CS61065: Theory and Applications of Blockchain

Ethereum

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Sandip Chakraborty
sandipc@cse.iitkgp.ac.in

TA – Utkalika Satapathy
utkalika.satapathy01@kgpian.iitkgp.ac.in

CONTENT

- Ethereum Introduction
- Ethereum Networks
- Ethereum Client
- Go-Ethereum Client (geth)
- Infura
- TestRPC
- Query using RPC over HTTP
- Solidity

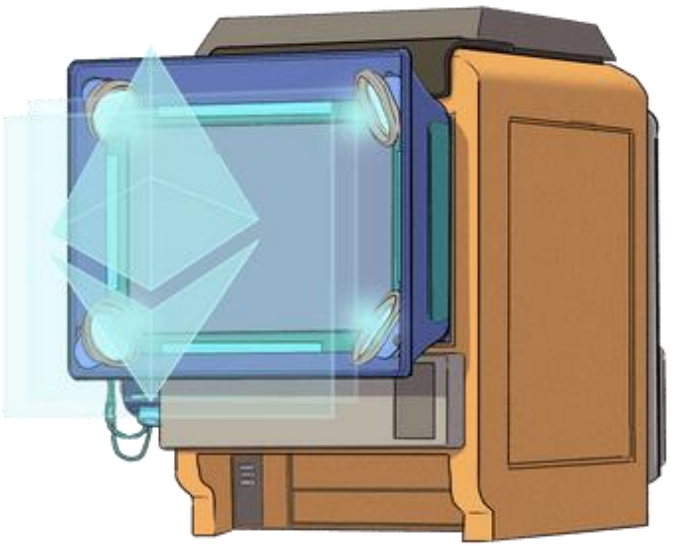
Ethereum

- Ethereum is the community-run technology powering the **Cryptocurrency ether (ETH)** and thousands of decentralized applications.
- Ethereum is a network of computers all over the world that follow a set of rules called the **Ethereum protocol**.
- It also powers applications that **everyone can use** and **no one can take down**.
- You can create an Ethereum account from anywhere, at any time, and explore a world of apps or build your own.
- The core innovation is that you can do all this **without trusting a central authority** that could change the rules or restrict your access.

source: https://ethereum.org/en/developers/docs/intro-to-ethereum/#what-is-ethereum

Ethereum

- Permissionless Blockchain capable of executing **Smart Contracts.**
- **Smart contracts are computer programs** living on the Ethereum blockchain.
- They execute when triggered by a transaction from a user.



source: https://ethereum.org/en/developers/docs/intro-to-ethereum/#what-is-ethereum

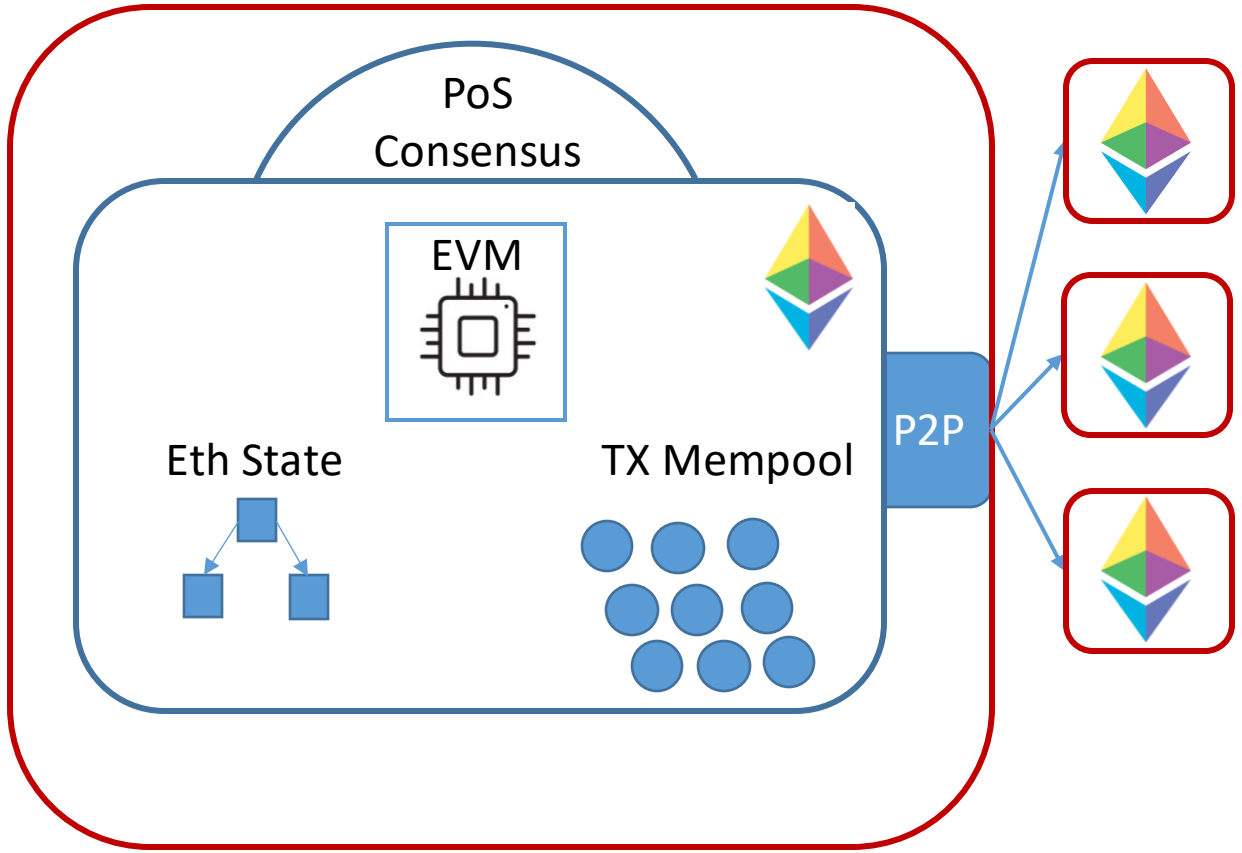
Ethereum

- Ethereum is a decentralized, open-source blockchain network with Turing-complete smart contract functionality.
- Ether (ETH) is the native cryptocurrency.
- Ethereum is used to build apps and organizations, hold assets, transact, and communicate without being controlled by a central authority.
- The Ethereum network has been upgraded from the original Proof of Work consensus mechanism to Proof of Stake during **The Merge.**
- **The Merge was executed successfully on September 15, 2022.**
- The Merge refers to the Ethereum upgrade that merged Ethereum Mainnet with the Beacon Chain Proof of Stake system. This marked the end of Proof of Work (PoW) for Ethereum, and the full transition to **Proof of Stake (PoS).**

source: https://docs.infura.io/learn/the-merge

Ethereum Network

- **Distributed network of computers**, known as **nodes** that can verify blocks and transaction data.
- An application, known as a **client**, running on your computer is a **node**.



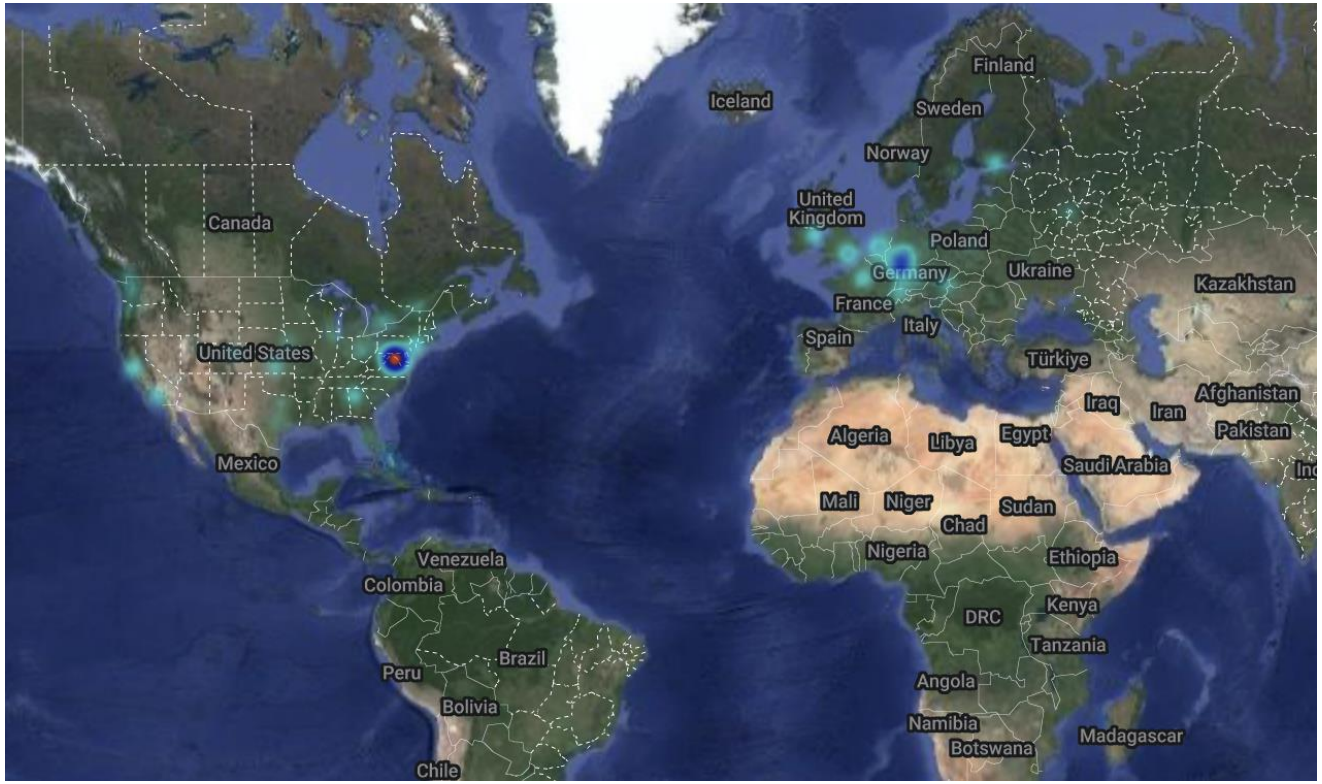
Ethereum Mainnet

MysticRyuujin-Besu-Lodestar	besu/v23.1.3/linux-x86_64/openjdk-java-17	52 ms	25	3870	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	163	0	6 s ago
MysticRyuujin-Besu-Teku	besu/v23.4.4/linux-x86_64/openjdk-java-17	58 ms	25	4096	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	163	0	0 s ago
EF-Bootnode-Helsinki	Geth/v1.12.2-stable-bed84606/linux-amd64/go1.20.7	93 ms	756	5142	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	0	0	7 s ago
EthPandaOps-geth-teku-berlin	Geth/v1.12.0-stable-e501b3b0/linux-amd64/go1.20.4	84 ms	50	5115	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	0	0	7 s ago
EF-Bootnode-Falkenstein	Geth/v1.12.2-stable-bed84606/linux-amd64/go1.20.7	85 ms	3279	5246	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	0	0	7 s ago
EthPandaOps-geth-prysm-berlin	Geth/v1.12.0-stable-e501b3b0/linux-amd64/go1.20.4	88 ms	50	5117	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	0	0	7 s ago
sunkist	Geth/v1.13.0-unstable-bce5c467/linux-amd64/go1.21.0	80 ms	50	5117	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	0	0	7 s ago
crush	Geth/v1.12.0-stable-e501b3b0/linux-amd64/go1.20.4	34 ms	50	5132	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	0	0	7 s ago
EthPandaOps-geth-nimbus-berlin	Geth/v1.12.0-stable-e501b3b0/linux-amd64/go1.20.4	86 ms	50	5117	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	0	0	7 s ago
EthPandaOps-geth-prysm-boulder	Geth/v1.12.0-stable-e501b3b0/linux-amd64/go1.20.4	28 ms	50	5115	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	0	0	7 s ago
EthPandaOps-geth-lighthouse-boulder	Geth/v1.12.0-stable-e501b3b0/linux-amd64/go1.20.4	27 ms	50	5124	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	0	0	7 s ago
EthPandaOps-geth-lighthouse-berlin	Geth/v1.12.0-stable-e501b3b0/linux-amd64/go1.20.4	114 ms	50	5115	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	0	0	7 s ago
EF-Bootnode-NorthVirginia	Geth/v1.12.2-stable-bed84606/linux-amd64/go1.20.7	37 ms	259	5133	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	0	0	7 s ago
EthPandaOps-geth-lodestar-boulder	Geth/v1.12.0-stable-e501b3b0/linux-amd64/go1.20.4	30 ms	50	5123	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	0	0	7 s ago
EF-Bootnode-Singapore	Geth/v1.12.2-stable-bed84606/linux-amd64/go1.20.7	94 ms	261	5139	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	0	0	7 s ago
EthPandaOps-geth-lodestar-berlin	Geth/v1.12.0-stable-e501b3b0/linux-amd64/go1.20.4	79 ms	50	5127	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	0	0	7 s ago
Foundation Benchmarking(bench05)	Geth/v1.13.0-unstable-d7786d0f-20230822/linux-amd64/go1.20.7	78 ms	50	5118	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	0	0	6 s ago
kinΞth-NUC	Geth/v1.12.2-stable-bed84606/linux-amd64/go1.20.7	80 ms	50	5118	#18,026,893	2dd1defe...fa60a59c	5.875000371659836e+22	0	0	5 s ago

- Display real time and historical statistics about the network and nodes.

Source: https://ethstats.dev

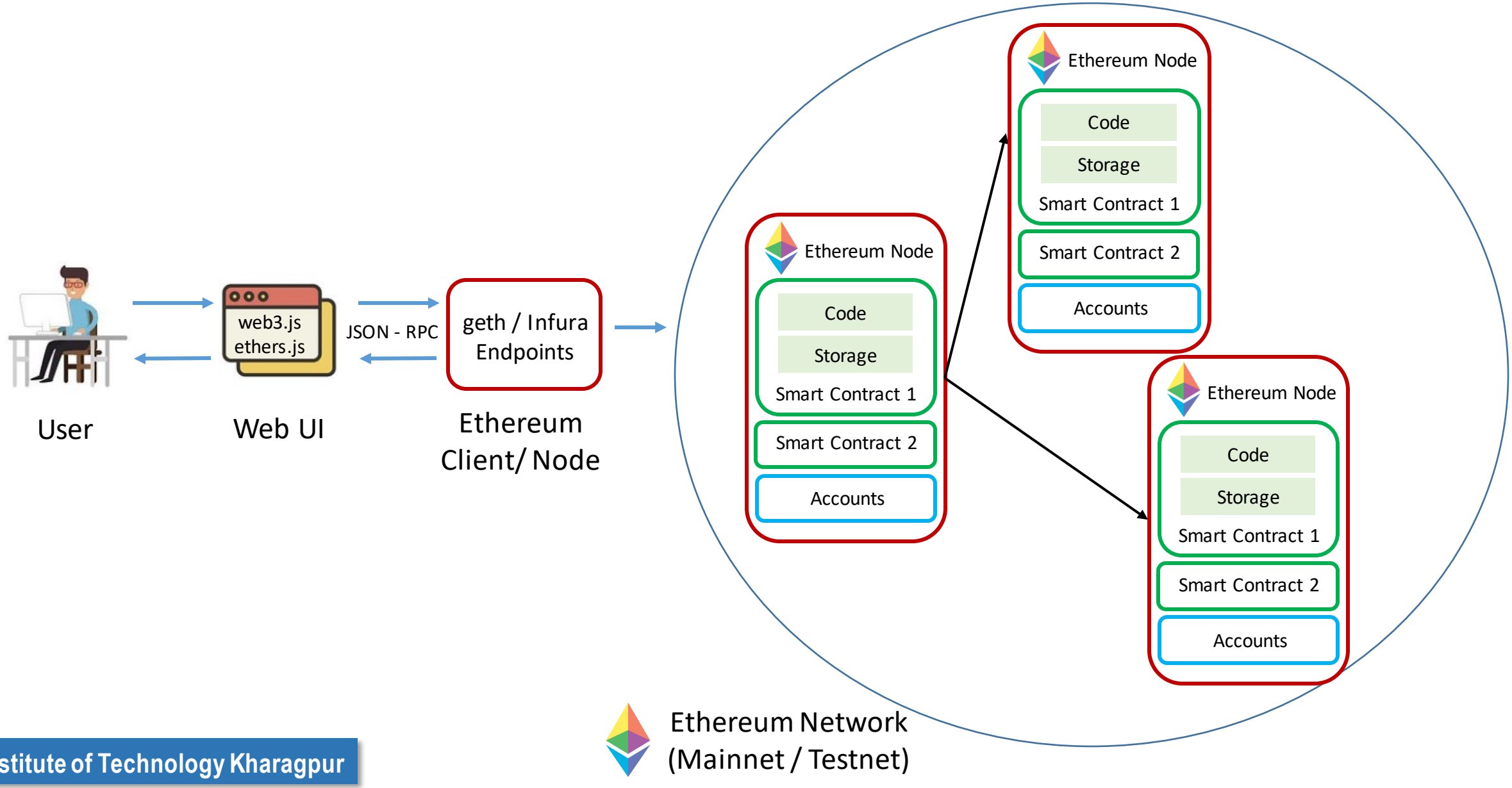
Ethereum Mainnet



- No central server
- Independent nodes connected in a P2P network.

Source: https://www.ethernodes.org/countries

Interacting with an Ethereum Network



Ethereum Client

- **Ethereum Client** allow you set up an Ethereum Node on your Computer.
- An Ethereum Client is like Java Virtual Machine (JVM) and allows you to run Ethereum Programs.
- With an Ethereum Client you can participate in the Ethereum network and perform tasks such as:
  - creating accounts and contracts
  - sending ether to other accounts
  - sending transactions to contracts
  - mining on Ethereum network and etc.

Ethereum Client

- Ethereum Clients supports:
  - Go (go-ethereum)
  - C++ (cpp-ethereum)
  - Python (pyethapp)
  - Javascript (ethereumjs-lib)
  - Ruby (ruby-ethereum)
  - Java (Ethereum(J))
  - Haskell (EthereumH)
  - Rust (Parity)

Go-Ethereum Client (geth)

- Official Go implementation of the Ethereum protocol
- Main Ethereum CLI client
- Entry point into the Ethereum network
  - main, test, or private net
- Capable of running as:
  - Full node (default)
  - Archive node (retaining all historical state)
  - Light node (retrieving data live).
- Provides JSON RPC endpoints exposed on top of HTTP, WebSocket and/or IPC transports.

Installing Go-Ethereum Client (geth)

- Ubuntu
  - Add Ethereum repository:

```
sudo apt-get install software-properties-common
sudo add-apt-repository -y ppa:ethereum/ethereum
```

  - Then install the stable version of go-ethereum:

```
sudo apt-get update
sudo apt-get install ethereum
```

Managing Ethereum Accounts

- USAGE:

```
geth account command [command options] [arguments...]
```

- COMMANDS:
  - `list` Print summary of existing accounts
  - `new` Create a new account
  - `update` Update an existing account
  - `import` Import a private key into a new account

Create an account

- USAGE:

```
geth account new
```

```
utkalikasatapathy@Utkalikas-MacBook-Pro Assignment1_2023 % geth account new
INFO [09-01|01:27:51.392] Maximum peer count                       ETH=50 LES=0 total=50
Your new account is locked with a password. Please give a password. Do not forget this password.
Password:
Repeat password:

Your new key was generated

Public address of the key:   0xcd98495985589AfcB8C29fEDDDDF4a603691A8b2
Path of the secret key file: /Users/utkalikasatapathy/Library/Ethereum/keystore/UTC--2023-08-31T19-57-56.261478000Z--cd98495985589afcb8c29feddddf4a603691a8b2

- You can share your public address with anyone. Others need it to interact with you.
- You must NEVER share the secret key with anyone! The key controls access to your funds!
- You must BACKUP your key file! Without the key, it's impossible to access account funds!
- You must REMEMBER your password! Without the password, it's impossible to decrypt the key!
```

List Accounts

- USAGE:

```
geth account list
```

```
utkalikasatapathy@Utkalikas-MacBook-Pro Assignment1_2023 % geth account list
INFO [09-01|01:30:01.865] Maximum peer count                       ETH=50 LES=0 total=50
Account #0: {cd98495985589afcb8c29feddddf4a603691a8b2} keystore:///Users/utkalikasatapathy/Library/Ethereum/keystore/UTC--2023-08-31T19-57-56.261478000Z--cd98495985589afcb8c29feddddf4a603691a8b2
```

- To See the content of the wallet

```
cat ~/.Ethereum/keystore/<Keyfile-Name>
```

```
utkalikasatapathy@Utkalikas-MacBook-Pro Assignment1_2023 % cat /Users/utkalikasa
tapathy/Library/Ethereum/keystore/UTC--2023-08-31T19-57-56.261478000Z--cd9849598
5589afcb8c29feddddf4a603691a8b2
{"address":"cd98495985589afcb8c29feddddf4a603691a8b2","crypto":{"cipher":"aes-12
8-ctr","ciphertext":"ba27712227bd4dabf9fad0e25c5355a09c0d19e44a001d5abdfe41ee133
78fc5","cipherparams":{"iv":"279ffb6b454d76867791047d580c2a4c"},"kdf":"scrypt","
kdfparams":{"dklen":32,"n":262144,"p":1,"r":8,"salt":"393f32d9e969b9a5c89175f645
721618fa18dd1fd5398c3dc838bdd90c139b8f"},"mac":"ccb03aefad9b99597593f74fe6750c89
3555c92da184e81a753d8d891e7b49ab"},"id":"883372b2-17aa-4760-8f2e-e75fb6a826e1","
version":3}%
```

Connect to a network

- Starting geth without any flag connects to the Ethereum mainnet.
- In addition to the mainnet, geth recognizes a few testnets which you can connect to via the respective flags:
- `--sepolia`: Sepolia Proof-of-Stake test Netwok
  https://sepolia.etherscan.io/
- `--ropsten`: Ropsten proof-of-work test network
  https://ropsten.etherscan.io/
- `--rinkeby`: Rinkeby proof-of-authority test network
  https://rinkeby.etherscan.io/
- `--goerli`: Goerli proof-of-authority test network
  https://goerli.etherscan.io/

Sync modes

- You can start Geth in one of three different sync modes using the **--syncmode "<mode>"** argument that determines what sort of node it is in the network:

  `full:` Downloads all blocks (including headers, transactions, and receipts) and generates the state of the blockchain incrementally by executing every block.

  `fast:` Downloads all blocks (including headers, transactions and receipts), verifies all headers, and downloads the state and verifies it against the headers.

  `snap (Default):` Same functionality as fast, but with a faster algorithm.

  `light:` Downloads all block headers, block data, and verifies some randomly.

Connecting to sepolia testnet

```
geth --sepolia --syncmode "light"
```

```
utkalikasatapathy@Utkalikas-MacBook-Pro ethchain % geth --sepolia --syncmode "light"
INFO [08-31|22:27:27.843] Starting Geth on Sepolia testnet...
INFO [08-31|22:27:27.843] Dropping default light client cache      provided=1024 updated=128
INFO [08-31|22:27:27.845] Maximum peer count                       ETH=0 LES=10 total=50
INFO [08-31|22:27:27.848] Set global gas cap                       cap=50,000,000
INFO [08-31|22:27:27.848] Initializing the KZG library             backend=gokzg
INFO [08-31|22:27:27.868] Using pebble as the backing database
INFO [08-31|22:27:27.868] Allocated cache and file handles         database=/Users/utkalikasatapathy/Library/Ethereum/sepolia/geth/lightchaindata cache=64.00MiB handles=5120
INFO [08-31|22:27:27.903] Using pebble as the backing database
INFO [08-31|22:27:27.903] Allocated cache and file handles         database=/Users/utkalikasatapathy/Library/Ethereum/sepolia/geth/les.client cache=16.00MiB handles=16
INFO [08-31|22:27:27.947]
INFO [08-31|22:27:27.947] ---------------------------------------------------------------------------------------------------------------------------------------------------------
INFO [08-31|22:27:27.947] Chain ID:  11155111 (sepolia)
INFO [08-31|22:27:27.947] Consensus: Beacon (proof-of-stake), merged from Ethash (proof-of-work)
INFO [08-31|22:27:27.947]
INFO [08-31|22:27:27.947] Pre-Merge hard forks (block based):
INFO [08-31|22:27:27.947]  - Homestead:                    #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/homestead.md)
INFO [08-31|22:27:27.947]  - Tangerine Whistle (EIP 150): #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/tangerine-whistle.md)
INFO [08-31|22:27:27.947]  - Spurious Dragon/1 (EIP 155): #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon.md)
INFO [08-31|22:27:27.947]  - Spurious Dragon/2 (EIP 158): #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon.md)
INFO [08-31|22:27:27.947]  - Byzantium:                    #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/byzantium.md)
INFO [08-31|22:27:27.947]  - Constantinople:               #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/constantinople.md)
INFO [08-31|22:27:27.947]  - Petersburg:                   #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/petersburg.md)
INFO [08-31|22:27:27.947]  - Istanbul:                     #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/istanbul.md)
INFO [08-31|22:27:27.947]  - Muir Glacier:                 #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/muir-glacier.md)
INFO [08-31|22:27:27.947]  - Berlin:                       #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/berlin.md)
INFO [08-31|22:27:27.947]  - London:                       #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/london.md)
INFO [08-31|22:27:27.947]
INFO [08-31|22:27:27.947] Merge configured:
INFO [08-31|22:27:27.947]  - Hard-fork specification:    https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/paris.md
INFO [08-31|22:27:27.947]  - Network known to be merged: true
INFO [08-31|22:27:27.947]  - Total terminal difficulty:  17000000000000000
INFO [08-31|22:27:27.947]  - Merge netsplit block:       #1735371
INFO [08-31|22:27:27.947]
INFO [08-31|22:27:27.947] Post-Merge hard forks (timestamp based):
INFO [08-31|22:27:27.947]  - Shanghai:                    @1677557088 (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/shanghai.md)
INFO [08-31|22:27:27.947]
INFO [08-31|22:27:27.947] ---------------------------------------------------------------------------------------------------------------------------------------------------------
INFO [08-31|22:27:27.947]
INFO [08-31|22:27:27.968] Loaded most recent local header          number=0 hash=25a5cc..3e6dd9 td=131,072 age=1y11mo1w
INFO [08-31|22:27:27.968] Gasprice oracle is ignoring threshold set threshold=2
INFO [08-31|22:27:27.973] Starting peer-to-peer node               instance=Geth/v1.12.2-stable/darwin-arm64/go1.20.7
INFO [08-31|22:27:27.996] IPC endpoint opened                      url=/Users/utkalikasatapathy/Library/Ethereum/sepolia/geth.ipc
WARN [08-31|22:27:27.996] Light client mode is an experimental feature
ERROR[08-31|22:27:27.996] Discovery v5 is not initialized
INFO [08-31|22:27:27.996] New local node record                    seq=1,693,497,893,662 id=e850f600260427d7 ip=127.0.0.1 udp=0 tcp=30303
INFO [08-31|22:27:27.996] Started P2P networking                   self="enode://4809f97d9ec60af631e913fd8b4db6b767293b3d7f3c01f14d2fd82559f21f451684d8781eb7db658162e787f6ed432ffd36df6e70d5403fe6b94345e7f6932f@12
7.0.0.1:30303?discport=0"
```

Interacting with Geth

- You can interact with Geth in two ways:
  - Directly with the node using the JavaScript console over IPC

  Or
  - Connecting to the node remotely over HTTP using RPC.
- **IPC** allows you to do more, especially when it comes to creating and interacting with accounts, but you need direct access to the node.
- **RPC** allows remote applications to access your node but has limitations and security considerations.

Using RPC over HTTP

```
geth --sepolia --syncmode "light" --http
```

ethchain — geth --sepolia --syncmode light --http — 212×61

```
utkalikasatapathy@Utkalikas-MacBook-Pro ethchain % geth --sepolia --syncmode "light" --http
INFO [08-31|22:36:12.782] Starting Geth on Sepolia testnet...
INFO [08-31|22:36:12.782] Dropping default light client cache      provided=1024 updated=128
INFO [08-31|22:36:12.783] Maximum peer count                       ETH=0 LES=10 total=50
INFO [08-31|22:36:12.786] Set global gas cap                       cap=50,000,000
INFO [08-31|22:36:12.786] Initializing the KZG library             backend=gokzg
INFO [08-31|22:36:12.807] Using pebble as the backing database
INFO [08-31|22:36:12.807] Allocated cache and file handles         database=/Users/utkalikasatapathy/Library/Ethereum/sepolia/geth/lightchaindata cache=64.00MiB handles=5120
INFO [08-31|22:36:12.847] Using pebble as the backing database
INFO [08-31|22:36:12.847] Allocated cache and file handles         database=/Users/utkalikasatapathy/Library/Ethereum/sepolia/geth/les.client cache=16.00MiB handles=16
INFO [08-31|22:36:12.875]
INFO [08-31|22:36:12.875] ---------------------------------------------------------------------------------------------------------------------------------------------------------
INFO [08-31|22:36:12.875] Chain ID:  11155111 (sepolia)
INFO [08-31|22:36:12.875] Consensus: Beacon (proof-of-stake), merged from Ethash (proof-of-work)
INFO [08-31|22:36:12.875]
INFO [08-31|22:36:12.875] Pre-Merge hard forks (block based):
INFO [08-31|22:36:12.875]  - Homestead:                    #0       (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/homestead.md)
INFO [08-31|22:36:12.875]  - Tangerine Whistle (EIP 150): #0       (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/tangerine-whistle.md)
INFO [08-31|22:36:12.875]  - Spurious Dragon/1 (EIP 155): #0       (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon.md)
INFO [08-31|22:36:12.875]  - Spurious Dragon/2 (EIP 158): #0       (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon.md)
INFO [08-31|22:36:12.875]  - Byzantium:                    #0       (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/byzantium.md)
INFO [08-31|22:36:12.875]  - Constantinople:               #0       (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/constantinople.md)
INFO [08-31|22:36:12.875]  - Petersburg:                   #0       (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/petersburg.md)
INFO [08-31|22:36:12.875]  - Istanbul:                     #0       (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/istanbul.md)
INFO [08-31|22:36:12.875]  - Muir Glacier:                 #0       (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/muir-glacier.md)
INFO [08-31|22:36:12.875]  - Berlin:                       #0       (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/berlin.md)
INFO [08-31|22:36:12.875]  - London:                       #0       (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/london.md)
INFO [08-31|22:36:12.875]
INFO [08-31|22:36:12.875] Merge configured:
INFO [08-31|22:36:12.875]  - Hard-fork specification:    https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/paris.md
INFO [08-31|22:36:12.875]  - Network known to be merged: true
INFO [08-31|22:36:12.875]  - Total terminal difficulty:  17000000000000000
INFO [08-31|22:36:12.875]  - Merge netsplit block:       #1735371
INFO [08-31|22:36:12.875]
INFO [08-31|22:36:12.875] Post-Merge hard forks (timestamp based):
INFO [08-31|22:36:12.875]  - Shanghai:                   @1677557088 (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/shanghai.md)
INFO [08-31|22:36:12.875]
INFO [08-31|22:36:12.875] ---------------------------------------------------------------------------------------------------------------------------------------------------------
INFO [08-31|22:36:12.875]
INFO [08-31|22:36:12.882] Loaded most recent local header          number=0 hash=25a5cc..3e6dd9 td=131,072 age=1y11mo1w
INFO [08-31|22:36:12.882] Gasprice oracle is ignoring threshold set threshold=2
INFO [08-31|22:36:12.887] Starting peer-to-peer node               instance=Geth/v1.12.2-stable/darwin-arm64/go1.20.7
INFO [08-31|22:36:12.911] IPC endpoint opened                      url=/Users/utkalikasatapathy/Library/Ethereum/sepolia/geth.ipc
INFO [08-31|22:36:12.912] HTTP server started                      endpoint=127.0.0.1:8545 auth=false prefix= cors= vhosts=localhost
WARN [08-31|22:36:12.912] Light client mode is an experimental feature
ERROR[08-31|22:36:12.912] Discovery v5 is not initialized
INFO [08-31|22:36:12.912] New local node record                    seq=1,693,497,893,663 id=e850f600260427d7 ip=127.0.0.1 udp=0 tcp=30303
INFO [08-31|22:36:12.912] Started P2P networking                   self="enode://4809f97d9ec60af631e913fd8b4db6b767293b3d7f3c01f14d2fd82559f21f451684d8781eb7db658162e787f6ed432ffd36df6e70d5403fe6b94345e7f6932f@12
7.0.0.1:30303?discport=0"
```

Query Balance

- Querying the balance of an account:
- POST request to the HTTP endpoint (default: 127.0.0.1:8545)
- JSON Payload:

```
{
"jsonrpc": "2.0",
"method": "eth_getBalance",
"params": [
   "0x9808f22453Ee87cc23eA76ca7Ed66a4F294d54D4",
   "latest”
],
"id": 1
}
```

Query using curl

- Querying the balance of an account:
- POST request using curl:

Request:

```
curl 127.0.0.1:8545 \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_gasPrice","params": [],"id":1}’
```

Response:

```
utkalikasatapathy@Utkalikas-MacBook-Pro ~ % curl 127.0.0.1:8545 \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_gasPrice","params": [],"id":1}'
{"jsonrpc":"2.0","id":1,"result":"0x77359400"}
```

Ethereum Units

Unit	Wei Value	Wei
wei	1 wei	1
kwei (babbage)	1e3 wei	1000
mwei (lovelace)	1e6 wei	1,000,000
gwei (shannon)	1e9 wei	1,000,000,000
microether (szabo)	1e12 wei	1,000,000,000,000
milliether (finney)	1e15 wei	1,000,000,000,000,000
ether (ETH)	1e18 wei	1,000,000,000,000,000,000

Ethereum Transactions

- Transactions are cryptographically signed instructions from accounts.
- An account will initiate a transaction to update the state of the Ethereum network.
- The simplest transaction is transferring ETH from one account to another.

Ethereum Transactions

- Transactions are cryptographically signed instructions from accounts.
- An account will initiate a transaction to update the state of the Ethereum network.
- The simplest transaction is transferring ETH from one account to another.

- Requires:
  1. Access to account private key
  2. Gas

Gas

- Each Ethereum transaction requires computational resources to execute.
- Each transaction requires a fee.
- **Gas** refers to the fee required to conduct a transaction on Ethereum successfully.
- Gas is paid as amounts of Ethers.

Source: https://docs.infura.io/getting-started

Unlock Account

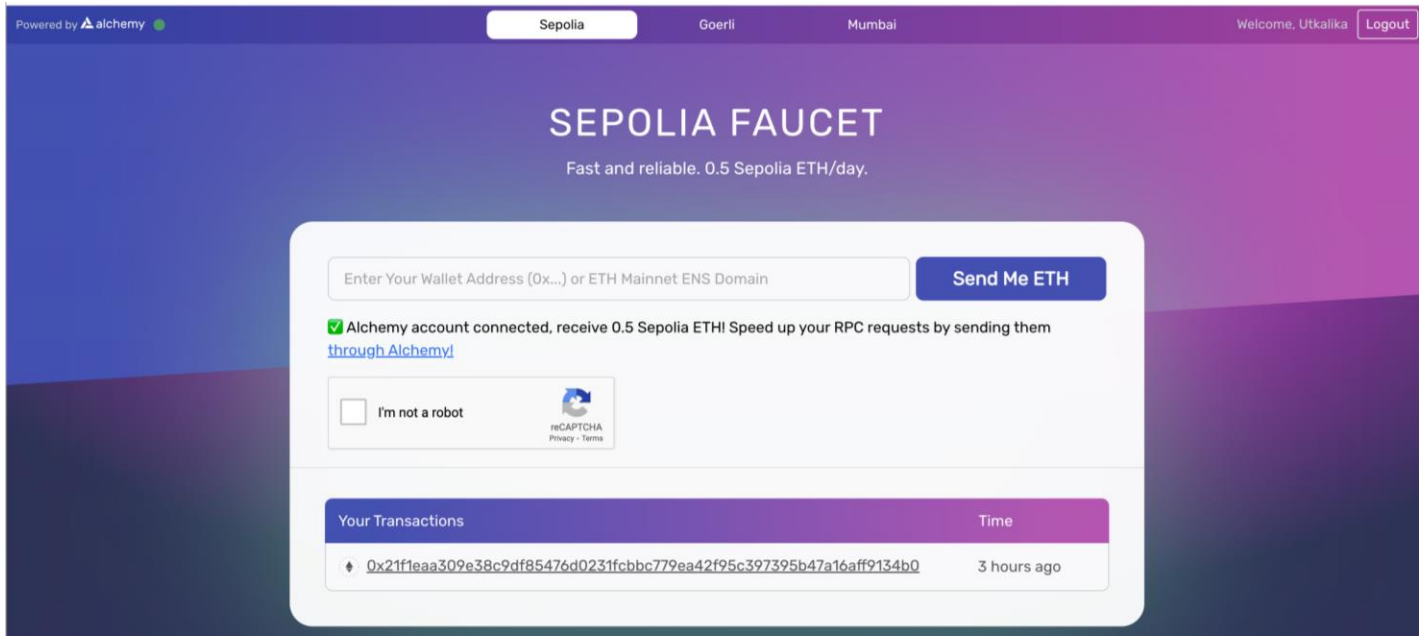
- Create new account for Sepolia testnet, suppose password is set to "12345"
- Write account password in a text file. Here we name it pwd.txt
  - `geth --sepolia account new`
  - `echo "12345" > pwd.txt`

- Unlock account while starting geth
  - `geth --sepolia --syncmode "light" --http --allow-insecure-unlock --unlock 0x3873DcDB0d5177B2Be28fFb911e08be6DbB40f7A --password pwd.txt`

Source: https://docs.infura.io/getting-started

Get ether in testnet

- For executing transactions in a testnet, we can obtain ethers from faucet.
- Sepolia faucet: https://sepoliafaucet.com/
- Goerli faucet:
  - Social faucet: https://faucet.goerli.mudit.blog/
  - Simple faucet: https://goerli-faucet.slock.it/
- Rinkeby faucet: https://faucet.rinkeby.io/

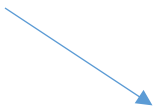


eth_sendTransaction

- POST sendTransaction: Creates new message call transaction or a contract creation, if the data field contains code.
- Parameters
  - object - The transaction object
  - from: DATA, 20 Bytes - The address the transaction is send from.
  - to: DATA, 20 Bytes -The address the transaction is directed to.
  - gas: QUANTITY - (optional, default: 90000) Integer of the gas provided for the transaction execution. It will return unused gas.
  - gasPrice: QUANTITY - (optional, default: To-Be-Determined) The gasPrice used for each paid gas
  - value: QUANTITY - (optional) Integer of the value sent with this transaction
  - data: DATA - The compiled code of a contract OR the hash of the invoked method signature and encoded parameters. For details see Ethereum Contract ABI
  - nonce: QUANTITY - (optional) Integer of a nonce. This allows to overwrite your own pending transactions that use the same nonce.
- Returns
  - DATA: 32 Bytes - the transaction hash, or the zero hash if the transaction is not yet available.

Prepare Transaction

```
{
"jsonrpc": "2.0",
"method": "eth_sendTransaction",
"params": [
      {
      "from": "04e7ad5d91bedef78e57ee6c56db10de39478232",
      "to": "3873dcdb0d5177b2be28ffb911e08be6dbb40f7a",
      "value": "0x2386F26FC10000"
      }
      ],
"id": 0
}
```

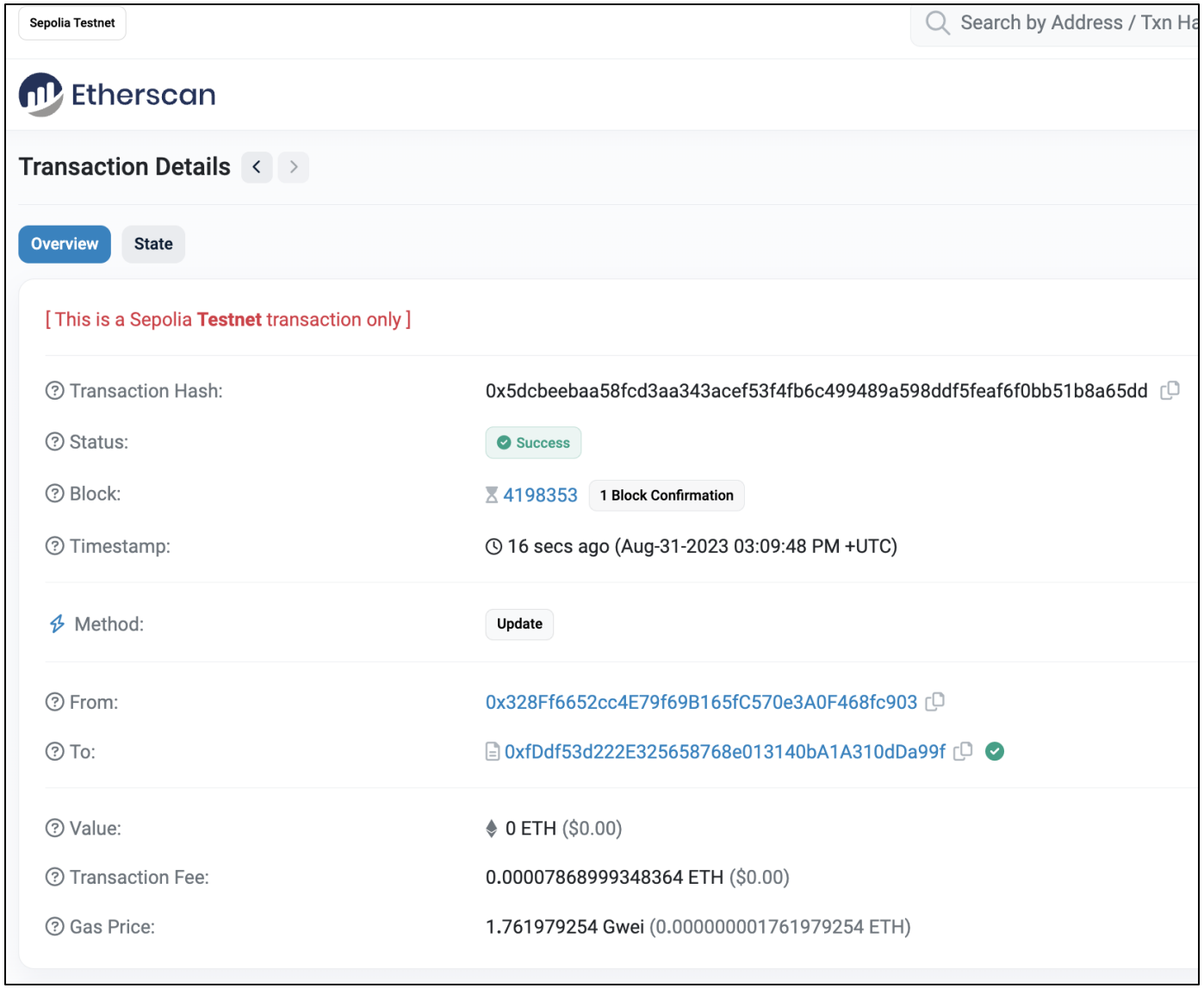


0.1 Ether

Submit Transaction

```
curl -X POST http://localhost:8545 \
-H "Content-Type: application/json" \
--data{"jsonrpc": "2.0", "method": "eth_sendTransaction", "params": [{"from":
"04e7ad5d91bedef78e57ee6c56db10de39478232", "to":
"3873dcdb0d5177b2be28ffb911e08be6dbb40f7a", "value": "0x2386F26FC10000"}],"id": 0}
```

View Transaction on Etherscan



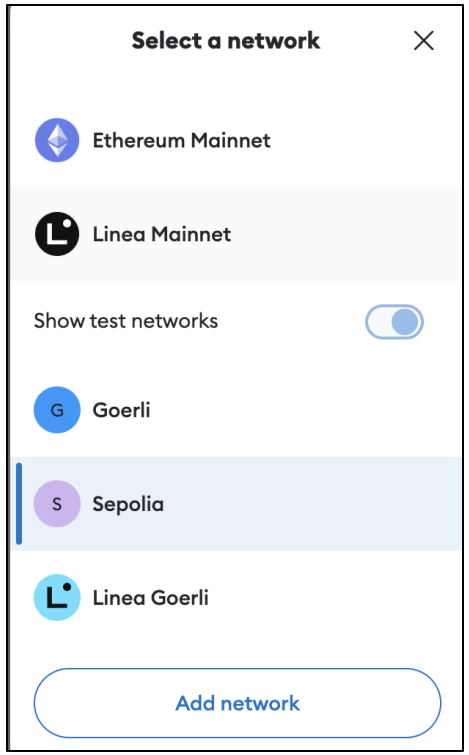
Many More RPC Methods

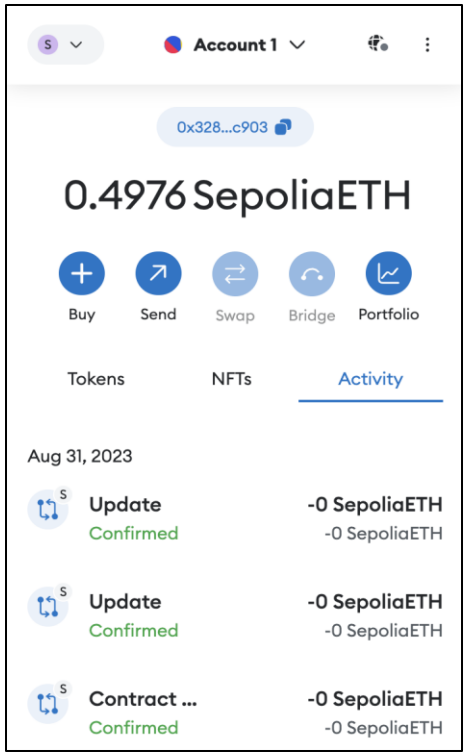
eth_accounts	Returns a list of addresses owned by client.
eth_blockNumber	Returns the number of most recent block.
eth_call	Executes a new message call immediately without creating a transaction on the block chain.
eth_chainId	Returns the chain ID of the current network.
eth_coinbase	Returns the client coinbase address.
eth_createAccessList	Generates an access list for a transaction.
eth_estimateGas	Generates and returns an estimate of how much gas is necessary to allow the transaction to complete.
eth_feeHistory	Transaction fee history
eth_gasPrice	Returns the current price per gas in wei.
eth_getBalance	Returns the balance of the account of given address.
eth_getBlockByHash	Returns information about a block by hash.
eth_getBlockByNumber	Returns information about a block by number.
eth_getBlockReceipts	Returns the receipts of a block by number or hash.
eth_getBlockTransactionCountByHash	Returns the number of transactions in a block from a block matching the given block ha

https://playground.open-rpc.org/?schemaUrl=https://raw.githubusercontent.com/ethereum/eth1.0-apis/assembled-spec/openrpc.json

MetaMask

- MetaMask is a software cryptocurrency wallet used to interact with the Ethereum blockchain.
- It allows users to access their Ethereum wallet through a browser extension or mobile app, which can then be used to interact with decentralized applications.





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

Extracting Private Key of Account in Geth

```
curl -LSs https://raw.githubusercontent.com/gochain/web3/master/install.sh | sh
```

```
web3 account extract --keyfile <Keyfile-Location> --password <Password>
```

Example:

```
web3 account extract --keyfile ~/.ethereum/sepolia/keystore/<KeyFileName> --password
<Password>
```

- This private key can used to import the account in Cryptocurrency wallet like MetaMask (Import Account -> Private Key)

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

INFURA

- Infura provides access to the Ethereum JSON-RPC API method library that interacts with the Ethereum blockchain.
- Methods include functionality for reading and writing data to the network, and executing smart contracts.

```
curl https://mainnet.infura.io/v3/<YOUR-API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_gasPrice","params": [],"id":1}'
```

Source: https://docs.infura.io/getting-started

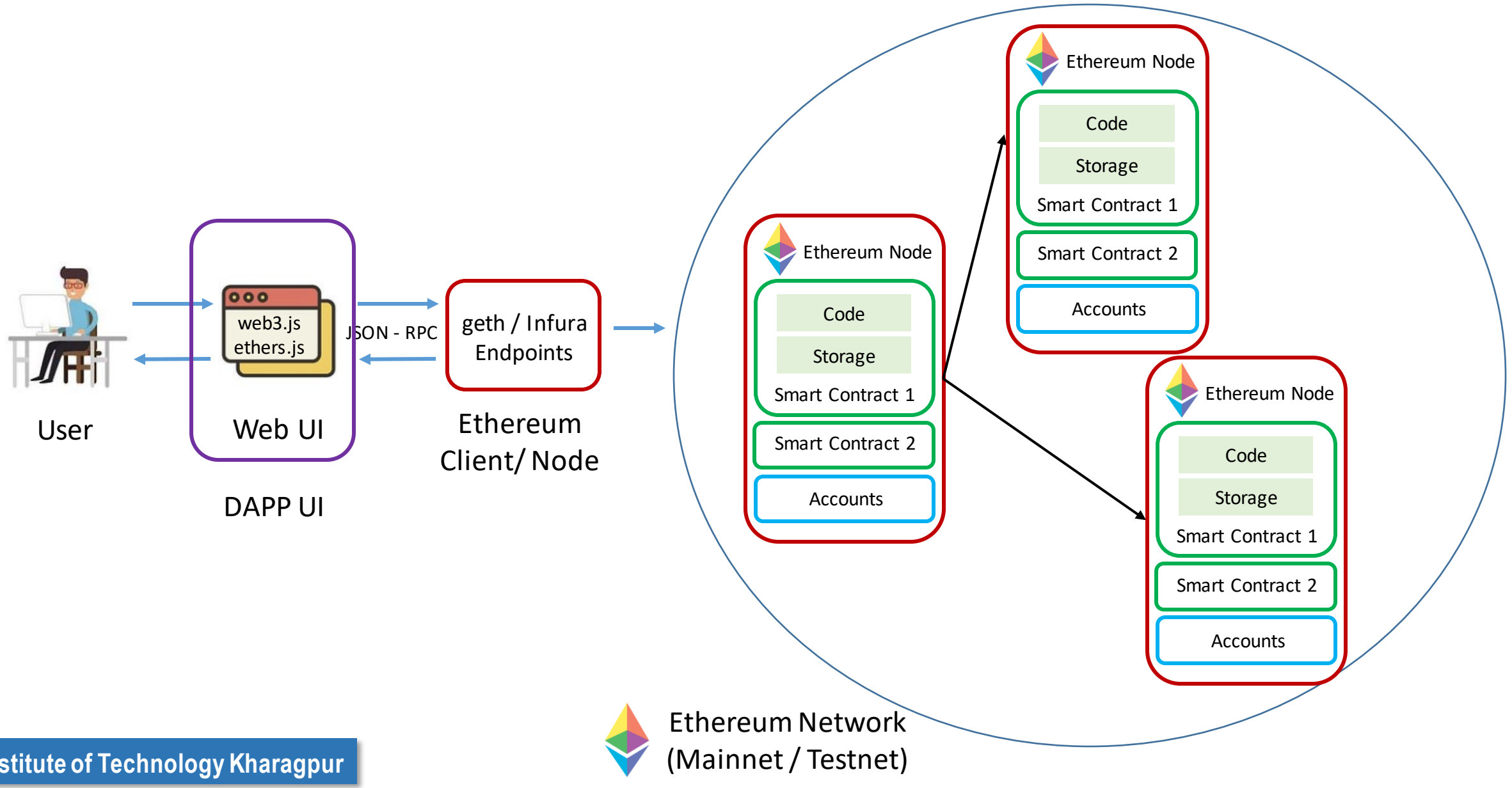
DAPPS in Ethereum

- DAPP - decentralized application
- Application that is built for decentralized networks like Ethereum

- Combines two components:
  - 1. Smart Contracts
  - 2. User interface for executing transactions and contracts

Source: https://docs.infura.io/getting-started

Programmatically access Ethereum networks



web3.js

- Ethereum JavaScript API
  - https://web3js.readthedocs.io/
- Collection of libraries that allow you to interact with a local or remote Ethereum nodes.
- It can connect using:
  - HTTP
  - IPC
  - WebSocket
- Install web3.js
  - `npm -i web3`
- Check node and npm version
  - `node --version`
  - `npm --version`

```
utkalikasatapathy@Utkalikas-MacBook-Pro ~ % node -v
v18.17.1
utkalikasatapathy@Utkalikas-MacBook-Pro ~ % npm -v
9.6.7
```

Connecting web3.js to Clients

Connecting to Infura:

```
var { Web3 } = require("web3");
var provider = "https://mainnet.infura.io/v3/<YOUR-API-KEY>";
var web3Provider = new Web3.providers.HttpProvider(provider);
var web3 = new Web3(web3Provider);
```

Connecting to Geth:

```
var { Web3 } = require("web3");
var provider = “http://localhost:8545";
var web3Provider = new Web3.providers.HttpProvider(provider);
var web3 = new Web3(web3Provider);
```

Query Balance

- Create a file with .js extension. Example: ”checkBalance.js”

```
var { Web3 } = require("web3");
var provider = "http://localhost:8545";
var web3Provider = new Web3.providers.HttpProvider(provider);
var web3 = new Web3(web3Provider);
console.log("Balance of the account 0x8cD5F171e45d899d7FeB56206a030d749a1A7257 is:");
web3.eth.getBalance("0x8cD5F171e45d899d7FeB56206a030d749a1A7257").then(console.log);
```

- Execute it in the terminal:

```
$ node checkBalance.js
```

```
utkalikasatapathy@Utkalikas-MacBook-Pro Assignment1_2023 % node checkBalance.js
Balance of the account 0x8cD5F171e45d899d7FeB56206a030d749a1A7257 is:
0n
```

Smart Contracts

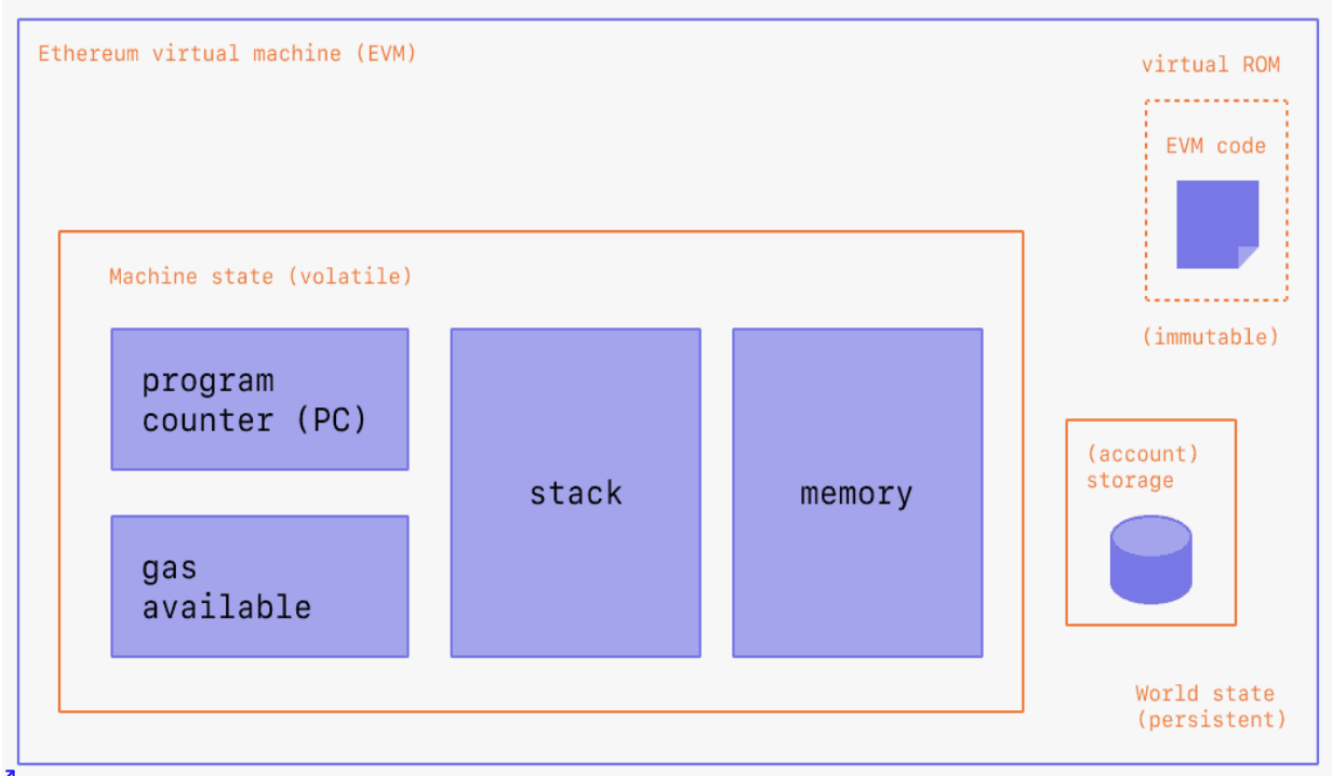
- Program that is executed in a decentralized setting.
- Acts on distributed ledger data.
- Output of the program depends on consensus of independent participants executing it.

Ethereum Smart Contracts

- Program that reads data from the Ethereum ledger, performs operations on it, and writes back the output to the ledger.
- Smart contracts are a type of Ethereum account.
  - have a balance
  - can send transactions over the network
  - not controlled by a user, run as programmed
- User accounts interact with a smart contract by submitting transactions that execute a function defined on the smart contract.

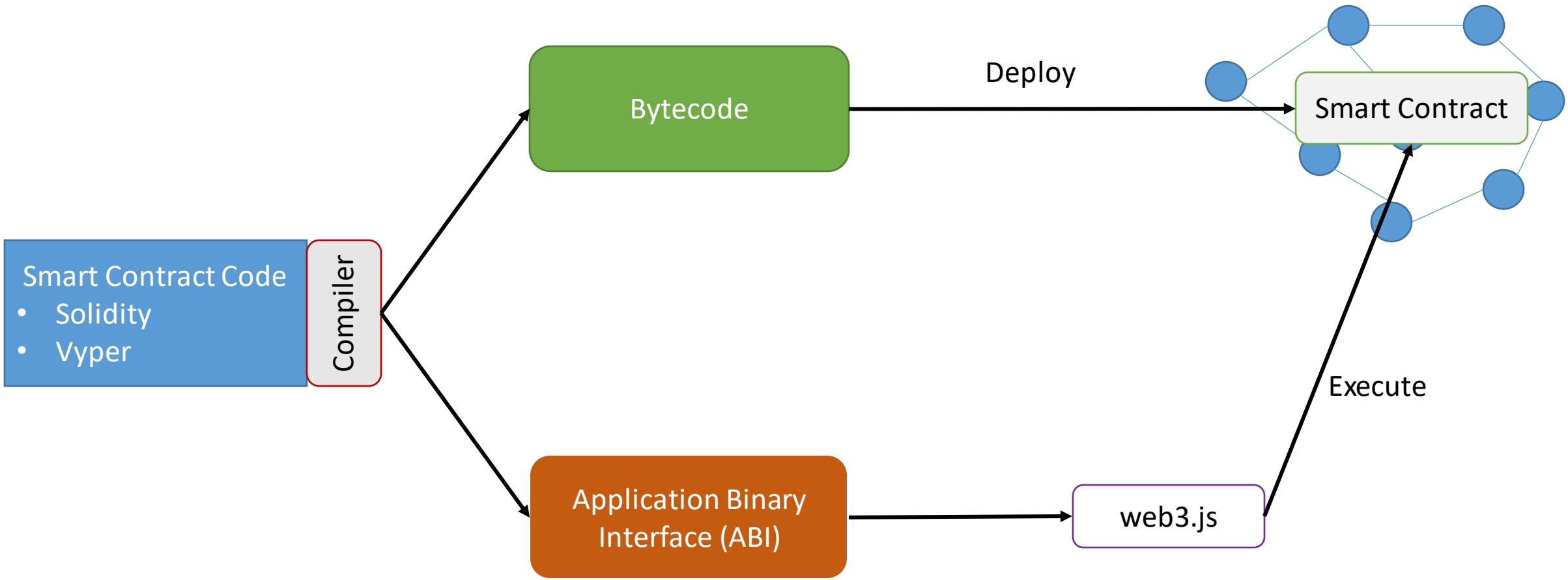
Ethereum Virtual Machine - EVM

- Ethereum implements a distributed state machine / replicated state machine.
- Ledger data holds the state - accounts, balances, and other variables.
- Transactions deterministically change the machine from one state to another.



Source: https://ethereum.org/en/developers/docs/evm/

Compiler



https://vyper.readthedocs.io/en/stable/
https://docs.soliditylang.org/en/v0.8.9/

Solidity

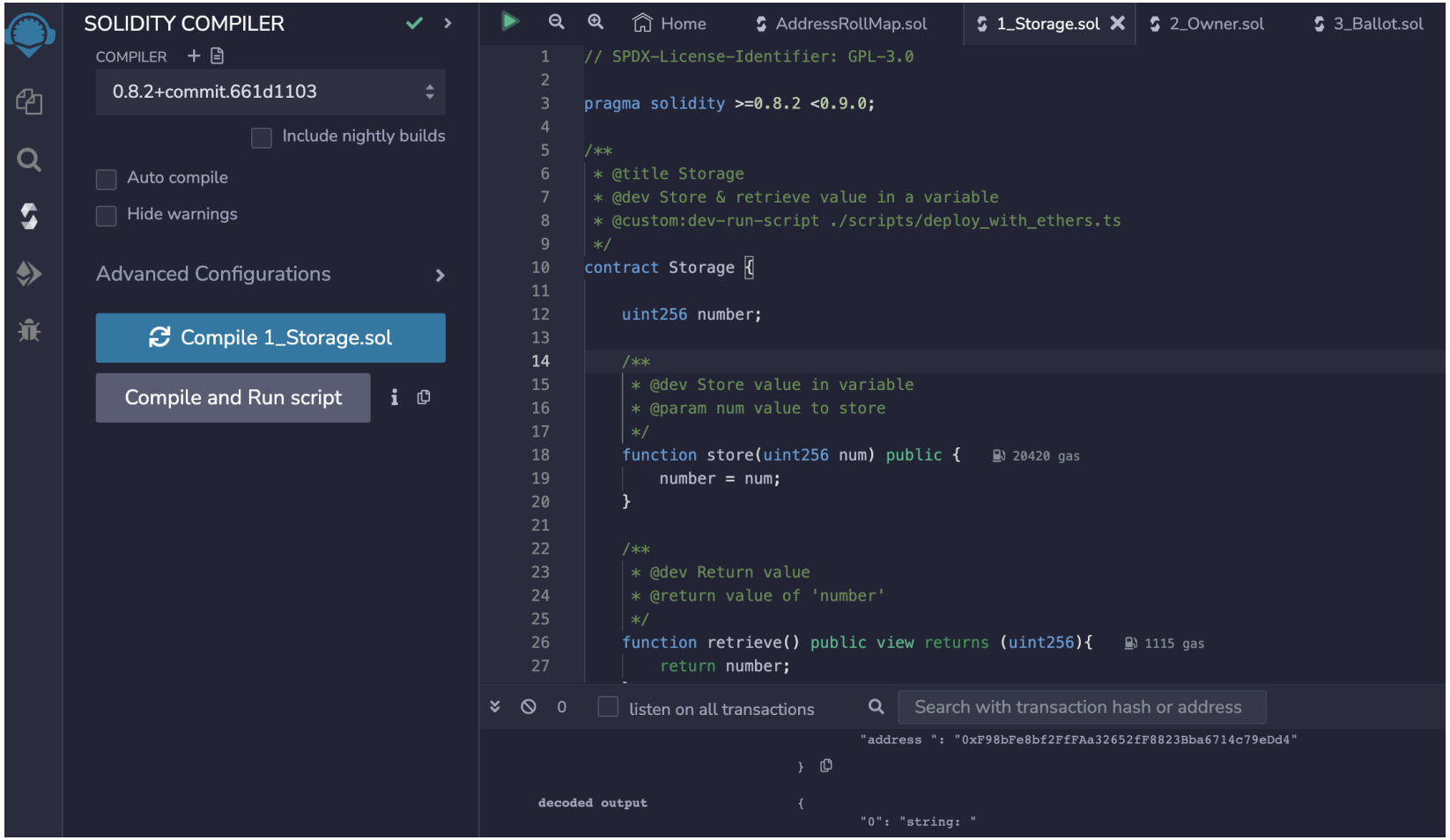
- High-level language for implementing smart contracts
  - Statically typed
  - Supports inheritance
  - Libraries
  - Complex user-defined types
- Syntax influenced by Javascript and C++

```
// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.8.2 <0.9.0;

/**
* @title Storage
* @dev Store & retrieve value in a variable
* @custom:dev-run-script ./scripts/deploy_with_ethers.ts
*/
contract Storage {

uint256 number;

/**
* @dev Store value in variable
* @param num value to store
*/
function store(uint256 num) public {
number = num;
}

/**
* @dev Return value
* @return value of 'number'
*/
function retrieve() public view returns (uint256){
return number;
}
}
```

https://docs.soliditylang.org/en/v0.8.9/

Compiling Solidity

- solc compiler
  - Compile .sol files to bytecode
  - Command line tool
- Remix editor
  - Browser based
  - Compile, emulate, deploy contracts
  - https://remix.ethereum.org/



https://docs.soliditylang.org/en/v0.8.9/installing-solidity.html#installing-solidity

Simple Storage Contract

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;
contract Storage {
        uint256 positivenumber;
        function store(uint256 inputnumber) public {
                positivenumber = inputnumber;
        }
        function readdata() public view returns (uint256){
                return positivenumber;
        }
}
```


https://docs.soliditylang.org/en/v0.8.9/installing-solidity.html#installing-solidity

Executing with Web3

```
var Web3 = require('web3');
var Contract = require('web3-eth-contract');
Contract.setProvider(new Web3.providers.HttpProvider('http://localhost:8545'));
var myContract = new Contract(<ABI>,
"0xb3f36458FFc0C686DB4f2FF6002a55bFD85C03C8",
{
from: "0xd84a0607843b28c3f468857f82f784d9ff743bf8",
gasPrice: "20000000000"
}
);

myContract.methods.readdata().call().then(function (output) { console.log(output) });
```

Read Data

https://docs.soliditylang.org/en/v0.8.9/installing-solidity.html#installing-solidity

Executing with Web3

```
var Web3 = require('web3');
var Contract = require('web3-eth-contract');
Contract.setProvider(new Web3.providers.HttpProvider('http://localhost:8545'));
var myContract = new Contract(<ABI>,
"0xb3f36458FFc0C686DB4f2FF6002a55bFD85C03C8",
{
from: "0xd84a0607843b28c3f468857f82f784d9ff743bf8",
gasPrice: "20000000000"
}
);

myContract.methods.store("11").send().then(function (output) { console.log(output) });
```

Write Data

https://docs.soliditylang.org/en/v0.8.9/installing-solidity.html#installing-solidity

References (Extra Resources)

- Ethereum Whitepaper - https://ethereum.org/en/whitepaper/
- Mastering Ethereum - https://github.com/ethereumbook/ethereumbook
- Ethereum EVM illustrated - https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf
- Go-Eth Command-line Options - https://geth.ethereum.org/docs/fundamentals/command-line-options
- TRUFFLE SUITE GANACHE - https://trufflesuite.com/ganache/
- Connect Geth Ethereum Private Blockchain to Metamask - https://blog.cryptostars.is/connect-geth-ethereum-private-network-to-metamask-7c58a0229eb6
- Beginners guide to Ethereum - https://medium.com/taipei-ethereum-meetup/beginners-guide-to-ethereum-3-explain-the-genesis-file-and-use-it-to-customize-your-blockchain-552eb6265145
- A PYTHON DEVELOPER'S INTRODUCTION TO ETHEREUM - https://ethereum.org/en/developers/tutorials/a-developers-guide-to-ethereum-part-one/
- INFURA - https://docs.infura.io/getting-started#1-sign-up-to-infura
- INFURA Make Request - https://docs.infura.io/networks/ethereum/how-to/make-requests
- Using Puppeth, the Ethereum Private Network Manage - https://www.sitepoint.com/puppeth-introduction/
- How to setup your own private Ethereum network - https://medium.com/@pradeep_thomas/how-to-setup-your-own-private-ethereum-network-f80bc6aea088
- Open-RPC Playground - https://playground.open-rpc.org/?schemaUrl=https://raw.githubusercontent.com/ethereum/eth1.0-apis/assembled-spec/openrpc.json
- Private Networks - https://geth.ethereum.org/docs/fundamentals/private-network

thank you!