

Chapter 3: Ethereum Clients.

An Ethereum client is a software application that implements the Ethereum specification and communicates over the peer-to-peer network with other Ethereum clients.

Ethereum is defined by a formal specification called the "Yellow Paper". The yellow paper gets updated as more changes are made to the Ethereum blockchain.

Some of Ethereum based networks - Ethereum, Ethereum Classic, Ella, Expanse, Ubiq, Musicoin, etc.

There are 6 main implementations of the Ethereum protocol:

- Parity, written in Rust.
- Geth, written in Go.
- cpp-ethereum, written in C++.
- pyethereum, written in Python.
- Mantis, written in Scala.
- Harmony, written in Java.

Remote client and wallet are almost the same but different, a remote client offers the services of a wallet, but with the addition of web3.js API or any other API.

Main and testnets have their own advantages and disadvantages.

Parity is an implementation of a full-node Ethereum client and DApp browser. Parity requires Rust version 1.27 or greater. Block 2,283,397, mined on September 18, 2016 marks the beginning of DoS attacks.

RPC - Remote Procedure Call, they are encoded in JSON, offered as a HTTP service on port 8545.

A JSON RPC request takes in 4 parameters: jsonrpc => 2.0; method => ""; params => ""; id => "";

Remote clients offer a subset of the functionality of a full client. They do not store the full Ethereum blockchain, so they are faster to set up and require far less data storage.

Examples of remote clients: Mobile wallets, Browser wallets,

Mist was the first Ethereum-enabled browser, built by the Ethereum foundation, and Fabian Vogelsteller is the author of ERC20.