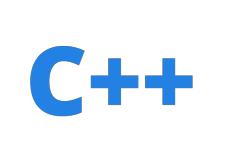
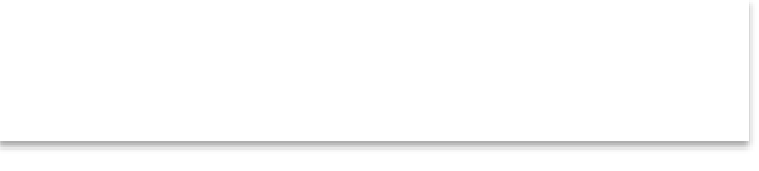
Ethereum Client

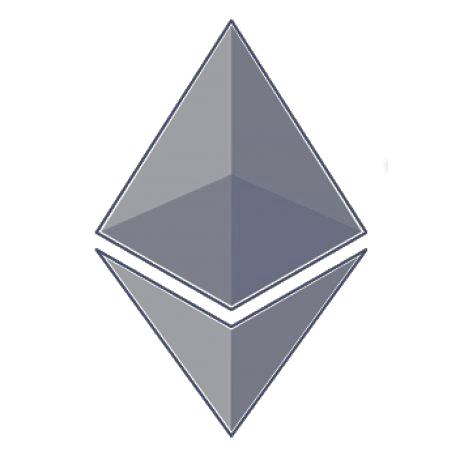
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | Client |  |
|  |  |  |  |  |  |  |  | or Node |  |
|  |  |  |  | 5. Deploy Contracts | |  |  |  |  |
|  |  |  |  | 6. Execute Contracts | |  |  |  |  |
|  | 4. Transactions | | |  |  | 1. Receives blocks | |  |  |
| Client |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| Dapp | |  |  | or Node |  |  |  |  |  |
|  | |  |  |  |  |  |  |
|  | |  |  | 3. Sends blocks | | |  |
|  |  |  |  | 2. Validate | |  |
|  | 8. Explore history | | |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  | 7. Mining | |  |  | Client |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | or Node |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

Explorers

|  |  |  |  |
| --- | --- | --- | --- |
| Client Implementation | Yellow paper = specs | <http://gavwood.com/paper.pdf> |  |
|  |  |
|  |  |  |  |



eth



pyethapp



geth

Third party

•

ethereum

**j**

•

ethereum**js**-vm

•

ethereum**h**

•

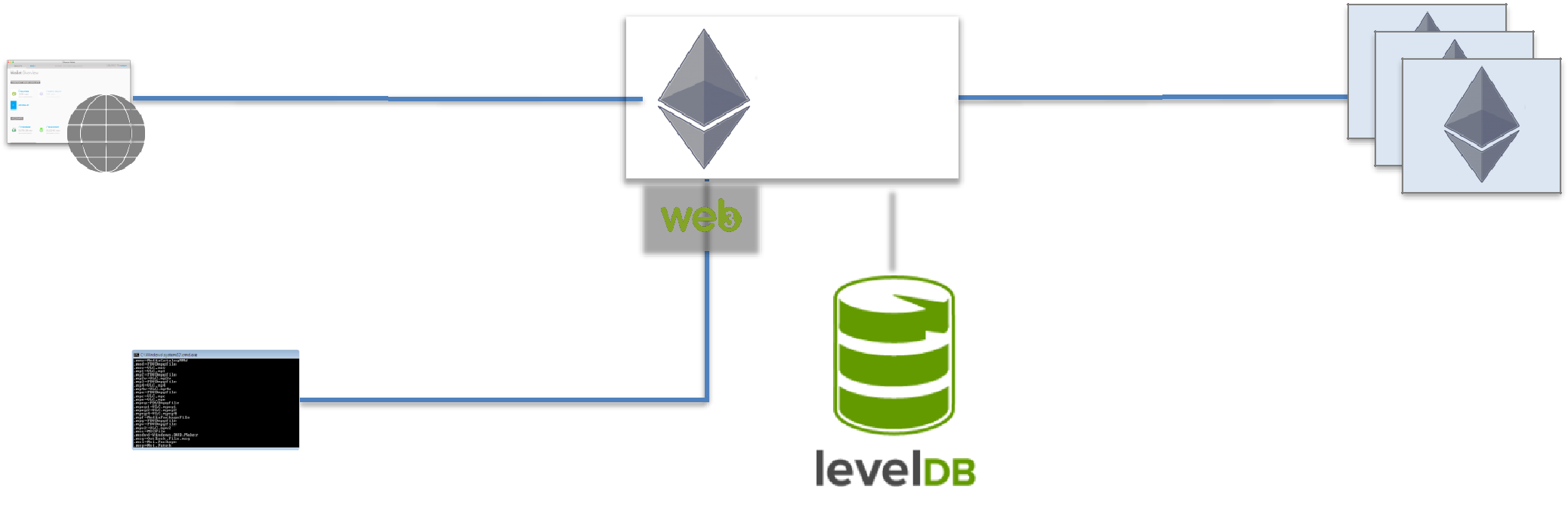
ruby-ethereum

•

node-blockchain-server

Working of the client

WS-RPC [8546]



|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | JSON-RPC [HTTP, 8545] | |  |  |  |  |  |  | ÐΞVp2p [TCP, 8546] |  |
|  |  |  |  | Client | |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  | or Node | |  | |  |
|  | IPC-RPC |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | *default* |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

• Fast key-value storage • Put/Get/Delete

> JS API Console

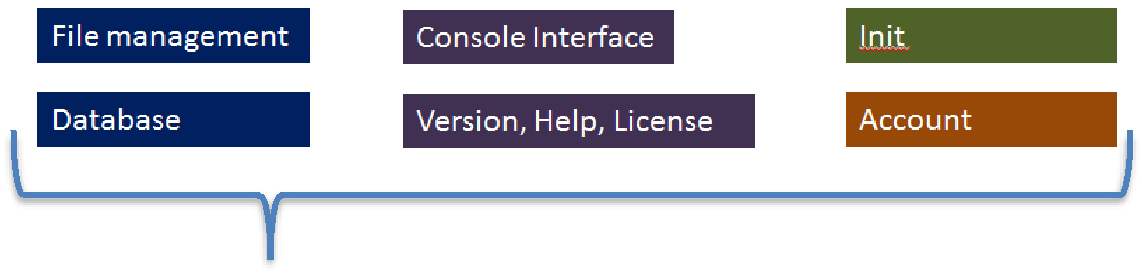
Geth



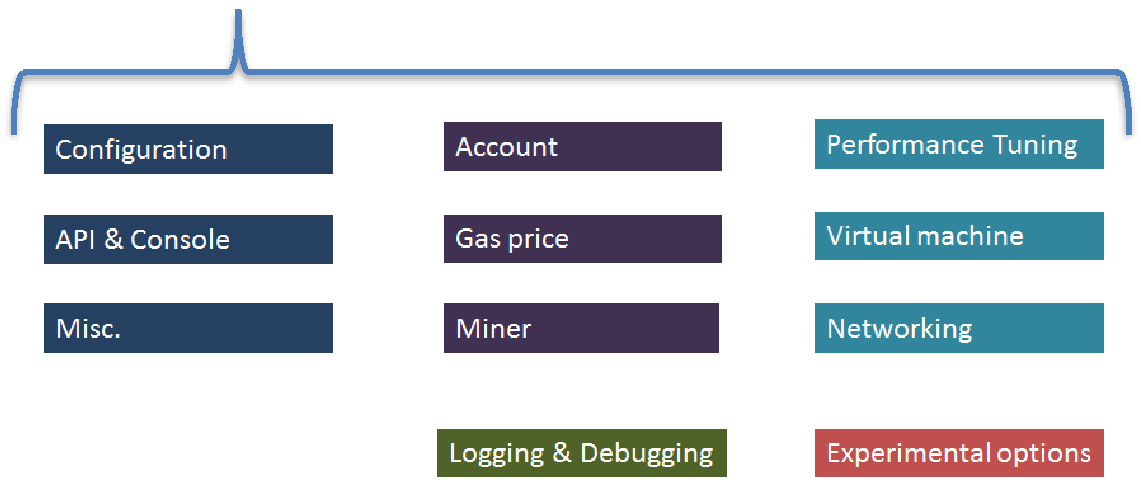
**Geth** is a Ethereum Client in written in

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

Geth CLI : Usage



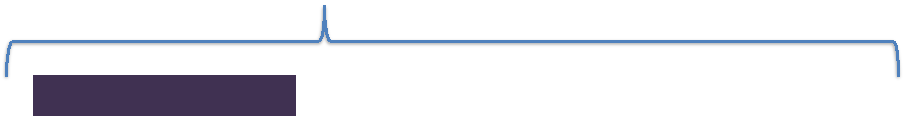
geth [**options**] **command** [command options] [args…]





<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

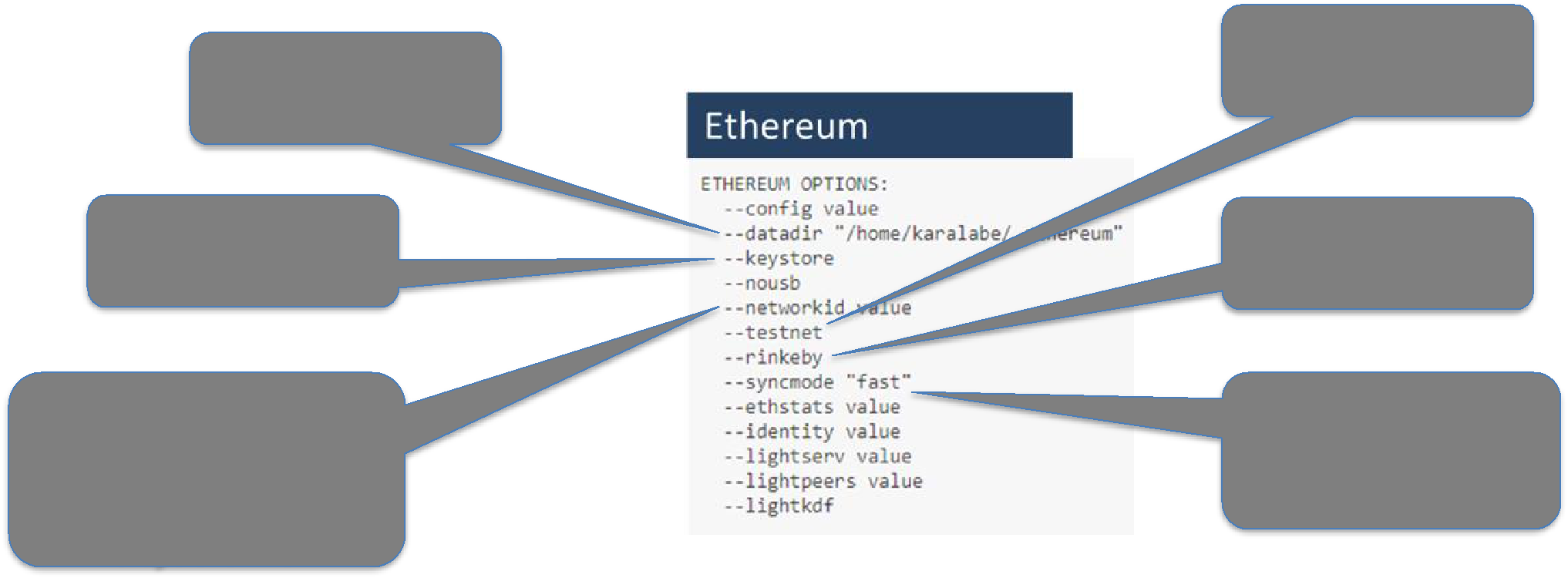
--fakepow, trace … Allows debugging & testing

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



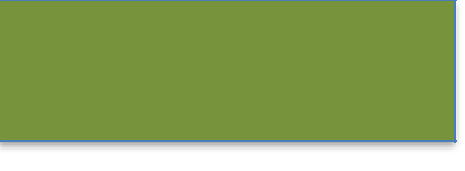
Configuring Geth

|  |  |
| --- | --- |
| Logging & Debugging | --verbosity *value* = 0 Silent To 6 Detailed |



|  |  |  |  |
| --- | --- | --- | --- |
| Chain data |  | Networkid=3 |  |
| directory |  |  |  |
| Keystore |  | Networkid=4 |  |
| Directory |  |  |
|  |  |  |
| Controls which nw | • | full |  |
| to connect | • | fast (*default*) |  |
|  | • | light |  |

*Default = 1*



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 Synch**

Course Videos developed with all 3 Sync Modes

geth [**options**] command [command options] [args…]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Configuration |  | API & Console |  | Misc. |
|  |  |  |  |  |



--identity *“MyTestNode”*

--datadir *“/data-dir”*

--keystore *“/key-dir”* --networkid *“value”* --testnet

--fast (use with --cache)

--dev

Custom identity

Directory for writing the chain data

Keystore location

Default=1 Live, Testnet=3, Private = User assigned

Connect the node to the testnet (ROPSNET)

Fast synching through state downloads

Developer mode; private network; multiple debug flags

Private Dev Network

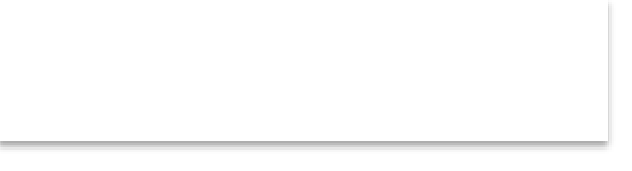
* Isolated network with 1 node

• Fast transactions • Extremely low storage needs

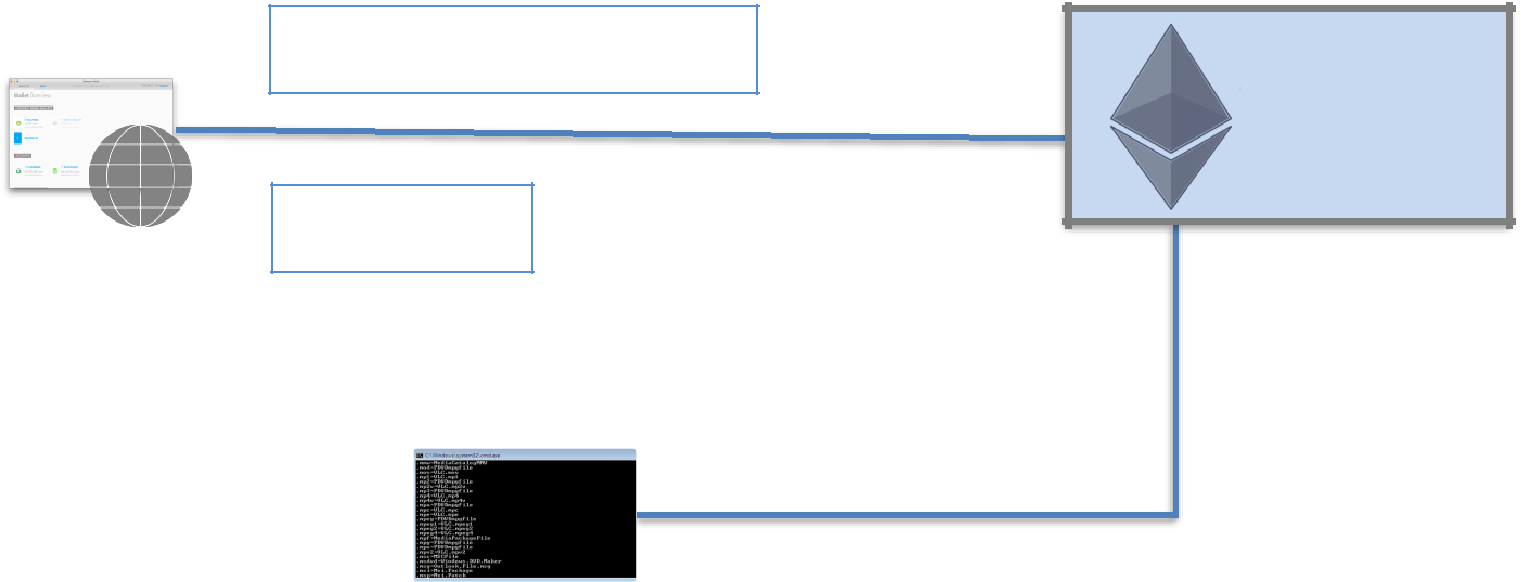
* Suitable for experimentation & learning
  + Mining needs to be ON for: (memory intensive)

• Trying out transactions • Getting ethers

|  |  |
| --- | --- |
| How it --dev work? | > geth **--dev** ….other options... |
|  |  |



WS-RPC [8546]



JSON-RPC [HTTP, 8545]

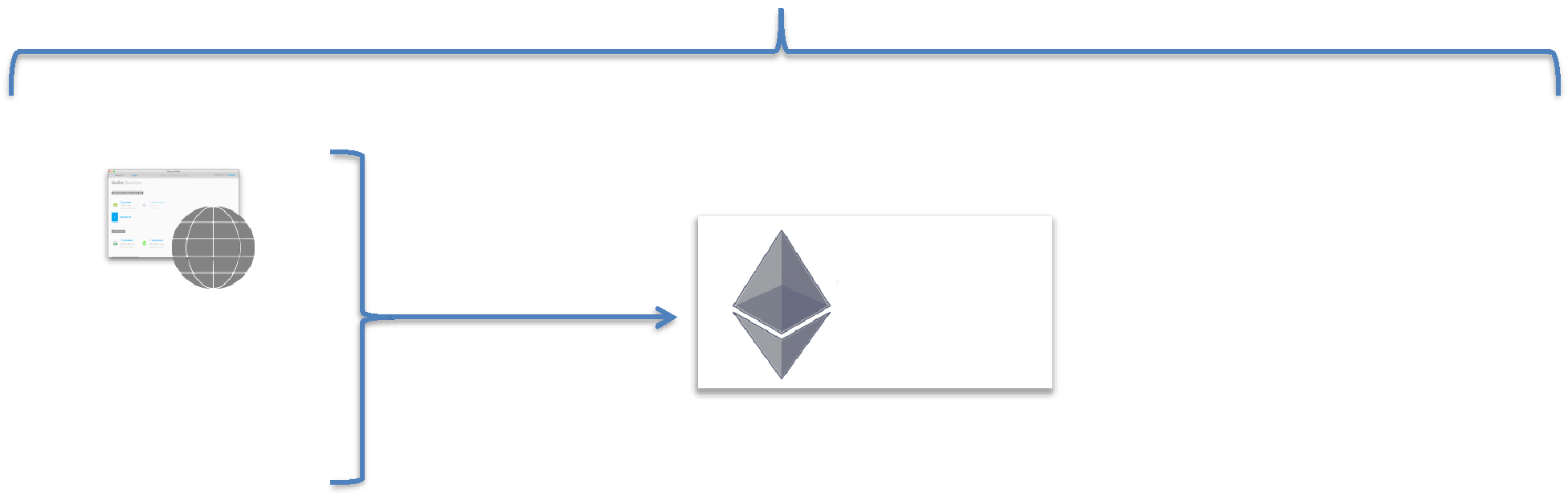
Client or Node

IPC-RPC

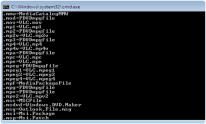
*default*

> JS API Console

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| geth [**options**] | | | command [command options] | | | | | [args…] |
|  |  |  |  |  |  |  |  |  |
| Configuration | | |  |  | API & Console |  |  | Misc. |
|  |  |  |  |  |  |  |  |  |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IPC-RPC |  |  | admin, eth, web3, |  |
|  | Client |  |
| *default* |  |  |  |
|  |  |  |  |
|  |  | or Node | personal, net, miner, |  |
|  |  | debug, shh, txpool |  |
|  |  |  |  |
|  |  |  |  |  |

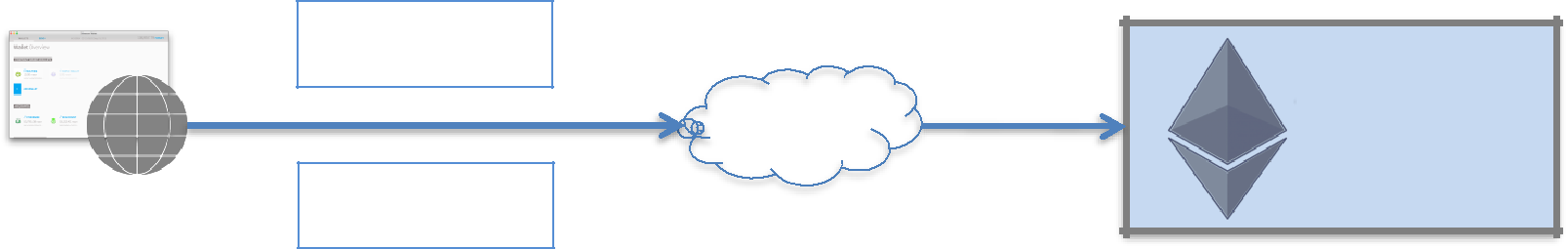


> JS API Console

|  |  |
| --- | --- |
| --ipcdisable | Enabled by default |
| --ipcvalue | All enabled by default |

--ipcpath “geth.ipc”

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| geth [**options**] | | | command [command options] | | | | | [args…] |
|  |  |  |  |  |  |  |  |  |
| Configuration | | |  |  | API & Console |  |  | Misc. |
|  |  |  |  |  |  |  |  |  |



JSON-RPC

Client

or Node

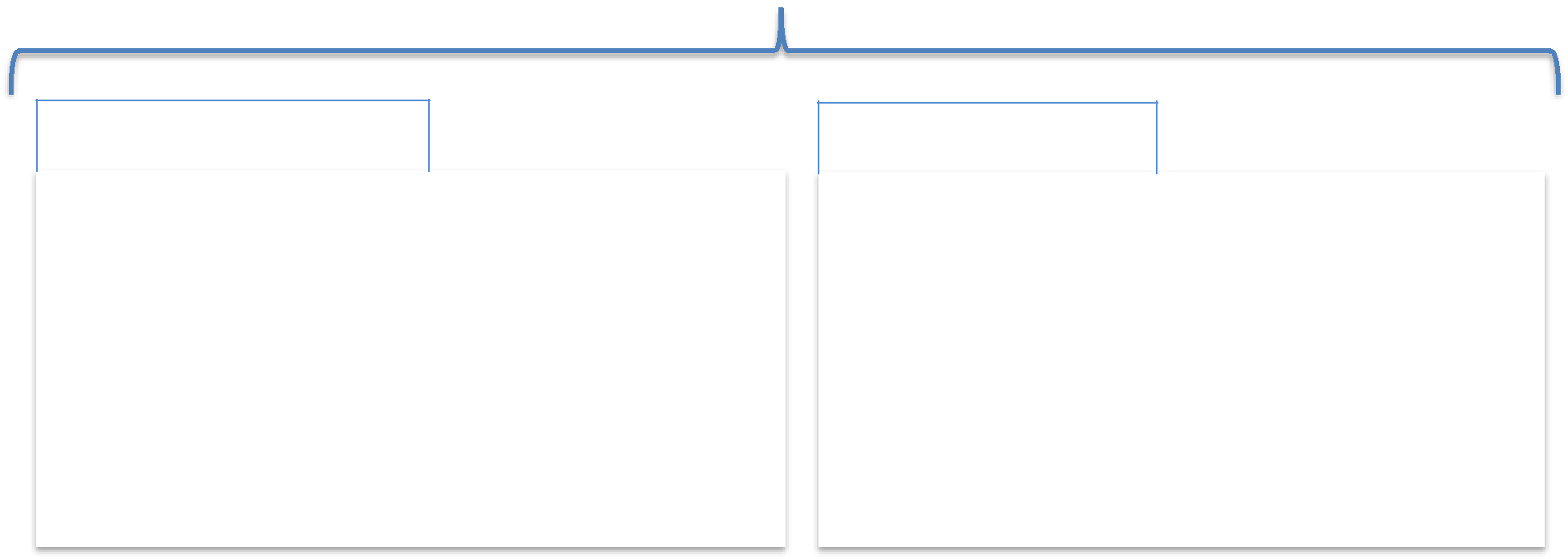
WS-RPC

eth, web3, net



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**]

WS-RPC [TCP, **8546**]

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| --rpc | Enable JSON-RPC | |  | --ws | Enable WS-RPC | |  |
| --rpcapi | Default=eth, web3, net | |  | --wsapi | Default=eth, web3, net | |  |
|  |  |  |  |  |
| --rpcaddr | “host” | |  | --wsaddr | “host” |  |  |
| --rpcport | “port” | |  |  |  |
|  | --wsport | “port” |  |  |
|  |  |  |  |  |  |
| --rpccorsdomain “domains , . ,” or “\*” | | |  | --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 | **new** | Create an account |
| account | **list** | Lists the accounts |
| account | **update** | Update the password |
| account | **import** | 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

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

--minerthreads

--etherbase

Enable mining

Number of threads for mining (default=8)

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 |
|  |  |



--gpomin, --gpomax *“value”*

--gpobasedown, --gpobaseup “value”

--gpofull “value”

--gpobasecf “value”

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

Miner

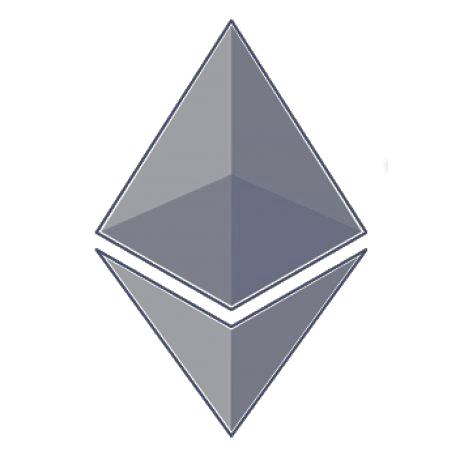
Min/Max suggested gas price

Suggested gas step down/up ratio

Full block threshold for gas calculation

Suggested price correction factor

Geth Javascript API



* Supports web3 Ðapp API

geth • Go-Ethereum supports additional API

* Available over JSON-RPC

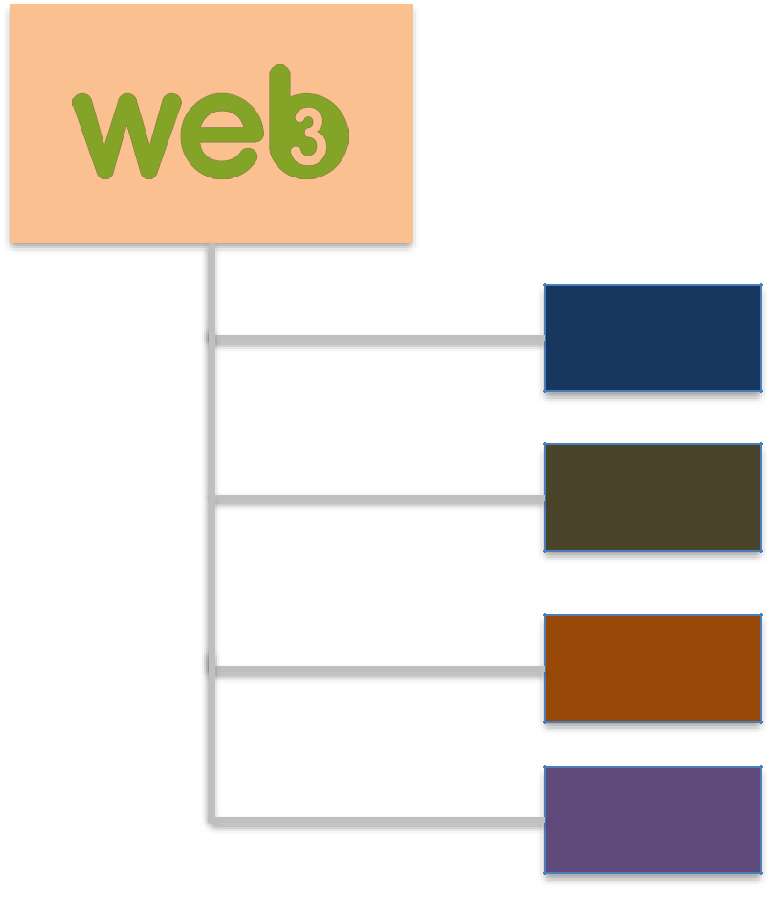
*geth* console API



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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  | eth | Ethereum blockchain related methods |  |
|  |  |  |  |
|  |  |  |  | Node’s network status |  |
|  |  |  |  |  |
|  |  |  | net |  |
|  |  |  |  |
|  |  |  |  | Get/put for local *LevelDB* |  |
|  |  |  |  |  |
|  |  |  | db |  |
|  |  |  |  |
|  |  |  |  | P2P messaging using *Whisper* |  |
|  |  |  |  |  |
|  |  |  | shh |  |
|  |  |  |  |
|  |  |  |  |  |  |

*geth* console API



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



eth

net

• API available as shorthand

|  |  |  |
| --- | --- | --- |
| db | eth, net, db, shh |  |
|  |  |

shh

*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 |
|  |  |

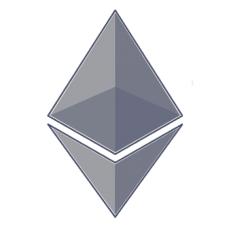


|  |  |
| --- | --- |
| debug | Node debugging |
|  |  |



|  |  |
| --- | --- |
| admin | Node Management |
|  |  |

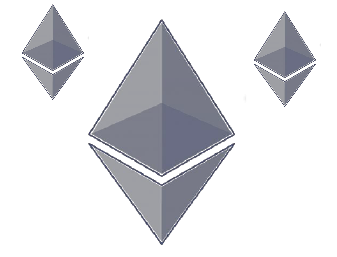
**Node Information**



nodeInfo

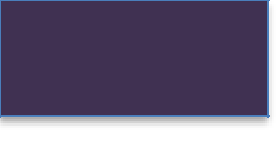
datadir

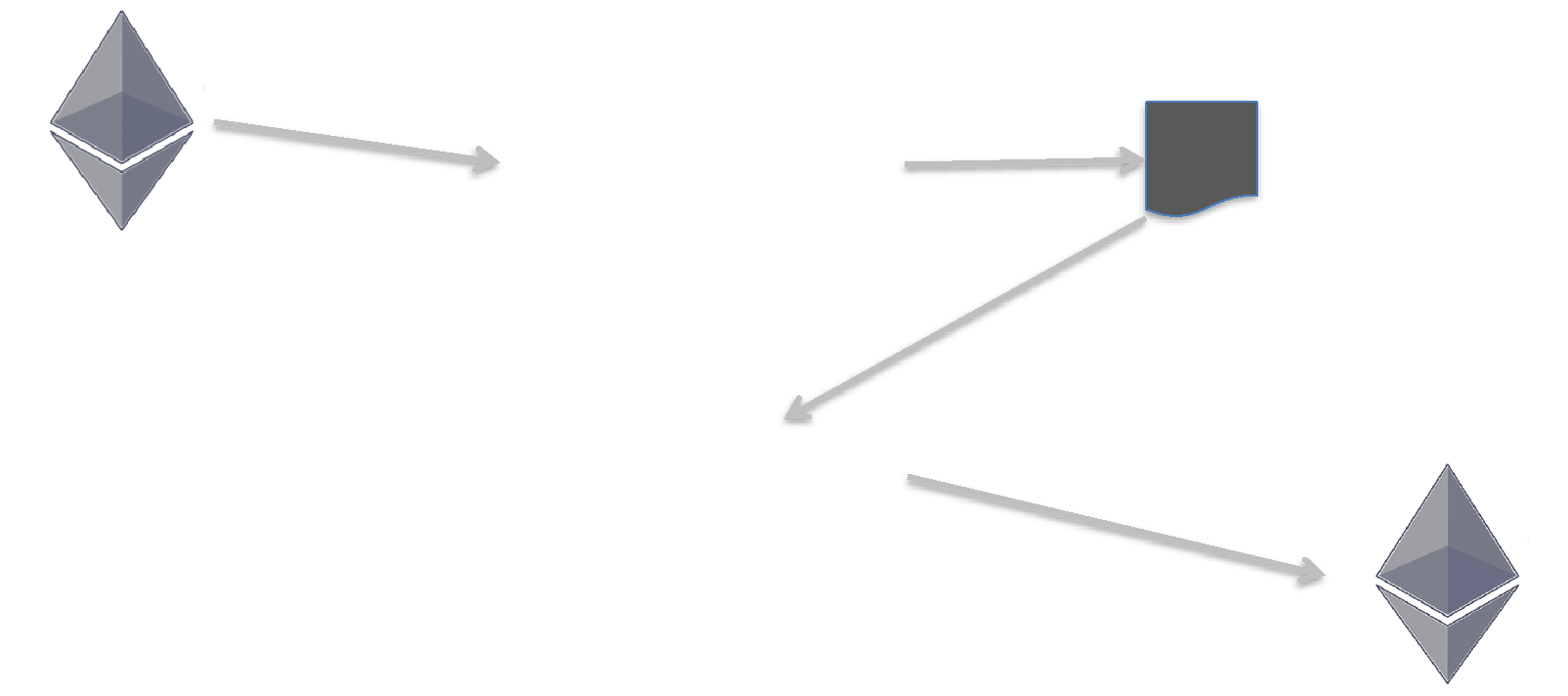
**Node Peers**



peers

addPeers(url)

admin Chain data management



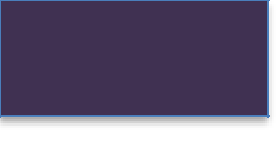
|  |  |  |
| --- | --- | --- |
| exportChain(file) | Chain data file |  |
|  |  |

Synchronized Node

importChain(file)

New Node

|  |  |  |
| --- | --- | --- |
| admin | Manage RPC interface |  |
|  |  |



JSON-RPC



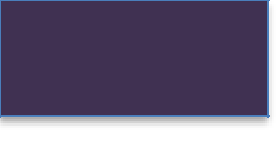
startRPC(…)

stopRPC()

WS-RPC

startWS(…)

stopWS()

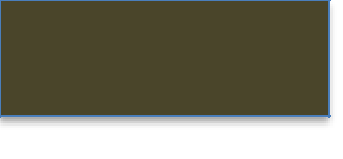
admin Chain contracts management

Compiler

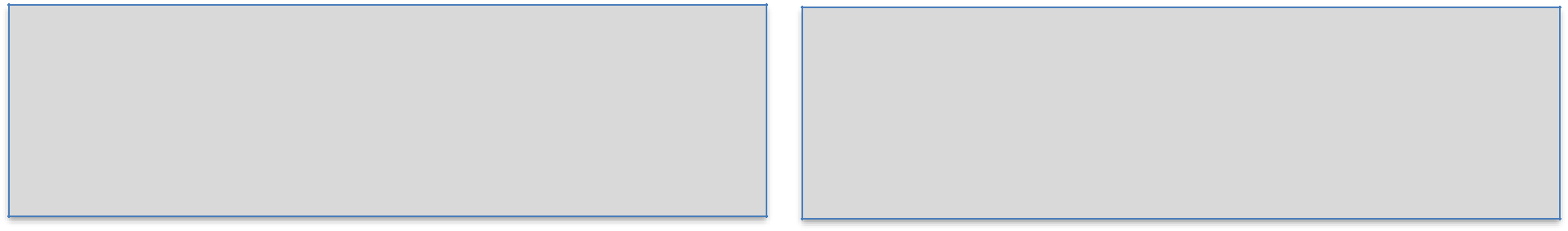
setSolc(‘path to Solidity compiler’)

getContractInfo(address)

saveInfo(contractInfo, file)

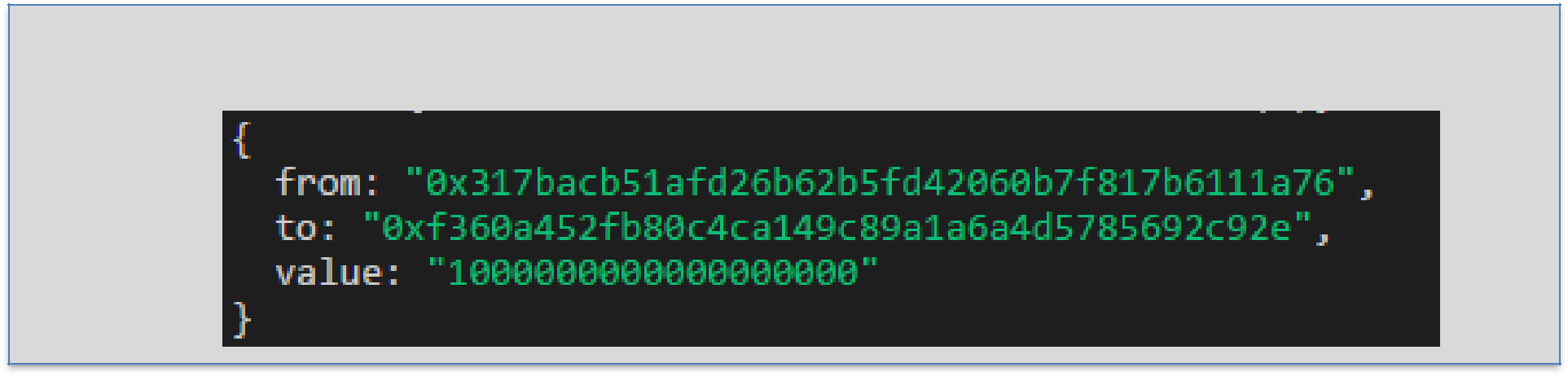
personal

* Manage accounts



**personal**.newAccount(“pass”) **personal.**unlockAccount()

**personal**.listAccounts **personal.**lockAccount()



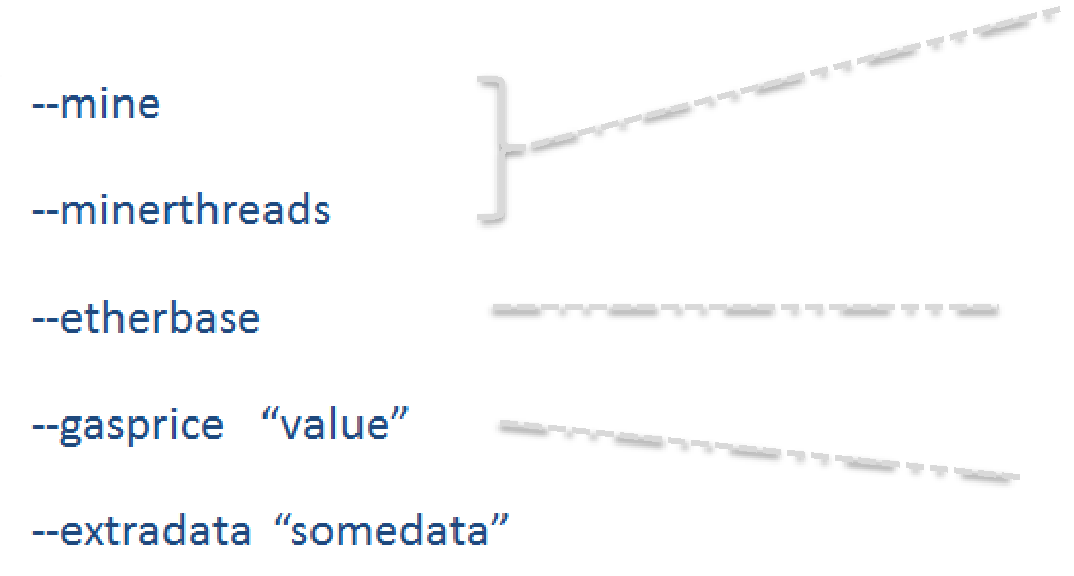
**personal.**sendTransaction(**txn**\_**object**, “paraphrase”)

Txn =

|  |  |
| --- | --- |
| miner | Miner management |
|  |  |



|  |  |
| --- | --- |
| geth CLI Miner options | **miner**.start(num\_threads) |
|  | **miner**.stop() |



**miner**.setEtherbase(addr)

**miner**.setGasPrice(num)

**miner**.setExtra(data)

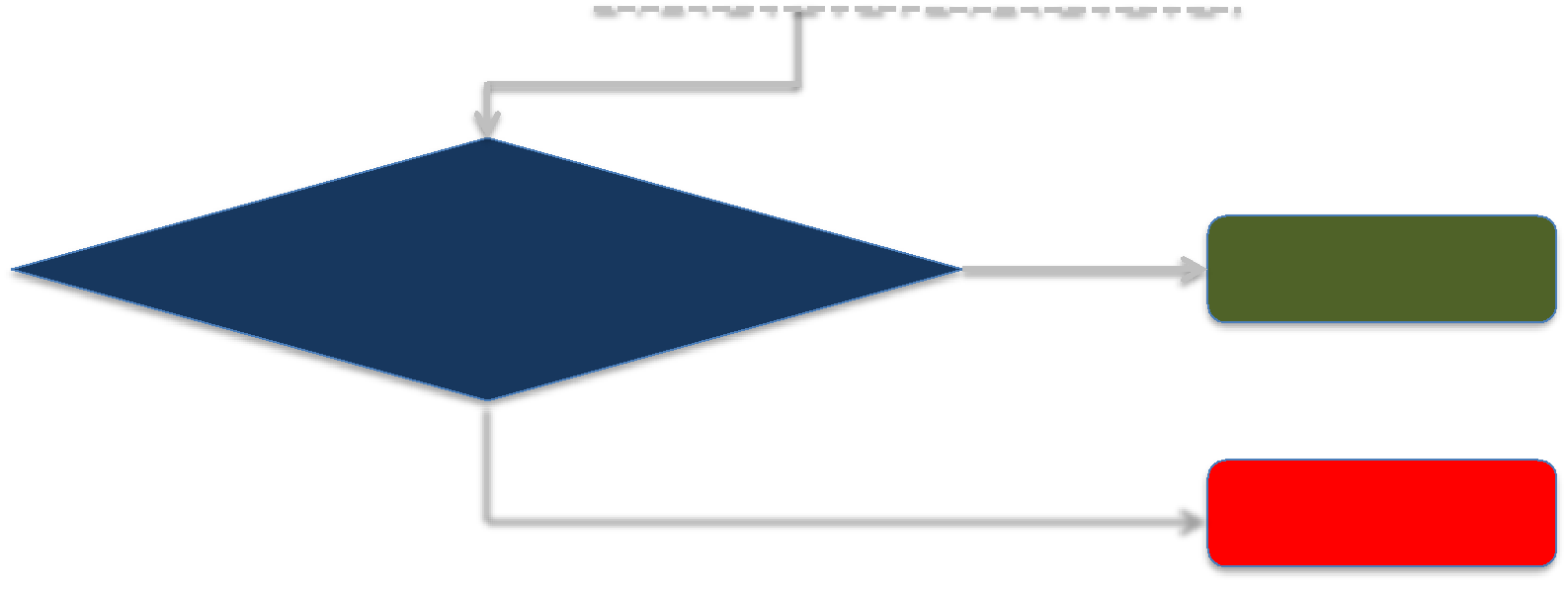
txpool

* 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



Nonce == Nonce of last

confirmed transaction for the

account **+ 1**

Yes

Pending pool

 Queued pool

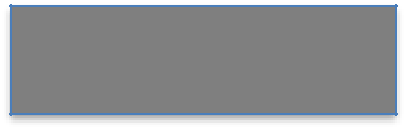
debug



|  |  |  |  |
| --- | --- | --- | --- |
| Information | • verbosity | • blocks | • transaction |
|  |  |  |  |



|  |  |  |
| --- | --- | --- |
| Performance | • GC Stats | • Mem Stats• CPU profile |
|  |  |  |



Go Code

* Code level debugging