Ethereum Client

Ethereum Client

- An 'Ethereum client' is just a term. It refers to any node able to parse and verify the blockchain, its smart contracts and everything related.
- It also allows you/provides interfaces to create transactions and mine blocks which is the key for any blockchain interaction.
- However, it can be very confusing for end-users, because there is no universal “Ethereum Installer” for them to use. So they can make use of various Ethereum clients. Various Ethereum clients are listed below.

Various Ethereum clients

Client	Language	Developers
go-ethereum	Go	Ethereum Foundation
parity	Rust	Ethcore
Aleth (cpp-ethereum)	C++	Ethereum Foundation
pyethapp	Python	Ethereum Foundation
ethereumjs-lib	Javacript	Ethereum Foundation
Ethereum(J)	Java	ether camp
ruby-ethereum	Ruby	Jan Xie
ethereumH	Haskell	BlockApp

Go-ethereum(Geth)

- Geth is a command-line tool written in Go which acts as a node or miners on an Ethereum network.
- Geth acts as an Ethereum client to connect to public and test networks.
- Geth can also be used to run and configure your own private network.
- Geth can be installed on Windows, Linux, and Mac as well.

Go-ethereum(Geth)

By installing and running geth, you can take part in the Ethereum live network and:

- mine real ether
- transfer funds between addresses
- create contracts and send transactions
- explore block history

Go-ethereum(Geth)

- Geth uses JSON RPC protocol.
- It defines the specification for remote procedure calls with payload encoded in JSON format.
- Geth allows connectivity to JSON RPC using the following three different protocols:

1. **Inter Process Communication (IPC)**: This protocol is used for inter process communication generally used within the same computer.
2. **Remote Procedure Calls (RPC):** This protocol is used for inter process communication across computers. This is generally based on TCP and HTTP protocol.
3. **Web Sockets (WS):** This protocol is used to connect to Geth using sockets.

Go-ethereum(Geth)

There are many commands, switches, and options for configuring Geth, which include the following:

1. Configuring IPC, RPC, and WS protocols
2. Configuring network types to connect- private, Ropsten, and Rinkeby
3. Mining options
4. Console and API
5. Networking
6. Debugging and logging

Go-ethereum(Geth)

The following are the network IDs used for connecting to the following different networks:

- **Chain ID 0:** represents Olympic Ethereum Public pre release testnetwork
- **Chain ID 1:** represents a Homestead public network
- **Chain ID 2:** represents Morden (not used anymore)
- **Chain ID 3:** represents Ropsten
- **Chain ID 4:** represents Rinkeby

Other Known Ethereum Network

- **Chain ID 61:** Classic, the (un)forked public Ethereum Classic main network.
- **Chain ID 2**: Expanse, an alternative Ethereum implementation.
- **Chain ID 2**: Morden, the public Ethereum testnet, now Ethereum Classic testnet.
- **Chain ID 5**: Guerli, the public Geth/Parity PoA testnet.
- **Chain ID 8**: Ubiq, the public Gubiq main network with flux difficulty.
- **Chain ID 77**: Sokol, the public POA Network testnet.

Other Known Ethereum Network (Cntnd...)

- Chain ID 99: Core, the public POA Network main network
- Chain ID 100: xDai, the public MakerDAO/POA Network main network
- Chain ID 401697: Tobalaba, the public Energy Web Foundation testnet
- Chain ID 7762959: Musicoin, the music blockchain
- Chain ID 61717561: Aquachain, ASIC resistant chain

- Chain ID [Other]: Could indicate that your connected to a local development test network.

Creating a Private chain

Installing from PPA(Ubuntu)

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

THANK YOU