

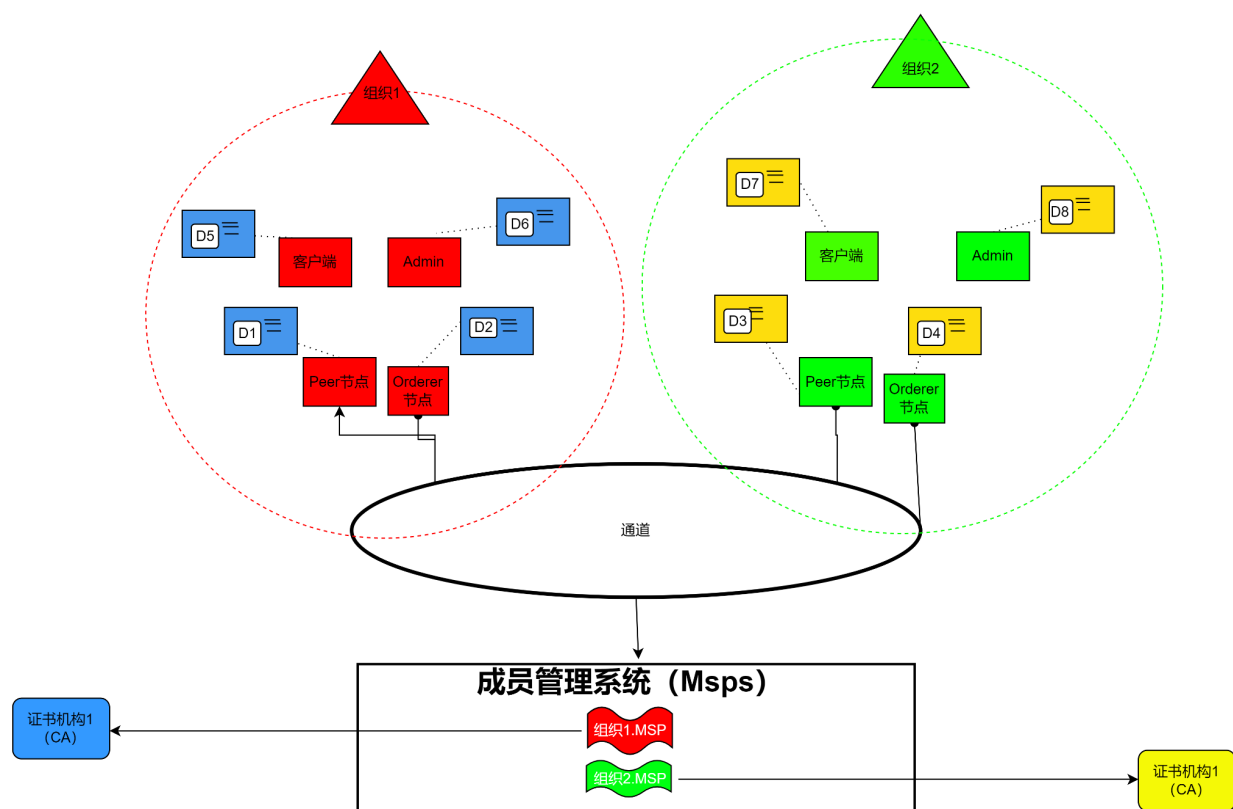
# 主要构成

Fabric 的联盟链是由一个个通道、组织组成。一个组织可加入多个通道, 但每个通道内的账本只允许该通道里所有组织写入、浏览。

组织里所有成员有可能拥有以下四种身份, 分别是管理员 (Admin)、客户端 (Client)、Peer 节点、和Orderer节点。

当新成员加入一个组织A之时, 该成员会发送X.509的电子证书申请到组织A自己的证书颁发机构 (CA-详情见下文)。当组织A同意此请求时, 组织A会发送X.509的电子证书 (公钥和私钥 ) 给该成员并把证书存放到自己组织的成员管理系统 (Msp)

一个完整区块链相互关系如图所示



## 客户端:

主要作用是允许客户与Fabric网络进行交互实现对区块链系统的操作。这些操作分为管理类和链码类的两种。管理类包括启停节点和配置网络等;链码类操作主要是链码的生命周期

管理, 如安装、实例化以及调用链码。最常用的客户端是命令行客户端(CLI), 此外是基于API开发的客户端程序。

Fabric早前提供了一整套Java SDK、用于实现身份标志注册、链码部署、调动等功能。在1.4版本之后, Fabric推出了Fabric gateway API, 以实现更轻便的编程风格, 但目前在功能上只支持在交易方面的操作, 并不支持更复杂的功能、例如通道创建, 身份注册等。

## Peer节点:

Fabric网络中的对等节点, 储存该通道所有账本并拥有执行和部署链码(智能合约)的权利。按照功能主要分为背书节点和确认节点。

确认节点主要负责检验交易合法性、并更新和维护区块链数据和账本状态

背书节点主要负责对于交易指定的链码执行、把结果进行签名打包、发还给客户端。在实际部署中, 背书节点和确认节点既可以部署在同一物理节点上, 也可以分开部署。

## Orderer排序节点:

主要职责是对各个节点发来的交易进行排序。在并发的情况下, 各个节点交易的先后时序需要通过排序节点来确定并达成共识。

排序节点按照一定规则确定交易顺序之后打包并发还给Peer节点、把交易上链到该通道的账本中。

排序节点和Peer节点一样隶属于单一组织。排序节点支持互相隔离的多个通道, 使得交易只发送给相关的节点(Peer), 真实产业应用时一般排序节点部署在盟主(联盟发起方)。

## 管理员(Admin):

管理员的权限是根据该通道的规则来决定的。一般来说, 管理员有资格颁布或取消其管理下成员的身份资格。

## 证书颁发机构(CA):

当我们创建一个通道时, 我们需要使用一个CA来颁布或取消每个节点、用户的电子身份证书。在成员身份明确的基础上, Fabric可以实现权限控制的管理。

每个组织都应该拥有他们自己的证书颁发机构。在创建机构时, 可以使用Fabric自己的Fabric CA、这样区块链会自动生成该组织的成员管理系统。如果使用第三方的证书颁发机构, 则需自己创建成员管理系统。

## 通道:

商业应用的一个重要的需求是私密性交易, 为此 Fabric 设计了通道(Channel)来提供成员之间的隐私保护。通道是部分网络成员之间拥有独立的通信渠道, 在通道中发送的交易只有属于通道的成员才可见, 因此通道可以看作是Fabric的网络中部分成员的私有通信“子网”。

通道由排序服务管理。在创建通道的时候, 需要定义它的成员和组织、peer节点和排序节点, 一条通道对应的区块链结构也同时生成, 用于记录账本的交易, 通道的初始配置信息记录在区块链的创世块(第一个区块)中。通道的配置信息可以用增加一个新的配置区块来更改。

每个组织可有多个节点加入同一个通道, 这些节点中可以指定一个Peer节点(或多个Peer节点做备份)。Peer节点代表本组织与其他组织的节点交互, 从而发现通道中的所有节点。另外, 同一组织的节点会选举或指定主导节点( leading peer ), 主导节点负责接收从排序服务发来的区块, 然后转发给本组织的其他节点。主导节点可以通过特定的算法选出, 因此保证了在节点数量不断变动的情况下仍能维持整个网络的稳定性。

在 Fabric 的网络中, 可能同时存在多个彼此隔离的通道, 每个通道包含一条私有的区块链和一个私有账本, 通道中可以实例化一个或多个链码, 以操作区块链上的数据。由此可见, Fabric 是以通道为基础的多链多账本系统。通道的引入, 增加了fabric商业化时的利用率, 降低了成本, 可以容纳更复杂的商业环境

## 分布式账本:

Fabric 里的数据以分布式账本的形式存储。账本由一系列有顺序和防篡改的记录组成, 记录包含着数据的全部状态改变。账本中的数据项以键值对的形式存放, 账本中所有的键值对构成了账本的状态, 也称为“世界状态”( World State )。

每个通道中有唯一的账本, 由通道中所有成员共同维护着这个账本, 每个确认节点上都保存了它所属通道的账本的一个副本, 因而是分布式账本。对账本的访问需要通过链码实现对账本键值对的增加、删除、更新和查询等的操作。

账本由区块链和状态数据库两部分组成。

区块链是一组不可更改的有序的区块(数据块), 记录着全部交易的日志。每个区块中包含若干个交易的数据, 不同区块所包含的交易数量可以不同。区块之间用哈希链( Hashed-link ) 关联: 每个区块头包含该区块所有交易的哈希值, 以及上一个区块头的哈希值。这样的链式架构可以确保每个区块的数据不可更改, 以及每个区块之间的顺序关系不可更改。这个特点决定了区块链的区块只可以添加在链的尾部。

状态数据库记录了账本中所有键值对的当前值, 相当于对当前账本的交易日志做了索引。链码执行交易的时候需要读取账本的当前状态, 从状态数据库可以迅速获取键值的最新状态。

如果没有状态数据库, 要获得某个键值时, 需要遍历整个区块链中和该键值相关的交易, 效率非常低, 因此, 读取状态数据库可以认为是快速定位和访问某个键值的方法。另外, 当状态数据库出现故障的时候, 可以通过遍历账本重新生成。

当一个区块附加到区块链尾部的时候, 如果区块中的有效交易修改了键值对, 则会在状态数据库中作相应的更新, 这样区块链和状态数据库始终保持一致。

区块链的数据块以文件形式保存在各个节点中。状态数据库原理上可以是各种键值数据库, Fabric 缺省使用的是 LevelDB, 也支持 CouchDB 的选项。CouchDB 除了支持键值数据之外, 也支持 JSON 格式的文档模型, 能够做复杂的查询。所以大家在操作Fabric上链数据时, 要区分你操作的是链上数据还是状态数据库的数据。

## 资料来源:

<https://blog.csdn.net/wj8023/article/details/126336537>

[https://blog.csdn.net/qq\\_38236620/article/details/128076140](https://blog.csdn.net/qq_38236620/article/details/128076140)

[https://hyperledger-fabric.readthedocs.io/en/release-2.5/orderer/ordering\\_service.html](https://hyperledger-fabric.readthedocs.io/en/release-2.5/orderer/ordering_service.html)