区块链技术应用题

(学习手册)

北京知链科技有限公司

目录

学习导入

今天，以云计算、大数据、人工智能、区块链等为代表的的新一轮科技革命，对金融业产生着前所未有的影响。新技术正义其独有的渗透性、冲击性、倍增性和创新性推动金融行业发展到一个全新节点。金融科技人才，是复合型创新人才，需要金融方面掌握扎实理论基础，科技方面尤其是区块链方面，掌握前沿技术，具备对金融产品和金融服务的设计、应用、基本开发、维护的能力。

区块链，作为金融科技核心技术之一，其“去中心化”“不可篡改”“公开透明”等特性在金融领域正在广泛应用，推动者金融服务模式和创新和变革，主要应用领域有：征信、跨境支付、供应链金融、证券交易、保险和租赁担保。这些领域的应用，避免不了的是这些领域中相关的企业如何上链、岗位如何上链、企业如何在链上发生业务、链在哪、链是如何搭建的等等一些的问题，会围绕在金融科技的四周。本学习手册，将全面讲解联盟链如何搭建、企业如何上链等一系列的知识，带领学生进入到联盟链的世界。

一、 什么是联盟链

1. 定义

部分公开。是指某个群体或组织内部使用的区块链，对加入的组织和单位有一定的限制和要求。在联盟链记账权的确定上，需要预先指定几个节点为记账人，而且记账人会不断变化。每个区块的生成由预选的所有记账人共同决定和打包，链上的其他节点可以交易，但是没有记账权。通过这种记账方式，联盟链上的区块产生更快，交易的确认效率达到秒级共识。

2. 联盟链的产生

联盟区块链产生，可以从联盟链使用的群体了解，他们的主要群体是银行、保险、证券、商业协会、集团企业及上下游企业。区块链的底层网络还是移动互联网，区块链对于进一步提升他们圈子的产业链条中的公证、结算清算业务和价值交换网络的效率很有帮助，但是在尝试使用现有区块链技术(如比特币代码家族和以太坊)发现，现有区块链的处理性能、隐私保护、合规性等都不能满足他们的业务需求;另一方面，他们意识到如果全面采用比特币的那一套完全公链的设计理念，会颠覆他们现有的商业模式和固有利益，且要负上很大的风险。于是他们开始改造适合他们的区块链体系，目前的联盟链形态，更多是分布式账本(DSL)为主，区块链的分布式账本和分布式共识为他们解决了主要核心问题，即联盟中多个参与方交互的信任问题。

3. 联盟链的优点

相比于公有链，联盟链在效率和灵活性上更有优势，主要体现为以下几点：

(1) 一是交易成本更便宜。链上节点发起的交易只需被指定的几个节点验证就可以了，而无需全网验证和达成共识。

(2) 二是节点之间的连接更稳定。存在故障的时候，可以迅速通过人工干预来修复，并允许使用共识算法减少区块时间，从而更快完成交易。

(3) 三是联盟链数据读取权限受到限制，可以对商业信息提供更好的隐私保护。

(4) 四是更灵活。如果需要的话，运行联盟链的组织或公司可以很容易地修改该区块链的规则，还原交易，修改交易等。

区块链的发展和演变很好地体现了经济金融史一再重现的中心化和去中心化的互相交锋渗透。区块链现实的应用取决于现实的需求，正如以太坊创始人所说“只有一种区域链能活下来的想法是完全的误导”。

二、 搭建联盟链

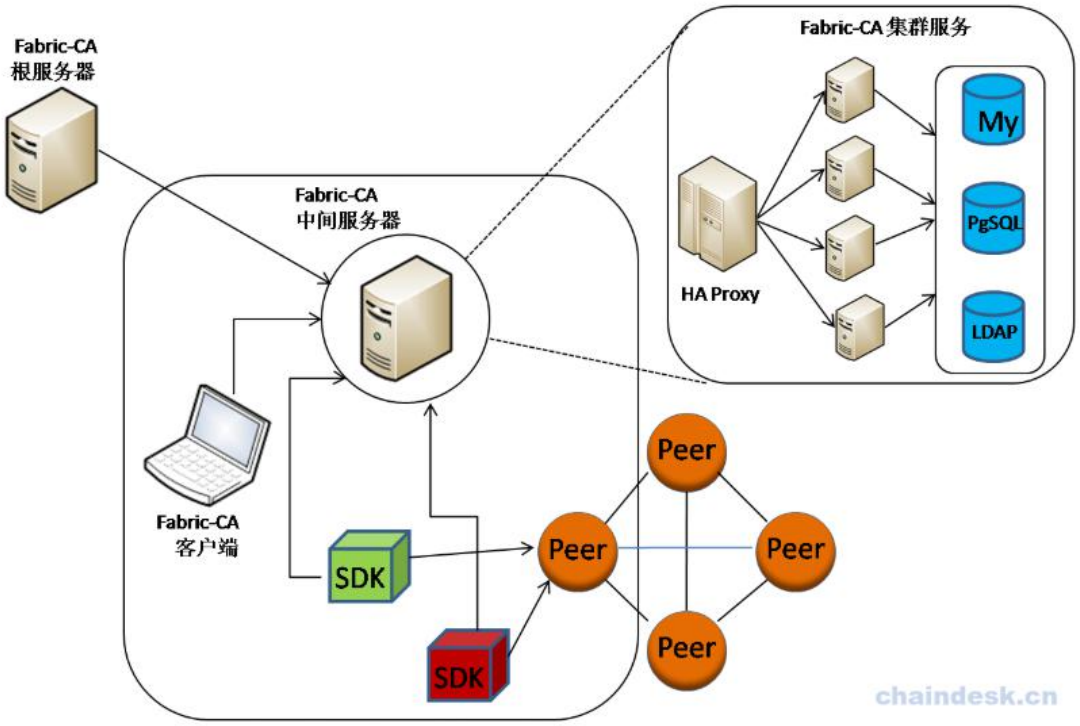
1. 构建联盟链链上认证服务

构建联盟链链上认证服务是搭建联盟链的第一步，链上的认证服务是用于给链上的其他节点发放证书、发放身份的机构。节点只有拥有的了证书，在链上才拥有身份，别的节点才能认可和识别。链上认证服务包括：CA Server（证书服务端）、CA Client（CA 客户端）、中间 CA 等组成。Ca（Certificate Authority）叫认证机构，是 Fabric 的证书颁发机构，由服务器（server）和客户端（client）组成。联盟链本来的应用场景就是几个成员组成联盟，共享需要共享的数据，而不同的成员又有不同的权限，这就需要成员有权限可以进入这个联盟链。所以 Ca 就是干这事，也可以说 Ca 是赋予成员进入联盟的权限机构。构建链上认证服务需要进行 CA Server 服务端设置、CA 数据库的设置、CA 客户端的配置、中间 CA 的证书服务设置以及 LDAP 目录访问协议的设置。

Fabric Server 端由一个服务器集群组成，以树形架构组织 CA Server 节点，包含一个 Root 节点和多个中间节点。每个 CA 要么是根 CA，要么是中间 CA。每个中间 CA 都有一个父 CA，它要么是根 CA，要么是另一个中间 CA。

可以通过 Client 或 SDK 与服务器集群中的 CA 服务器进行交互。客户端首先路由到 HA 代理，由代理进行负载均衡，将客户端连接至某一服务器的集群成员。

包括前端的一个高可用的代理服务器，连接着若干个 CA Server 集群，这些集群将数据共同存放在同一个数据服务器上。数据库可能是 MySQL、LDAP、PostgresSQL 或者 SQLite（集群环境中不推荐使用 SQLite）。



1.1 CA Sever

CA Server 服务端的配置，需要首先要对服务端的基本信息进行设置，创建出服务端的账户和访问密码，同时将服务端的安装包下载和安装。然后进行 CA Server 服务的初始化，经过初始化，各种字段参数将会被设置为系统默认值，防止前期错误使用时造成不可预知的错误。若出现错误，我们可以进行初始化操作，将默认值对比自己的参数值，就可以方便出现错误的时候找到原因。其次是启动 CA Server，将安装好的 CA Server 客户端进行启动已发挥作用。最后再完成证书信息相关的设置，以完成账户、撤销证书、授权中间 CA、获取证书列表等权限。下方是证书设置中相关的权限列表：

步骤	配置项	填写内容（建议）	说明
Ca 证书配置	名称	默认值：Fabric CA Root Server	当前设置的组件名称
	秘钥文件	系统默认路径	秘钥文件的存储地址

	证书文件	系统默认路径	证书文件的存储地址
	证书链文件	系统默认路径	证书链文件的存储地址
注册表	最大登记数	-1	-1 表示不限登记个数；0 表示特定登记个数
	hf.Registrar.Roles	默认值：peer,orderer,client,user	本配置项设置可以进行管理员注册的节点类型
	hf.Registrar.DelegateRoles	默认值：peer,orderer,client,user	本配置项设置能够进行 hf.Registrar 属性配置的节点类型
	hf.Revoker	true	true 表示本 Ca 能够能够撤销用户身份或用户证书
	hf.IntermediateCA	true	true 表示本 Ca 可以作为中间 Ca 注册
	hf.GenCRL	true	true 表示本 Ca 能够生成 CRL(证书吊销列表)
	hf.Registrar.Attributes	默认值：*	允许注册商注册的属性列表，*表示允许所有节点注册
	hf.AffiliationMgr	true	true 表示本 Ca 可以管理联盟
组织结构配置	组织结构配置	系统默认值	前面设置的最大登记数为-1 时，本配置项不需要修改
签发证书	是否签署 Ecert	true	true 表示可以签署身份证书
	证书域	默认值：ca.fabric.com	无

	根证书超时时间	20000	这里设根证书的超时时间，若一节点的根证书超时，其签署的未过期证书仍有效。 就算这一节点证书过期，但也可在该联盟链中追溯到该节点，所以过期不会涉及其下属节点。 若这一节点退出该联盟链，其签署的未过期证书就会无效。
	使用者证书域	系统默认值	可以使用该设置权限的节点
	吊销列表签名	系统默认值	记录被吊销或过期的证书
	中间证书超时时间	20000	若签署中间 Ca ，这里设置中间证书的超时时间
	是否是 CA	true	true 表示为可以设置为证书认证机构
	是否能签署中间 CA	true	true 表示为具有签署中间 Ca 的权限

CA Server 的功能

1、负责 Fabric 网络内所有实体（Identity）身份的注册。
2、负责对数字证书的签发，包括 ECerts（身份证书）、TCerts（交易证书）。
3、证书的续签或吊销。

1.2 CA Client

客户端（Client）是用户与 Fabric 网络组件发送请求进行交互的接口，包括 FabricCa 客户端和 Fabric 客户端，本任务创建的是 FabrciCa 客户端，用于对后续的加入联盟链的组织进行审核。

对于 CA 的客户端，与 CA 服务器提供数据存储不同，其主要功能涉及联盟链节点的操作，比如登记启动身份、注册新的身份、登记对等节点、获取 CA 验证链和获取用户的身份密文等。

步骤	配置项	填写内容（建议）	说明
Fabric Ca Client 基本信息	组件 Id	默认值：FabricCaClient	组件的唯一标识，不

			建议修改。
	组件名称	FabricCaClient	组件的名称，可自定义
	组件描述	FabricCa 客户端，用于审核加入联盟的组织	组件的描述，可自定义
编译安装 Fabric Ca Client	下载路径	系统默认路径	客户端文件的存储地址
	安装状态	自动安装	无

CA 客户端直接与 CA Server 服务端直接关联，通过客户端完成服务端相关的证书功能。在 CA 客户端处核心在于登记了管理员的角色，管理员可以进入到 CA 的客户端中进行审核组织管理员、发放节点身份、管理证书列表、授权中间 CA、撤销用户证书等功能。

1.3 ICA Server

CA Server 的证书颁发机构，在联盟链中颁发的是根证书的角色，其为整个联盟链上的所有节点进行证书服务，会增加其负载压力。就需要在联盟链中配置一个中间 CA 的服务集群，其作用是作为中间 CA 为各个节点发放证书，减小根 CA 的负载压力。ICA 相当于中间代理商，其权利需要根 CA 确认才行，其要受根 CA 管理。其直接与 CA Server（RCA）进行关联，每次中间 CA 颁发的证书都会带有根 CA 的证书签名，以表明证书的可信任性和授权性。最终，所有的中间 CA 颁发的证书都可以找到根 CA 的签名信息，以形成了证书链的形式。

证书链由两个环节组成—信任锚（CA 证书）环节和已签名证书环节。自我签名的证书仅有一个环节的长度—信任锚环节就是已签名证书本身。

1.4 LDAP

轻型目录访问协议（英文：Lightweight Directory Access Protocol，缩写：LDAP，/ˈɛldæp/）是一个开放的，中立的，工业标准的应用协议，通过 IP 协议提供访问控制和维护分布式信息的目录信息。LDAP 的一个常用用途是单点登录，用户可以在多个服务中使用同一个密码，通常用于公司内部网站的登录中（这样他们可以在公司计算机上登录一次，便可以自动在公司内部网上登录）。

Hyperledger Fabric CA 的身份信息保存在数据库或 LDAP 中。目前 Fabric CA 支持的数据库有 MySQL、PostgreSQL、SQLite；默认使用 SQLite 数据库。如果配置了 LDAP，则身份信息将保留在 LDAP 而不是数据库中。

LDAP（Light Directory Access Portocol），它是基于 X.500 标准的轻量级目录访问协议。

目录是一个为查询、浏览和搜索而优化的数据库，它成树状结构组织数据，类似文件目录一样。

目录数据库和关系数据库不同，它有优异的读性能，但写性能差，并且没有事务处理、回滚等复杂功能，不适于存储修改频繁的数据。所以目录天生是用来查询的，就好象它的名字一样。

LDAP 目录服务是由目录数据库和一套访问协议组成的系统。

1.5 CA 数据库

CA 数据库记录着很多关于证书的数据，是不可缺少的部分之一。在联盟链系统中，提供了三种数据库类型，分别是 SQLite、PostgreSQL 和 MySQL。默认的数据库是 SQLite，默认的数据库文件是 Fabric CA 服务器所在目录的 fabric-ca-server.db 文件。在 CA 数据库中重要的是进行基础信息的配置，完成与 CA Server 的关联和数据库本身的访问权限，包括：账户和密码。下方是 CA 数据库相关的配置参数：

配置项	填写内容（建议）	说明
数据库 ID	默认值：Ca Sever	Id 是唯一的，用于识别组件
数据库名称	RootCa 数据库	组件的名称，可以自定义名称
数据库描述	用于存储 Ca 机构向组织颁发的证书	当前数据的的描述，可以自定义内容
数据库类型	Sqlite	Sqlite:一个轻量级别数据库。 PostgreSql:一种特性非常齐全的自由软件的对象-关系型数据库管理系。 Mysql:是最流行的关系型数据库管理系统之一。
数据库文件	系统默认路径	数据库文件的存储地址
Ca Server 服务器	FabricCaRootServer	这里就是前一任务中创建的 Ca Server 组件 Id

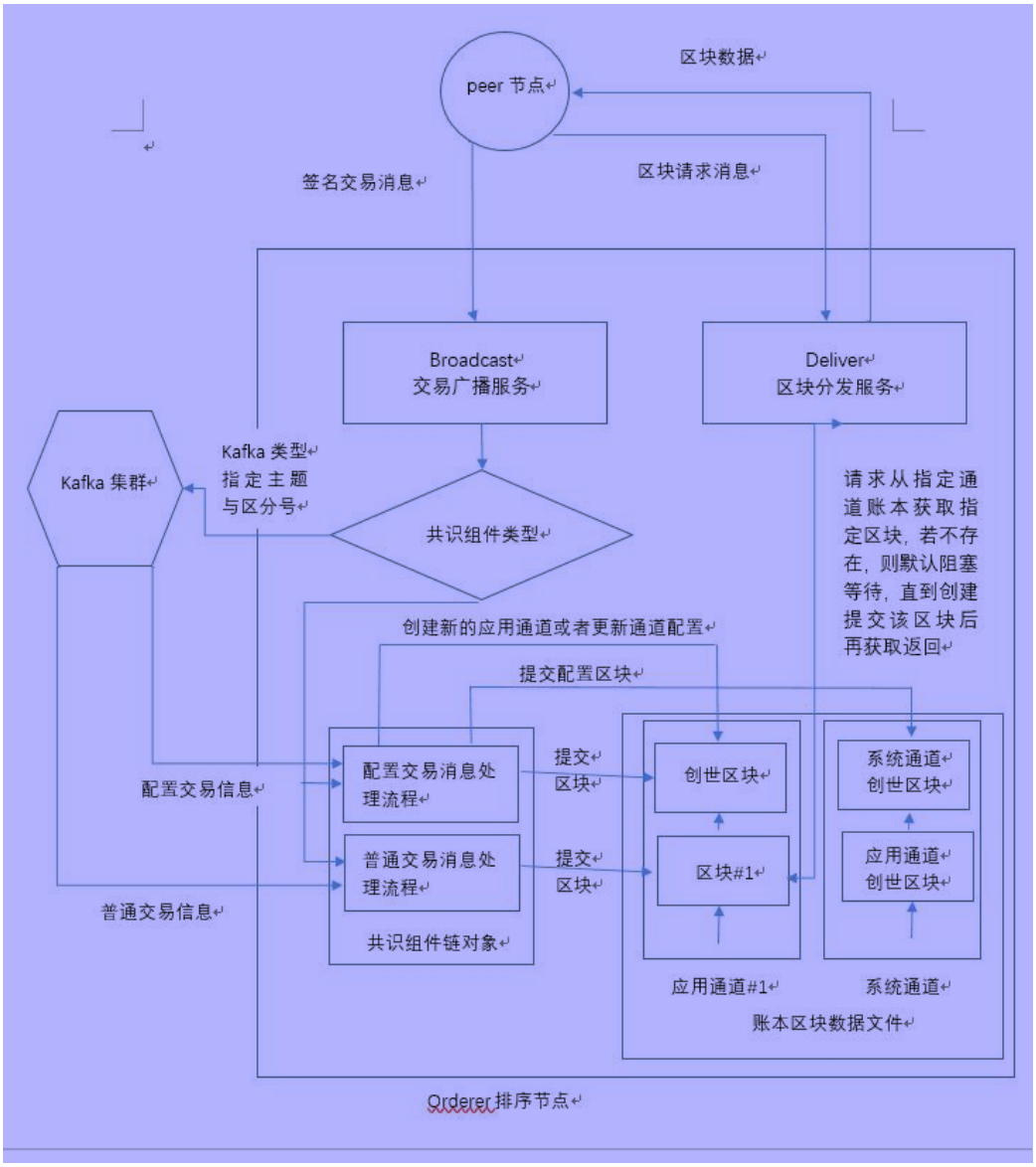
端口号	自定义四位数字	端口号是用于传输数据的出入口，不能够重复
Ca 数据库账户	SqlUser	登陆 Ca 数据库的账户，该账户对本组件进行直接管理，有最高权限，可以自定义，字符+数字的组合
Ca 数据库密码	SqlUser123	登陆 Ca 数据库账户的密码，可以自定义，字符+数字的组合

2. 构建 Fabric 排序与共识服务

区块链中核心的技术是共识算法，一个优秀的共识算法直接决定了链上出块的速度和交易达成共识的时间，也是奠定链上信息可信的基础。在 Fabric 中有自己独有的排序服务和共识共识服务，其中，Orderer 排序节点在 Hyperledger Fabric 系统结构中处于核心角色地位，管理者系统通道与所有应用通道，负责通道创建、通道配置更新等操作，并处理客户端提交的交易消息请求，对交易进行排序并按规则打包新区块，提交账本并维护通道账本数据，为全网节点提供 Broadcast 交易广播服务、Orderer 共识排序服务、Deliver 区块分发服务等。

2.1 Orderer 排序服务

Hyperledger Fabric 启动时需要先启动 Orderer 排序节点，创建系统通道一共正常的服务后，再启动其他角色的 peer 节点进入正常工作状态，因此，orderer 排序节点相当于 Hyperledger Fabric 系统的“中枢神经”，其服务模块关系与架构示意图，如图所示



排序服务节点接收到网络中所有通道发出的交易信息，读取交易信封获取通道名称，按各个通道上交易的接收时间顺序对交易信息进行排序（多通道隔离），生成区块。（在这个过程中，排序服务节点不会关心交易是否正确，只是负责排序和打包。

2.2 Kafka 共识服务

交易在通过背书节点进行模拟运行并且背书后，将交易返回给请求节点，请求节点将提交交易信息为请求排序，排序服务内部的排序节点对交易进行排序，按规则构造新区块。注意，这里的按规则，就会使用到 kafka 共识机制。

kafka 是一个分布式消息队列。具有高性能、持久化、多副本备份、横向扩展能力。Kafka 包括生产者、消费者和分区消费者。Kafka 生产者按照主题 (Topic) 生产消息并进行发布，Kafka 服务器集群自动对消息主题进行分类。同一个主题

的消息都会被收集到一个或多个分区中，按照 FIFO（先进先出）的顺序追加到文件尾部，并且每个消息在分区中都会有一个 OFFSET 位置偏移量作为该消息的唯一标识 ID。

2.3 Broadcast 接口

Hyperledger Fabric 提供了 Broadcast 交易广播接口，接受客户端提交的签名交易消息请求，交由共识组件链对象对交易进行排序与执行通道管理，按照交易出块规则切割打包，构造新区块并提交账本。同时，通过 Deliver 区块分发服务接口，将区块数据发送给通道组织内发送请求的 Leader 主节点，再基于 Gossip 消息协议到组织内其他节点上，从而实现广播交易消息的目的。

2.4 Deliver 接口

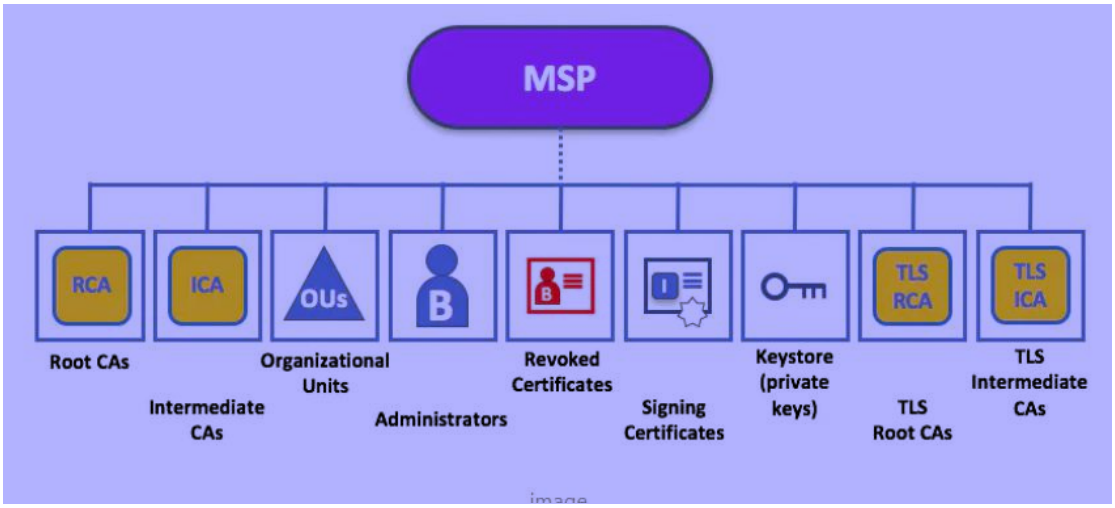
Orderer 排序服务器提供了区块分发服务接口 Deliver，接受客户端提交的区块信息，根据该消息封装的区块搜索信息对象，构造对应请求范围查询结果迭代，读取 Orderer 节点指定通道账本上的区块数据，同时，建立消息处理循环，基于该结果迭代依次读取请求的区块数据结果，发送给组织 Leader 主节点等请求节点，由主节点同步同组织下的其他节点，最终的结果是同一通道下的所有组织中的节点的信息得到同步。

3. 构建联盟组织

3.1 组织上链

Fabric 中的组织在现实世界中可以是一个公司、一个企业，或者一个协会。在 Fabric 中，组织是承担着数据信用责任的区块链系统参与方。可以对组织进行组织类型、组织名称、组织部门、组织岗位、国家、省份、城市进行设置。组织的上链要完成组织身份的发放，组织可以向 CA Server 或者 ICA Server 进行证书申请，以获取自身的 MSP 身份文件。

MSP（Membership Service Provider）：成员服务提供商，是 Hyperledger Fabric 1.0 版本开始抽象出来的一个模块化组件。用于定义身份验证，进行身份验证和允许访问网络的规则。更确切地说，MSP 是 Hyperledger Fabric 对网络中的组成成员进行身份管理与验证的模块组件。



在组织上链时，同时对组织内的成员进行权限设置，包括：访问权限、写入权限、管理权限。1、访问：定义组织内哪些节点可以访问组织有关信息
2、写入：定义组织内哪些节点可以修改组织内配置文件信息
3、管理：定义组织内哪些节点可以有管理其他节点权限
组织内的成员包括：client、admin、peer 三种成员

3.2 组织关联排序服务

当该组织中的节点在关联的通道中发生交易后，首先形成提案发送给背书节点进行背书并模拟运行，之后获得背书提案原路返回，再需要将提案发送给排序服务，若干个交易形成一个区块。我们知道，某一 kafka 关联某一排序服务，本任务是将某一排序服务关联某一组织。

将组织直接和排序服务中的某一排序节点直接关联。同时，在关联排序服务时，需要验证组织的身份，包括：安全传输协议私钥文件、安全传输协议证书文件、安全传输协议根证书文件，来确保该组织的连接安全性。

4. 创建组织节点

4.1 节点的基本信息

节点功能模块在 Hyperledger Fabric 架构中提供了用户与系统交互接口，支持 node 子命令，用于启动 peer 节点功能服务器提供服务，其功能涉及 Deliver 事件服务器、链码支持服务、Admin 管理服务器、Endorser 背书服务器、Gossip 消息服务器等。同时，Peer 节点功能模块还支持 channel、chaincode、logging、version 等子命令，用于创建 Hyperledger Fabric 系统上的应用通道、链码、日志等对象并进行管理。

节点是区块链的通信实体，是一个逻辑概念，不同类型的多个节点可以运行在同一个物理服务器上。

节点主要有以下四种类型：如图所示。

1、客户端节点：客户端必须连接到某一个 peer 节点或排序服务节点上才能与区块链网络进行通信。

2、普通节点（**本章涉及到的节点类型**）：根据所承担的角色又可以分为记账节点（committer）、背书节点（endorser）、主节点（leader）和锚节点（anchor）。

3、排序服务节点：接收包含背书签名的交易，对未打包的交易进行排序生成区块，广播给 peer 节点。

4、CA 节点：CA 节点接收客户端的注册申请，返回注册密码用于用户登录，以便获取身份证书。

4.2 节点加入组织

节点加入组织的过程中，需要获取链上的身份，向 CA Server 或者 ICA Server 进行身份申请。同时，要经过组织内的节点和管理员进行审核，审核通过后方可进入组织。节点加入组织后，在组织中扮演着某种身份进行执行。联盟链上，节点一共有四种类型：主节点、记账节点、背书节点、锚节点。

- 主节点：主节点（leader peer），能与排序服务节点通信，负责从排序服务节点获取最新的区块并在组织内部同步。
- 背书节点：验证提案的合法性，对提案进行背书操作，并对结果进行签名背书。
- 记账节点：所有的 peer 节点都是记账节点（committer），负责验证排序服务节点区块里的交易，维护状态和总账 (Ledger) 的副本。该节点会定期从 orderer 节点获取包含交易的区块，在对这些区块进行核发验证之后，会把这些区块加入到区块链中。
- 锚节点：锚节点主要负责代表组织和其他组织进行信息交换。每个组织都有一个锚节点，锚节点对于组织来说非常重要，如果锚节点出现问题，当前组织就会与其他组织失去联系。用于许多不同的跨组织通信场景。channel 上的每个成员都有一个 anchor peer(或多个 anchor peer 来防止单点故障)，允许属于不同成员的 peer 发现 channel 上的所有现有 peer。

4.3 节点加入通道

将 peer 节点加入到指定的应用通道，通过请求默认的 Endorser 背书节点执行 CSCC 系统链码，在本地 peer 节点上创建绑定该通道的链结构对象，并注册到节点的链结构字典中，以管理本地通道上的账本、通道配置资源，同时，初始化该通道上的 Gossip 消息服务模块。其中 Leader 主节点会启动 Deliver 服务实例，代表组织向 Orderer 服务节点请求该通道账本上的区块数据，并基于 Gossip 消息协议分发到组织内的其他 peer 节点上。普通节点（非 leader 节点）只能接收从其

他节点转发的账本区块。join 命令执行成功意味着该 peer 节点可以正常接收该通道账本上的所有区块数据。

5. 创建联盟通道服务

5.1 创建通道

通道是 Fabric 中非常重要的概念，它实质是由排序节点划分和管理的私有原子广播通道，目的是对通道的信息进行隔离，使得通道外的实体无法访问通道内的信息，从而实现交易的隐私性。

目前通道分为系统通道（System Channel）和应用通道（Application Channel）。排序节点通过系统通道来管理应用通道，用户的交易信息通过应用通道传递。对一般用户来说，通道是指应用通道。

通道通常指 Orderer 排序节点管理的彼此隔离的原子广播渠道，提供隔离 Peer 节点信息的重要机制。

Peer 节点在加入应用通道时会主动创建关联通道的链结构对象，接收来自 Orderer 节点的通道账本数据，通道上的数据只会发给加入通道的合法组织成员，从而隔离未经授权的数据访问，保护数据隐私性。

1、**应用通道**为上层应用程序处理交易提供隔离机制，在指定通道组织成员间共享账本数据，应用通道账本上保存了应用通道的创始区块，配置区块和普通交易区块。

2、**系统通道**保存 Orderer 配置，基于系统通道配置与应用通道配置交易信息创建新的应用通道，并将其注册到 Orderer 节点的多通道注册管理器 Registrar 对象上，系统通道账本保存了系统通道的创世区块、所有应用通道的创世区块及其更新的配置区块。

通道的配置信息都被打包到一个区块中，并存放在通道的共享账本中。该区块除了配置信息外不包含其他交易信息，称之为通道的配置区块（Configuration Block）。通道可以使用配置区块来更新配置，因此在账本中每新添加一个配置区块，通道就按照最新配置区块的定义来修改配置。通道账本的首个区块一定是配置区块，也称为创始区块（Genesis Block）。

在创世区块中规定了当前通道的最大交易数、有效块签名、访问策略、写入策略、管理策略。同时规定了排序服务的节点，进行关联。

5.2 通道系统链码

- 系统链码之一——CSCC，此链码主要功能是管理 peer 上通道相关的信息以及执行通道配置交易。链码分为系统链码和用户链码。系统链码运行是为了支持联盟链管理的一系列操作，使联盟链能进行基本的组件操作，与具体的业务联系不大，且系统链码几乎一样，除非联盟链版本不同，基本的系统链码有五种；用户链码则是在系统链码能支持联盟链正常运行的基础上做到和

具体业务相关，按业务程序步骤运行，业务不一样，用户链码也就不一样，用户链码有多种，数量不限。 和用户链码不同的是，系统链码不需要使用 SDK 或 CLI 的提案安装和实例化。它在 peer 节点启动的时候注册和部署。

- LSCC，即生命周期系统链码，它是一种系统链码，功能的对象是用户链码，负责对用户链码的生命周期进行管理，支持被从链外进行调用。链码的生命周期包括安装、部署、升级、权限管理、获取信息等环节。
- 在 fabric 中，交易在通道中进行，交易形成提案，再形成区块，最后同步到区块链账本中。若想知道此时区块的情况，就需要 QSCC 来帮助了。Query System Chaincode，简称 QSCC，查询系统链码，负责提供一些账本和链信息的查询方法。
- VSCC 系统链码用于验证交易背书策略，要求检查背书信息的有效性（即使用有效证书进行签名）、是否存在恰当的（满足要求的）背书数量与来自预期的背书节点（指定组织和角色）等，可以通过客户端命令行或 SDK 指定背书策略。
- Endorsement System Chaincode，即背书管理系统链码。负责背书（签名）过程，并可以支持对背书策略进行管理，仅支持链内系统调用。ESCC 主要代码实现在 core/scc/escc 路径下，目前提供了 Invoke 方法，会对传入的链码提案的模拟运行结果进行签名，之后创建响应消息返回给客户端。