

# Ethereum Node

- Concepts
- Working
- Management API
- ...

## Discount Coupon Links to UDEMY courses:



<https://www.udemy.com/hyperledger/?couponCode=DKHLF1099>



<https://www.udemy.com/ethereum-dapp/?couponCode=DKETH1099>



<https://www.udemy.com/rest-api/?couponCode=DKRST1099>



mentoring, seeking Blockchain part time work, project guidance, advice ... ..

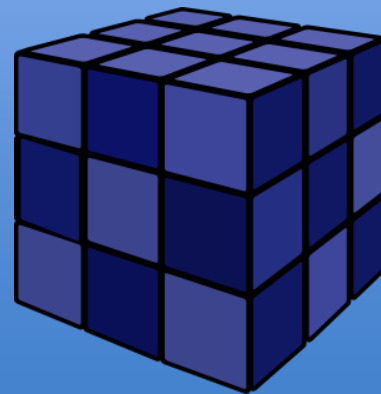
<http://www.bcmentors.com>

raj@acloudfan.com



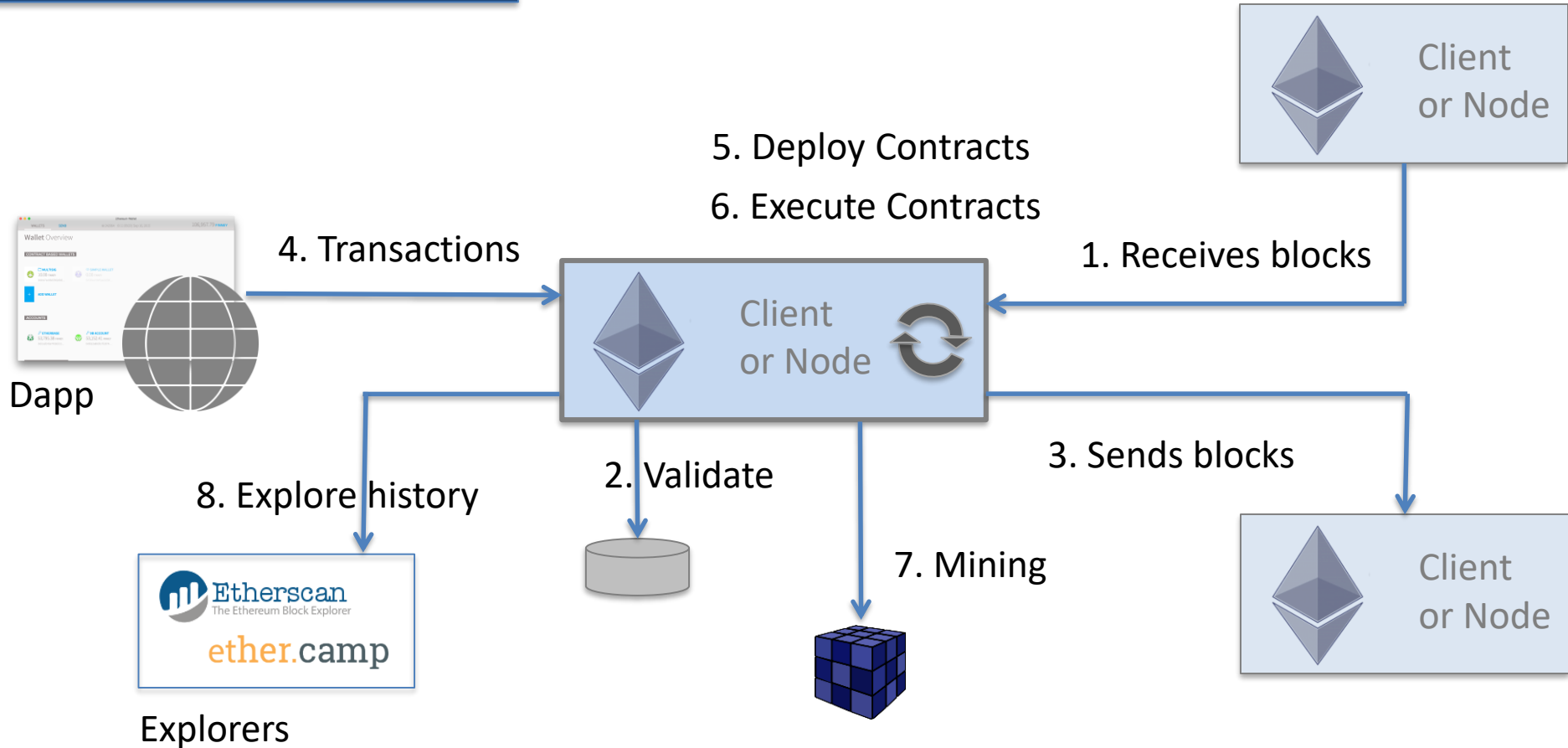
@acloudfan

<http://ACloudFan.com>



This deck is part of a online course on [“Ethereum: Design and Development of Decentralized Apps.”](#)

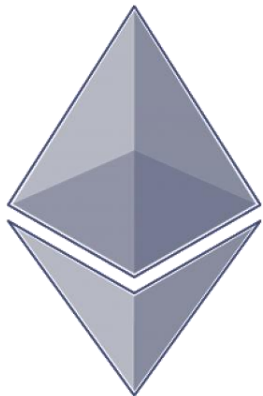
# Ethereum Client



# Client Implementation

Yellow paper = specs

<http://gavwood.com/paper.pdf>



C++

eth



pyethapp

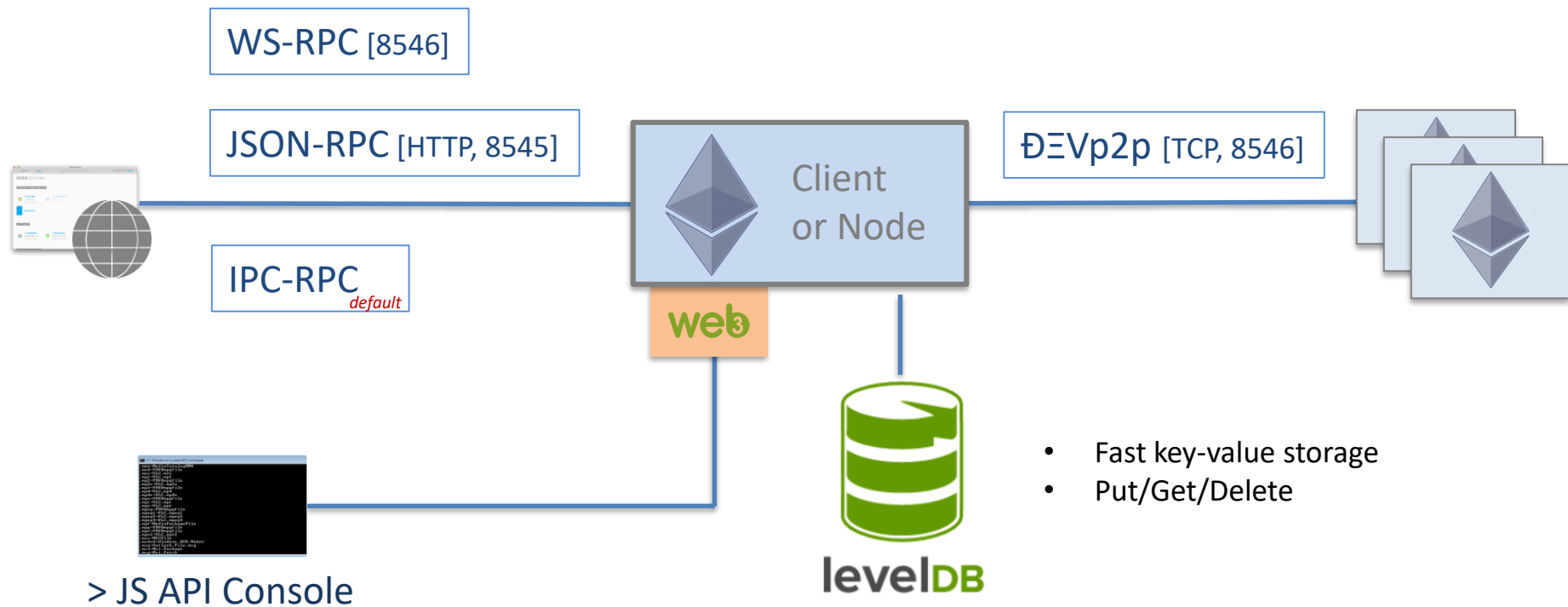


geth

Third party

- ethereumj
- ethereumjs-vm
- ethereumh
- ruby-ethereum
- node-blockchain-server

# Working of the client

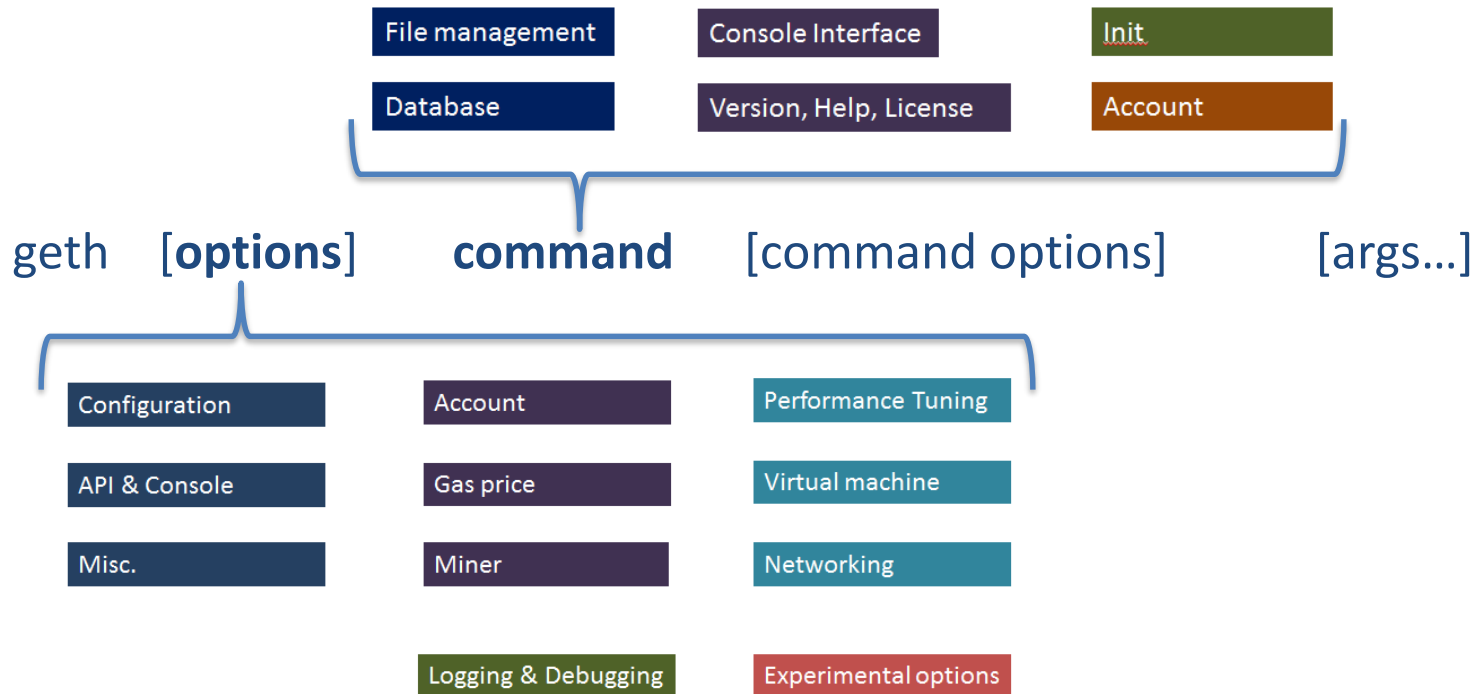


# Geth

**Geth** is a Ethereum Client in written in  ***golang***

> **geth** [options] command [command options] [args...]

# Geth CLI : Usage





<https://github.com/acloudfan/Blockchain-Course-Geth.git>

geth [options] command [command options] [args...]



Version, Help, License

version                      Prints the version of the geth client

help                         Prints the help (same as `-help`)

license                      Prints the license information



# Geth CLI : [options]

geth [**options**] command [command options] [args...]

Configuration

API & Console

Misc.

--help -h

Help on usage

--solc "value"

Solidity compiler command to be used

--netrestrict "value"

Restricts network communication to given IP networks

geth [options] command [command options] [args...]

## Logging & Debugging

--verbosity *value*

0=silent, 1=error, 2=warn, 3=info, 4=core, 5=debug,  
6=detail      default=3

--fakepow, trace ...

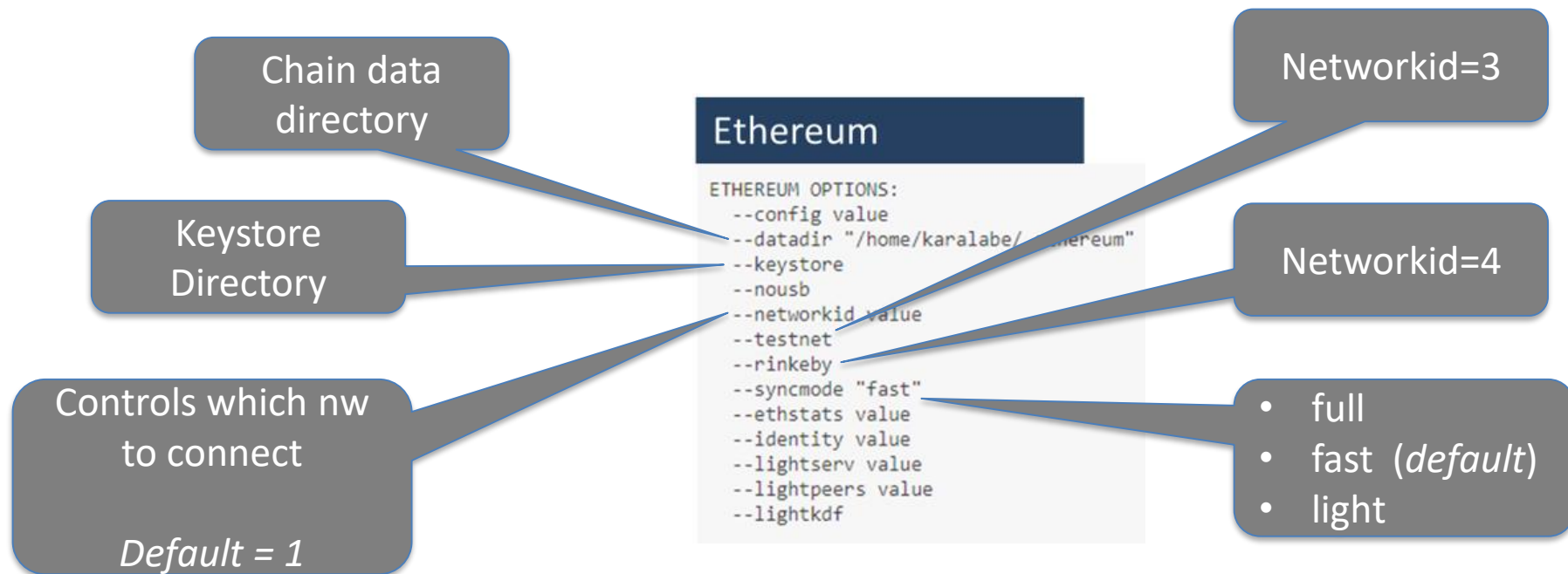
Allows debugging & testing

**PS:** *There are multiple options which are mostly geared towards the ethereum developers*

# Configuring Geth

## Logging & Debugging

--verbosity *value* = 0 Silent To 6 Detailed



# Light Sync



- **EXPERIMENTAL** as of February 2018

```
>_ geth --syncmode light
```



- Works **MOSTLY** with Rinkeby (Syncs < 15 min)
- Somewhat **Flaky** with Ropsten

For all experimentation & Learning : feel free to **Light Synchrony**


Course Videos developed with all 3 Sync Modes

geth [options] command [command options] [args...]

Configuration

API & Console

Misc.



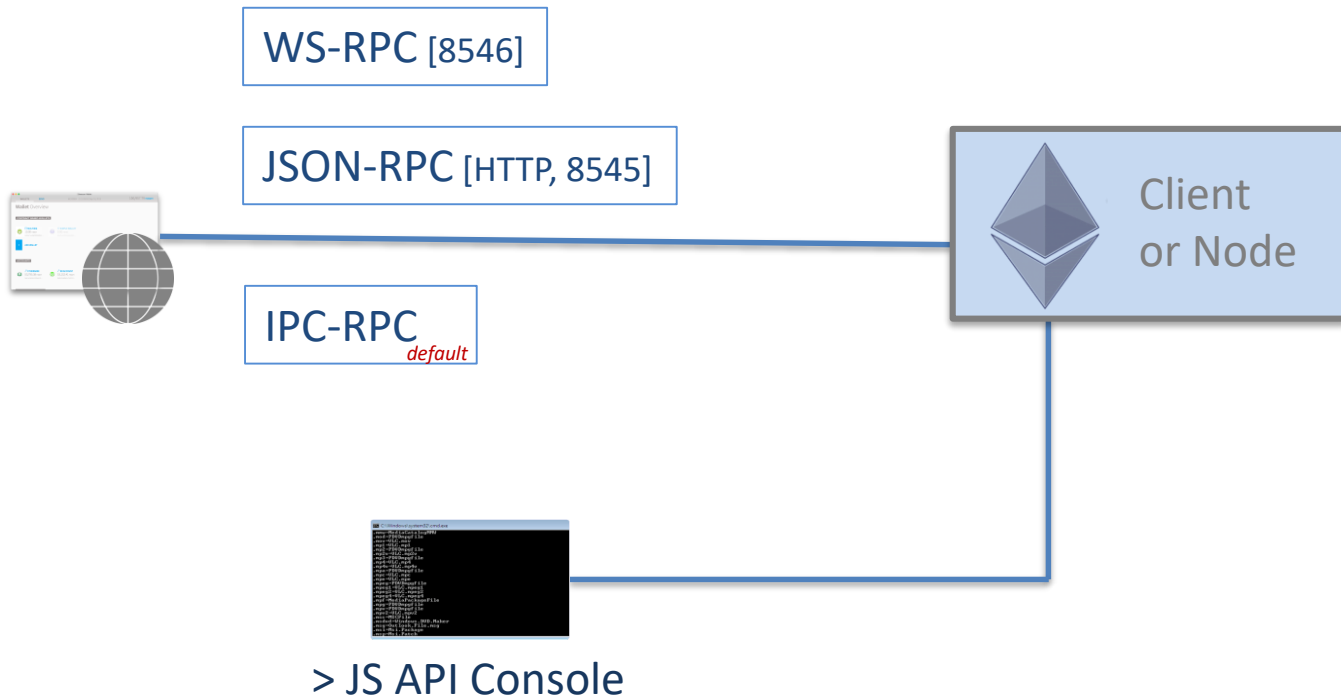
--identity	<i>"MyTestNode"</i>	Custom identity
--datadir	<i>"/data-dir"</i>	Directory for writing the chain data
--keystore	<i>"/key-dir"</i>	Keystore location
--networkid	<i>"value"</i>	Default=1 Live, Testnet=3, Private = User assigned
--testnet		Connect the node to the testnet (ROPSNET)
--fast	(use with --cache)	Fast synching through state downloads
--dev		Developer mode; private network; multiple debug flags

# Private Dev Network

- Isolated network with 1 node
  - Fast transactions
  - Extremely low storage needs
- Suitable for experimentation & learning
- Mining needs to be ON for: (memory intensive)
  - Trying out transactions
  - Getting ethers

# How it --dev work?

> geth --dev ....other options...

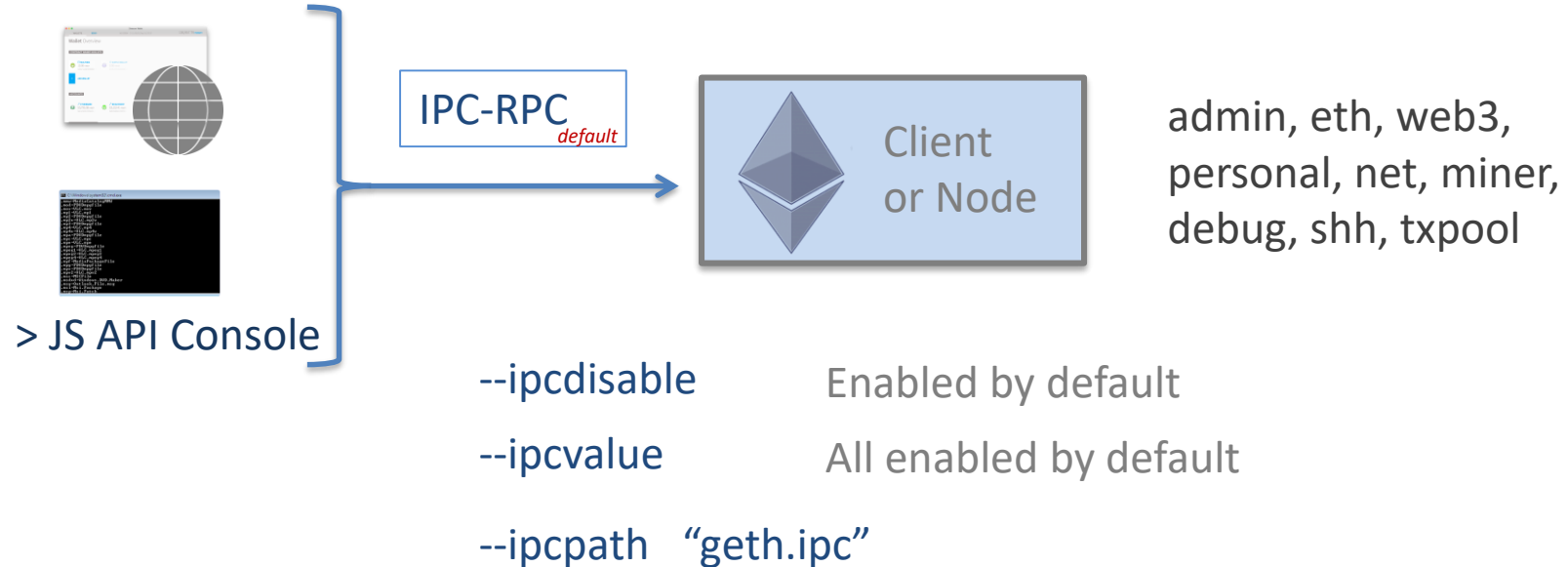


geth [options] command [command options] [args...]

Configuration

API & Console

Misc.



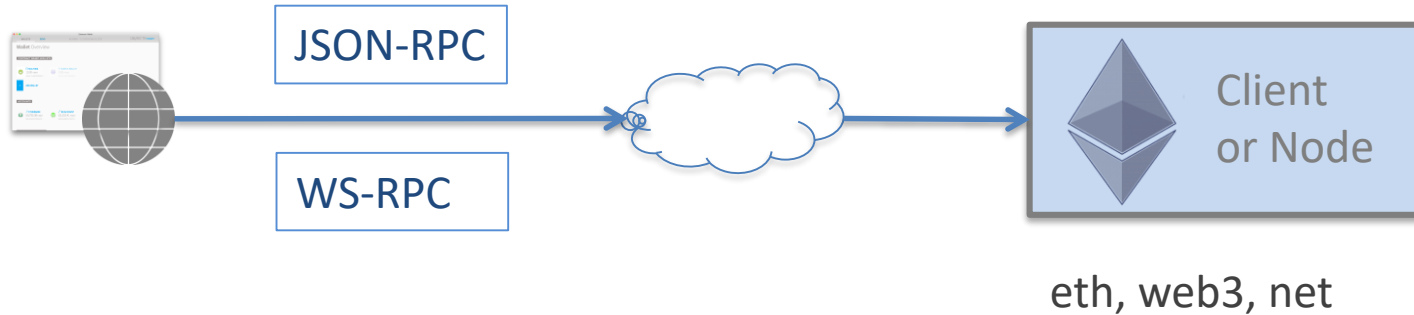


geth [options] command [command options] [args...]

Configuration

API & Console

Misc.



JSON or WS RPC need to be set up for remote apps

geth [options] command [command options] [args...]

Configuration

API & Console

Misc.

JSON-RPC [HTTP, 8545]

--rpc Enable JSON-RPC  
--rpcapi Default=eth, web3, net  
--rpcaddr "host"  
--rpcport "port"  
--rpccorsdomain "domains , . , " or "\*"

WS-RPC [TCP, 8546]

--ws Enable WS-RPC  
--wsapi Default=eth, web3, net  
--wsaddr "host"  
--wsport "port"  
--wsorigin "origin" to accept request

# Javascript Runtime

- Geth implements a Javascript Runtime Environment
  - Execute the javascript code
  - Invoke the **Management API**  
*web3, eth, personal, admin, miner, personal ...*
- Supports interactive as well as non-interactive mode

geth [options] command [command options] [args...]

Configuration

API & Console

Misc.

--preload "files,.."      Used with **console** command

--jspath "loadscript"      JS script path

--exec "JS statement"      Used with **console** & **attach** commands

```
> geth --exec "personal.listAccounts" console
```

geth [options] command [command options] [args...]



Console Interface

console

Start an interactive console session

attach "pipe address" Interactive session by connecting with a already running client

attach "http://....:8545"

attach "ws://....:8546"

geth [options] command [command options] [args...]



## Account

- |         |               |                     |
|---------|---------------|---------------------|
| account | <b>new</b>    | Create an account   |
| account | <b>list</b>   | Lists the accounts  |
| account | <b>update</b> | Update the password |
| account | <b>import</b> | Import key file     |

# Account Keyfiles

- On account creation a key file gets added to the geth client
  - `<DATADIR>/keystore`
  - `UTC--<created_at UTC ISO8601>--<address hex>`
- Ok to copy keystore file to a different instance of client
  - Switching clients e.g., Mac to PC \*or\* Geth to Wallet

└─ keystore

UTC--2017-01-14T11-57-39.276261600Z--8b4180f38fbb42f42c4d6b0cbccd3e01b8a5f521

# Keyfiles Practices

- Good practices:
  - Periodically backup all your key files
  - Use strong passwords; DO NOT forget the passwords.
  - If managing passwords in a file make sure they are hidden and accessible to only you



geth [options] command [command options] [args...]

Account

Gas price

Miner

--unlock      *“Comma seperated list of address/index”*      Account locked by default

--password    *“password file”*      Used for non-interactive password input

geth [options] command [command options] [args...]

Account

Gas price

Miner

--mine

Enable mining

--minerthreads

Number of threads for mining (default=8)

--etherbase

Public address for mining rewards (default=#0)

--gasprice "value"

Minimal gas accepted for mining a txn (default=2E10)

--extradata "somedata"

Block extra data set by miner (default=client miner)

--targetgaslimit "value"

Target gas floor for the blocks to mine (max gas spent in block)

geth [options] command [command options] [args...]

Account

Gas price

Miner

--gpomin, --gpomax    *"value"*

Min/Max suggested gas price

--gpobasedown, --gpobaseup    *"value"*

Suggested gas step down/up ratio

--gpofull            *"value"*

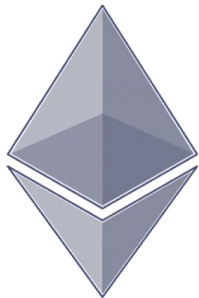
Full block threshold for gas calculation

--gpobasecf        *"value"*

Suggested price correction factor

<https://github.com/ethereum/go-ethereum/wiki/Gas-Price-Oracle>

# Geth Javascript API



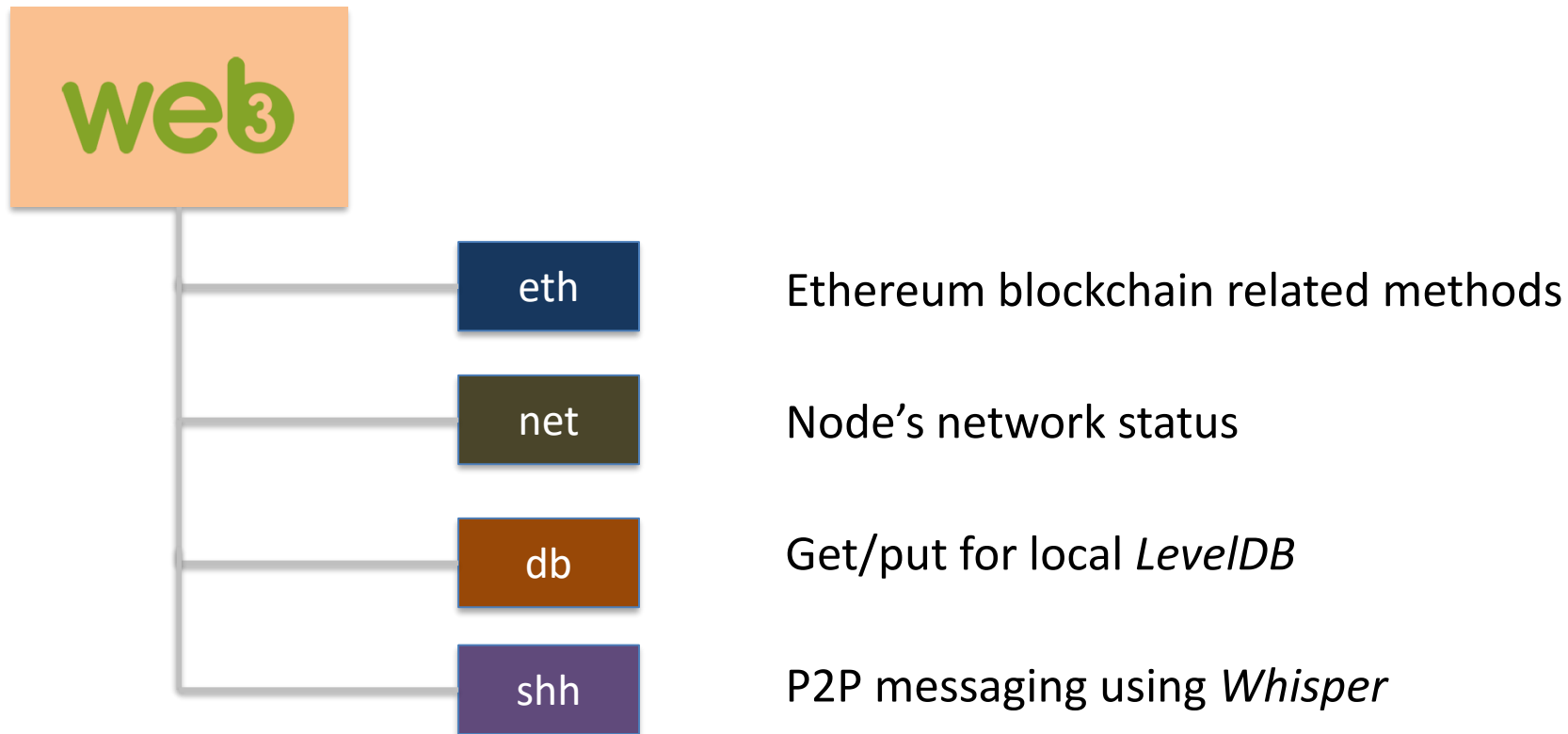
- Supports web3 Dapp API

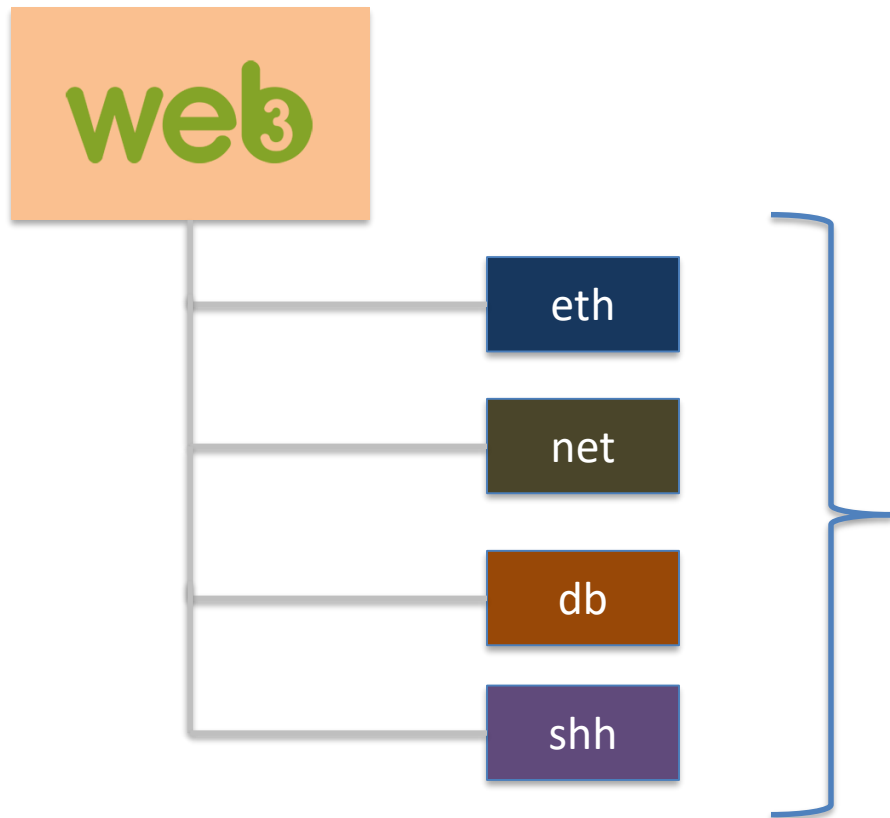
geth

- Go-Ethereum supports additional API
  - Available over JSON-RPC

# geth console API

<https://github.com/ethereum/wiki/wiki/JavaScript-API>





- API available as shorthand  
eth, net, db, ssh

# Management API

<https://github.com/ethereum/go-ethereum/wiki/Management-API>  
<https://github.com/ethereum/go-ethereum/wiki/JavaScript-Consoles>

admin

Node management

personal

Account management

miner

Miner management

txpool

Transaction pool

debug

Node debugging



## Node Information

nodeInfo

```
> admin.nodeInfo
```

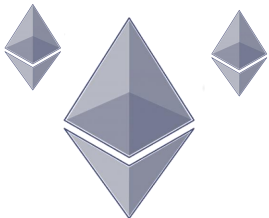
datadir

## Node Peers

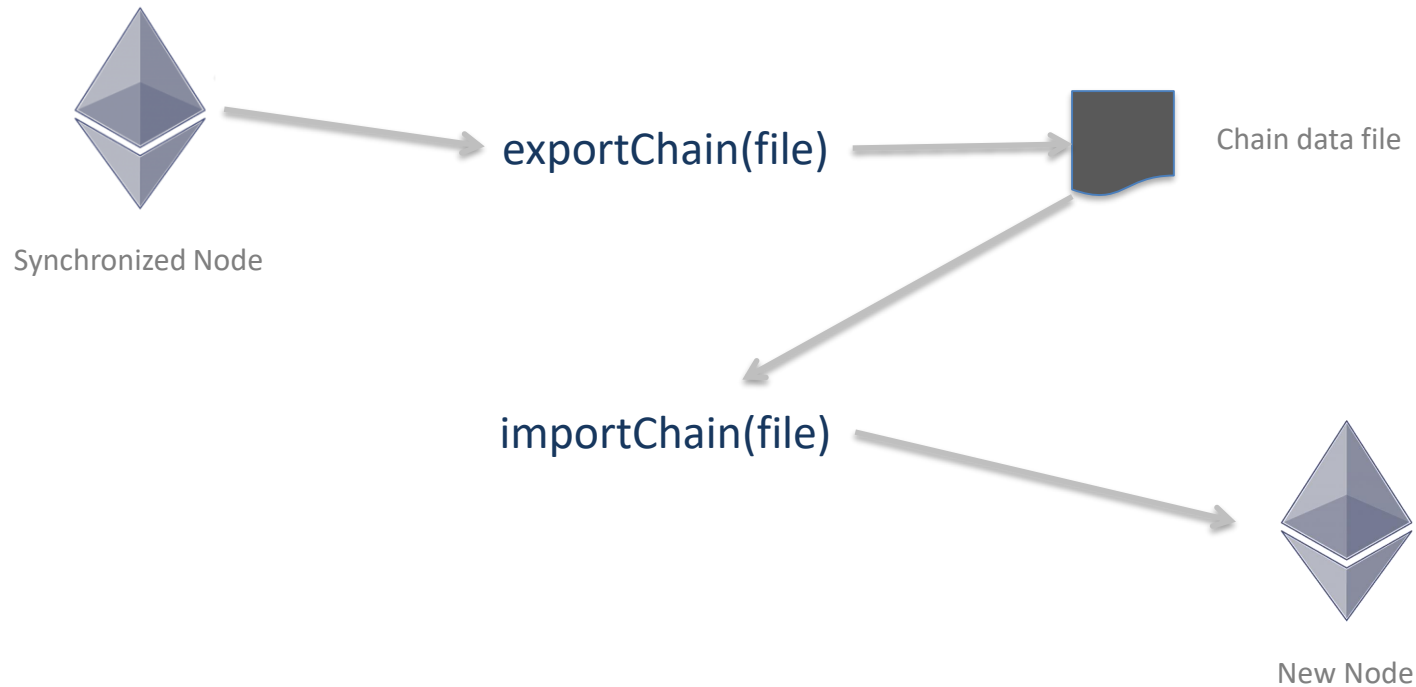
peers

```
> admin.peers
```

addPeers(url)







JSON-RPC

startRPC(...)

```
> admin.startRPC( "localhost", 8545, "web3")
```

stopRPC()

WS-RPC

startWS(...)

stopWS()

admin

## Chain contracts management

Compiler

`setSolc('path to Solidity compiler')`

`getContractInfo(address)`

`saveInfo(contractInfo, file)`

## personal

- Manage accounts

```
personal.newAccount("pass")
```

```
personal.listAccounts
```

```
personal.unlockAccount()
```

```
personal.lockAccount()
```

```
personal.sendTransaction(txn_object, "paraphrase")
```

Txn =

```
{  
  from: "0x317bacb51afd26b62b5fd42060b7f817b6111a76",  
  to: "0xf360a452fb80c4ca149c89a1a6a4d5785692c92e",  
  value: "10000000000000000000"  
}
```

## geth CLI Miner options

`--mine``--minerthreads``--etherbase``--gasprice "value"``--extradata "somedata"``miner.start(num_threads)``miner.stop()``miner.setEtherbase(addr)``miner.setGasPrice(num)``miner.setExtra(data)`

- Nonce = Running transaction counter per account
- A transaction ends up in *queued* pool if nonce is skipped
- A valid transaction ends up in the *pending* pool & gets mined

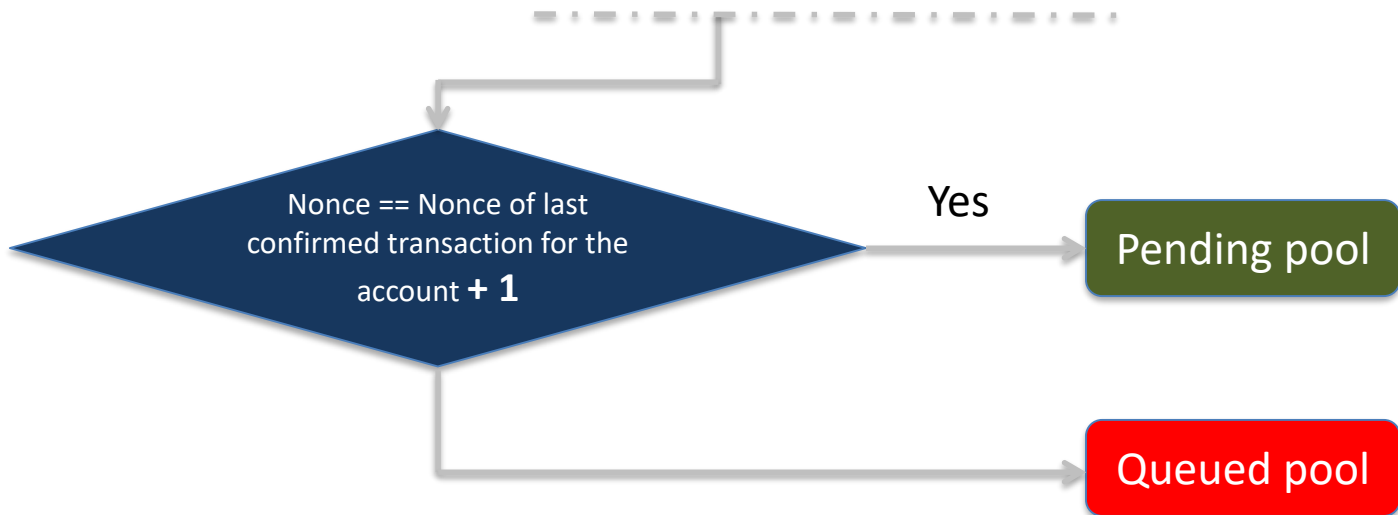
`txpool.status`

`txpool.inspect`

`txpool.content`

## Account nonce

- Transaction counter in each account
  - Every transaction under the account has a unique nonce
  - The transaction object in **personal.sendTransaction** can specify nonce



# debug

## Information

- verbosity
- blocks
- transaction

## Performance

- GC Stats
- Mem Stats
- CPU profile

## Go Code

- Code level debugging