超级账本 Fabric 的架构与设计

目前，超级账本 Fabric 架构上核心特性主要包括：解耦了原子排序环节与其他复杂处理环节，消除了网络处理瓶颈，提高可扩展性；解耦交易处理节点的逻辑角色为背书节点、确认节点，可以根据负载进行灵活部署；加强了身份证书管理服务，作为单独的 Fabric CA 项目，提供更多功能；支持多通道特性，不同通道之间的数据彼此隔离，提高隔离安全性；支持可拔插的架构，包括共识、权限管理、加解密、账本机制都模块，支持多种类型；引入系统链码来实现区块链系统的处理，支持可编程和第三方实现。

Fabric 为应用提供了 gRPC API，以及封装 API 的 SDK 供应用调用。应用可以通过 SDK 访问 Fabric 网络中的多种资源，包括账本、交易、链码、事件、权限管理等。应用开发者只需要跟这些资源打交道即可，无需关心如何实现。其中，账本是最核心的结构，记录应用信息，应用则通过发起交易来向账本中记录数据。交易执行的逻辑通过链码来承载。整个网络运行中发生的事件可以被应用访问，以触发外部流程甚至其他系统。权限管理则负责整个过程中的访问控制。账本和交易进一步地依赖核心的区块链结构、数据库、共识机制等技术；链码则依赖容器、状态机等技术；权限管理利用了已有的 PKI 体系、数字证书、加解密算法等诸多安全技术。底层由多个节点组成 P2P 网络，通过 gRPC 通道进行交互，利用 Gossip 协议进行同步。层次化结构提高了架构的可扩展和可插拔性，方便开发者以模块为单位进行开发。超级账本 Fabric 根据交易过程中不同环节的功能，在逻辑上将节点角色解耦为 Endorser 和 Committer，让不同类型节点可以关注处理不同类型的工作负载。在整个交易过程中，各个组件的功能主要为：客户端（App）：客户端应用使用 SDK 来跟 Fabric 网络打交道。首先，客户端从 CA 获取合法的身份证书来加入到网络内的应用通道。发起正式交易前，需要先构造交易提案提交给 Endorser 进行背书；客户端收集到足够的背书支持后可以利用背书构造一个合法的交易请求，发给 Orderer 进行排序处理。客户端还可以通过事件机制来监听网络中消息，来获知交易是否被成功接收。命令行客户端的主要实现代码在 peer/chaincode 目录下。Endorser 节点：主要提供 ProcessProposal（ctx context.Context,signedProp*pb.SignedProposal）（*pb.ProposalResponse,error）方法供客户端调用，完成对交易提案的背书处理。收到来自客户端的交易提案后，首先进行合法性和 ACL 权限检查，检查通过则模拟运行交易，对交易导致的状态变化进行背书并返回结果给客户端。注意网络中可以只有部分节点担任 Endorser 角色。主要代码在 core/endorser 目录下；Committer 节点：负责维护区块链和账本结构。该节点会定期地从 Orderer 获取排序后的批量交易区块结构，对这些交易进行落盘前的最终检查。检查通过后执行合法的交易，将结果写入账本，同时构造新的区块，更新区块中 BlockMetadata [2] 记录交易是否合法等信息。同一个物理节点可以仅作为 Committer 角色运行，也可以同时担任 Endorser 和 Committer 这两种角色。主要实现代码在 core/committer 目录下；Orderer：仅负责排序。为网络中所有合法交易进行全局排序，并将一批排序后的交易组合生成区块结构。Orderer 一般不需要跟账本和交易内容直接打交道。主要实现代码在 orderer 目录下。对外主要提供 Broadcast（srv ab.AtomicBroadcast_BroadcastServer）error 和 Deliver（srv ab.AtomicBroadcast_DeliverServer）error 两个 RPC 方法；CA：负责网络中所有证书的管理，实现标准的 PKI 架构。主要代码在单独的 fabric-ca 项目中。CA 在签发证书后，自身不参与到网络中的交易过程。

超级账本 Fabric 采用了模块化功能设计，面向不同的开发人员提供了不同层面的功能，自下而上可以分为 3 层：网络层：面向系统管理人员。实现 P2P 网络，提供底层构建区块链网络的基本能力，包括代表不同角色的节点和服务；共识机制和权限管理：面向联盟和组织的管理人员。基于网络层的连通，实现共识机制和权限管理，提供分布式账本的基础；业务层：面向业务应用开发人员。基于分布式账本，支持链码、交易等跟业务相关的功能模块，提供更高一层的应用开发支持。网络层通过软、硬件设备，实现了对分布式账本结构的连通支持，包括节点、排序者、客户端等参与角色，还包括成员身份管理、Gossip 协议等支持组件。节点（Peer）的概念最早来自 P2P 分布式网络，意味着在网络中担任一定职能的服务或软件。节点功能可能是对等一致的，也可能是分工合作的。在超级账本 Fabric 网络中，Peer 意味着在网络中负责接受交易请求、维护一致账本的各个 fabric-peer 实例。这些实例可能运行在裸机、虚拟机

甚至容器中。节点之间彼此通过 gRPC 消息进行通信。按照功能角色划分，Peer 可以包括 3 种类型：Endorser：负责对来自客户端的交易提案进行检查和背书；Committer：负责检查交易请求，执行交易并维护区块链和账本结构；Submitter：负责接收交易，转发给排序者，目前未单独出现。这些角色是功能上的划分，彼此并不相互排斥。一般情况下，网络中所有节点都具备 Committer 功能；部分节点具有 Endorser 功能；Submitter 功能则往往集成在客户端进行实现。Peer 节点相关的主要数据结构包括 PeerEndpoint 和 endorserClient。前者代表一个 Peer 节点在网络中的接入端点；后者实现 EndorserClient 接口，代表连接到 Peer 节点的客户端句柄，提供对 Endorser 角色实现的 ProcessProposal（ctx context.Context,signedProp *pb.SignedProposal）（*pb.ProposalResponse, error）方法的访问。排序者，或称为排序节点，负责对所收到的交易在网络中进行全局排序。Orderer 主要提供了 Broadcast（srv ab.AtomicBroadcast_BroadcastServer）error 和 Deliver（srv ab.AtomicBroadcast_DeliverServer）error 两个接口。前者代表客户端将数据发给 Orderer，后者代表从 Orderer 获取到排序后构造的区块结构。客户端可以使用 atomicBroadcastClient 结构访问这两个接口。atomicBroadcastClient 结构维持了一个 gRPC 的双向通道。Orderer 可以支持多通道。不同通道之间彼此隔离，通道内交易相关信息将仅发往加入到通道内的 Peer，从而提高隐私性和安全性。在目前的设计中，所有的交易信息都会从 Orderer 经过，因此，Orderer 节点在网络中必须处于可靠、可信的地位。从功能上看，Orderer 的目的是对网络中的交易分配全局唯一的序号，实际上并不需要交易相关的具体数据内容。因此为了进一步提高隐私性，发往 Orderer 的可以不是完整的交易数据，而是部分信息，比如交易加密处理后的结果，或者仅仅是交易的 Hash 值、Id 信息等。这些改进设计会降低对 Orderer 节点可靠性和安全性的需求。

客户端是用户和应用跟区块链网络打交道的桥梁。客户端主要包括两大职能：操作 Fabric 网络：包括更新网络配置、启停节点等；操作运行在网络中的链码：包括安装、实例化、发起交易调用链码等。这些操作需要跟 Peer 节点和 Orderer 节点打交道。特别是链码实例化、交易等涉及到共识的操作，需要跟 Orderer 交互，因此，客户端往往也需要具备 Submitter 的能力。网络中的 Peer 和 Orderer 等节点则对应提供了 gRPC 远程服务访问接口，供客户端进行调用。目前，除了基于命令行的客户端之外，超级账本 Fabric 已经拥有了多种语言的 SDK。这些 SDK 封装了对底层 gRPC 接口的调用，可以提供更完善的客户端和开发支持，包括 Node.Js、Python、Java、Go 等多种实现。CA 节点负责对 Fabric 网络中的成员身份进行管理。Fabric 网络目前采用数字证书机制来实现对身份的鉴别和权限控制，CA 节点则实现了 PKI 服务，主要负责对身份证书进行管理，包括生成、撤销等。需要注意的是，CA 节点可以提前签发身份证书，发送给对应的成员实体，这些实体在部署证书后即可访问网络中的各项资源。后续访问过程中，实体无须再次向 CA 节点进行请求。因此，CA 节点的处理过程跟网络中交易的处理过程是完全解耦开的，不会造成性能瓶颈。Fabric 网络中的节点之间通过 Gossip 协议来进行状态同步和数据分发。Gossip 协议是 P2P 领域的常见协议，用于进行网络内多个节点之间的数据分发或信息交换。由于其设计简单，容易实现，同时容错性比较高，而被广泛应用到了许多分布式系统。Gossip 协议的基本思想十分简单，数据发送方从网络中随机选取若干节点，将数据发送过去；接收方重复这一过程。这一过程持续下去，网络中所有节点最终都会达到一致。数据传输的方向可以是发送方发送或获取方拉取。

在 Fabric 网络中，节点会定期地利用 Gossip 协议发送它看到的账本的最新的数据，并对发送消息进行签名认证。通过使用该协议，主要实现如下功能：通道内成员的探测：新加入通道的节点可以获知其他节点的信息，并发送 Alive 信息宣布在线；离线节点经过一段时间后可以被其他节点感知。节点之间同步数据：多个节点之间彼此同步数据，保持一致性。另外，Leader 节点从 Orderer 拉取区块数据后，也可以通过 Gossip 传播给通道内其他、节点。