NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

Bishakh Chandra Ghosh

Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Lecture 37: Hyperledger Fabric 4

CONCEPTS COVERED

- Hyperledger Foundation
- Hyperledger Fabric Introduction
- Fabric Architecture
- Hyperledger Fabric Installation

KEYWORDS

- Ethereum
- Geth
- Testnets

Ethereum

- "Ethereum is a technology that lets you send **cryptocurrency** to anyone for a small fee. It also **powers applications that everyone can use and no one can take down**."

- **Permissionless** blockchain **capable of executing smart contracts**.

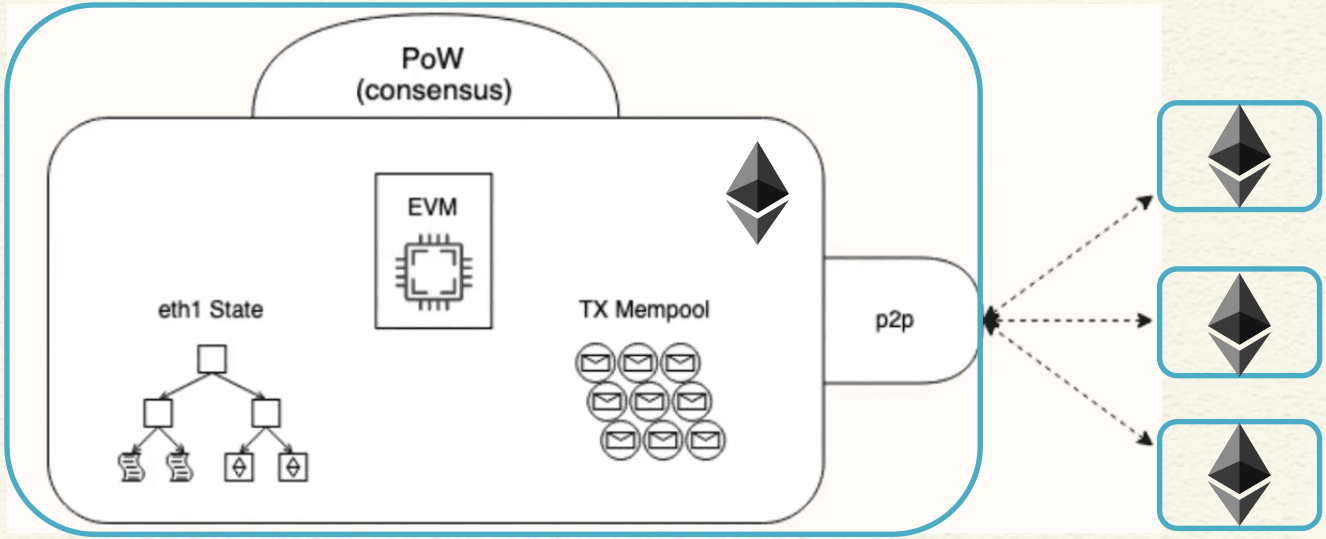
https://ethereum.org/en/what-is-ethereum/

Ethereum Network

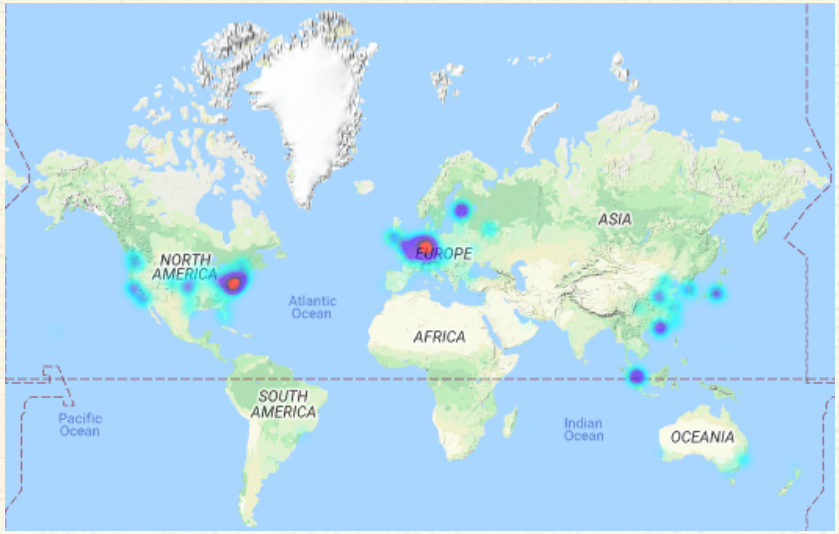
- **Distributed network of computers**, known as **nodes** that can verify blocks and transaction data.

- An application, known as a **client**, running on your computer is a node.

Ethereum Network



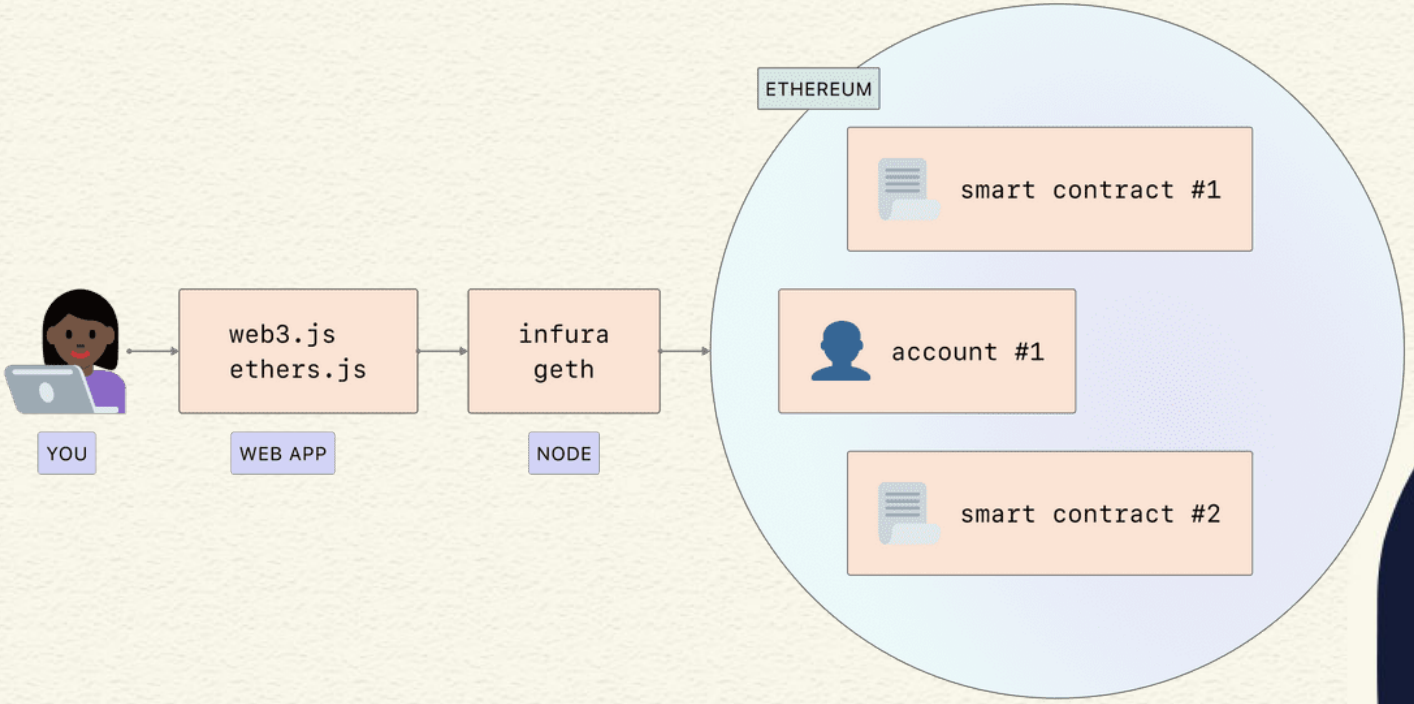
Ethereum Mainnet



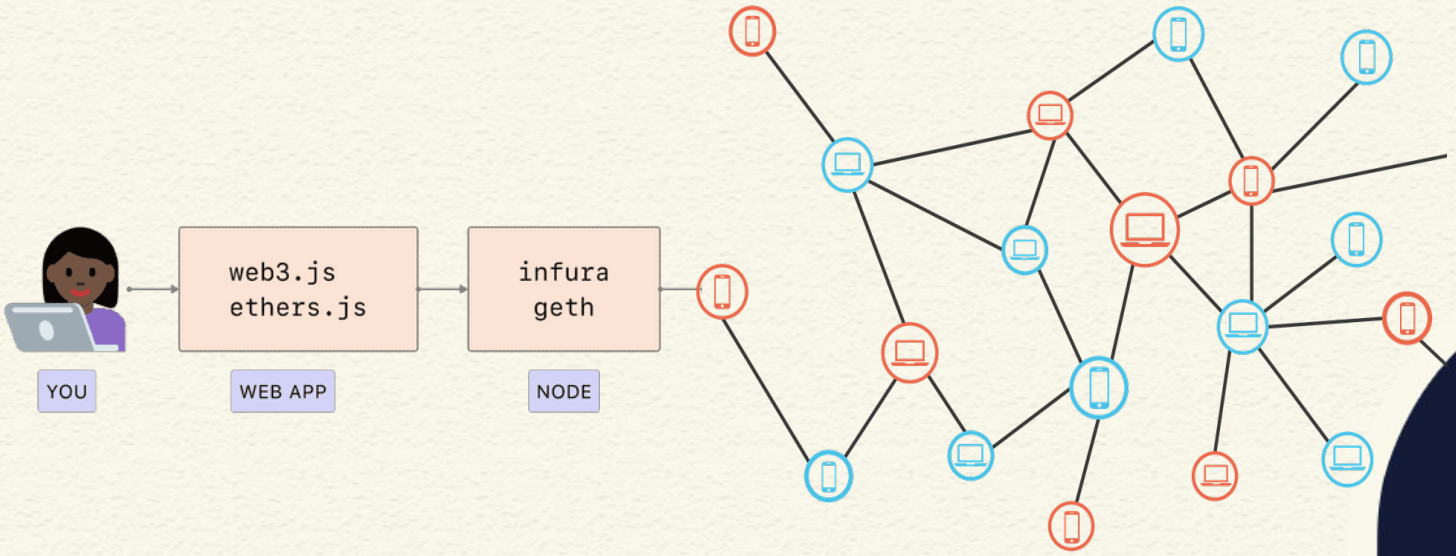
- **No central server**
- Independent nodes connected in a P2P network.

https://ethstats.net/
https://www.ethernodes.org/countries

Interacting with an Ethereum Network



Interacting with an Ethereum Network



Go Ethereum (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 Geth

For Ubuntu

1. Add ethereum repository:

```
sudo add-apt-repository -y
ppa:ethereum/ethereum
```

2. 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

geth account new

```
~/nptel geth account new
INFO [10-27|01:53:10.892] Maximum peer count                       ETH=50 LES=0 total=50
INFO [10-27|01:53:10.893] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no su
ch file or directory"
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:   0x8010319e8e3115EA522Dc2f00ab9bC28A5AbBf80
Path of the secret key file: /home/bishakh/.ethereum/keystore/UTC--2021-10-26T20-23-14.304161080Z--801031
9e8e3115ea522dc2f00ab9bc28a5abbf80

- 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

geth account list

```
~/nptel geth account list
INFO [10-27|01:53:46.704] Maximum peer count                       ETH=50 LES=0 total=50
INFO [10-27|01:53:46.704] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no su
ch file or directory"
INFO [10-27|01:53:46.705] Set global gas cap                       cap=50,000,000
Account #0: {8010319e8e3115ea522dc2f00ab9bc28a5abbf80} keystore:///home/bishakh/.ethereum/keystore/UTC--2
021-10-26T20-23-14.304161080Z--8010319e8e3115ea522dc2f00ab9bc28a5abbf80
```

cat ~/.ethereum/keystore/<key file>

```
~/nptel cat ~/.ethereum/keystore/UTC--2021-10-26T20-23-14.304161080Z--8010319e8e3115ea522dc2f00ab9bc28
a5abbf80
{"address":"8010319e8e3115ea522dc2f00ab9bc28a5abbf80","crypto":{"cipher":"aes-128-ctr","ciphertext":"8adc
e6d4bb5200e47d41115200020de93eb1b4e529777ead7235c769613e748e","cipherparams":{"iv":"d1b38535f133c9cc8d787
2b06417fd4d"},"kdf":"scrypt","kdfparams":{"dklen":32,"n":262144,"p":1,"r":8,"salt":"99dcda6aa950e2c3fb6bf
adbe00ba12a6cf4dd72f5a90116b9587f7388c908a4"},"mac":"1d492f1a132284a38ccada2c6ab0cea697a9529df52489dc8ddf
8ecd3af84628"},"id":"403b8b04-886f-4fc5-b8cf-f12f26ee0565","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:

  - --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 Goerli testnet

geth –goerli --syncmode "light"

```
~ geth --goerli --syncmode "light"
INFO [11-02|12:47:58.049] Starting Geth on Görli testnet...
INFO [11-02|12:47:58.049] Dropping default light client cache      provided=1024 updated=128
INFO [11-02|12:47:58.053] Maximum peer count                       ETH=0 LES=10 total=50
INFO [11-02|12:47:58.053] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
INFO [11-02|12:47:58.053] Set global gas cap                       cap=50,000,000
INFO [11-02|12:47:58.054] Allocated cache and file handles         database=/home/bishakh/.ethereum/goerli/geth/lightchaindata
cache=64.00MiB handles=2048
INFO [11-02|12:47:58.100] Allocated cache and file handles         database=/home/bishakh/.ethereum/goerli/geth/les.client cach
e=16.00MiB handles=16
INFO [11-02|12:47:58.164] Persisted trie from memory database      nodes=361 size=51.17KiB time=4.610792ms gcnodes=0 gcsize=0.0
0B gctime=0s livenodes=1 livesize=0.00B
INFO [11-02|12:47:58.166] Initialised chain configuration          config="{ChainID: 5 Homestead: 0 DAO: <nil> DAOSupport: true
 EIP150: 0 EIP155: 0 EIP158: 0 Byzantium: 0 Constantinople: 0 Petersburg: 0 Istanbul: 1561651, Muir Glacier: <nil>, Berlin: 446
0644, London: 5062605, Engine: clique}"
INFO [11-02|12:47:58.180] Added trusted checkpoint                 block=5,275,647 hash=b5a666..34b5a5
INFO [11-02|12:47:58.180] Loaded most recent local header          number=5,777,531 hash=fcf2f2..274e52 td=8,469,843 age=42s
INFO [11-02|12:47:58.181] Configured checkpoint oracle             address=0x18CA0E045F0D772a851BC7e48357Bcaab0a0795D signers=5
 threshold=2
```

Copy account to testnetwork

cp ~/.ethereum/keystore/UTC--2021-10-26T20-23-14.304161080Z--8010319e8e3115ea522dc2f00ab9bc28a5abbf80 ~/.ethereum/rinkeby/keystore/

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 --goerli --syncmode "light" --http

```
~ geth --goerli --syncmode "light" --http
INFO [11-02|12:49:39.576] Starting Geth on Görli testnet...
INFO [11-02|12:49:39.576] Dropping default light client cache      provided=1024 updated=128
INFO [11-02|12:49:39.578] Maximum peer count                       ETH=0 LES=10 total=50
INFO [11-02|12:49:39.578] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
INFO [11-02|12:49:39.579] Set global gas cap                       cap=50,000,000
INFO [11-02|12:49:39.579] Allocated cache and file handles         database=/home/bishakh/.ethereum/goerli/geth/lightchaindata cache=64.00MiB han
dles=2048
INFO [11-02|12:49:39.614] Allocated cache and file handles         database=/home/bishakh/.ethereum/goerli/geth/les.client cache=16.00MiB handles
=16
INFO [11-02|12:49:39.665] Persisted trie from memory database      nodes=361 size=51.17KiB time=1.009893ms gcnodes=0 gcsize=0.00B gctime=0s liven
odes=1 livesize=0.00B
INFO [11-02|12:49:39.666] Initialised chain configuration          config="{ChainID: 5 Homestead: 0 DAO: <nil> DAOSupport: true EIP150: 0 EIP155:
 0 EIP158: 0 Byzantium: 0 Constantinople: 0 Petersburg: 0 Istanbul: 1561651, Muir Glacier: <nil>, Berlin: 4460644, London: 5062605, Engine: cliqu
e}"
```

```
WARN [11-02|12:49:39.668] Unclean shutdown detected                booted=2021-09-08T13:04:34+0530 age=1mo3w3d
WARN [11-02|12:49:39.668] Unclean shutdown detected                booted=2021-10-27T03:48:19+0530 age=6d9h1m
INFO [11-02|12:49:39.668] Starting peer-to-peer node               instance=Geth/v1.10.8-stable-26675454/linux-amd64/go1.16.4
INFO [11-02|12:49:39.720] New local node record                    seq=69 id=c0aaffb0082e603d ip=127.0.0.1 udp=30303 tcp=30303
INFO [11-02|12:49:39.720] Started P2P networking                   self=enode://c768b3ba0b37788ddf7ebcaa4c7677df0e3a40a9c5a92a28361ca8df726f47544
928724209513b7ac96ef51eb8ae85d9b8dda8b7f6b9936b0ff9f7bf0f7b5d7f0127 0.0.0.1:30303
INFO [11-02|12:49:39.721] IPC endpoint opened                      url=/home/bishakh/.ethereum/goerli/geth.ipc
INFO [11-02|12:49:39.721] HTTP server started                      endpoint=127.0.0.1:8545 prefix= cors= vhosts=localhost
WARN [11-02|12:49:39.721] Light client mode is an experimental feature
INFO [11-02|12:49:49.883] Block synchronisation started
```

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:

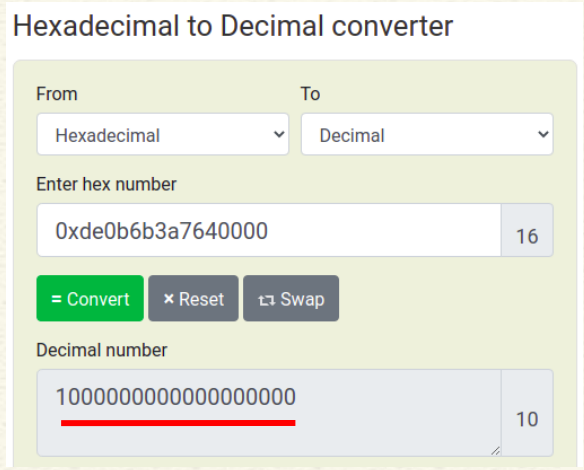
POST request using curl:

```
curl -X POST http://localhost:8545 \
-H "Content-Type: application/json" \
--data '{"jsonrpc":"2.0", "method":"eth_getBalance", "params":
["0x5342722156fcd4b0e0b140c4fb9cd63dcea347f4","latest"],
"id":1}'
```

```
~/nptel curl -X POST http://localhost:8545 \
    -H "Content-Type: application/json" \
   --data '{"jsonrpc":"2.0", "method":"eth_getBalance", "params":["0x5342722156fcd4b0e0b140c4fb9cd63dcea3
47f4","latest"], "id":1}'
{"jsonrpc":"2.0","id":1,"result":"0xde0b6b3a7640000"}
```

Ethereum Units

Unit	Wei Value	Wei
wei	1 wei	1
Kwei (babbage)	1e3 wei	1,000
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	1e18 wei	1,000,000,000,000,000,000



Conclusion

- Ethereum
  - Permissionless, smart contract support
  - Go ethereum
  - Main network, test networks
  - Accounts
- Query using RPC over HTTP


Thank
you