



# 以太坊客户端

2018.10



# 什么是以太坊客户端

- 以太坊客户端是一个软件应用程序，它实现以太坊规范并通过p2p网络与其他以太坊客户端进行通信。如果不同的以太坊客户端符合参考规范和标准化通信协议，则可以进行相互操作。
- 以太坊是一个开源项目，由“黄皮书”正式规范定义。除了各种以太坊改进提案之外，此正式规范还定义了以太坊客户端的标准行为。
- 因为以太坊有明确的正式规范，以太网客户端有了许多独立开发的软件实现，它们之间又可以彼此交互。



# 基于以太坊规范的网络

- 存在各种基于以太坊规范的网络，这些网络基本符合以太坊“黄皮书”中定义的形式规范，但它们之间可能相互也可能不相互操作。
- 这些基于以太坊的网络中有：以太坊，以太坊经典，Ella，Expanse，Ubiq，Musicoin等等。
- 虽然大多数在协议级别兼容，但这些网络通常具有特殊要求，以太坊客户端软件的维护人员、需要进行微小更改、以支持每个网络的功能或属性。



# 以太坊的多种客户端

- go-ethereum ( Go )  
官方推荐，开发使用最多  
地址: <https://github.com/ethereum/go-ethereum>
- parity ( Rust )  
最轻便客户端，在历次以太坊网络攻击中表现卓越  
地址: <https://github.com/ethcore/parity/releases>
- cpp-ethereum (C++)  
地址: <https://github.com/ethereum/cpp-ethereum>
- pyethapp (python)  
地址: <https://github.com/heikoheiko/pyethapp>
- ethereumjs-lib ( javascript )  
地址: <https://github.com/ethereumjs/ethereumjs-lib>
- EthereumJ / Harmony ( Java )  
地址: <https://github.com/ethereum/ethereumj>



# 以太坊全节点

- 全节点是整个主链的一个副本，存储并维护链上的所有数据，并随时验证新区块的合法性。
- 区块链的健康和扩展弹性，取决于具有许多独立操作和地理上分散的全节点。每个全节点都可以帮助其他新节点获取区块数据，并提供所有交易和合约的独立验证。
- 运行全节点将耗费巨大的成本，包括硬件资源和带宽。
- 以太坊开发不需要在实时网络（主网）上运行的全节点。我们可以使用测试网络的节点来代替，也可以用本地私链，或者使用服务商提供的基于云的以太坊客户端；这些几乎都可以执行所有操作。



# 远程客户端和轻节点

- 远程客户端

不存储区块链的本地副本或验证块和交易。这些客户端一般只提供钱包的功能，可以创建和广播交易。远程客户端可用于连接到现有网络，MetaMask 就是一个这样的客户端。

- 轻节点

不保存链上的区块历史数据，只保存区块链当前的状态。轻节点可以对块和交易进行验证。



# 全节点的优缺点

## 优点

- 为以太坊网络的灵活性和抗审查性提供有力支持。
- 权威地验证所有交易。
- 可以直接与公共区块链上的任何合约交互。
- 可以离线查询区块链状态（帐户，合约等）。
- 可以直接把自己的合约部署到公共区块链中。

## 缺点

- 需要巨大的硬件和带宽资源，而且会不断增长。
- 第一次下载往往需要几天才能完全同步。
- 必须及时维护、升级并保持在线状态以同步区块。



# 公共测试网络节点的优缺点

## 优点

- 一个 testnet 节点需要同步和存储更少的数据，大约10GB，具体取决于不同的网络。
- 一个 testnet 节点一般可以在几个小时内完全同步。
- 部署合约或进行交易只需要发送测试以太，可以从“水龙头”免费获得。
- 测试网络是公共区块链，有许多其他用户和合约运行（区别于私链）。

## 缺点

- 测试网络上使用测试以太，它没有价值。因此，无法测试交易对手的安全性，因为没有任何利害关系。
- 测试网络上的测试无法涵盖所有的真实主网特性。例如，交易费用虽然是发送交易所必需的，但由于gas免费，因此 testnet 上往往不会考





# 本地私链的优缺点

## 优点

- 磁盘上几乎没有数据，也不同步别的数据，是一个完全“干净”的环境。
- 无需获取测试以太，你可以任意分配以太，也可以随时自己挖矿获得。
- 没有其他用户，也没有其他合约，没有任何外部干扰。

## 缺点

- 没有其他用户意味与公链的行为不同。发送的交易并不存在空间或交易顺序的竞争。
- 除自己之外没有矿工意味着挖矿更容易预测，因此无法测试公链上发生的某些情况。
- 没有其他合约，意味着你必须部署要测试的所有内容，包括所有的



# 运行全节点的要求

- **最低要求**

- 双核以上CPU
- 硬盘存储可用空间至少80GB
- 如果是SSD，需要4GB 以上 RAM，如果是HDD，至少8GB RAM
- 8 MB/s 下载带宽

- **推荐配置**

- 四核以上的快速CPU
- 16GB 以上 RAM
- 500GB 以上可用空间的快速SSD
- 25+ MB/s 下载带宽



# Geth ( Go-Ethereum )

- Geth是由以太坊基金会积极开发的 Go 语言实现，因此被认为是以太坊客户端的“官方”实现。
- 通常，每个基于以太坊的区块链都有自己的Geth实现。
- 以太坊的 Geth github 仓库链接：

<https://github.com/ethereum/go-ethereum>



# JSON-RPC

- 以太坊客户端提供了 API 和一组远程调用 (RPC) 命令，这些命令被编码为 JSON。这被称为 JSON-RPC API。本质上，JSON-RPC API 就是一个接口，允许我们编写的程序使用以太坊客户端作为网关，访问以太坊网络和链上数据。
- 通常，RPC 接口作为一个 HTTP 服务，端口设定为 8545。出于安全原因，默认情况下，它仅限于接受来自 localhost 的连接。
- 要访问JSON-RPC API，我们可以使用编程语言编写的专用库，例如JavaScript的 web3.js。
- 或者也可以手动构建HTTP请求并发送/接收JSON编码的请求，如：

```
$ curl -X POST -H "Content-Type: application/json" --data \
'{"jsonrpc":"2.0","method":"web3_clientVersion","params":[],"id":1
}' \ http://localhost:8545
```



# Q&A



尚硅谷

