

# 术语表

专业术语很重要，因为所有 Hyperledger Fabric 项目的用户和开发人员都要一致的理解我们所说的每个术语的含义。比如什么是链码。本文档将会按需引用这些术语，如果你愿意的话也可以阅读整个文档，这会非常有启发！

## 锚节点

gossip 用来确保在不同组织中的 Peer 节点能够知道彼此。

当提交一个包含锚节点更新的配置区块时，Peer 节点会联系锚节点并从它们那里获取该节点知道的所有 Peer 节点的信息。一旦每个组织中至少有一个 Peer 节点已经联系到一个或多个锚节点的话，锚节点就会知道这个通道中的每个 Peer 节点。因为 gossip 通信是不变的，并且 Peer 节点总是会要求告知他们不知道的 Peer 节点，这样就建立起一个通道成员的通用视图。

比如，我们可以假设在通道中有三个组织：`A`、`B`、`C`，有一个为组织 `C` 定义的锚节点 `peer0.orgC`。当 `peer1.orgA`（来自组织 `A`）联系 `peer0.orgC` 的时候，它会告诉 `peer0.orgC` 关于 `peer0.orgA` 的信息。然后当 `peer1.orgB` 联系 `peer0.orgC` 的时候，后者会告诉前者关于 `peer0.orgA` 的信息。之后，组织 `A` 和 `B` 就可以开始直接地交换成员信息而不需要任何来自 `peer0.orgC` 的帮助了。

由于组织间的通信要基于 gossip 来工作，所以在通道配置中至少要定义一个锚节点。为了高可用和冗余，非常建议每个组织提供他们自己的一组锚节点。

## ACL

ACL（Access Control List，访问控制列表）将特定节点资源（例如系统链代码 API 或事件服务）的访问与 策略（指定需要多少和哪些类型的组织或角色）关联在一起。ACL 是通道配置的一部分。因此，它会保留在通道的配置区块中，并可使用标准的配置更新机制进行更新。

ACL 的格式为键值对列表，其中键标识我们希望控制其访问的资源，值标识允许访问它的通道策略（组）。例如，`lscc/GetDeploymentSpec: /Channel/Application/Readers` 定义对生命周期链代码 `GetDeploymentSpec` API（资源）的访问可由满足 `/Channel/Application/Readers` 策略的身份访问。

`configtx.yaml` 文件中提供了一组默认 ACL，`configtxgen` 使用该文件来构建通道配置。可以在 `configtx.yaml` 的顶级“Application”部分中设置默认值，也可以在“Profiles”部分中按每个配置文件覆盖默认值。

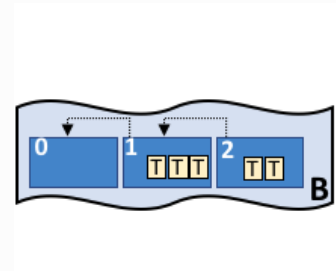
## 区块

一个区块包含了一组有序的交易。他们以加密的方式与前一个区块相连，并且他们也会跟后续的区块相连。在这个链条中的第一个区块被称为 **创世区块**。区块是由排序服务创建的，并且由 Peer 节点进行验

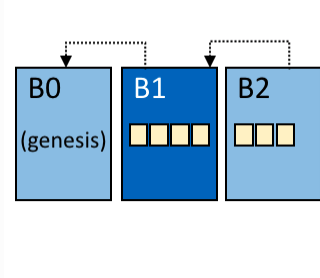
证和提交。

## 链

账本的链是交易区块经过哈希连接结构化的交易日志。Peer 节点从排序服务收到交易区块，基于背书策略和并发冲突来标注区块的交易为有效或者无效状态，并且将区块追加到 Peer 节点文件系统的哈希链中。



区块链 B 包含了区块 0, 1, 2



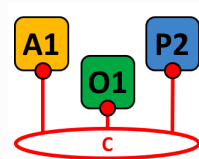
区块 B1 是连接到区块 B0 的。区块 B2 是连接到区块 B1 的。

## 链码

请查看 [智能合约](#)。

## 通道

通道是基于数据隔离和保密构建的一个私有区块链。特定通道的账本在该通道中的所有 Peer 节点共享，交易方必须通过该通道的正确验证才能与账本进行交互。通道是由“配置区块”来定义的。



通道 C 连接了应用程序 A1，Peer 节点 P2 和排序服务 O1。

## 提交

通道中每个“Peer节点\_”都会验证交易的有序区块，然后将区块提交（写入或追加）到该通道上“账本\_”的各个副本。Peer 节点也会标记每个区块中的每笔交易的状态为有效或者无效。

## 并发控制版本检查

并发控制版本检查（Concurrency Control Version Check，CCVC）是保持通道中各节点间状态同步的一种方法。Peer 节点并行的执行交易，在交易提交至账本之前，节点会检查交易在执行期间读到的数据是否被修改。如果读取的数据在执行和提交之间被改变，就会引发 CCVC 冲突，该交易就会在账本中被标记为无效，而且值不会更新到状态数据库中。

## 配置区块

包含为系统链（排序服务）或通道定义成员和策略的配置数据。对某个通道或整个网络的配置修改（比如，成员离开或加入）都将导致生成一个新的配置区块并追加到适当的链上。这个配置区块会包含创始区块的内容，再加上增量。

# 共识

贯穿交易流程的一个广泛的概念，用于对区块中的交易生成一致的顺序并确保其正确性。

## 共识者集合

Raft 排序服务中，在一个通道的共识机制中会有多个活动的排序节点参与其中。如果其他排序节点存在于系统通道，但是没有加入通道，它就不属于这个通道的共识者集合。

## 联盟

联盟是区块链网络上非定序的组织集合。这些是组建和加入通道及拥有节点的组织。虽然区块链网络可以有多个联盟，但大多数区块链网络都只有一个联盟。在通道创建时，添加到通道的所有组织都必须是联盟的一部分。但是，未在联盟中定义的组织可能会被添加到现有通道中。

## 链码定义

链码定义用于组织在将链码应用在通道之前协商链码参数。每个想使用链码背书交易或者查询账本的通道成员都需要为他们的组织批准链码定义。一旦有足够多的通道成员批准了链码定义，使其满足了生命周期背书策略（默认是通道中的多数组织），链码定义就可以提交到通道中了。定义提交之后，链码的第一个调用（或者，必要的话调用 Init 方法）就会在通道上启动链码。

## 动态成员

Hyperledger Fabric 支持成员、节点、排序服务节点的添加或移除，而不影响整个网络的操作性。当业务关系调整或因各种原因需添加或移除实体时，动态成员至关重要。

## 背书

背书是指特定节点执行链码交易并返回一个提案响应给客户端应用的过程。提案响应包含链码执行后返回的消息、结果（读写集）和事件，同时也包含证明该节点执行链码的签名。链码应用具有相应的背书策略，其中指定了背书节点。

## 背书策略

定义了通道上必须执行依赖于特定链码的交易的节点，和必要的组合响应（背书）。背书策略可指定特定链码应用交易背书的最小背书节点数、百分比或全部节点。背书策略可以基于应用程序和节点对于抵御（有意无意）不良行为的期望水平来组织管理。提交的交易在被执行节点标记成有效前，必须符合背书策略。

## 跟随者

在基于领导者的共识协议中，比如 Raft，有一些节点复制领导者生产的日志条目。在 Raft 中，跟随者也接受领导者的“心跳”信息。当领导者停止发送这些信息达到配置中的时间是，跟随者会初始化一个领导者选举并选举出一个领导者。

# 初始区块

初始化排序服务的配置区块，也是链上的第一个区块。

## Gossip 协议

Gossip数据传输协议有三项功能： 1) 管理节点发现和通道成员； 2) 在通道上的所有节点间广播账本数据； 3) 在通道上的所有节点间同步账本数据。 详情参考 [Gossip](#) 话题。

## Hyperledger Fabric CA

Hyperledger Fabric CA 是默认的证书授权组件，用于向网络成员组织和他们的用户发行基于 PKI 的证书。CA 向每一个成员发行一个根证书（rootCert）并向每一个授权的用户发行一个注册证书（ECert）。

## 初始化

初始化链码应用的方法。所有的链码都需要有一个 Init 方法。默认情况下，该方法不会被执行。但是你可以在链码定义中请求执行 Init 方法来初始化链码。

## 安装

将链码放到 Peer 节点文件系统的过程。

## 实例化

在特定通道上启动和初始化链码应用的过程。实例化完成后，装有链码的节点可以接受链码调用。

**NOTE:** *This method i.e. Instantiate was used in the 1.4.x and older versions of the chaincode lifecycle. For the current procedure used to start a chaincode on a channel with the new Fabric chaincode lifecycle introduced as part of Fabric v2.0, see Chaincode-definition\_.*

## 调用

用于调用链码内的函数。客户端应用通过向节点发送交易提案来调用链码。节点会执行链码并向客户端应用返回一个背书提案。客户端应用会收集充足的提案响应来判断是否符合背书策略，之后再将交易结果递交到排序、验证和提交。客户端应用可以选择不提交交易结果。比如，调用只查询账本，通常情况下，客户端应用是不会提交这种只读性交易的，除非基于审计目的，需要记录访问账本的日志。调用包含了通道标识符，调用的链码函数，以及一个包含参数的数组。

## Leader

In a leader based consensus protocol, like Raft, the leader is responsible for ingesting new log entries, replicating them to follower ordering nodes, and managing when an entry is considered committed.

This is not a special **type** of orderer. It is only a role that an orderer may have at certain times, and then not others, as circumstances determine.

## Leading Peer

每一个“组织 [Organization](#)”在其订阅的通道上可以拥有多个节点，其中一个节点会作为通道的主导节点，代表该成员与网络排序服务节点通信。排序服务将区块传递给通道上的主导节点，主导节点再将此区块分发给同一成员集群下的其他节点。

## Ledger

账本由两个不同但相关的部分组成——“区块链”和“状态数据库”，也称为“世界状态”。与其他账本不同，区块链是 **不可变** 的——也就是说，一旦将一个区块添加到链中，它就无法更改。相反，“世界状态”是一个数据库，其中包含已由区块链中的一组经过验证和提交的交易添加，修改或删除的键值对集合的当前值。



认为网络中每个通道都有一个 **逻辑** 账本是有帮助的。实际上，通道中的每个节点都维护着自己的账本副本——通过称为共识的过程与所有其他节点的副本保持一致。术语 **分布式账本技术**（DLT）通常与这种账本相关联——这种账本在逻辑上是单一的，但在一组网络节点（节点和排序服务）上分布有许多相同的副本。

## Log entry

The primary unit of work in a Raft ordering service, log entries are distributed from the leader orderer to the followers. The full sequence of such entries known as the “log”. The log is considered to be consistent if all members agree on the entries and their order.

## Member

参见 [Organization](#) 。

## Membership Service Provider

成员服务提供者（MSP）是指为客户端和节点加入超级账本Fabric网络，提供证书的系统抽象组件。客户端用证书来认证他们的交易；节点用证书认证交易处理结果（背书）。该接口与系统的交易处理组件密切相关，旨在定义成员服务组件，以这种方式可选实现平滑接入，而不用修改系统的交易处理组件核心。



## Membership Services

成员服务在许可的区块链网络上做认证、授权和身份管理。运行于节点和排序服务的成员服务代码均会参与认证和授权区块链操作。它是基于PKI的抽象成员服务提供者（MSP）的实现。

## Ordering Service

Also known as **orderer**. A defined collective of nodes that orders transactions into a block and then distributes blocks to connected peers for validation and commit. The ordering service exists independent of the peer processes and orders transactions on a first-come-first-serve basis for all channels on the network. It is designed to support pluggable implementations beyond the out-of-the-box Kafka and Raft varieties. It is a common binding for the overall network; it contains the cryptographic identity material tied to each **Member**.

## Organization

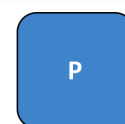
也被称为“成员”，组织被区块链服务提供者邀请加入区块链网络。通过将成员服务提供程序（**MSP**）添加到网络，组织加入网络。**MSP**定义了网络的其他成员如何验证签名（例如交易上的签名）是由该组织颁发的有效身份生成的。**MSP**中身份的特定访问权限由策略控制，这些策略在组织加入网络时也同意。组织可以像跨国公司一样大，也可以像个人一样小。组织的交易终端点是节点 **Peer**。一组组织组成了一个联盟 **Consortium**。虽然网络上的所有组织都是成员，但并非每个组织都会成为联盟的一部分。



An organization, 'ORG'

## Peer

一个网络实体，维护账本并运行链码容器来对账本做读写操作。节点由成员所有，并负责维护。



A peer, 'P'

## Policy

策略是由数字身份的属性组成的表达式，例如：`Org1.Peer OR Org2.Peer`。它们用于限制对区块链网络上的资源的访问。例如，它们决定谁可以读取或写入某个通道，或者谁可以通过ACL使用特定的链码API。在引导排序服务或创建通道之前，可以在 `configtx.yaml` 中定义策略，或者可以在通道上实例化链码时指定它们。示例 `configtx.yaml` 中提供了一组默认策略，适用于大多数网络。

## Private Data

存储在每个授权节点的私有数据库中的机密数据，在逻辑上与通道账本数据分开。通过私有数据收集定义，对数据的访问仅限于通道上的一个或多个组织。未经授权的组织将在通道账本上拥有私有数据的哈希作为交易数据的证据。此外，为了进一步保护隐私，私有数据的哈希值通过排序服务 **Ordering-Service** 而不是私有数据本身，因此这使得私有数据对排序者保密。

## Private Data Collection (Collection)

用于管理通道上的两个或多个组织希望与该通道上的其他组织保持私密的机密数据。集合定义描述了有权存储一组私有数据的通道上的组织子集，这通过扩展意味着只有这些组织才能与私有数据进行交易。

## Proposal

一种通道中针对特定节点的背书请求。每个提案要么是链码的实例化，要么是链码的调用（读写）请求。

## Query

A query is a chaincode invocation which reads the ledger current state but does not write to the ledger. The chaincode function may query certain keys on the ledger, or may query for a set of keys on the ledger. Since queries do not change ledger state, the client application will typically not submit these read-only transactions for ordering, validation, and commit. Although not typical, the client application can choose to submit the read-only transaction for ordering, validation, and commit, for example if the client wants auditable proof on the ledger chain that it had knowledge of specific ledger state at a certain point in time.

## Quorum

This describes the minimum number of members of the cluster that need to affirm a proposal so that transactions can be ordered. For every consenter set, this is a **majority** of nodes. In a cluster with five nodes, three must be available for there to be a quorum. If a quorum of nodes is unavailable for any reason, the cluster becomes unavailable for both read and write operations and no new logs can be committed.

## Raft

New for v1.4.1, Raft is a crash fault tolerant (CFT) ordering service implementation based on the [etcd library](#) of the [Raft protocol](#). Raft follows a “leader and follower” model, where a leader node is elected (per channel) and its decisions are replicated by the followers. Raft ordering services should be easier to set up and manage than Kafka-based ordering services, and their design allows organizations to contribute nodes to a distributed ordering service.

## Software Development Kit (SDK)

超级账本Fabric客户端软件开发包（SDK）为开发人员提供了一个结构化的库环境，用于编写和测试链码应用程序。SDK完全可以通过标准接口实现配置和扩展。它的各种组件：签名加密算法、日志框架和状态存储，都可以轻松地被替换。SDK提供APIs进行交易处理，成员服务、节点遍历以及事件处理。

目前，两个官方支持的SDK用于Node.js和Java，而另外两个——Python和Go——尚非正式，但仍可以下载和测试。

## 智能合约

智能合约是代码——由区块链网络外部的客户端应用程序调用——管理对 [World State](#) 中的一组键值对的访问和修改。在超级账本Fabric中，智能合约被称为链码。智能合约链码安装在节点上并实例化为一个或多个通道。

## State Database



为了从链码中高效的读写查询，当前状态数据存储存储在状态数据库中。支持的数据库包括levelDB和couchDB。

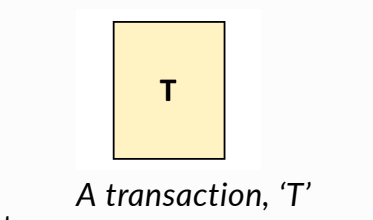
## System Chain

一个在系统层面定义网络的配置区块。系统链存在于排序服务中，与通道类似，具有包含以下信息的初始配置：MSP（成员服务提供者）信息、策略和配置详情。全网中的任何变化（例如新的组织加入或者新的排序节点加入）将导致新的配置区块被添加到系统链中。

系统链可看做是一个或一组通道的公用绑定。例如，金融机构的集合可以形成一个财团（表现为系统链），然后根据其相同或不同的业务计划创建通道。

## Transaction

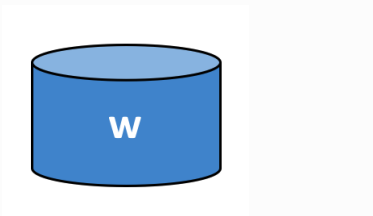
Transactions are created when a chaincode is invoked from a client application to read or write data from the ledger. Fabric application clients submit transaction proposals to endorsing peers for execution and endorsement, gather the signed (endorsed) responses from those endorsing peers, and then package the results and endorsements into a transaction that is submitted to the ordering service. The ordering service orders and places transactions in a block that is broadcast to the peers which validate and commit the transactions to the ledger and update world state.



A transaction, 'T'

## World State

世界状态也称为“当前状态”，是 HyperLedger Fabric **Ledger** 的一个组件。世界状态表示链交易日志中包含的所有键的最新值。链码针对世界状态数据执行交易提案，因为世界状态提供对这些密钥的最新值的直接访问，而不是通过遍历整个交易日志来计算它们。每当键的值发生变化时（例如，当汽车的所有权——“钥匙”——从一个所有者转移到另一个——“值”）或添加新键（创造汽车）时，世界状态就会改变。因此，世界状态对交易流程至关重要，因为键值对的当前状态必须先知道才能更改。对于处理过的区块中包含的每个有效事务，节点将最新值提交到账本世界状态。



The World State, 'W'