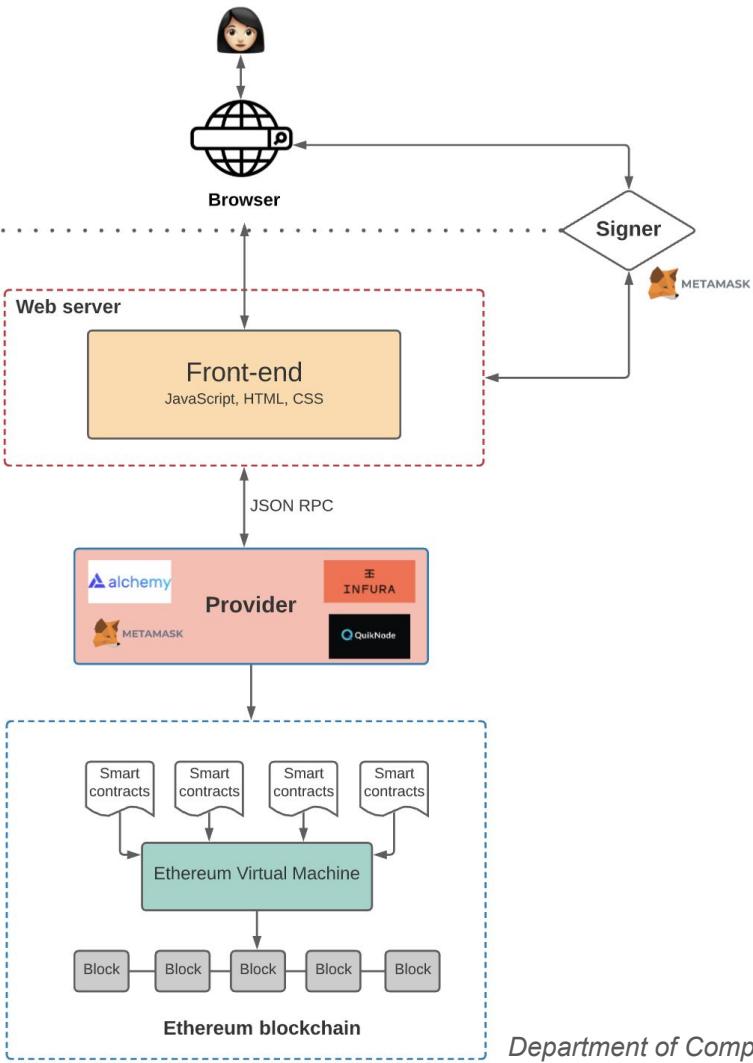


When we want to interact with the data and code on a blockchain, we need to interact with one of these nodes. This is because any node can broadcast a request for a transaction to be executed on the EVM. A miner will then execute the transaction and propagate the resulting state change to the rest of the network.

There are two ways to broadcast a new transaction:

1. Set up your own node which runs the Ethereum blockchain software
2. Use nodes provided by third-party services like [Infura](#), [Alchemy](#), and [Quicknode](#)

Every Ethereum client (i.e. provider) implements a JSON-RPC specification. This ensures that there's a uniform set of methods when frontend applications want to interact with the blockchain.



Frontend Code Communicate

When a user wants to publish a new post onto the chain, our DApp would ask the user to “sign” the transaction using their private key — only then would the DApp relay the transaction to the blockchain. Otherwise, the nodes wouldn’t accept the transaction.

This “signing” of transactions is where **Metamask** typically comes in.

Storage on the Blockchain

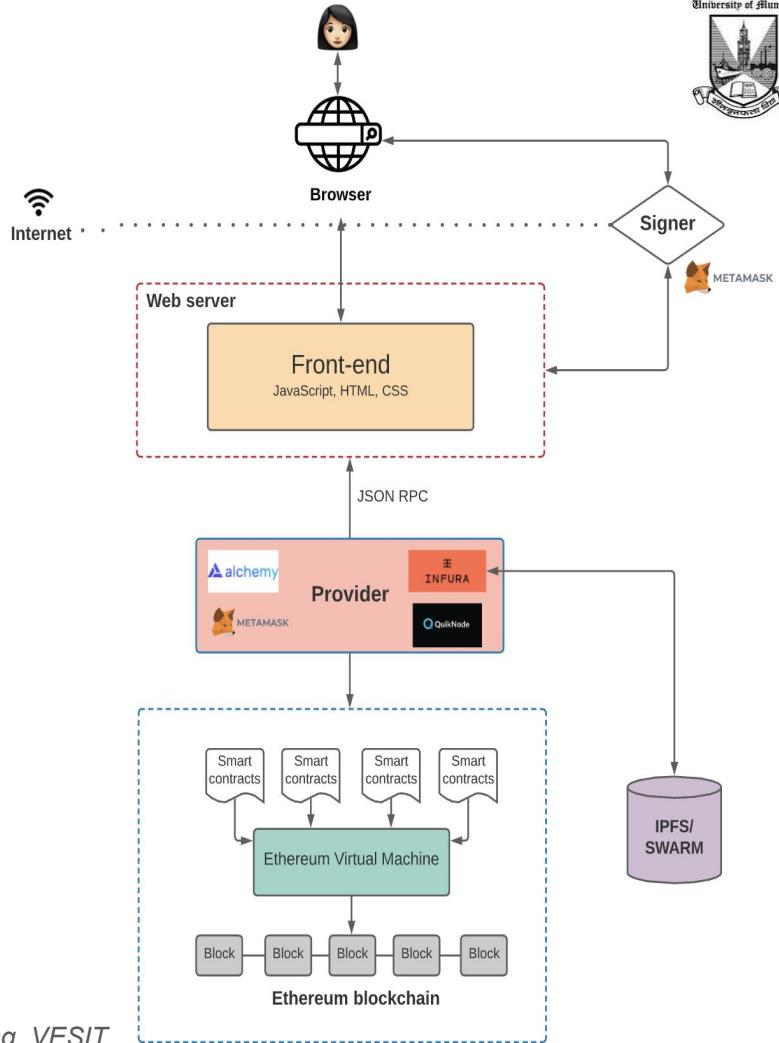
Storing everything on the blockchain gets really expensive,

Users pay extra for using your DApp every time their transaction requires adding a new state is not the best user experience. One way to combat this is to use a decentralized off-chain storage solution, like [IPFS](#) or [Swarm](#).

IPFS is a distributed file system for storing and accessing data. So, rather than storing data in a centralized database, the IPFS system distributes and stores the data in a peer-to-peer network. This makes it easy for you to retrieve when you need to.

IPFS also has an incentive layer known as “Filecoin.” This layer incentivizes nodes around the world to store and retrieve this data. You can use a provider like Infura (which provides you with an IPFS node) or Pinata (which provides an easy-to-use service where you can “pin” your files to IPFS and take the IP hash and store that on the blockchain).

Swarm is similar in that it’s a decentralized storage network, but there’s a notable difference. While Filecoin is a separate system, Swarm’s incentive system is built-in and enforced through smart contracts on the Ethereum blockchain for storing and retrieving data.

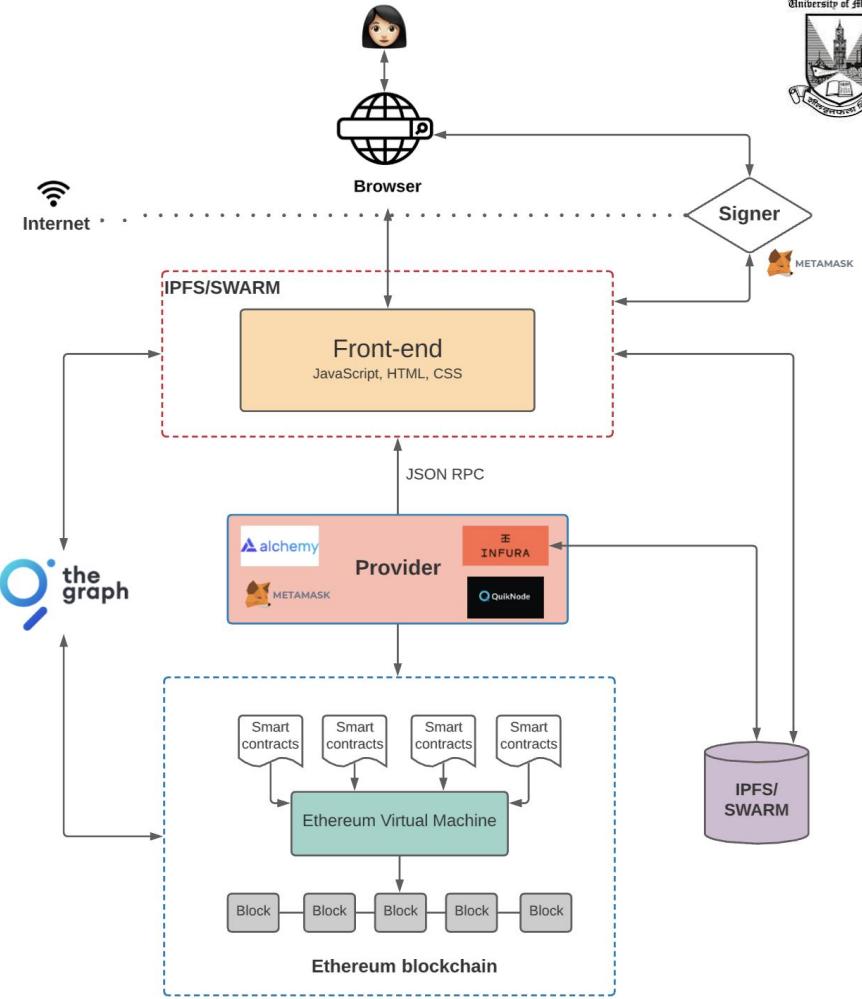


Querying the Blockchain

Smart Contract Events

You can use the Web3.js library to query and listen for smart contract events. You can listen to specific events and specify a callback every time the event is fired. For instance, if you have a smart contract that sends a continuous payment stream from person A to person B every block, then you can emit an event every time a new payment is made to person B. Your frontend code can listen to events being fired by the smart contract and carry out specific actions based on it.

The Graph is an off-chain indexing solution that makes it easier to query data on the Ethereum blockchain. The Graph allows you to define which smart contracts to index, which events and function calls to listen to, and how to transform incoming events into entities that your frontend logic (or whatever is using the API) can consume. It uses GraphQL as a query language, which many frontend engineers love because of how expressive it is compared to traditional REST APIs.





Ethereum frameworks

- Building a full-fledged dapp requires different pieces of technology. Software frameworks include many of the needed features or provide easy plugin systems to pick the tools you desire.
- Frameworks come with a lot of out-of-the-box functionality, like:
 - Features to spin up a local blockchain instance.
 - Utilities to compile and test your smart contracts.
 - Client development add-ons to build your user-facing application within the same project/repository.
 - Configuration to connect to Ethereum networks and deploy contracts, whether to a locally running instance, or one of Ethereum's public networks.
 - Decentralized app distribution - integrations with storage options like IPFS.



- [Truffle](#) - A development environment, testing framework, build pipeline, and other tools.
- [Hardhat](#) - Ethereum development environment for professionals.
- [Ape](#) - The smart contract development tool for Pythonistas, Data Scientists, and Security Professionals.
- [Brownie](#) - Python-based development environment and testing framework.
- [Web3j](#) - A platform for developing blockchain applications on the JVM.
- [OpenZeppelin SDK](#) - The Ultimate Smart Contract Toolkit: to help you develop, compile, upgrade, deploy and interact with smart contracts.
- [Create Eth App](#) - Create Ethereum-powered apps with one command. Comes with a wide offering of UI frameworks and DeFi templates to choose from.
- [Scaffold-Eth](#) - Ethers.js + Hardhat + React components and hooks for web3
- [Tenderly](#) - Web3 development platform that enables blockchain developers to build, test, debug, monitor, and operate smart contracts and improve dapp UX.
- [The Graph](#) - The Graph for querying blockchain data efficiently.
- [Alchemy](#) - Ethereum Development Platform.
- [Foundry](#) - A blazing fast, portable and modular toolkit for Ethereum application development written in Rust.
- [NodeReal](#) - Ethereum Development Platform.
- [thirdweb SDK](#) - Build web3 applications that can interact with your smart contracts using our powerful SDKs and CLI.
- [Chainsack](#) - Web3 (Ethereum and otherwise) Development Platform.



Case study of Ganache for Ethereum blockchain

- Ganache is a personal blockchain for rapid Ethereum and Filecoin distributed application development.
- One can use Ganache across the entire development cycle;
- enabling you to develop, deploy, and test your dApps in a safe and deterministic environment.
- Ganache comes in two flavors: a UI and CLI.
 - Ganache UI is a desktop application supporting Ethereum and Filecoin technology.
 - More robust command-line tool, ganache, is available for Ethereum development. It offers:
 - console.log in Solidity
 - Zero-config Mainnet and testnet forking
 - Fork any Ethereum network without waiting to sync
 - Ethereum JSON-RPC support
 - Snapshot/revert state
 - Mine blocks instantly, on demand, or at an interval
 - Fast-forward time
 - Impersonate any account (no private keys required!)
 - Listens for JSON-RPC 2.0 requests over HTTP/WebSockets
 - Programmatic use in Node.js
 - Pending Transactions

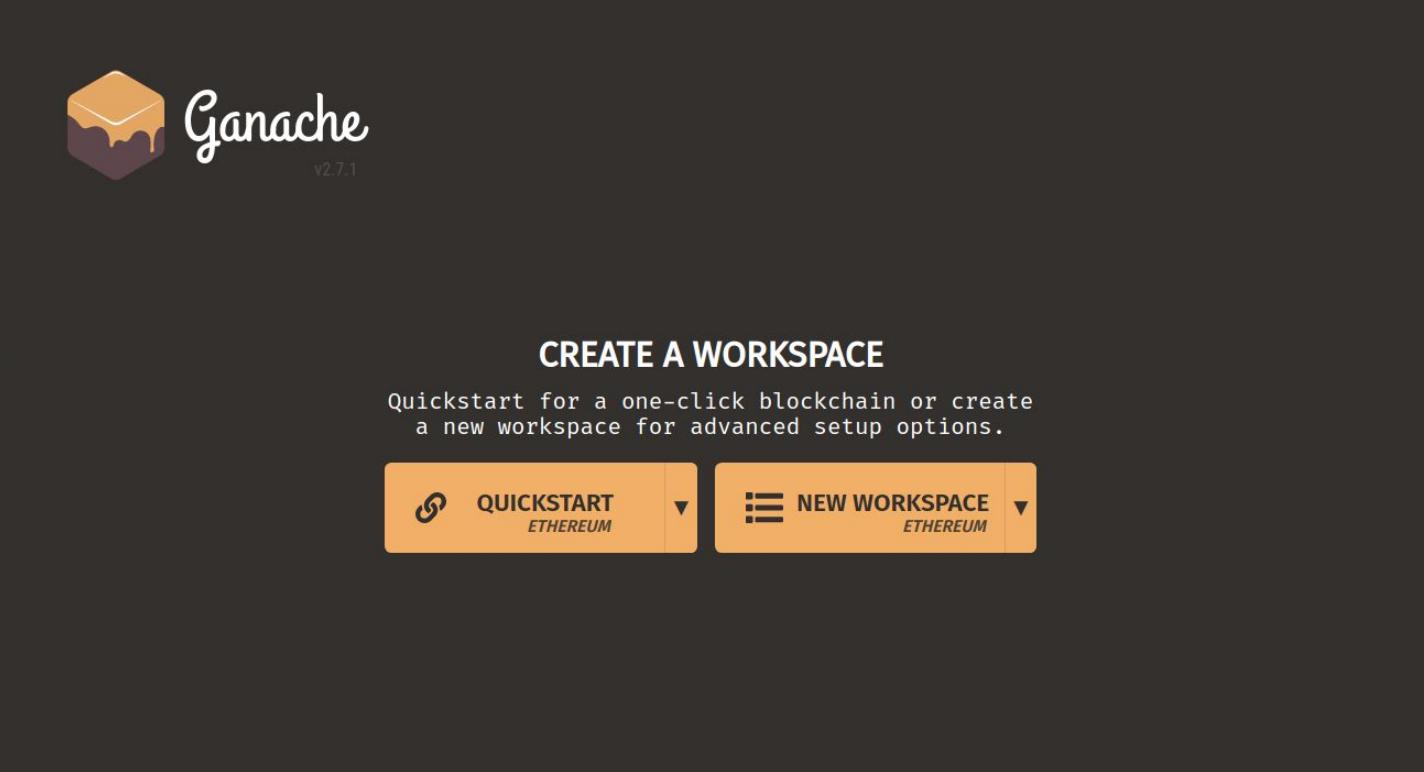
All versions of Ganache are available for Windows, Mac, and Linux.





Case study of Ganache for Ethereum blockchain

Flash screen of the Ganache after installation, Click on **Quickstart** to create a blockchain





Case study of Ganache for Ethereum blockchain



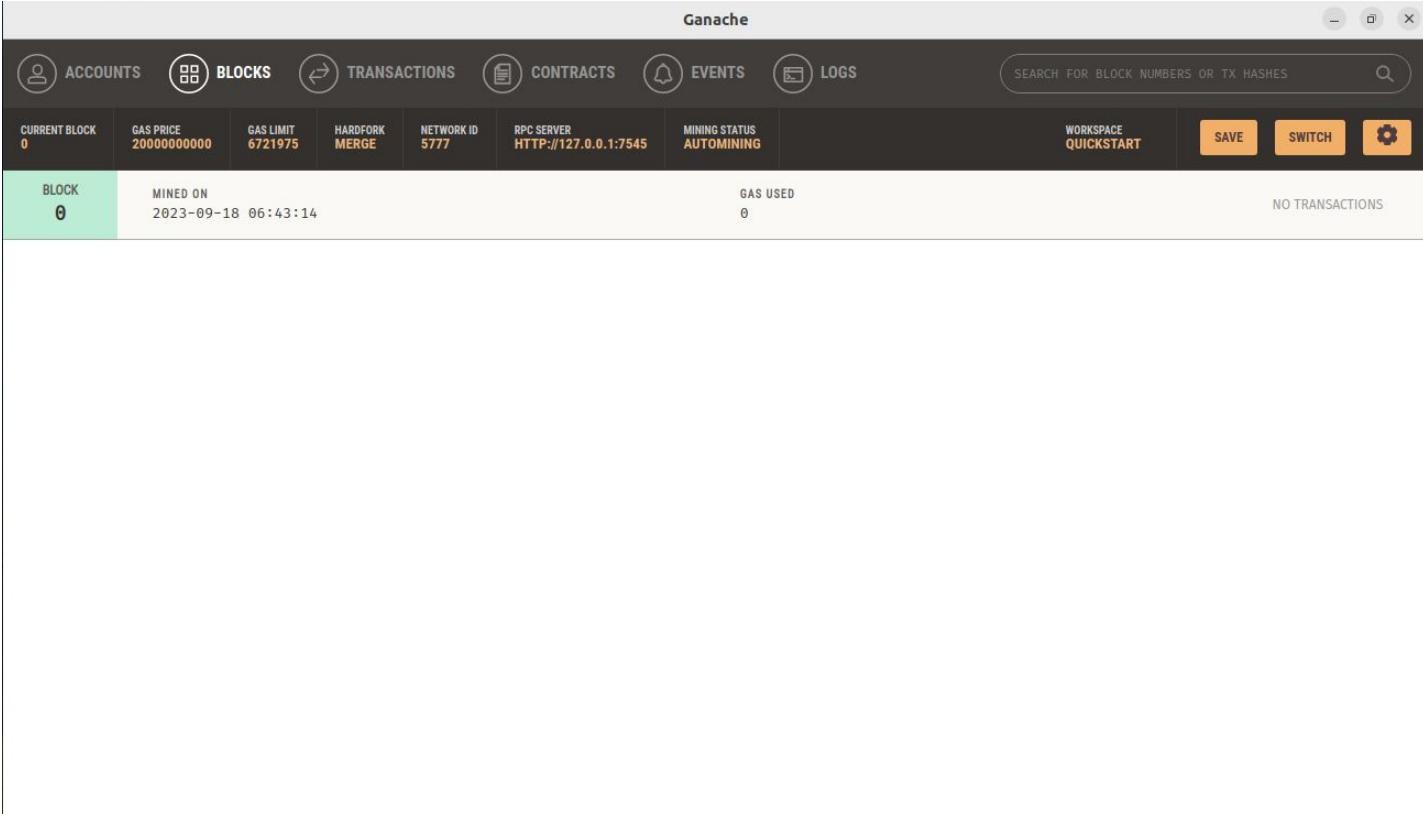
On the Accounts tab, enlists the 10 Accounts created each with 100 Ethers

The screenshot shows the Ganache interface with the 'ACCOUNTS' tab selected. At the top, there are tabs for ACCOUNTS, BLOCKS, TRANSACTIONS, CONTRACTS, EVENTS, and LOGS. A search bar is also present. Below the tabs, various network parameters are displayed: CURRENT BLOCK (0), GAS PRICE (20000000000), GAS LIMIT (6721975), HARDFORK (MERGE), NETWORK ID (5777), RPC SERVER (HTTP://127.0.0.1:7545), and MINING STATUS (AUTOMINING). Buttons for WORKSPACE (QUICKSTART), SAVE, SWITCH, and SETTINGS are located on the right.

MNEMONIC	HD PATH			
outer tiger monitor doctor guess success select run globe fluid wish attend	m44'60'0'0account_index			
ADDRESS 0x3EF48Bde2FeD5515d6bA2bf7b12444F9fb05113f	BALANCE 100.00 ETH	TX COUNT 0	INDEX 0	
ADDRESS 0x59CB64541a112B554ED0beB8BB76Cac4A3264e7A	BALANCE 100.00 ETH	TX COUNT 0	INDEX 1	
ADDRESS 0xa2E518AC650a803D7d0Ca390287de4B85f8Ffa2C	BALANCE 100.00 ETH	TX COUNT 0	INDEX 2	
ADDRESS 0x7F191E74166dDab416877fEb83373BDe2484c857	BALANCE 100.00 ETH	TX COUNT 0	INDEX 3	
ADDRESS 0xB73E56fA400acD07221818109d1f87CC1741FDEE	BALANCE 100.00 ETH	TX COUNT 0	INDEX 4	
ADDRESS 0x2873707f7508406BefC0da18B11D597A8B8539C3	BALANCE 100.00 ETH	TX COUNT 0	INDEX 5	

Case study of Ganache for Ethereum blockchain

On the Blocks Tab, we could see the Genesis Block created



The screenshot shows the Ganache application interface. At the top, there is a navigation bar with tabs: ACCOUNTS, BLOCKS (which is selected), TRANSACTIONS, CONTRACTS, EVENTS, and LOGS. A search bar is located above the main content area. Below the navigation bar, there is a header with various configuration options: CURRENT BLOCK (0), GAS PRICE (20000000000), GAS LIMIT (6721975), HARDFORK (MERGE), NETWORK ID (5777), RPC SERVER (HTTP://127.0.0.1:7545), MINING STATUS (AUTOMINING), WORKSPACE (QUICKSTART), and buttons for SAVE, SWITCH, and SETTINGS.

The main content area displays the details of the Genesis Block:

BLOCK	MINED ON	GAS USED	NO TRANSACTIONS
0	2023-09-18 06:43:14	0	



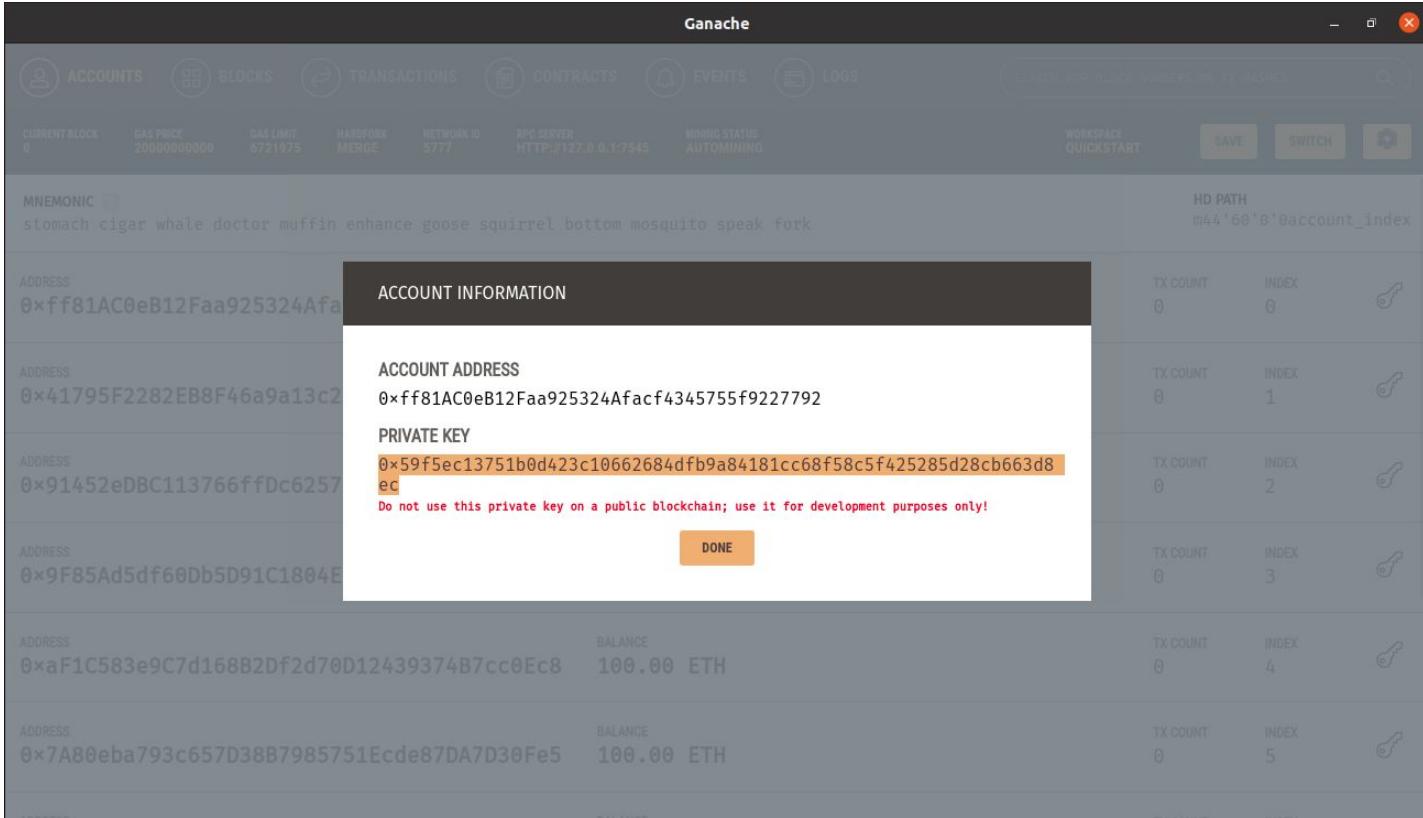
Case study of Ganache for Ethereum blockchain

On the Metamask, add Ganache Network as follows:

The screenshot shows the MetaMask extension settings page. The left sidebar has a 'Networks' section highlighted. The main area displays a list of networks, including Ethereum Mainnet, Lined Mainnet, Goerli test network, Sepolia test network, Lined Goerli test network..., and Ganache. A modal window is open on the right side, titled 'Add a network'. It contains fields for 'Network name' (set to 'Ganache'), 'New RPC URL' (set to 'HTTP://127.0.0.1:7545'), 'Chain ID' (set to '1337'), and 'Currency symbol' (set to 'ETH'). There is also an optional 'Block explorer URL' field which is empty. At the bottom of the modal are 'Cancel' and 'Save' buttons.

Case study of Ganache for Ethereum blockchain

From one account, copy the Private key

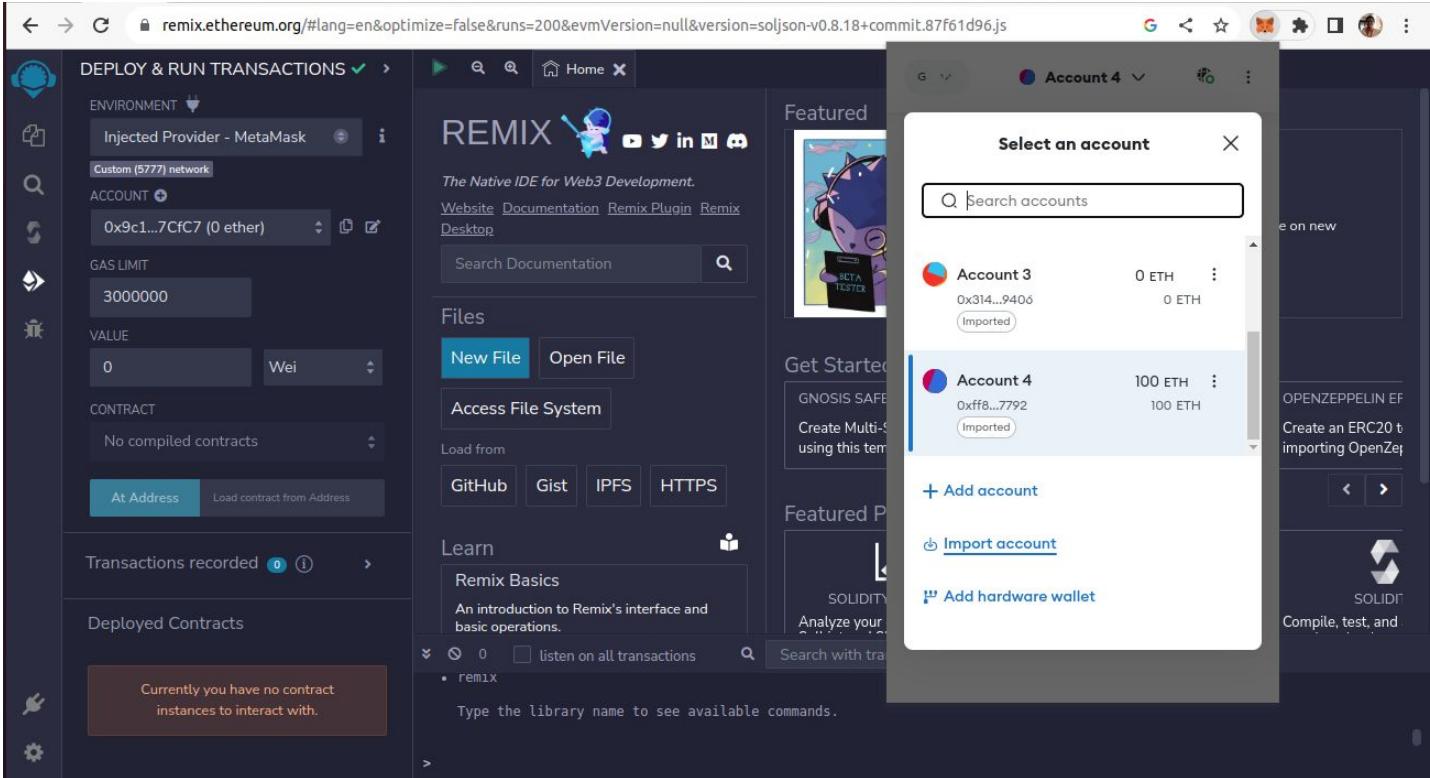


The screenshot shows the Ganache interface with the following details:

- MNEMONIC:** stomach cigar whale doctor muffin enhance goose squirrel bottom mosquito speak fork
- HD PATH:** m/44'/60'/0'/0/account_index
- ACCOUNT INFORMATION:** Address: 0x`ff81AC0eB12Faa925324Afa`, Private Key: 0x`59f5ec13751b0d423c10662684dfb9a84181cc68f58c5f425285d28cb663d8ec`
- Do not use this private key on a public blockchain; use it for development purposes only!**
- DONE** button
- ACCOUNTS:** ADDRESS: 0x`41795F2282EB8F46a9a13c2`, ADDRESS: 0x`91452eDBC113766ffDc6257`, ADDRESS: 0x`9F85Ad5df60Db5D91C1804E`
- BALANCE:** 100.00 ETH
- TRANSACTIONS:** TX COUNT: 0, INDEX: 0, TX COUNT: 0, INDEX: 1, TX COUNT: 0, INDEX: 2, TX COUNT: 0, INDEX: 3, TX COUNT: 0, INDEX: 4, TX COUNT: 0, INDEX: 5
- CONTRACTS:** NETWORK ID: 5777, RPC SERVER: HTTP://127.0.0.1:7545, MINING STATUS: AUTOMINING
- BLOCKS:** CURRENT BLOCK: 0, GAS PRICE: 20000000000, GAS LIMIT: 6721975, HARDFORK MERGE

Case study of Ganache for Ethereum blockchain

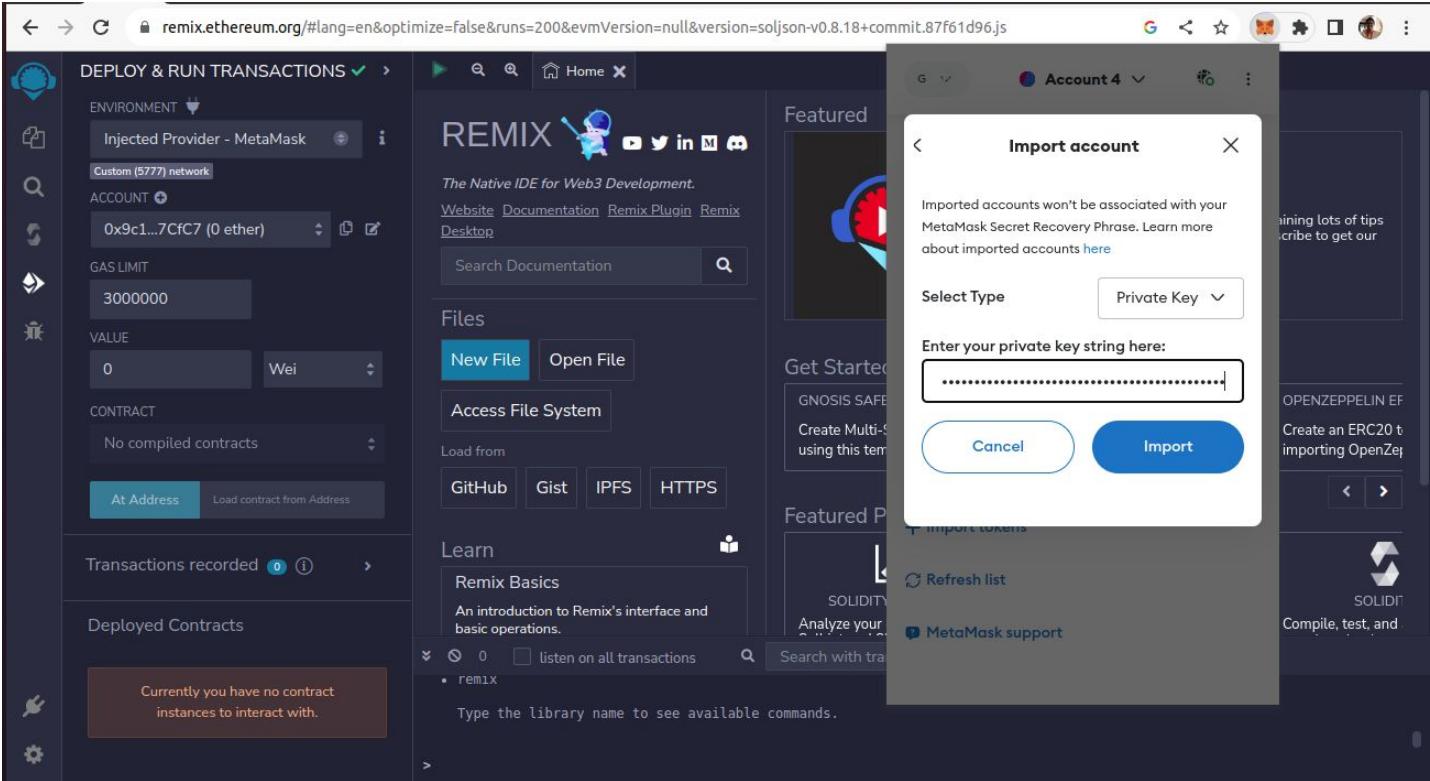
On the Metamask, select Import Account



The screenshot shows the Remix IDE interface. On the left, there's a sidebar for "DEPLOY & RUN TRANSACTIONS" with fields for ENVIRONMENT (Injected Provider - MetaMask), ACCOUNT (0x9c1...7Cfc7 (0 ether)), GAS LIMIT (3000000), and VALUE (0 Wei). Below this are sections for CONTRACT (No compiled contracts) and Transactions recorded (0). Deployed Contracts are listed as "Currently you have no contract instances to interact with." The main area features the REMIX logo and a search bar. A "Featured" section includes a "Get Started" button and links to Gnosis Safe and Create Multi-Sig. At the bottom, there's a "Learn" section with "Remix Basics" and a terminal-like interface for running commands like "remix". A "Select an account" modal is open in the center-right, listing "Account 3" (0 ETH, Imported) and "Account 4" (100 ETH, Imported). It also has buttons for "+ Add account", "+ Import account", and "+ Add hardware wallet".

Case study of Ganache for Ethereum blockchain

While importing Account on the Metamask, paste the Private key



The screenshot shows the Remix IDE interface on a web browser. The left sidebar contains options for deploying and running transactions, including environment selection (Injected Provider - Metamask), account selection (0x9c1...7CfC7 (0 ether)), gas limit (3000000), value (0 Wei), and contract selection (No compiled contracts). The main area features the REMIX logo and a search bar. A central modal window titled "Import account" is open, prompting the user to enter a private key string. The modal includes a note about imported accounts not being associated with the Metamask secret recovery phrase, a "Select Type" dropdown set to "Private Key", and a large input field for the private key string, which is currently filled with dots. Below the input field are "Cancel" and "Import" buttons.



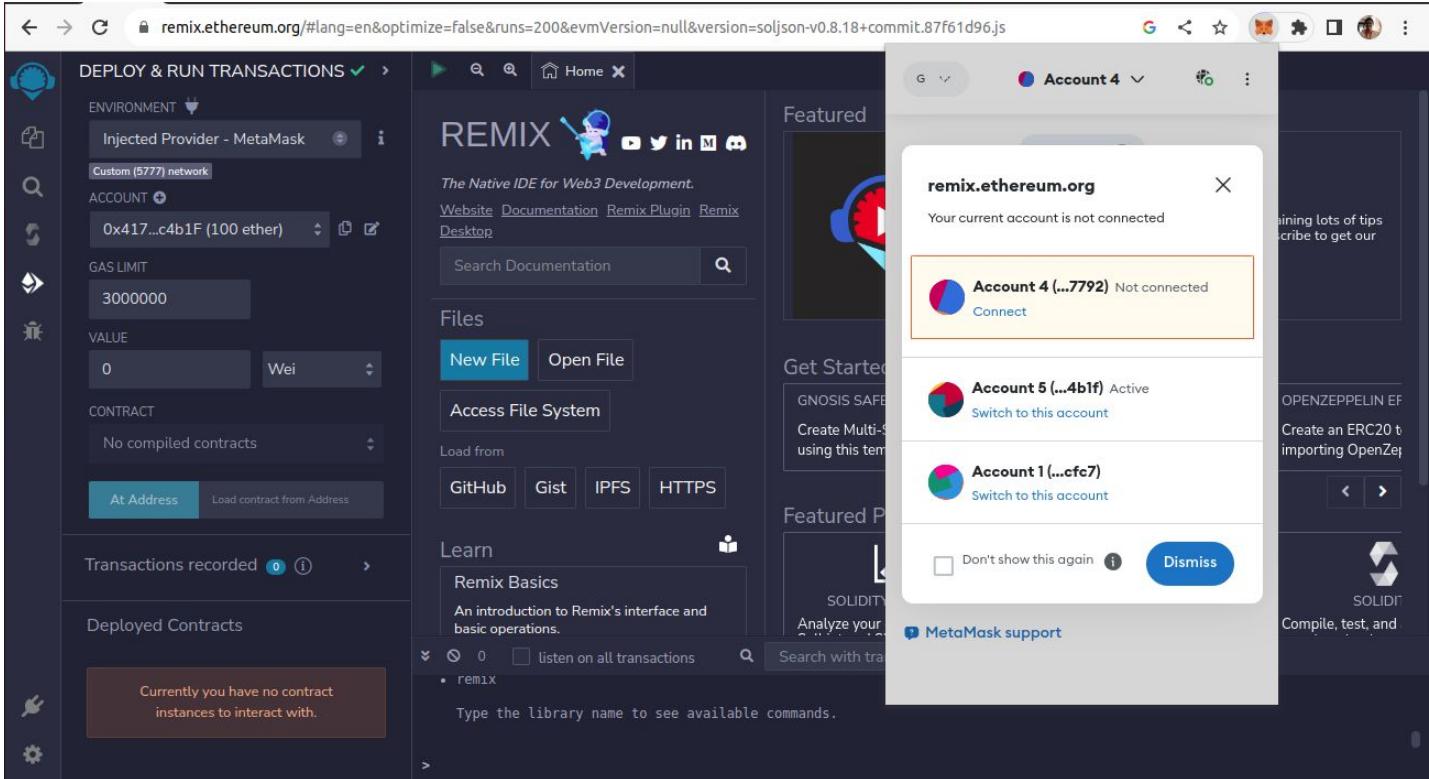
Case study of Ganache for Ethereum blockchain

Enlists the Account imported

The screenshot shows the Remix IDE interface at [remix.ethereum.org](https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js). The left sidebar displays deployment and transaction settings, including an environment set to "Injected Provider - MetaMask" and an account selected as "0x9c1...7Cfc7 (0 ether)". The main area features the REMIX logo and various development tools like GitHub, Gist, IPFS, and HTTPS. A central modal window titled "Select an account" lists two imported accounts: "Account 4" (0xff8...7792) and "Account 5" (0x417...4b1F), both showing a balance of 100 ETH.

Case study of Ganache for Ethereum blockchain

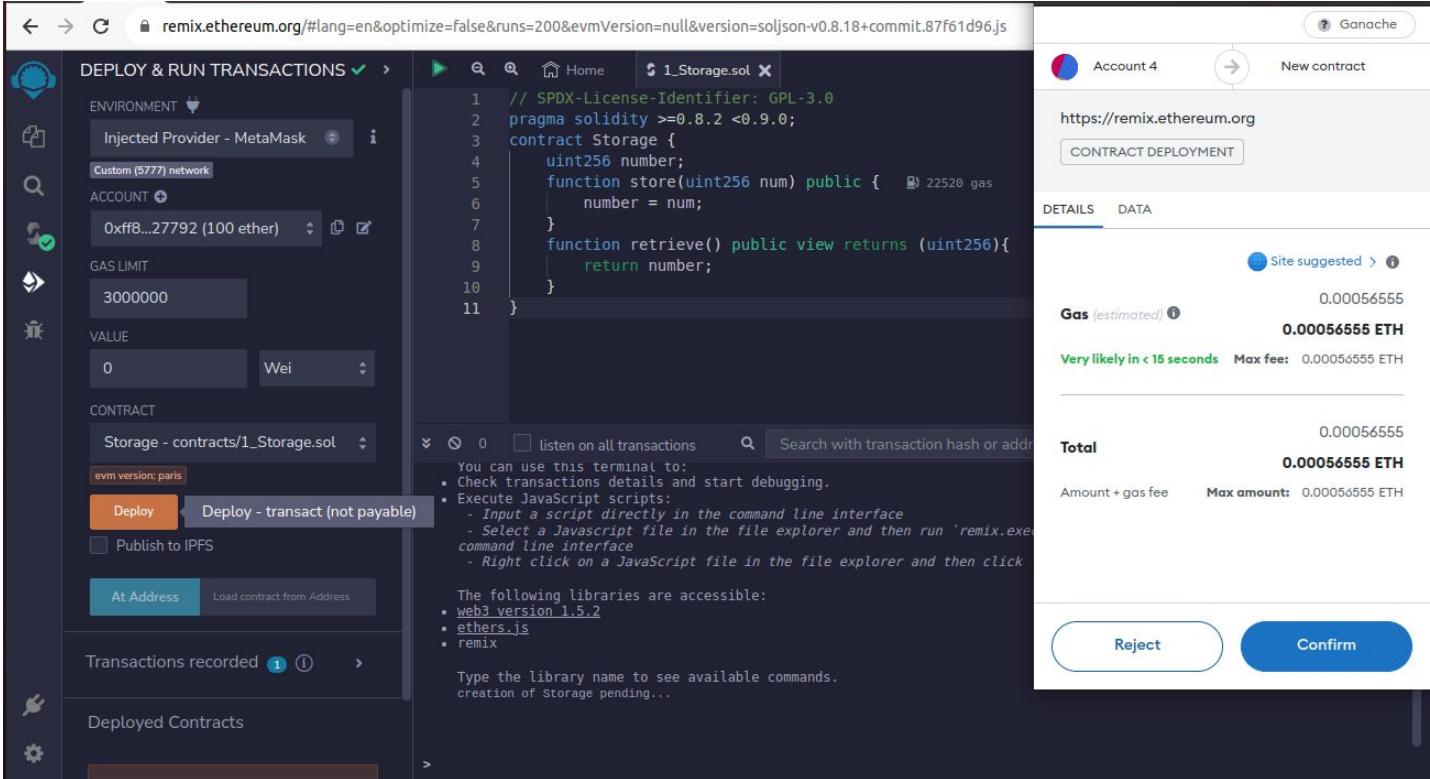
Select the Account from Metamask to be connected with Remix IDE



The screenshot shows the Remix IDE interface on a web browser. On the left, there's a sidebar for "DEPLOY & RUN TRANSACTIONS" with fields for ENVIRONMENT (set to "Injected Provider - MetaMask"), ACCOUNT (set to "0x417...c4b1F (100 ether)"), GAS LIMIT (set to 3000000), and VALUE (set to 0 Wei). Below these are sections for CONTRACT (No compiled contracts) and Transactions recorded (0). The main area is titled "REMIX" and features a search bar, documentation links, and a "Search Documentation" button. A "Files" section includes "New File" and "Open File" buttons, along with "Access File System" options for GitHub, Gist, IPFS, and HTTPS. A "Learn" section provides an introduction to Remix Basics. A central "Featured" section displays a "remix.ethereum.org" pop-up dialog. This dialog shows three accounts: "Account 4 (...7792)" (Not connected, with a "Connect" button), "Account 5 (...4b1f)" (Active, with a "Switch to this account" button), and "Account 1 (...fcf7)" (with a "Switch to this account" button). There are also "Dismiss" and "MetaMask support" buttons at the bottom of the dialog. The background of the Remix interface shows various Ethereum-related components like Gnosis Safe, OpenZeppelin, and Solidity.

Case study of Ganache for Ethereum blockchain

Deploy the Smart Contract



The screenshot shows the Remix IDE interface for deploying a Solidity smart contract named `1_Storage.sol`. The code defines a `Storage` contract with two functions: `store(uint256 num)` and `retrieve()`.

Left Panel (DEPLOY & RUN TRANSACTIONS):

- ENVIRONMENT:** Injected Provider - MetaMask, Custom (5777) network.
- ACCOUNT:** 0x... (100 ether)
- GAS LIMIT:** 3000000
- VALUE:** 0 Wei
- CONTRACT:** Storage - contracts/1_Storage.sol
- Deploy** button is highlighted.
- Deploy - transact (not payable)** button.
- Publish to IPFS** checkbox.
- At Address** and **Load contract from Address** buttons.
- Transactions recorded** section.
- Deployed Contracts** section.

Middle Panel (Code View):

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.8.2 <0.9.0;
contract Storage {
    uint256 number;
    function store(uint256 num) public {
        number = num;
    }
    function retrieve() public view returns (uint256){
        return number;
    }
}
```

Right Panel (Deployment Details):

- Account 4** (Ganache)
- New contract** button.
- URL:** https://remix.ethereum.org
- CONTRACT DEPLOYMENT** button.
- DETAILS** tab selected.
- Gas (estimated):** 0.00056555 ETH (Very likely in < 15 seconds, Max fee: 0.00056555 ETH)
- Total:** 0.00056555 ETH (Amount + gas fee, Max amount: 0.00056555 ETH)
- Buttons:** Reject and Confirm.



Case study of Ganache for Ethereum blockchain

On the Ganache Environment, the Transaction count is updated

The screenshot shows the Ganache application window. At the top, there are tabs for ACCOUNTS, BLOCKS, TRANSACTIONS, CONTRACTS, EVENTS, and LOGS. Below these are status indicators: CURRENT BLOCK (1), GAS PRICE (2000000000), GAS LIMIT (6721975), HARDFORK (MERGE), NETWORK ID (5777), RPC SERVER (HTTP://127.0.0.1:7545), and MINING STATUS (AUTOMINING). There are also buttons for WORKSPACE (SAVE, SWITCH, GEAR).

MNEMONIC: stomach cigar whale doctor muffin enhance goose squirrel bottom mosquito speak fork

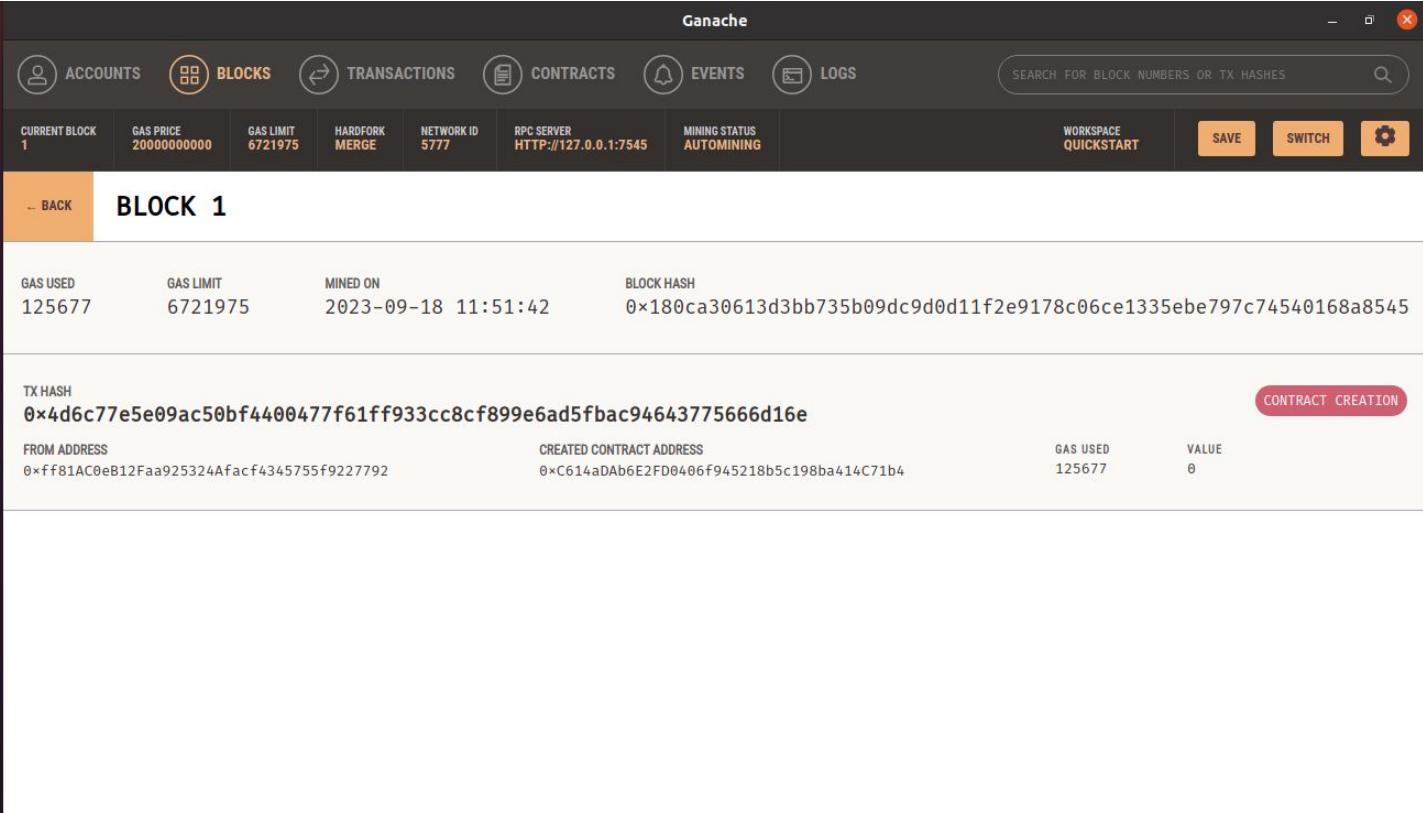
HD PATH: m/44'/60'0'@account_index

ADDRESS	BALANCE	TX COUNT	INDEX	Copy
0xff81AC0eB12Faa925324Afacf4345755f9227792	100.00 ETH	1	0	copy
0x41795F2282EB8F46a9a13c28dDed6215D65c4b1F	100.00 ETH	0	1	copy
0x91452eDBC113766ffDc6257fbC519E2E253D6dD4	100.00 ETH	0	2	copy
0x9F85Ad5df60Db5D91C1804E5789DE8610F1401DD	100.00 ETH	0	3	copy
0xaF1C583e9C7d168B2Df2d70D12439374B7cc0Ec8	100.00 ETH	0	4	copy
0x7A80eba793c657D38B7985751Ecde87DA7D30Fe5	100.00 ETH	0	5	copy



Case study of Ganache for Ethereum blockchain

Block 1 is added to the Blockchain which displays the Contract Created



The screenshot shows the Ganache interface with the following details:

Block 1 Details:

GAS USED	GAS LIMIT	MINED ON	BLOCK HASH
125677	6721975	2023-09-18 11:51:42	0x180ca30613d3bb735b09dc9d0d11f2e9178c06ce1335ebe797c74540168a8545

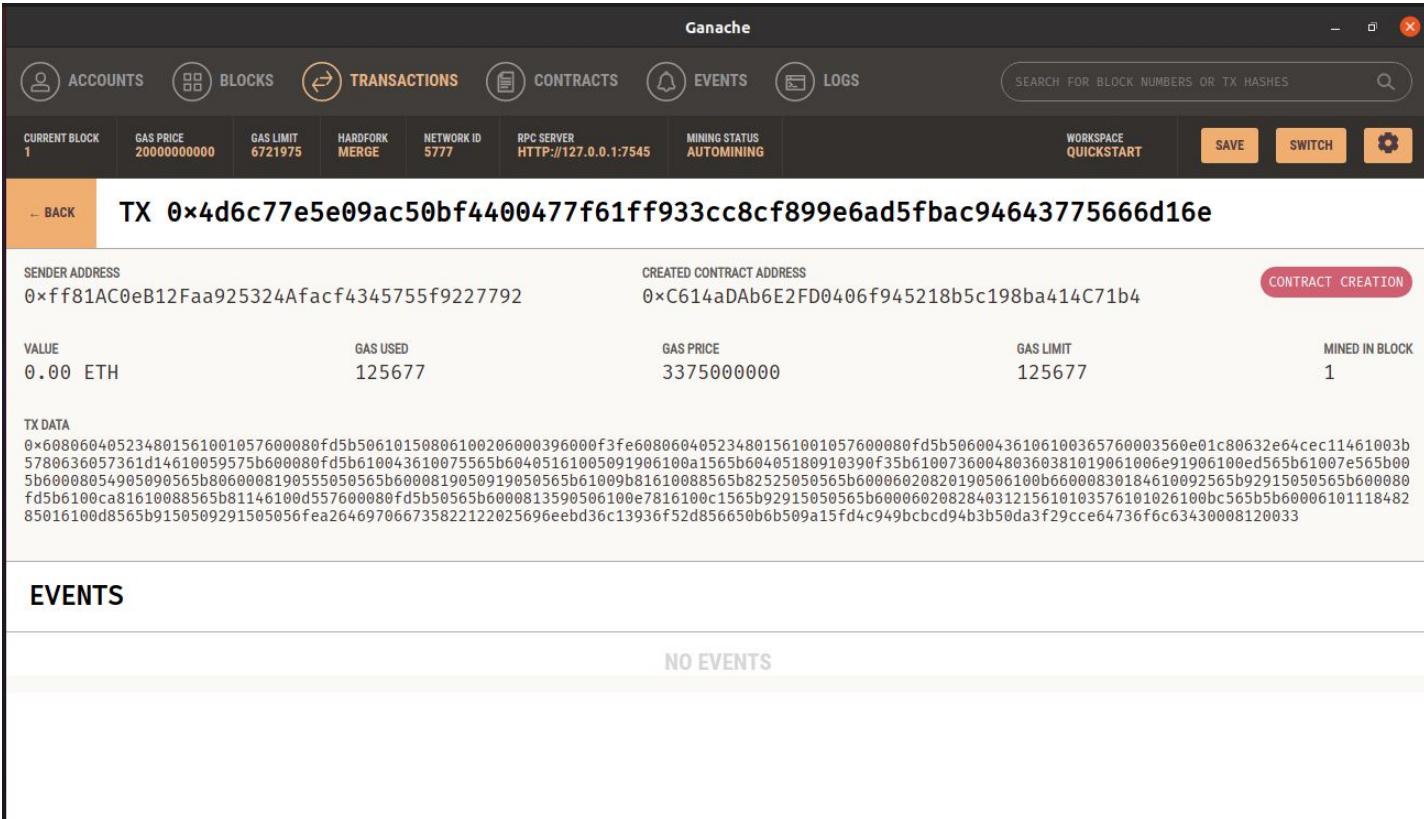
Contract Creation:

FROM ADDRESS	CREATED CONTRACT ADDRESS	GAS USED	VALUE
0xff81AC0eB12Faa925324Afacf4345755f9227792	0xC614aDAb6E2FD0406f945218b5c198ba414C71b4	125677	0



Case study of Ganache for Ethereum blockchain

Transaction details of the Contract is displayed on Ganache Environment



The screenshot shows the Ganache interface with the following details:

Header: Ganache

Menu Bar: ACCOUNTS, BLOCKS, TRANSACTIONS (highlighted), CONTRACTS, EVENTS, LOGS, SEARCH FOR BLOCK NUMBERS OR TX HASHES.

Toolbar: CURRENT BLOCK 1, GAS PRICE 20000000000, GAS LIMIT 6721975, HARDFORK MERGE, NETWORK ID 5777, RPC SERVER HTTP://127.0.0.1:7545, MINING STATUS AUTOMINING, WORKSPACE QUICKSTART, SAVE, SWITCH, GEAR.

Transaction Details:

- TX:** 0x4d6c77e5e09ac50bf4400477f61ff933cc8cf899e6ad5fbac94643775666d16e
- SENDER ADDRESS:** 0xf81AC0eB12Faa925324Afacf4345755f9227792
- CREATED CONTRACT ADDRESS:** 0xC614aDAbE2FD0406f945218b5c198ba414C71b4
- Contract Creation:** CONTRACT CREATION
- Value:** 0.00 ETH
- Gas Used:** 125677
- Gas Price:** 3375000000
- Gas Limit:** 125677
- Mined In Block:** 1

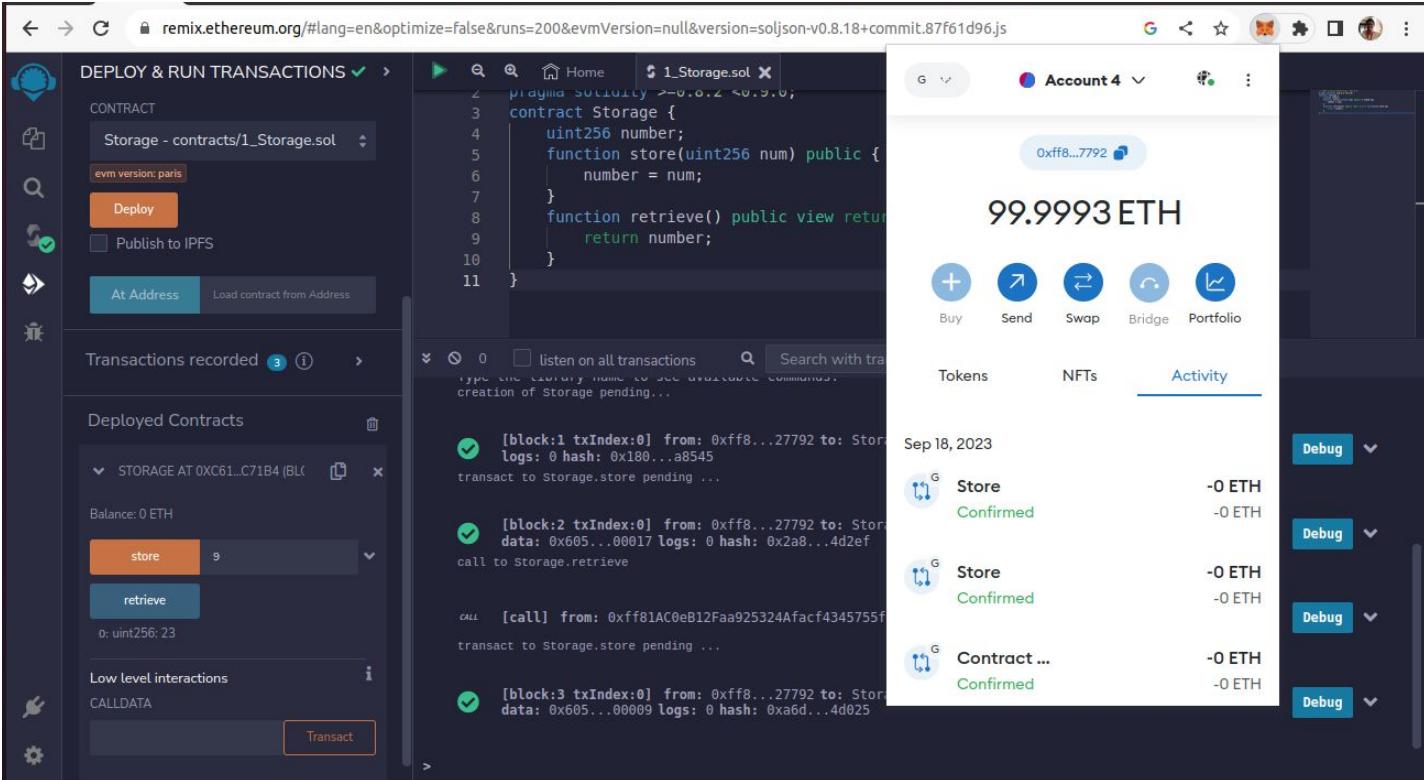
TX DATA:

```
0x608060405234801561001057600080fd5b50610150806100206000396000f3fe608060405234801561001057600080fd5b50600436106100365760003560e01c80632e64cec11461003b5780636057361d14610059575b600080fd5b610043610075565b60405161005091906100a1565b60405180910390f35b61007360048036038101906100e91906100ed565b61007e565b005b60008054905090565b8060008190555050565b6000819050919050565b61009b81610088565b82525050565b60006020820190506100b66000830184610092565b92915050565b600080fd5b6100ca81610088565b81146100d557600080fd5b50565b6000813590506100e7816100c1565b92915050565b600060208284031215610103576101026100b565b56000610111848285016100d8565b9150509291505056fea264697066735822122025696eebd36c13936f52d856650b6b509a15fd4c949bc94b3b50da3f29cce64736f6c63430008120033
```

Events: NO EVENTS

Case study of Ganache for Ethereum blockchain

After interacting with the Smart Contract, funds are updated on the Metamask



The screenshot displays the Remix IDE interface for interacting with a deployed Ethereum smart contract. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar shows the deployed contract 'Storage - contracts/1_Storage.sol'. The main central area shows the Solidity code for the Storage contract:

```
pragma SOLIDITY >=0.0.2 <0.9.0;
contract Storage {
    uint256 number;
    function store(uint256 num) public {
        number = num;
    }
    function retrieve() public view returns (uint256) {
        return number;
    }
}
```

The 'Transactions recorded' section shows four interactions with the contract:

- [block:1 txIndex:0] from: 0xff8...27792 to: Storage at 0xC61...C71B4 (Block 1) logs: 0 hash: 0x180...a8545 transact to Storage.store pending ...
- [block:2 txIndex:0] from: 0xff8...27792 to: Storage at 0xC61...C71B4 (Block 2) data: 0x605...00017 logs: 0 hash: 0x2a8...4d2ef call to Storage.retrieve
- [call] from: 0x81AC0eB12Faa925324Afacf4345755f transact to Storage.store pending ...
- [block:3 txIndex:0] from: 0xff8...27792 to: Storage at 0xC61...C71B4 (Block 3) data: 0x605...00009 logs: 0 hash: 0xa6d...4d025 transact to Storage.store pending ...

The 'Low level interactions' section includes 'CALLDATA' and a 'Transact' button.

To the right, the Metamask extension interface shows the account balance for 'Account 4' (0xff8...7792) which has increased to 99.9993 ETH. The activity tab shows the transaction history:

Date	Type	From	To	Value
Sep 18, 2023	Store	0xff8...27792	Storage at 0xC61...C71B4	-0 ETH
Sep 18, 2023	Store	0xff8...27792	Storage at 0xC61...C71B4	-0 ETH
Sep 18, 2023	Contract ...	0xff8...27792	Storage at 0xC61...C71B4	-0 ETH



Case study of Ganache for Ethereum blockchain

On the Ganache, the Contract call is listed

The screenshot shows the Ganache interface with the following details:

TX HASH: 0x66bed44db0b872d860eef0be07ea553815b9d7609464837b819983a65c735e5f

FROM ADDRESS: 0xff81AC0eB12Faa925324Afacf4345755f9227792

TO CONTRACT ADDRESS: 0xC614aDAb6E2FD0406f945218b5c198ba414C71b4

GAS USED: 43724

VALUE: 0

CONTRACT CALL

TX HASH: 0x4d6c77e5e09ac50bf4400477f61ff933cc8cf899e6ad5fbac94643775666d16e

FROM ADDRESS: 0xff81AC0eB12Faa925324Afacf4345755f9227792

CREATED CONTRACT ADDRESS: 0xC614aDAb6E2FD0406f945218b5c198ba414C71b4

GAS USED: 125677

VALUE: 0

CONTRACT CREATION



Exploring etherscan.io



etherscan.io

ETH Price: \$1,847.19 (-0.04%) Gas: 11 Gwei



Home Blockchain ▾ Tokens ▾ NFTs ▾ Resources ▾ Developers ▾ More ▾ | [Sign In](#)

The Ethereum Blockchain Explorer

All Filters



Search by Address / Txn Hash / Block / Token / Domain Name



Sponsored:  QREDO: Looking to access DeFi with TOP-TIER SECURITY for free? Qredo Web3 Wallets! Try it now!



 ETHER PRICE

\$1,847.19 @ 0.06286 BTC (-0.04%)

 TRANSACTIONS

2,062.22 M (10.4 TPS)

MED GAS PRICE

11 Gwei (\$0.43)

TRANSACTION HISTORY IN 14 DAYS



 MARKET CAP

\$221,929,476,488.00

 LAST FINALIZED BLOCK

17911193

LAST SAFE BLOCK

17911225

Latest Blocks

 17911273
11 secs ago

Fee Recipient [Fee Recipient: 0xe...](#)
86 txns in 12 secs

0.00783 Eth

Latest Transactions

 0x07c8be652ab4...
11 secs ago

From [0x704dDD...FA59B669](#)
To [0xD35af...68C1749e](#)

0 Eth



Exploring etherscan.io and ether block structure

What is Etherscan ?

- effective block explorer
- abstracts the complexity of retrieving data on the blockchain
- by providing a simple interface for Ethereum blockchain.
- allows users to look through the transactions, smart contracts, wallet addresses, blocks, and other on-chain data related to Ethereum-based trades and assets.
- all Ethereum interactions are publicly recorded on Blockchain.
- Helps to visualize past transactions on the Ethereum network that spans from transferring coins to purchasing non-fungible tokens (NFTs).



Exploring [etherscan.io](#) and ether block structure

Why Do You need Etherscan?

- allows users to **search and explore** the Ethereum blockchain
 - for transactions, tokens, addresses, and other activities.
- Provides **Token Tracker** for **Ether** by providing information,
 - such as price, holders, total supply, and transfers.
- It allows to **monitor high-profile NFT holders' portfolios.**
 - track what a certain NFT project or company is doing with its funds.
 - Eg : Gamestop dumped \$42 million worth of NFX tokens, shortly after it received a \$100 million grant from Immutable X. The proof of this incident was provided by Etherscan.
- Etherscan **helps the victims of crypto scams.**
 - By tracking the transaction of NFT projects' potential rug pulls and NFT scams,
- Etherscan **tracks the “whales” wallet which holds a large amount of ETH.**
 - helpful as you can predict the price movement with this information.



Exploring etherscan.io and ether block structure

Why Do You need Etherscan?

- The use cases may involve:
 - Searching and viewing wallets and transactions
 - Tracking gas fees
 - Interacting with smart contracts
 - Revoking or reviewing your token approvals
 - Using **Token ignore list feature to hide your tokens**
 - Staying up-to-date with the Ethereum ecosystem

Note:

- **Etherscan is not a wallet;**
- Use Etherscan to search your Ethereum wallet and track the transactions
- If you want to **enter into a transaction or use ETH**, access your crypto exchange or wallet.



Exploring etherscan.io - Blocks

ETH Price: \$1,846.87 (-0.10%) Gas: 11 Gwei

Search by Address / Txn Hash / Block / Token / Domain Name

Blocks

Sponsored: 🦇 Wall Street Memes Raises \$10million, next Pepe Coin? [Buy \\$WSM Now!](#)

NETWORK UTILIZATION (24H)
50.9%

LAST SAFE BLOCK
17911513

PRODUCED BY MEV BUILDERS (24H)
85.3%

BURNT FEES 🔥
3,535,172.50 ETH

Total of 17,911,565 blocks
(Showing blocks between #17911540 to #17911564)

Block	Age	Txn	Fee Recipient	Gas Used	Gas Limit	Base Fee	Reward	Burnt Fees (ETH)
17911564	13 secs ago	135	builder0x69	13,339,995 (44.47% -11%)	30,000,000	11.66 Gwei	0.06822 ETH	0.155558 (69.51%)
17911563	25 secs ago	118	beaverbuild	12,680,964 (42.27% -15%)	30,000,000	11.89 Gwei	0.0204 ETH	0.150787 (88.08%)
17911562	37 secs ago	148	MEV Builder: 0xBaF...e19	17,297,338 (57.66% +15%)	30,000,000	11.66 Gwei	0.0868 ETH	0.201815 (69.92%)
17911561	49 secs ago	161	rsync-builder	13,555,963 (45.19% -10%)	30,000,000	11.8 Gwei	0.02233 ETH	0.160090 (87.76%)
17911560	1 min ago	124	Flashbots: Builder	11,301,329 (37.67% -25%)	30,000,000	12.18 Gwei	0.03133 ETH	0.137708 (81.46%)
17911559	1 min ago	163	beaverbuild	27,806,483 (92.69% +85%)	30,000,000	11.01 Gwei	0.03 ETH	0.306152 (91.07%)





Exploring etherscan.io - ether block structure



← → G 🔒 etherscan.io/block/17911273



ETH Price: \$1,846.89 (-0.06%) Gas: 11 Gwei

Search by Address / Txn Hash / Block / Token / Domain Name



Home Blockchain ▾ Tokens ▾ NFTs ▾ Resources ▾ Developers ▾ More ▾ | Sign In

Block #17911273

Overview Consensus Info Comments

② Block Height: 17911273 < >

② Status: Unfinalized

② Timestamp: ① 5 mins ago (Aug-14-2023 06:24:23 AM +UTC)

② Proposed On: Block proposed on slot [7097520](#), epoch [221797](#)

② Transactions: 86 transactions and 27 contract internal transactions in this block

② Withdrawals: 16 withdrawals in this block

② Fee Recipient: Fee Recipient: 0xeBe...Acf [in 12 secs](#)

② Block Reward: 0.007831604199409442 ETH (0 + 0.088851720809664918 - 0.081020116610255476)

② Total Difficulty: 58,750,003,716,598,352,816,469

② Size: 34,993 bytes





Exploring etherscan.io - ether block structure (after Finalized)

ETH Price: \$1,847.09 (-0.09%) Gas: 13 Gwei

Search by Address / Txn Hash / Block / Token / Domain Name

② Status: Finalized

② Timestamp: 1 hr 2 mins ago (Aug-14-2023 06:24:23 AM +UTC)

② Proposed On: Block proposed on slot 7097520, epoch 221797

② Transactions: 86 transactions and 27 contract internal transactions in this block

② Withdrawals: 16 withdrawals in this block

② Fee Recipient: Fee Recipient: 0xeBe...Acf in 12 secs

② Block Reward: 0.007831604199409442 ETH (0 + 0.088851720809664918 - 0.081020116610255476)

② Total Difficulty: 58,750,003,716,598,352,816,469

② Size: 34,993 bytes

② Gas Used: 6,500,852(21.67%) -57% Gas Target

② Gas Limit: 30,000,000

② Base Fee Per Gas: 0.000000012462999713 ETH (12.462999713 Gwei)

② Burnt Fees: 0.081020116610255476 ETH

② Extra Data: geth go1.20.4 linux (Hex:0xd883010c00846765746888676f312e32302e34856c696e7578)





Exploring etherscan.io - ether block structure (Transactions)

ETH Price: \$1,846.49 (-0.12%) Gas: 11 Gwei

Search by Address / Txn Hash / Block / Token / Domain Name

A total of 86 transactions found

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0x07c8be652ab449a0...	Transfer	17911273	56 mins ago	0x704dDD...FA59B669	Ecomi: OMI Token	0 ETH	0.00052014
0x92d63652a7dc1cea...	Approve	17911273	56 mins ago	0x06744d...2f8d00F6	0xC93CEb...34F54Eb0	0 ETH	0.00033326
0x8533e97ea1047622...	Swap Exact E...	17911273	56 mins ago	0x095aF7...f915EAe5	0x2Ec705...0661ADd1	0.215 ETH	0.00212728
0x0b33bb45e099736a...	Execute Trans...	17911273	56 mins ago	0x3a0eDE...CA5f7966	0x832154...6F8A3DD5	0 ETH	0.00379136
0x3dd148de9bf4b08d5...	Transfer	17911273	56 mins ago	0xAf581a...c98f7309	0x51A7C1...66D7c3eb	0.001015472 ETH	0.00026274
0xc580b9bc1b280f528...	Transfer	17911273	56 mins ago	0xEAbfcE...D6490170	0xca0049...fA862431	0.0392 ETH	0.00026382
0xb19a1161145388c2...	Claim Asset	17911273	56 mins ago	0x01Be90...7bb56C91	Polygon (Matic): zkEV...	0 ETH	0.00114389
0xc393b0ced53d543e...	Transfer	17911273	56 mins ago	0x155EDf...7B646B51	0x572E31...13fb9F1D	0.971238617 ETH	0.00026382
0x467c9e2fe81721502...	Approve	17911273	56 mins ago	0xF040A1...631b18e2	Tether: USDT Stablecoin	0 ETH	0.00061007
0xcc25f74892dfbee70...	Execute	17911273	56 mins ago	0xfC9150...46faa39E	Uniswap: Universal Ro...	1 ETH	0.00166895
0x782fb156459aff847...	Approve	17911273	56 mins ago	0x6A69e8...AA80f45C	0x55c7Eb...19E89599	0 ETH	0.0005822





Exploring etherscan.io - ether block structure (Consensus Info)



ETH Price: \$1,846.89 (-0.06%) Gas: 11 Gwei

Search by Address / Txn Hash / Block / Token / Domain Name

Etherscan Home Blockchain Tokens NFTs Resources Developers More | Sign In

Block #17911273

Overview Consensus Info Comments

Slot: 7097520

Epoch: 221797

Proposer Index: 789004

Slot Root Hash: 0xbd18eefec045645b886155eb2c1fd9957bd1345be238aaeb921e792ce836af8

Beacon Chain Deposit Count: 908298

Slot Graffiti: 0x (Hex:Null)

Block Randomness: 0xbc1e1c64da8533dfd6f02958b447cde2a73ab4b9bf7e35886fb9a94edb2d607c

Randao Reveal: 0x95f8fd2475f95a1530fd4090fc0448e3bb6136b526a7c9acd48a6d3fbce7c845e46da6c62524c6d6ca8c4886f32338610d12ada8c12042f180140c693791283f065d3af900efdd92b070c52c6760ca7e51d5a3d7ec495015261c72e9865ff3c9



Exploring [etherscan.io](#) and ether block structure (Consensus Info - validator)

beaconscan.com/validator/789004

Mainnet | Last Epoch 221789 at Slot 7097279

Search by Epoch / Slot / PublicKey

Overview Proposals (0) Slasher Inclusions (0) Deposits (1) Stats

⑦ Index	789004
⑦ Balance Snapshot	32.01223515 ETH
⑦ Effective Balance	32 ETH
⑦ Deposits Received	32 ETH
⑦ Status	Active
⑦ Public Name Tag	-

⑦ Total Income	0.012190529 ETH (est APR of 1.28%)
----------------	------------------------------------

⑦ Daily Income Chart
2023-07-16 - 2023-08-14

Over the last 30 days, there were ▲ 11 days which had a positive income and ▼ 1 day with a negative income



⑦ Validator Performance	-
-------------------------	---

⑦ Eligible Epoch	209889 ① 52 days 22 hrs ago (Jun-22-2023 08:09:59 AM)
------------------	---

⑦ Activation Epoch	219348 ① 10 days 21 hrs ago (Aug-03-2023 09:07:35 AM)
--------------------	---





Exploring [etherscan.io](#) and ether block structure (Consensus Info - validator)

beaconscan.com/validator/789004#deposits

Home | Mainnet | Last Epoch 221789 at Slot 7097279 Search by Epoch / Slot / PublicKey

Overview Proposals (0) Slasher Inclusions (0) Deposits (1) Stats

(Step 1 of 2) To participate in the network, a minimum of 32 ETH from the Eth_1 network has to be sent to a validator deposit contract

Eth 1 A total of 32 Ether in Deposits was originally sent to the Ethereum Mainnet Eth_1 deposit contract for funding this validator					
AGE	ETH_1 TXHASH	ETH_1 FROM ADDRESS	SIGNATURE	AMOUNT	STATUS
53 days 16 hrs ago	0x8a76f4...cdf3c0c1	0xa40dfe....17838703	0xa1f590...b2cf77dc	32 Eth	Valid



(Step 2 of 2) Deposits sent from the Eth_1 network (above) are processed and included in the Beacon chain. Each validator needs a minimum balance of 32 ETH to get activated.

Eth 2.0 Beacon A total of 32 Ether in Deposits was processed by the Eth_2 Beacon Chain network

EPOCH	SLOT	AGE	WITHDRAWAL CREDENTIALS	SIGNATURE	AMOUNT
209888	6716431	52 days 22 hrs ago	0x010000...e4722a1b	0xa1f590...b2cf77dc	32 Eth





Exploring [etherscan.io](#) - Checking Transactions

- **Keeping track of Ethereum blockchain transactions** (most fundamental use case of Etherscan).
- If a user understands how to follow cryptocurrency, he will be able to unlock the rest of the cryptocurrency information.
- Steps to see the history of a particular crypto transaction on Ethereum
 1. Find the transaction hash(Tx Hash) or transaction ID (TxID).
(identify a particular transaction on the blockchain.)
 2. Go to the Etherscan search bar and enter the TxID or Tx Hash and click “search”
 3. A page will display information related to that transaction, such as its status (successful, failed, pending), time and date of transaction, and the number of blocks in which the transaction was recorded.
 4. You can also find senders' and receivers' addresses, gas prices, and transaction fees. Click “see more” to get these additional details.

Note :

- One can also access the transaction using the recipient's wallet address.
- Insert the recipient's wallet address in the search bar.
- All the transaction details associated with a particular wallet address will be displayed at the bottom of the page.



Exploring etherscan.io - Checking Transactions



← → C etherscan.io/tx/0x68abd8bd5fca2edd4eb68696914663bec334d1b04d6c647d7f5024430f398147 ⌂

ETH Price: \$1,847.24 (-0.08%) Gas: 12 Gwei

Search by Address / Txn Hash / Block / Token / Domain Name

Overview State Comments More ▾

Transaction Hash: 0x68abd8bd5fca2edd4eb68696914663bec334d1b04d6c647d7f5024430f398147 ⌂

Status: Success

Block: 17911461 4 Block Confirmations

Timestamp: 40 secs ago (Aug-14-2023 07:01:59 AM +UTC)

Transaction Action: Transfer 0.041200695896691378 Ether To 0x51a144...1712De97

 BC.GAME  CHAMPIONS LEAGUE

100% CASHBACK
ON YOUR LOSE UP TO \$100



From: 0x95222290DD7278Aa3Ddd389Cc1E1d165CC4BAfe5 (beaverbuild) ⌂

To: 0x51a1449b3B6D635EddeC781cD47a99221712De97 (Fee Recipient: 0x51...e97) ⌂

Value: 0.041200695896691378 ETH (\$76.11)

Transaction Fee: 0.000257826073869 ETH (\$0.48)



Exploring etherscan.io - Checking Transactions

ETH Price: \$1,847.24 (-0.08%) Gas: 12 Gwei Search by Address / Txn Hash / Block / Token / Domain Name

② Value: ♦ 0.041200695896691378 ETH (\$76.11)

② Transaction Fee: 0.000257826073869 ETH (\$0.48)

② Gas Price: 12.277432089 Gwei (0.00000012277432089 ETH)

② Gas Limit & Usage by Txn: 32,000 | 21,000 (65.63%)

② Gas Fees: Base: 12.277432089 Gwei | Max: 12.277432089 Gwei | Max Priority: 0 ETH

② Burnt & Txn Savings Fees:  Burnt: 0.000257826073869 ETH (\$0.48)  Txn Savings: 0 ETH (\$0.00)

② Other Attributes: Txn Type: 2 (EIP-1559)Nonce: 333092Position In Block: 105

② Input Data: 0x

More Details: — Click to show less

② Private Note: To access the Private Note feature, you must be [Logged In](#)

 A transaction is a cryptographically signed instruction that changes the blockchain state. Block explorers track the details of all transactions in the network. Learn more about transactions in our [Knowledge Base](#).



Exploring etherscan.io - Checking Transactions (pending)

ETH Price: \$1,846.55 (-0.12%) Gas: 12 Gwei

A total of 165,175 pending txns found
(Showing the last 10000 records)

First < Page 2 of 200 > Last

Txn Hash	Method	Nonce	Last Seen	Gas Limit	Gas Price	From	To	Amount
0xa53290aa10504e22...	Transfer	17	6 secs ago	21000	0.3447 Gwei	0xC30D91...b1f4ecF5	0x777777...d553C0e1	777 wei
0x2b8cab5a9ab6a39b...	Transfer	0	6 secs ago	21055	0 0 Gwei	0x9e5Bd3...fB9d320c	0x92e929...8BB2b786	0.015 ETH
0x8d7c0cdaffa018832...	Redeem	1	6 secs ago	192020	1 Gwei	0x9cB3fC...42004a7F	Compound: cETH Token	0 ETH
0x28313a4faefb479a1...	Remove Liqui...	4	6 secs ago	285750	1 Gwei	0x7E8F78...bbF7646C	Uniswap V2: Router 2	0 ETH
0x5161cb7028ee045e...	Transfer	7	6 secs ago	71000	1 Gwei	0xF42847...e83a2355	QuadrigaCX 2	0.00011846
0x2af97953349a64bf2...	Deposit	49	6 secs ago	33170	15.3747 0.1 Gwei	0x3107B6...E54B6fCe	Blur: Bidding	0.33 ETH
0xa95ddd23abc6c58c...	Transfer	0	6 secs ago	34470	0.0234 Gwei	0x01A62A...e546694A	Amepay: AME Token	0 ETH
0xd652c68956728356...	Approve	35	6 secs ago	46052	15.3747 0.1 Gwei	0xa2608e...47208353	Wrapped Ether	0 ETH
0x82488f7cff36c57eb...	Execute	270	6 secs ago	162944	16.3474 0.1 Gwei	0x9fa331...189b2f14	Uniswap: Universal Ro...	0.24 ETH
0x4aea330aa4486d00...	Mint	1	6 secs ago	100887	1 1 Gwei	0xc7252C...05788672	0xE45119...B74D368f	0 ETH
0xdd2dbd2d8471eea8...	Approve	1338	6 secs ago	46577	26.0237 1.5 Gwei	0x1ac24F...eBb79318	0x628443...3A96f2B8	0 ETH





Exploring [etherscan.io](#) - Checking Smart Contracts

1. Helpful for **users who are regularly in contact with smart contracts** of decentralized applications.
 - They can confirm the transfer of funds to the right contract.
 - The **smart contract addresses** contain information, such as
 - token logic, how the token was transferred, and a “code” tab to access smart contracts’ source code.
 - To **access information related to smart contracts**.
 - Users need to **copy and paste the contract address** on the search icon
2. Allows users to read contracts and edit them.
 - Click on the “Read Contract” and “Write Contract” option on Etherscan.
3. **Minting NFTs** from smart contracts is another use case of Etherscan.
 - The verified contracts feature a checkmark.
 - **One will not be able to interact if the contract is not verified.**
 - If you have access to the application binary interface (ABI) of a contract, you can interact with an unverified contract as well.
4. Allows users to search for the contract.
 - You can see the code of contract, its balance, and transactions made to that contract.





Exploring etherscan.io - Checking Smart Contracts

ETH Price: \$1,847.34 (-0.07%) Gas: 12 Gwei

Search by Address / Txn Hash / Block / Token / Domain Name

Etherscan Home Blockchain Tokens NFTs Resources Developers More Sign In

Verified Contracts

Sponsored: TOKEN2049 Singapore - Attend Asia's Premier Crypto Event - [Register Now](#)

Filter by | Latest 500 Verified Contracts

Showing the last 500 verified contracts source code

Address	Contract Name	Compiler	Version	Uncles	Created	Verified	Audit	License	Similar Contract
0x784730...5a36d92E	GIFT	Solidity(Json)	0	Top Accounts	8/14/2023	-	-	Q Search	
0x93E490...164b63A6	PHOUSE	Solidity(Json)	0	Verified Contracts	8/14/2023	-	-	Q Search	
0x7b4001...a03DcD36	Proxy	Solidity(Json)	0.8.20	0.082 ETH 2	View Edit	8/14/2023	-	-	Q Search
0xfC99B4...568FBA25	Proxy	Solidity(Json)	0.8.20	0 ETH 0	View Edit	8/14/2023	-	-	Q Search
javascrip...;	TaskToken	Solidity	0.8.20	0 ETH 1	View Edit	8/14/2023	-	MIT	Q Search





Exploring [etherscan.io](#) - Checking Smart Contracts

ETH Price: \$1,847.37 (-0.07%) Gas: 13 Gwei

Search by Address / Txn Hash / Block / Token / Domain Name

Transactions Internal Transactions Token Transfers (ERC-20) Contract Events Analytics Comments

Code Read Contract Write Contract Read as Proxy Write as Proxy Search Source Code

Contract Source Code Verified (Exact Match)

Contract Name: GIFT Optimization Enabled: Yes with 300 runs

Compiler Version v0.8.17+commit.8df45f5 Other Settings: default evmVersion

Contract Source Code (Solidity Standard Json-Input format)

File 1 of 5: GIFT.sol

```
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity ^0.8.0;
4
5 /// @title: OUVR3 GIFT
6 /// @author: manifold.xyz
7
8 import "./manifold/ERC1155Creator.sol";
9
10 ///////////////////////////////////////////////////////////////////
11 //
12 //
13 //
14 //
15 //
16 //
17 javascript://
```



Exploring [etherscan.io](#) - Checking Gas Prices

- “Gas” is a unit used in the Ethereum platform to **describe the amount of computational power needed to execute the transaction.**
- One need to pay a certain amount to perform a certain transaction on Ethereum.
- This amount is variable as it depends on the network traffic and the number of blocks included in the transaction.

Etherscan offers a **dedicated page to track the gas fees of a specific transaction.**

- This page shows the average gas fee for the last seven days and the last few blocks.
- This can help you decide the best time for making a transaction.
- The “**gas tracker**” feature shows the difference in time and price at different gas prices.

Etherscan also roughly **predicts the gas fee based on the amount to be transferred and network traffic.**

- You will find the average gas fee on Etherscan.
- Below that you will get an estimated cost of the transaction.
- **Higher gas prices** reflect the higher speed of the transactions..



Exploring etherscan.io - Checking Gas Prices

ETH Price: \$1,848.26 (-0.02%) Gas: 15 Gwei

Etherscan

Home Blockchain Tokens NFTs Resources Developers More | Sign In

Tools & Services

Discover more of Etherscan's tools and services in one place.

Sponsored

 Blockscan

Tools

Unit Converter CSV Export Account Balance Checker

Explore

Gas Tracker DEX Tracker Node Tracker Label Cloud Domain Name Lookup

Services

Token Approvals Verified Signature Input Data Messages (IDM) Advanced Filter Blockscan Chat

Low 15 gwei Average 15 gwei High 19 gwei

Base: 14 | Priority: 1
\$0.50 | ~ 3 mins: 0 secs

Base: 14 | Priority: 1
\$0.50 | ~ 3 mins: 0 secs

Base: 14 | Priority: 5
\$0.62 | ~ 30 secs

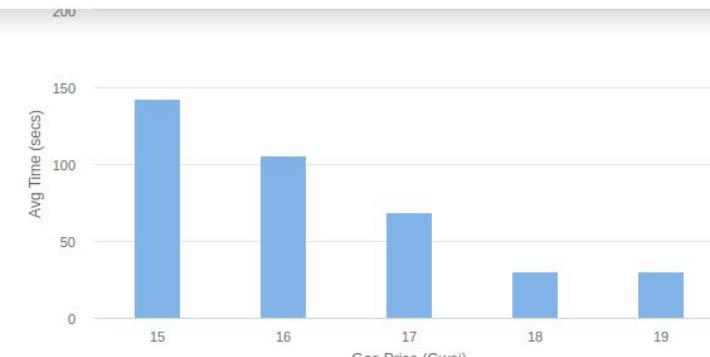
Estimated Cost of Transaction Actions:

Action	Low	Average	High
OpenSea: Sale	\$1.85	\$1.85	\$2.25

View APIs

Avg Time (secs)

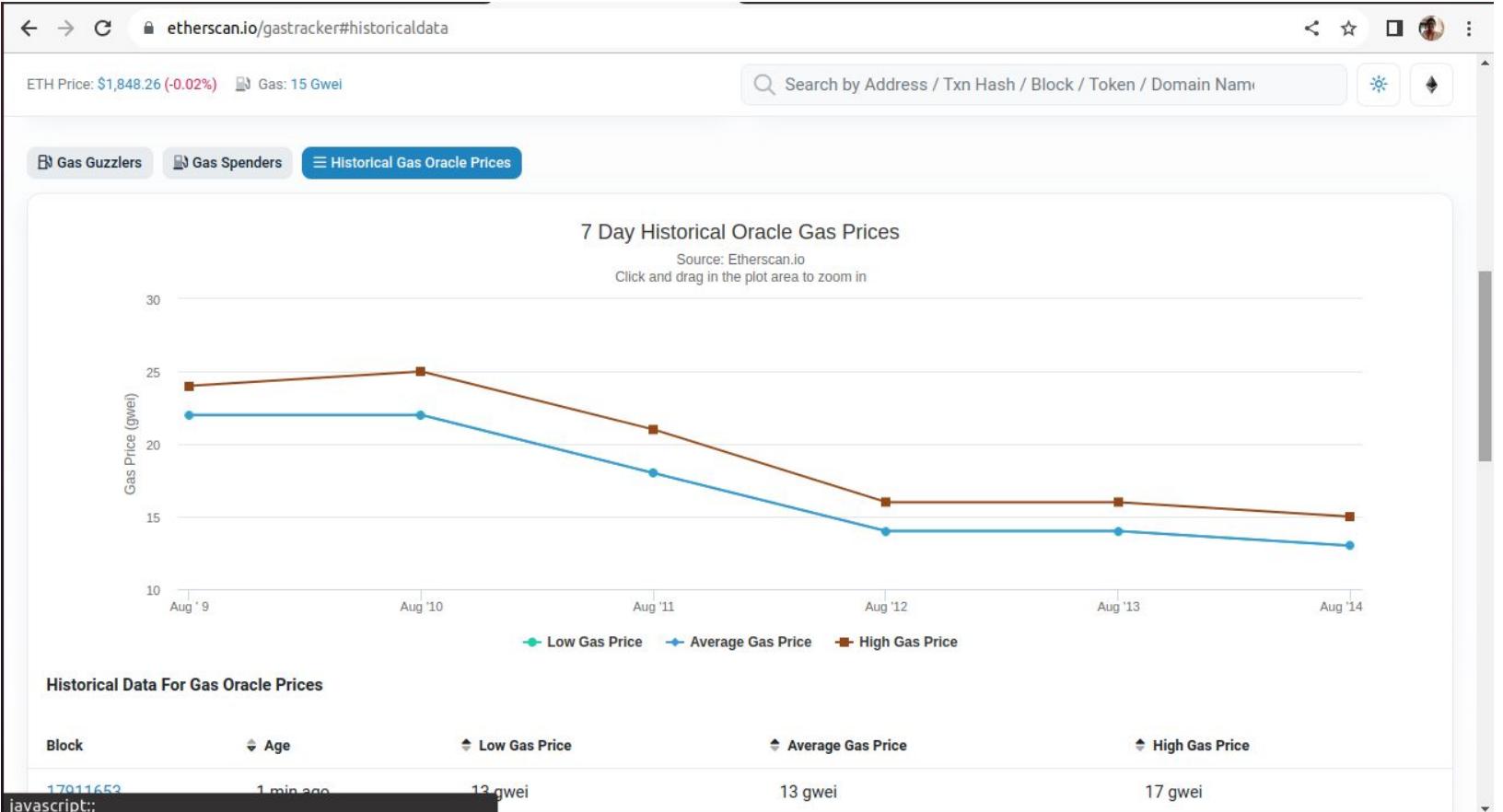
Gas Price (Gwei)



Gas Price (Gwei)	Avg Time (secs)
15	~145
16	~105
17	~70
18	~35
19	~35



Exploring etherscan.io - Checking Gas Prices



- Feature launched in 2021 by which you can directly check your NFTs.
- This **allows users to buy and sell their digital collectibles increasing transparency on transactions and ownership.**

Steps for checking NFTs on Etherscan:

1. Click the Etherscan [homepage](#), and find the search bar at the top of the page.
2. Insert your wallet address in the search icon and hit “Search”.
3. In the mid-screen, click ‘Erc721 Token Txns’ to see a list of all NFTs associated with the given wallet address.
4. In order to access additional details, go to the detailed column and click “View NFTs”.





Exploring etherscan.io - top NFT's

ETH Price: \$1,847.21 (-0.04%) Gas: 12 Gwei

Search by Address / Txn Hash / Block / Token / Domain Name

Etherscan Home Blockchain ▾ Tokens ▾ NFTs ▾ Resources ▾ Developers ▾ More ▾ Sign In

Top NFTs

1h 6h 12h 1d 7d 30d

#	Collection	Type	Volume	Change (%)	Min Price (24H) ⓘ	Max Price (24H) ⓘ	Sales	Transfers	Owners	Total
1	Saints	ERC-721	290.4 ETH	0%	0.81 ETH	26.4 ETH	11	12,473	747	12,0
2	Nouns	ERC-721	60.79 ETH	-7.02%	29.29 ETH	31.5 ETH	2	3,721	397	812
3	YOU THE REAL MVP	ERC-721	40.69 ETH	0%	35.69 ETH	40.69 ETH	1	1,375	283	420
4	Taciturn-robot	ERC-721	24.021 ETH	-0.02%	3.99 ETH	4.02 ETH	6	8,491	485	7,62
5	OpenSea Shared Storefront	ERC-1155	20,4329 ETH	-99.94%	0.0000 ETH	2.75 ETH	69	3,811,106 ⓘ	718,347	0 ⓘ
6	Parallel	ERC-1155	20,2100 ETH	-99.95%	0.0000 ETH	27.5 ETH	00	1,001,512 ⓘ	60,446	0 ⓘ





Exploring [etherscan.io](#) - Working

- Etherscan serves as a search engine for ethereum.
- Analogy : consider it as a Google but for the cryptocurrency world.
- In fact, these block explorers are called “**blockchain browsers**” or “**blockchain search engines**”.

Note :

- One can access wallet, transactions, and smart contract
- Can also identify any suspicious activities.
- However, **Etherscan is not a wallet**,
 - it is a source of information that allows users to track their transactions on the Ethereum network, index this information, and add it to its website.

Etherscan

- **combines different technologies, such as blockchain explorers, decentralized applications (Dapps), and Web3 libraries,**
- **displays the information in an easy-to-access format.**



Exploring etherscan.io - Working

The working of Etherscan is based on three parts:

1. Data Retrieval:

- Etherscan uses an Application Programming Interface (API) to access data.
- APIs allow one software to interact with another.
- API connects block explorers to the Ethereum blockchain.
- Etherscan **uses JSON-RPC 2.0 specification,**
 - which means it utilizes JSON (JavaScript Object Notation) to **communicate with Ethereum.**
 - These APIs provide users access to the transactions and blocks on Ethereum.

2. Data Storage:

- Etherscan stores the data in the form of relational tables in the “SQL Database”.
- The relational format means the data in one table is related to the data in other tables.
- This relationship helps explorers to display information easily.

3. Data Production:

- Data displayed in the Database is converted to a human-understandable language.
- Etherscan takes the help of different programming languages, such as HTML, CSS, and JavaScript.
 - HTML structures the data on block explorers.
 - CSS gives styles to the content.
 - JavaScript helps data to interact with the user.



Is Etherscan free?

- Yes! Etherscan is a free tool that allows people to check their wallet and transaction status without any service charges.
- It does not store or hold your cryptocurrency as it is **not a cryptocurrency wallet service provider.**
- It just offers an interface to view and track your past transactions.

Do you need an account to use Etherscan?

- No! You do not need to create an account to access your ether transactions using Etherscan.
- However, you can open an account to access certain Etherscan features.
- For instance, If you want to build data streams,
 - utilize toolchains, and
 - set notifications to alert income or expenditure.
- These **add-ons are only available if you open an account on Etherscan.**
- This search engine is **managed autonomously by dedicated individuals to make decentralized information and technical apps feasible.**





Exploring etherscan.io - Live Demo (Google CoLab)



Extracting Live data with API Keys

- Go to <https://etherscan.io/>
- Sign Up - create an account
- Verify your account from email
- Login to your account
- Go to API Keys
- Create App by giving name
- Copy the API Key generated into the code

```
import requests

def get_latest_block(api_key):
    url = "https://api.etherscan.io/api"
    params = {
        "module": "proxy",
        "action": "eth_getBlockByNumber",
        "tag": "latest",
        "boolean": "true",
        "apikey": api_key,
    }

    try:
        response = requests.get(url, params=params)
        if response.status_code == 200:
            data = response.json()
            return data["result"]
        else:
            print("Request failed with status code:", response.status_code)
    except requests.RequestException as e:
        print("Request failed:", str(e))

    return None

# Replace "YOUR_API_KEY" with your actual API key
api_key = "EXK36NERHPD5487RG5F2RQI669A79VQUHR" # Paste your API Key
```



Extracting Live data with API Keys

- Go to <https://etherscan.io/>
- Sign Up - create an account
- Verify your account from email
- Login to your account
- Go to API Keys
- Create App by giving name
- Copy the API Key generated into the code

```
print ("Current date and time : ", now.strftime("%d-%B-%Y"))
latest_block = get_latest_block(api_key)
#print(latest_block)
if latest_block is not None:
    print("Latest block information:")
    print("Block Number:", int(latest_block["number"], 16))
    print("Timestamp:", int(latest_block["timestamp"], 16))
    print("Miner Address:", latest_block["miner"])
    print("Difficulty:", int(latest_block["difficulty"], 16))
    print("Total Difficulty:", int(latest_block["totalDifficulty"], 16))
    print("Gas Limit:", int(latest_block["gasLimit"], 16))
    print("Gas Used:", int(latest_block["gasUsed"], 16))
    print("Transaction Count:", len(latest_block["transactions"]))
    print("Transactions:", latest_block["transactions"])
    # Add more fields as per your requirements
    # print("Transactions:")
    # for tx in latest_block["transactions"]:
    #     print(tx)
```

```
Current date and time : 13-August-2023
Latest block information:
Block Number: 17902335
Timestamp: 1691886227
Miner Address: 0x1f9090aae28b8a3dceadf281b0f12828e676c326
Difficulty: 0
Total Difficulty: 58750003716598352816469
Gas Limit: 30000000
Gas Used: 12180476
Transaction Count: 123
```



Exploring infura.io - Live Demo (Google CoLab)

```
import requests

# Infura HTTP endpoint
infura_url = 'https://mainnet.infura.io/v3/7329beded7a74ad085a6144b63645314'

# Make a request to retrieve the latest block number
response = requests.post(
    infura_url,
    json={
        "jsonrpc": "2.0",
        "method": "eth_blockNumber",
        "params": [],
        "id": 1
    }
)
if response.status_code == 200:
    result = response.json()
    latest_block_number = int(result["result"], 16) # Convert hexadecimal to decimal

    print("Latest Block Number:", latest_block_number)
    # Print the desired information
    print("Miner Address:", latest_block["miner"]) # Error
    print("Difficulty:", int(latest_block["difficulty"], 16))
    print("Total Difficulty:", int(latest_block["totalDifficulty"], 16))
    print("Gas Limit:", int(latest_block["gasLimit"], 16))
    print("Gas Used:", int(latest_block["gasUsed"], 16))
    print("Transaction Count:", len(latest_block["transactions"]))
```

For extracting Live data,
[Infura.io - Getting Started](#)

Current date and time : 14-August-2023
Latest Block Number: 17911780
Miner Address: 0x690b9a9e9aa1c9db991c7721a92d351db4fac990
Difficulty: 0
Total Difficulty: 58750003716598352816469
Gas Limit: 29970705
Gas Used: 13555004
Transaction Count: 138

