ETHEREUM 2.0 MASTERY PROGRAM

Instructor: Raja Rizwan Saleem

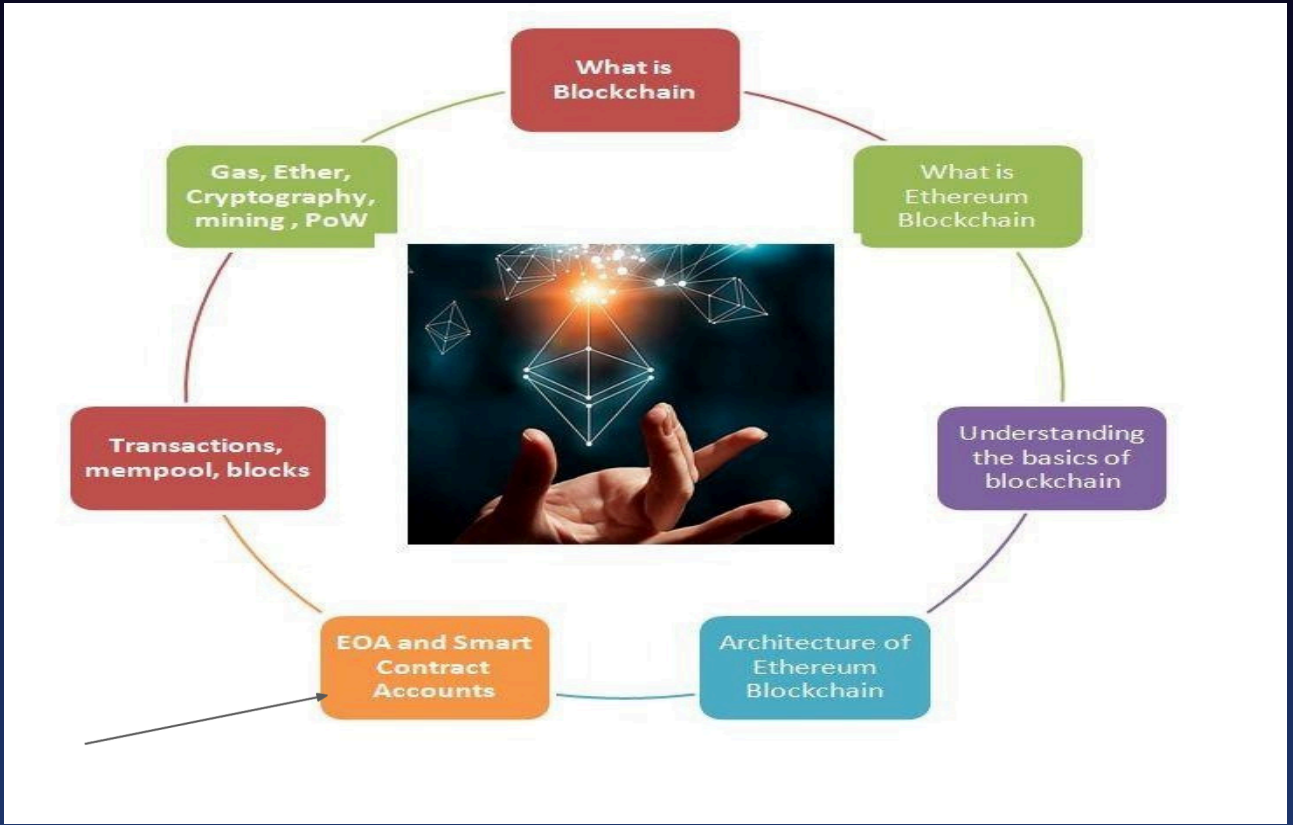
MODULE-2

BLOCKCHAIN AND SMART CONTRACT BASICS

Raja Rizwan Saleem
Lead Blockchain Trainer

Class-04

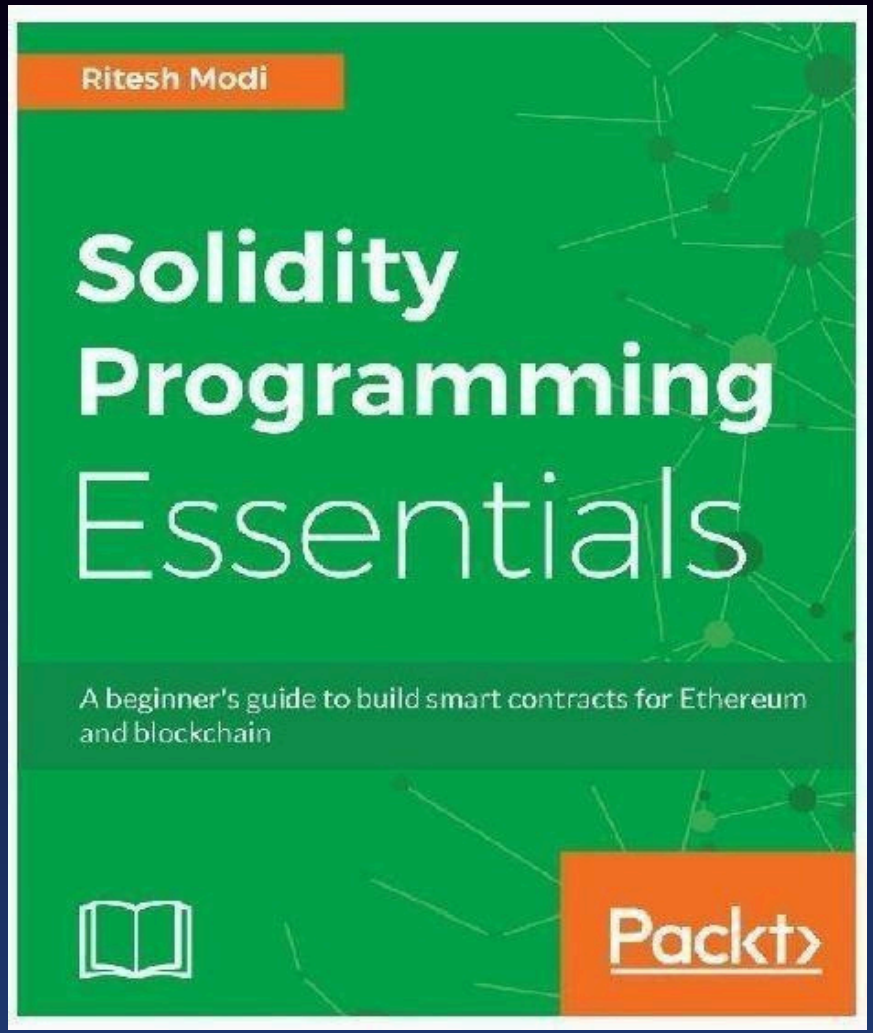
Eth Developers' Basic knowledge kit



Externally Owned Account


BLOCKCHAIN
BLOCKCHAIN
SOLIDITY
IN BLOCKCHAIN

Book we will follow



SOLIDITY

- Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs which govern the behaviour of accounts within the Ethereum state.
- Solidity is a curly-bracket language designed to target the Ethereum Virtual Machine (EVM). It is influenced by C++, Python and JavaScript.

SOLIDITY

- Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features.
- With Solidity you can create contracts for uses such as voting, crowdfunding, blind auctions, and multi-signature wallets.
- When deploying contracts, you should use the latest released version of Solidity.
- Apart from exceptional cases, only the latest version receives security fixes.
- Furthermore, breaking changes as well as new features are introduced regularly.

Smart Contract Structure

Solidity contracts are similar to object-oriented language classes. Contract can include **state variables**, **functions**, **function modifiers**, **events**, **struct types**, and types of **enum declarations**. However, contracts may be inherited from other contracts as well.

- **State variables** are quantities that are stored indefinitely in contract storage.
- **Functions** are the units of code, executable inside a contract.
- **Function modifiers** may be used to modify function semantics in a declarative manner.
- **Events** are channels of comfort for the EVM logging services.
- **Structs** are categories that are custom specified and can group multiple variables.
- **Enums** can be combined with a finite set of 'constant values' to construct custom types.

Variables

Solidity supports 3 variable types.

- **State variables** – Variables whose values are stored permanently in a contract storage.
- **Local variables** – Variables whose values are present before executing function.
- **Global variables** – The generic namespace used to provide knowledge about the blockchain includes special variables.

Solidity is a statically typed language, which means defining the type of state or local variable during declaration. Every variable that was declared also has a default value depending on its type. There is no 'undefined' or 'null' definition.

You can change the accessibility of the variable and control who can access its values.

Reference Types

Reference types include:

- **Arrays** - Arrays are sets of the same type of variables in which each particular variable has a specific location called the index. You can access the attribute by the use of the index position. The array scale can be set or adjustable.
- **Structs** - Solidity helps users to build their own Structure-type. Struct is the category of various forms as a member
  of its own type. Struct is a variable of reference type, which can include both-value types & reference types.
- **Mapping** - Mapping types are the most widely used type of reference; they are used to store data in a key-value pair, where the key maybe some form of built-in value or byte and string. Like in any other language,you might think of it as a hashtable or dictionary in which a user might store data in a key-value format and retrieve data by name.

The only difference between value type and reference type is the location of the data. Arrays and Structs have additional locations of data, which determines where data (variable value) should be stored.

Ethereum Networks

1. Ethereum is an Open Source Platform for Creating and Deploying Distributed Applications.
2. Ethereum is backed up by a large number of computers (nodes)
3. All Nodes are interconnected and storing data in a Distributed ledger (each node).
4. Developer can choose an appropriate network based on their requirements and use cases.

Ethereum Networks

5. Different networks help in deploying solutions and smart contracts on networks that do not actually cost any Ether or money.
6. There are networks that are free of cost while there are ones that require its users to pay in terms of Ether or other currencies for its usage.

Main network

1. The main Ethereum network is a global public network that anybody can use.
2. Everybody's free to create an account and deploy their solutions and smart contracts.
3. Main network incurs costs in terms of gas.
4. The main network is known as Homestead.
5. Main network was earlier known as Frontier.

Main network

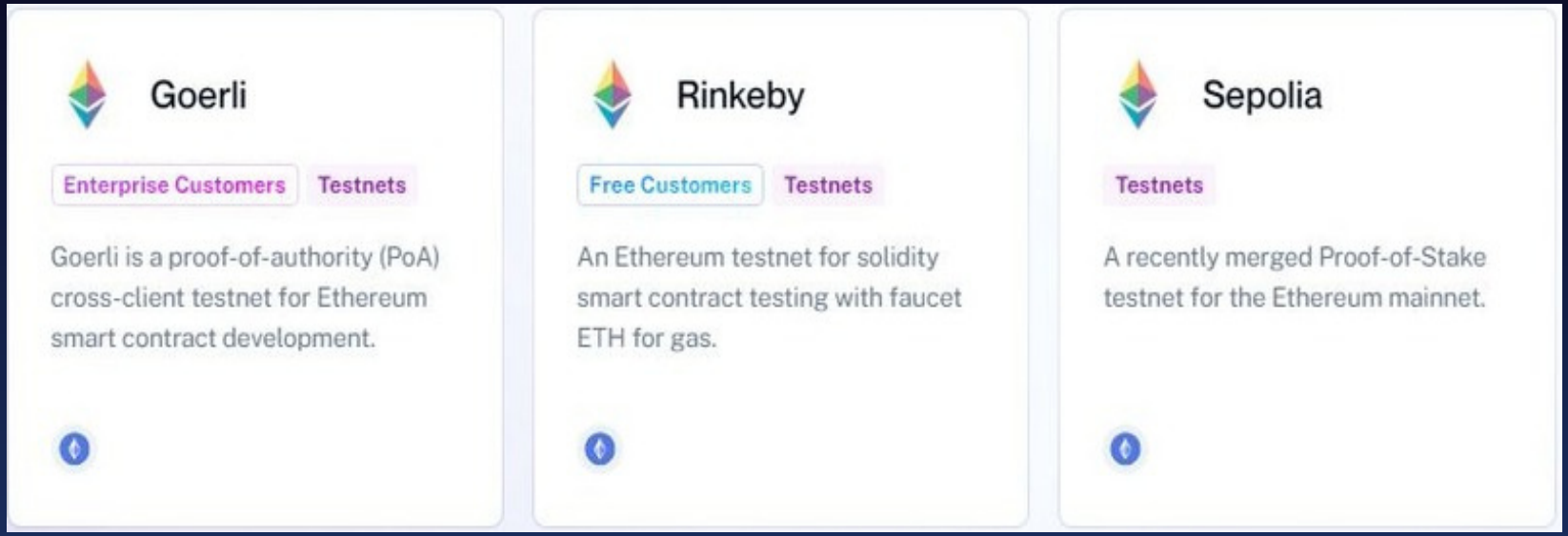
6. This is a public chain accessible over the internet.
7. Anybody can connect to it and access both data and transactions stored in it.

Test network

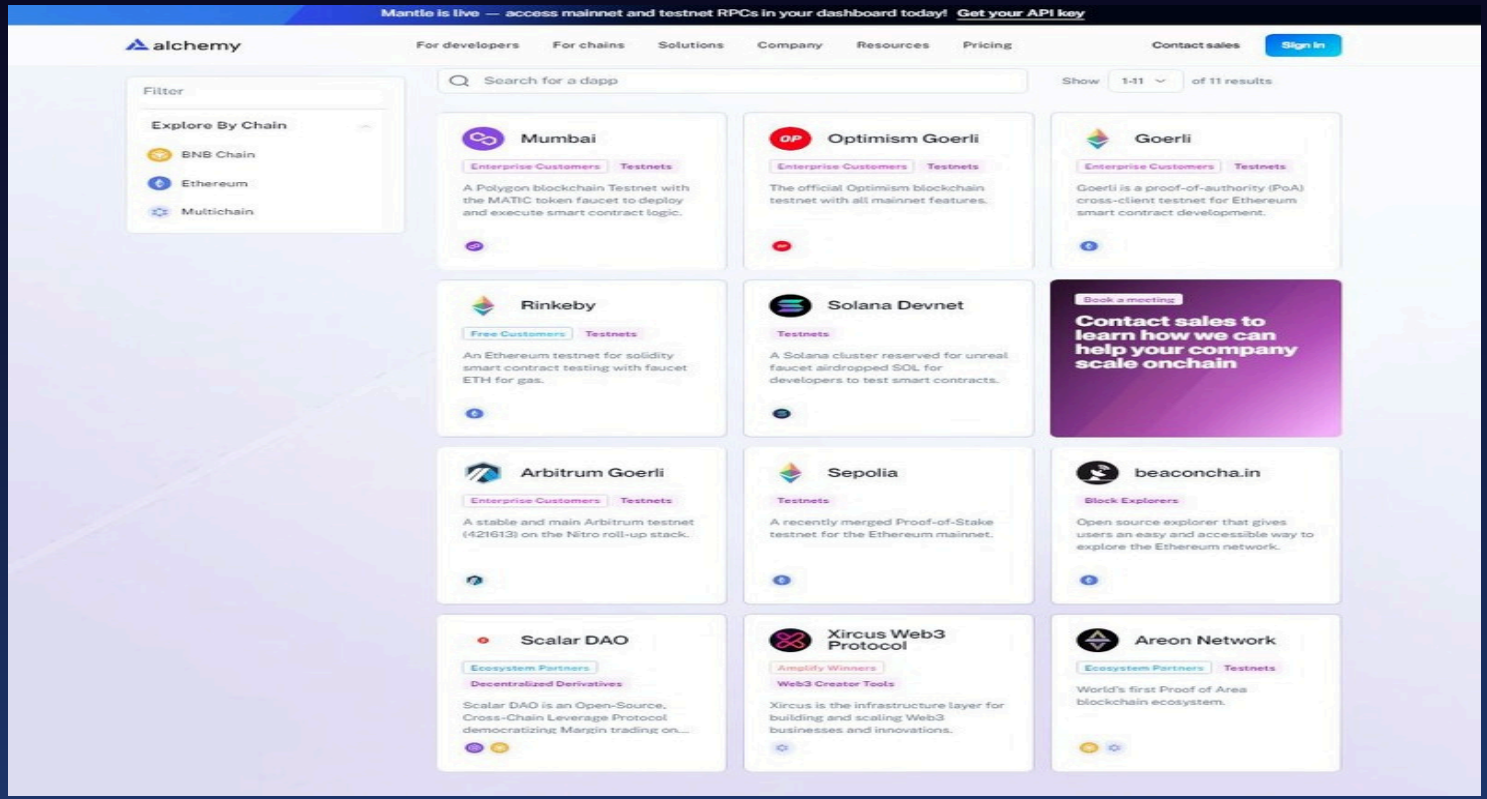
1. A test network exists to help facilitate and increase adoption of the Ethereum blockchain.
2. Exact replica of the main network.
3. Does not cost anything for deployment and usage of contracts.
4. Test Ethers can be generated using faucets and used on these networks.
5. Multiple test networks available, such as Sepolia, Ropsten, Kovan, and Rinkeby etc

Test network

Test network

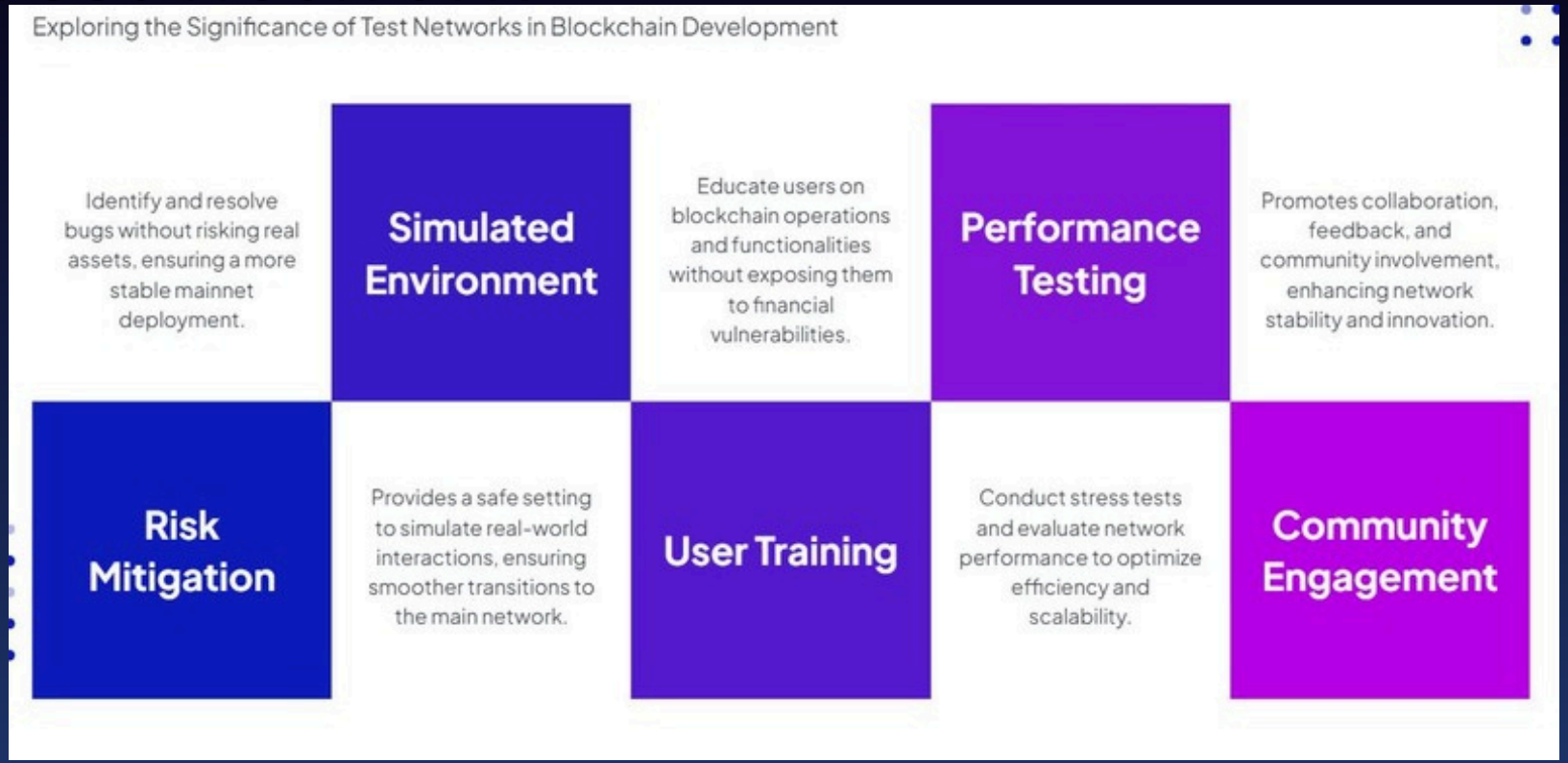


Test network



Test network

Test network



Private network

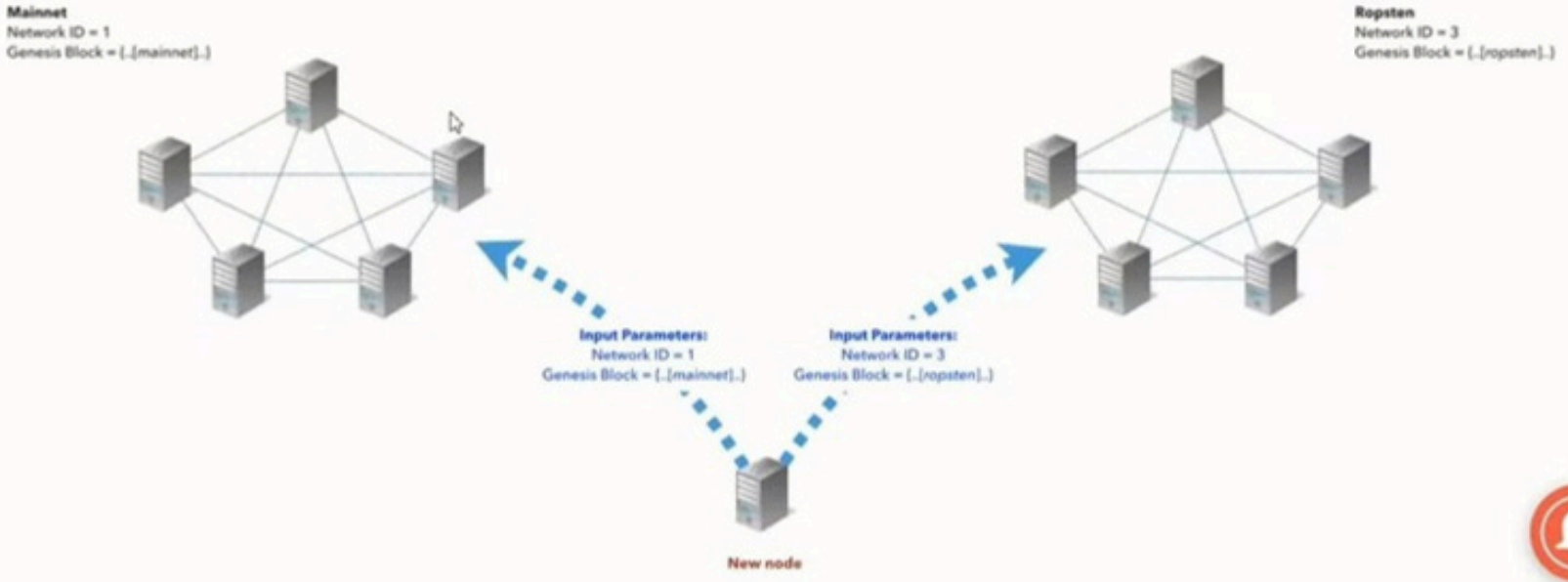
1. Created and hosted on a private infrastructure. Controlled by a single organization and they have full Control over it.
2. There are solutions, contracts, and use cases that an organization might not want to put on a public network even for test purposes.
3. They want to use private chains for development, testing, and production environments.

Consortium network

1. It is also a private network, however, with a difference.
2. The consortium network comprises nodes, each managed by a different organization.
3. In effect, no organization has a control over the data and chain.
4. However, it is shared within the organization and everyone from these organizations can view and modify the current state.
5. These might be accessible through the internet or completely private networks using VPN.

Mainnet vs Testnet

A mainnet and a testnet are two separate networks that operate independently from each other. Here's an illustration from the context of Ethereum:



Mainnet vs Testnet

Mainnet	Testnet
Used for actual transactions with value.	Used for testing smart contracts and decentralized applications.
Mainnet's network ID is 1.	Testnets have network IDs of 3, 4, and 42.
Example - Ethereum	Example – Rinkeby Test Network

Case Study: Ethereum Test Networks

Exploring Different Ethereum Test Networks and Their Significance in Blockchain Development

Ropsten: Proof of Work

01

Ropsten operates similarly to the Ethereum mainnet, utilizing proof of work for consensus.

Goerli: Large State Capabilities

03

Goerli is favored for complex interactions due to its capability to handle large state requirements effectively.

Different Testnets, Unique Purposes

05

Each Ethereum test network serves distinct purposes, catering to various testing needs and scenarios.

Rinkeby: Proof of Authority

02

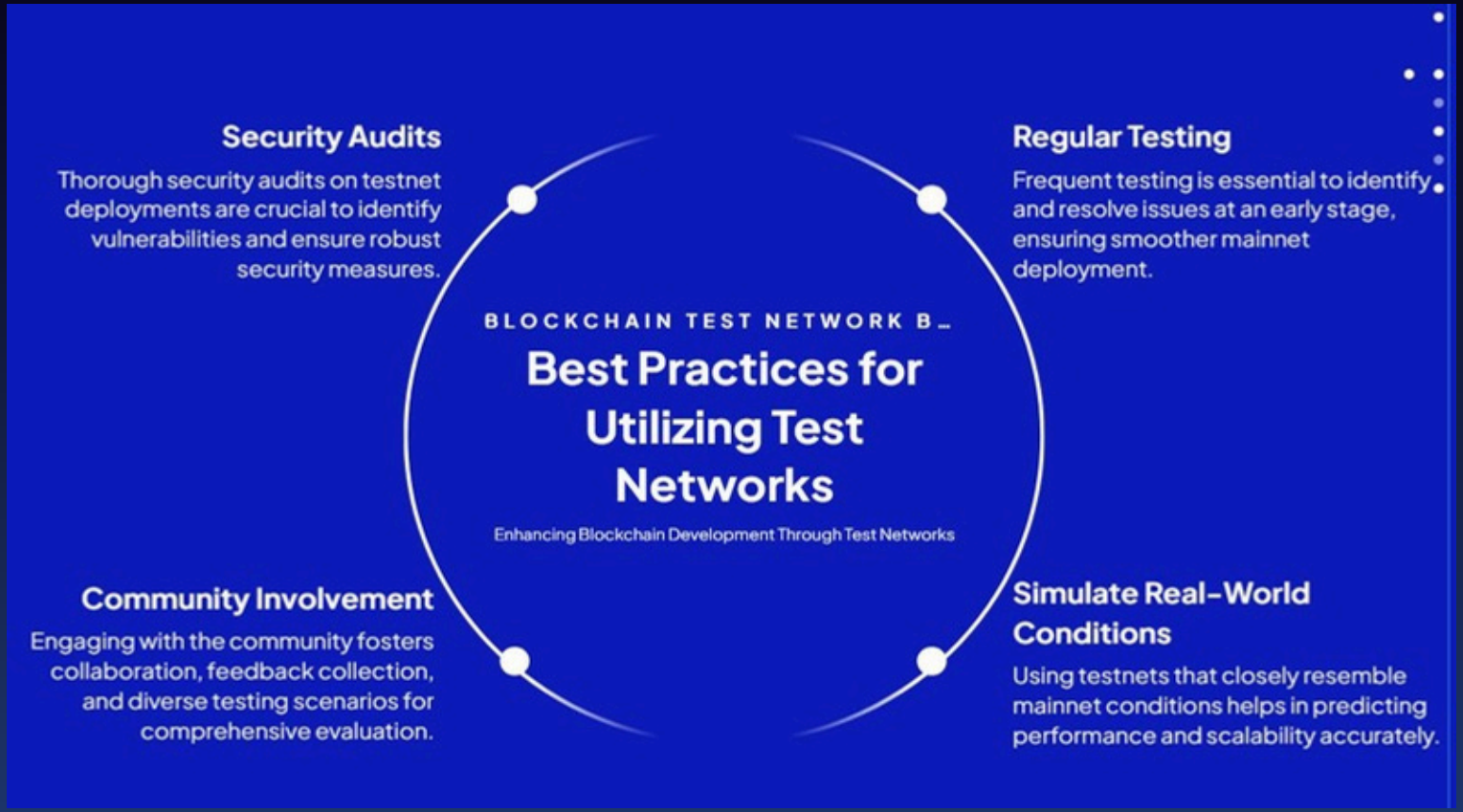
Rinkeby employs proof of authority consensus mechanism with trusted nodes for faster block confirmations.

Sepolia: Robust Testing Environment

04

Sepolia is designed to replicate challenging network conditions for rigorous testing of Ethereum applications.

Test networks



Tools Required

- MetaMask
- Remix IDE

MetaMask

Brings Ethereum to your browser

MetaMask

1. MetaMask is a lightweight Chrome browser-based extension that helps in interacting with Ethereum networks.
2. It is also a wallet that helps in sending and receiving Ether.
3. It is able to connect to a variety of Ethereum nodes and test blockchains.
4. Since MetaMask runs in a browser, it does not download the entire chain data locally; instead, it stores it centrally and helps users connect to their store using the browser.

Install MetaMask

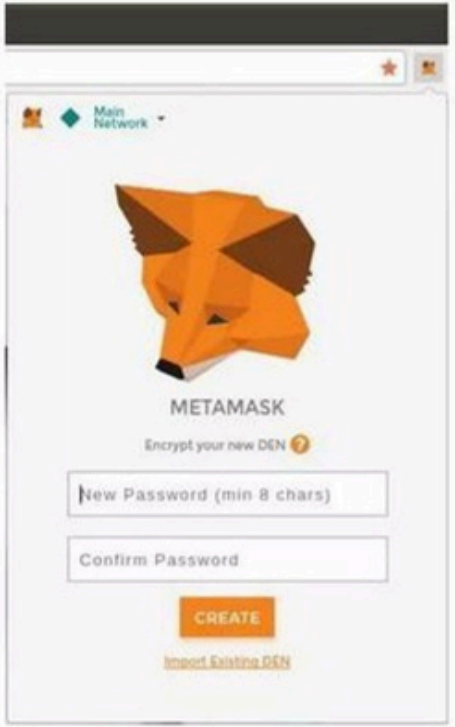
1. Install MetaMask extension in browser:
   a. https://metamask.io/

Metamask Extension
https://metamask.io/download/

MetaMask

Getting Started with MetaMask

1. Once MetaMask is installed you should see a new icon in your browser's toolbar.
2. Click on it to get started.
3. You will be asked to accept the terms and conditions and then to create your new Ethereum wallet by entering a password



MetaMask

Switching Networks

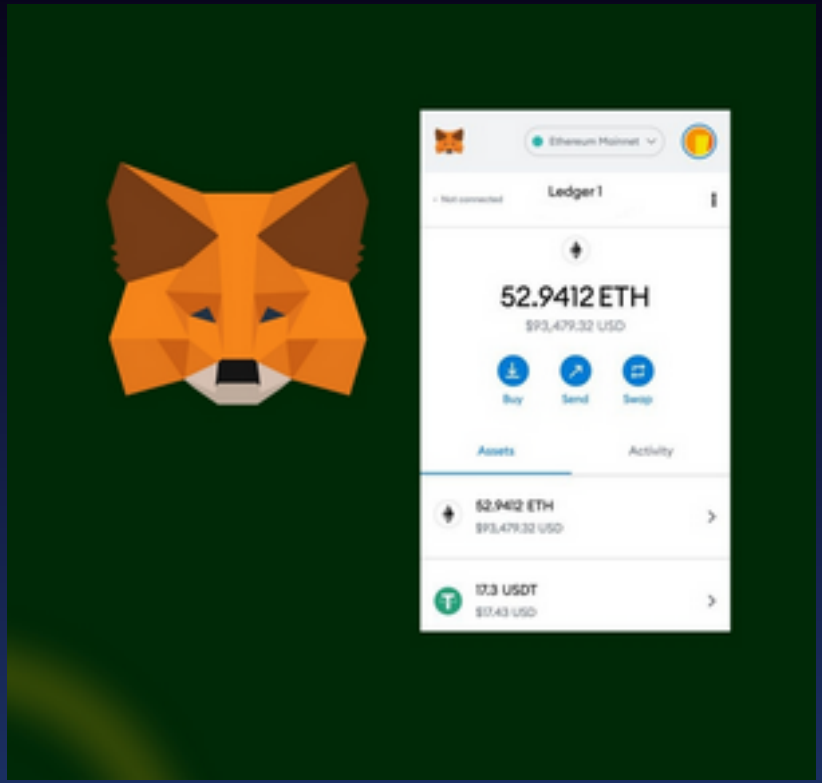
1. **Main Ethereum Network**
   a. By default, MetaMask will try to connect to the main public network.
   b. Real ETH, real value, and real consequences.
2. **Ropsten Test Network**
   a. Ethereum public test blockchain and network.
   b. ETH on this network has no value.
3. **Kovan Test Network**
   a. Ethereum public test blockchain and network using consensus protocol - Proof of authority ETH on this network has no value. The Kovan test network is supported by Parity only.

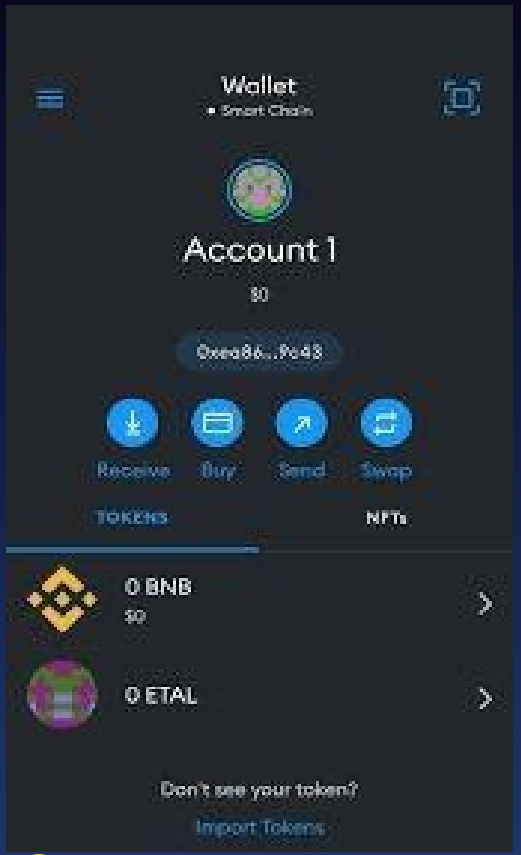
MetaMask

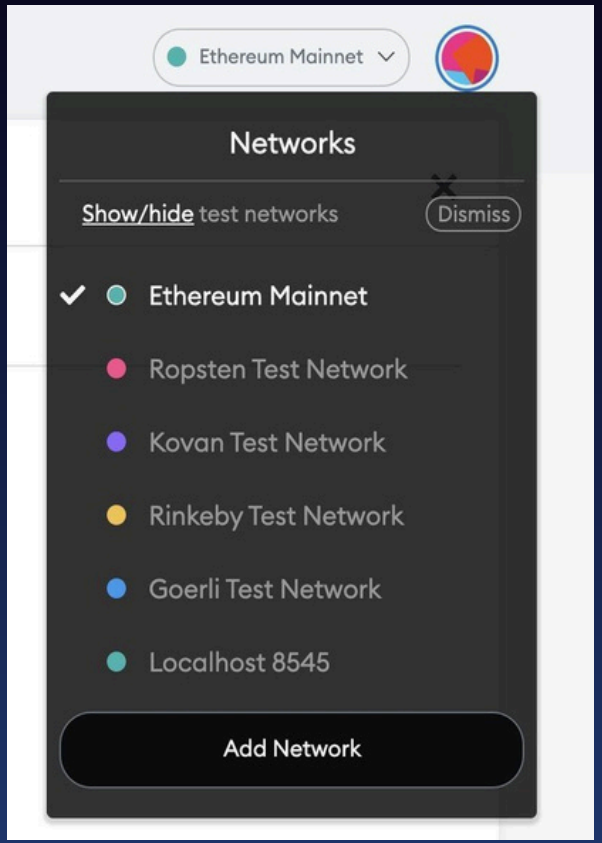
Switching Networks

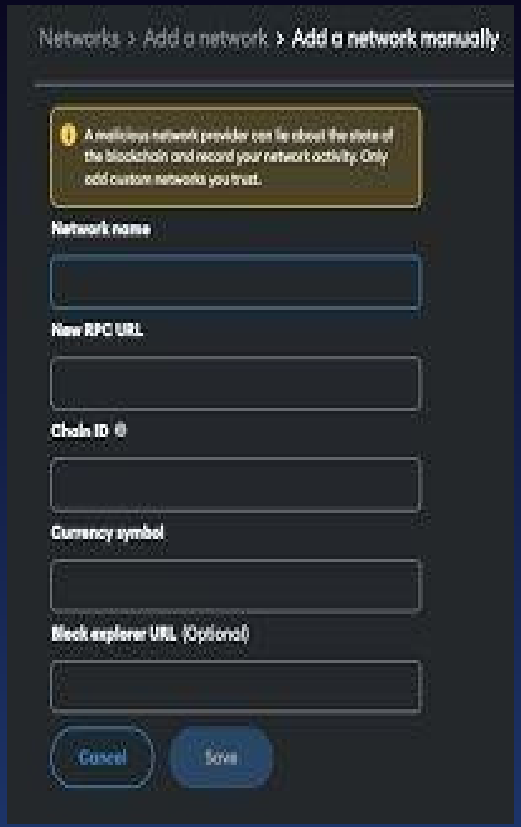
4. **Rinkeby Test Network**
   a. Ethereum public test blockchain and network, using consensus protocol - proof of authority
   b. ETH on this network has no value.
5. **Localhost 8545**
   a. Connects to a node running on the same computer as the browser.
   b. The node can be part of any public blockchain (main or testnet), or a private testnet.
6. **Custom RPC**
   a. Allows you to connect MetaMask to any node with a Geth-compatible Remote Procedure Call (RPC) interface. The node can be part of any public or private blockchain

Demo MetaMask




Wallet
Smart Chain
Account 1
$0
0xea86...9c48
Receive
Buy
Send
Swap
TOKENS
NFTs
0 BNB
$0
0 ETAL
Don't see your token?
Import Tokens


Ethereum Mainnet
Networks
Show/hide test networks
Dismiss
Ethereum Mainnet
Ropsten Test Network
Kovan Test Network
Rinkeby Test Network
Goerli Test Network
Localhost 8545
Add Network


Networks > Add a network > Add a network manually
A malicious network provider can lie about the state of the blockchain and record your network activity. Only add custom networks you trust.
Network name
New RPC URL
Chain ID
Currency symbol
Block explorer URL (Optional)
Cancel
Save

Remix

Browser-based compiler and IDE

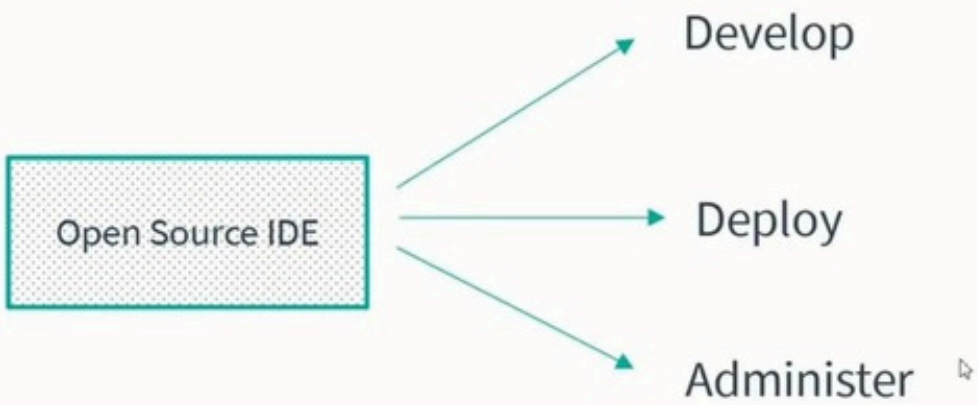
Remix IDE

- The easiest and fastest way to develop smart contracts is to use a browser-based tool known as Remix.
- Remix is available on http://remix.ethereum.org.
- Remix is a new name and was earlier known as browser-solidity.
- Remix provides a rich integrated development environment in a browser for authoring, developing, deploying, and troubleshooting contracts written using the Solidity language.
- All contract management related activities such as authoring, deploying, and troubleshooting can be performed from the same environment without moving to other windows or tabs

Remix IDE

- Remix is a browser-based compiler and IDE that enables users to build Ethereum contracts with Solidity language and to debug transactions. Remix IDE is an IDE for Solidity dApp developers, powered byRemix.
- Online version is available at https://remix.ethereum.org.
- You can also install remix on you machine
    a. npm install remix-ide -g

Remix IDE

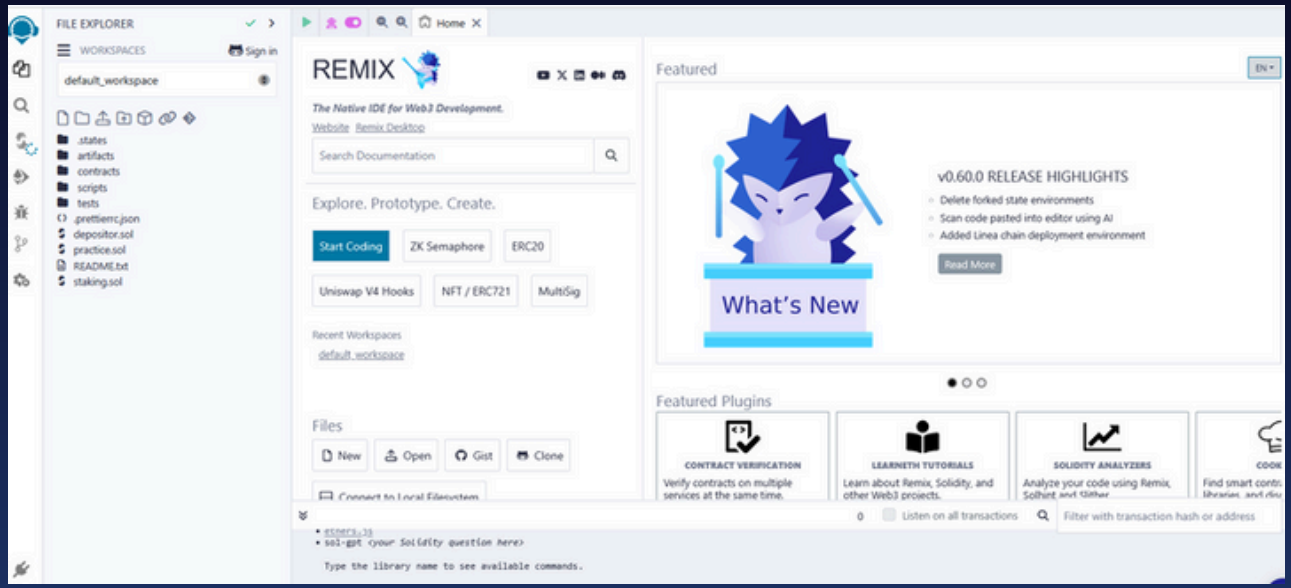


Remix IDE

Some important to know about Remix IDE-

- **Language Support -** Solidity and Vyper
- **Written in -** JavaScript
- **Modules -** Testing , Debugging ,Deploy

Demo Remix


FILE EXPLORER
WORKSPACES
Sign in
default_workspace
.states
artifacts
contracts
scripts
tests
.prettierrc.json
depositor.sol
practice.sol
README.txt
staking.sol
Home
REMIX
The Native IDE for Web3 Development.
Website Remix Desktop
Search Documentation
Explore. Prototype. Create.
Start Coding
ZK Semaphore
ERC20
Uniswap V4 Hooks
NFT / ERC721
MultiSig
Recent Workspaces
default_workspace
Files
New
Open
Gist
Clone
Featured
What's New
v0.60.0 RELEASE HIGHLIGHTS
Delete forked state environments
Scan code pasted into editor using AI
Added Linea chain deployment environment
Read More
Featured Plugins
CONTRACT VERIFICATION
LEARNETH TUTORIALS
SOLIDITY ANALYZERS
Listen on all transactions
Filter with transaction hash or address
sol-gpt <your Solidity question here>
Type the library name to see available commands.

THANK-YOU