5.2 联盟链的开拓者Fabric

01 Hyperledger项目介绍

1.项目起源

Hyperledger是由Linux基金会主导，成员包括金融、银行、物联网、供应链、制造业和科技产业的领头羊，**旨在促进跨行业的区块链技术的一个开源的全球合作项目。**

2015 年 12 月，开源世界的旗舰——Linux 基金会牵头，联合 30 家初始企业成员（包括 IBM、Accenture、Intel、J.P.Morgan、R3、DAH、DTCC、FUJITSU、HITACHI、SWIFT、Cisco 等），共同宣告了Hyperledger项目的成立。**该项目试图打造一个透明、公开、去中心化的分布式账本项目，作为区块链技术的开源规范和标准，让更多的应用能更容易的建立在区块链技术上。**项目官方信息网站是hyperledger.org，目前已经有超过 120 家全球知名企业和机构（大部分均为各自行业的领导者）宣布加入Hyperledger项目。

02 Hyperledger项目介绍

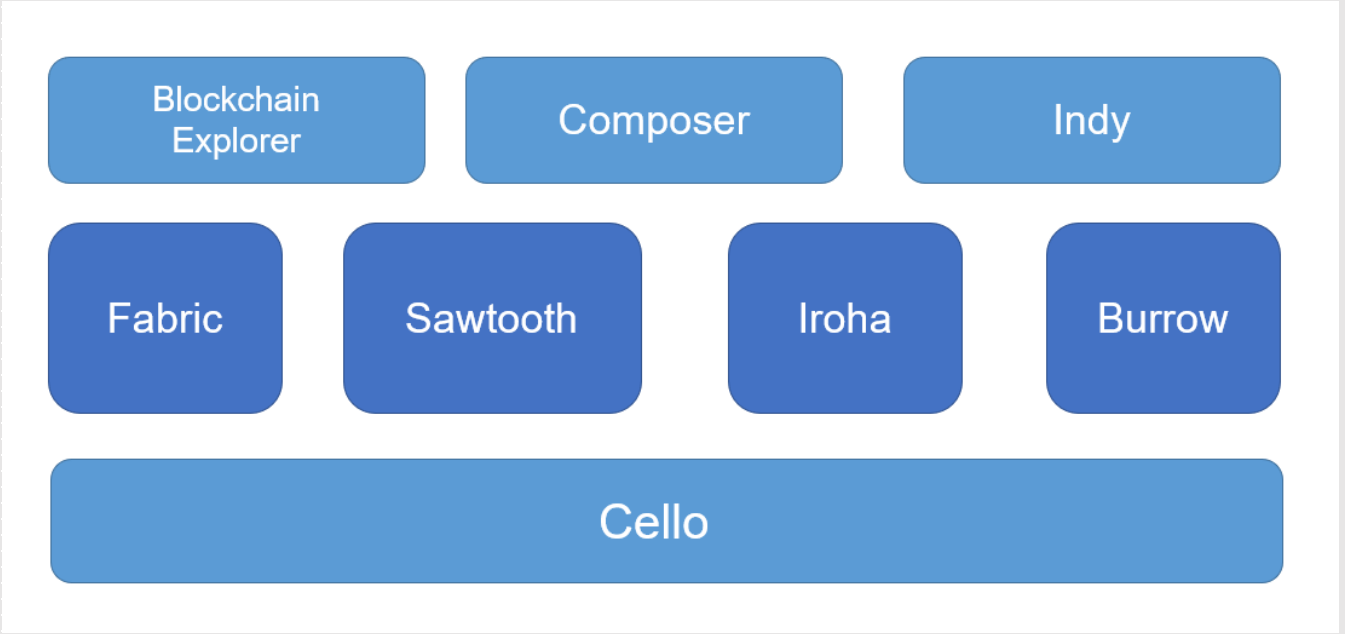
2.Hyperledger的目标

- 创建企业级、开源、分布式账本框架和代码库，以支持商业交易；
- 提供中立、开放和社区驱动的基础设施，由技术和商业合作支持；
- 建设技术社区，开发区块链和共享账本的概念验证（POC）、应用案例并实地跟踪和部署；
- 向大众普及区块链技术，开拓区块链技术的市场。

03 Hyperledger项目介绍

3.Hyperledger的区块链平台

- 主要有8个顶级子项目



04 Hyperledger项目介绍

3.Hyperledger的区块链平台

- BlockChain Explorer 提供Web 操作界面，通过界面快速查看查询绑定区块链的状态（区块个数、交易历史）信息等
- Fabric 包括Fabric 、Fabric CA 、Fabric SDK和fabric-api 等，目标是区块链的基础核心平台，支持PBFT 等新的共识机制，支持权限管理
- Sawtooth 高度模块化的分布式账本平台，支持全新的基于硬件芯片的共识机制ProofofElapsed Time (PoET)
- Iroha 账本平台项目 基于 C++ 实现的轻量级的分布式账本，侧重于Mobile

05 Hyperledger项目介绍

3.Hyperledger的区块链平台

- Cello一个部署管理平台，提供区块链平台的部署和运行时管理功能。使用Cello ，管理员可以轻松部署和管理多条区块链；应用开发者可以无需关心如何搭建和维护区块链，
- Indy ： 提供基于分布式账本技术的数字身份管理机制
- Composer ： 提供面向链码（区块链运行的程序称为链码）开发的高级语言支持，自动生成链码等
- Burrow ： 提供以太坊虚拟机的支持，实现支持高效交易的带权限的区块链平台

06 Hyperledger项目介绍

- 以比特币为代表的货币区块链技术为 1.0，以以太坊为代表的合同区块链技术为 2.0，那么实现了完备的权限控制和安全保障**的 Hyperledger 项目代表着 3.0 时代的到来。**

- 该项目的出现，实际上宣布区块链技术已经不再是仅面向“社会实验”性质的应用场景，它已经正式被主流机构和企业市场认可；

- 同时，**Hyperledger首次提出和实现的完备权限管理、创新的一致性算法和可拔插、可扩展的框架**，对于区块链相关技术和产业的发展都将产生深远的影响。

07 Fabric简述

- Fabric 是最早加入到超级账本项目中的顶级项目，由IBM 、DAH 等企业于2015年底提交到社区。该项目的定位是面向企业的分布式账本平台，创新地引入了权限管理支持，设计上支持可插拔、可扩展，是首个面向联盟链场景的开源项目。**已经先后发布了两个大的版本，0.6实验版本（2016年9月）和1.0正式版本（2017年7月）。**

- **Hyperledger Fabric与其它区块链系统最大的不同体现在私有和许可。**与开放无需许可的网络系统允许未知身份的参与者加入网络不同（需要通过工作量证明协议来保证交易有效并维护网络的安全），Hyperledger Fabric通过Membership Service Provider(MSP)来登记所有的成员。

08 Fabric简述

- Fabric**提供了多个可拔插选项**。账本数据可被存储为多种格式，共识机制可被接入或者断开，同时支持多种不同的MSP。

- Fabric**提供了建立 channel的功能**，这允许参与者为交易新建一个单独的账本。当网络中的一些参与者是竞争对手时，这个功能变得尤为重要。因为这些参与者并不希望所有的交易信息——比如提供给部分客户的特定价格信息——都对网络中所有参与者公开。只有在同一个channel中的参与者，才会拥有该channel中的账本，而其他不在此channel中的参与者则看不到这个账本。

09 Fabric简述

共享账本

Hyperledger Fabric包含一个账本子系统，**这个子系统包含两个组件：世界状态(world state)和交易记录**。并且在Hyperledger Fabric网络中的每一个参与者都拥有一个账本的副本。

- 世界状态组件描述了账本在特定时间点的状态，它是账本的数据库。
- 交易记录组件记录了产生世界状态当前值的所有交易，它是世界状态的更新历史。因此，账本则是世界状态数据库和交易历史记录的集合。

10 Fabric简述

智能合约

Hyperledger Fabric智能合约被称为chaincode，当一个区块链外部的一个应用程序需要访问账本时，就会调用chaincode。大多数情况下，chaincode只会访问账本的数据库组件和世界状态(world state)（比如查询），但不会查询交易记录。

隐私

Hyperledger Fabric支持构建隐私保护严格的网络，也支持构建相对开放的网络。

11 Fabric简述

共识

在网络中，不同的参与者写入的交易必须按照产生顺序依次被写入账本中。要实现这一目标，交易顺序必须被正确的建立并且必须包含拒绝错误交易的方法。

Hyperledger Fabric被设计为允许网络构建者依据业务需求来选择采用哪种共识机制。比如业务上要考虑隐私性，就会有一连串的需求，网络架构可能会从高度结构化的网络调整为更加去中心化的网络。

12 Fabric主要组件

Fabric 的组件包括**客户端（Client），网络节点（Peer），CA（Certificate Authority）节点和排序节点（Orderer）**。

- **客户端：**主要作用是和 Fabric 系统交互，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），此外是用 Fabric SDK 开发的应用客户端。用户通过不同的客户端使用 Fabric 系统的功能。

13 Fabric主要组件

Fabric 的组件包括**客户端（Client），网络节点（Peer），CA（Certificate Authority）节点和排序节点（Orderer）**。

- **网络节点（Peer）**：是区块链去中心化网络中的对等节点，按照功能主要分为背书节点（Endorser）和确认节点（Committer）。背书节点主要对交易预案进行校验、模拟执行和背书。确认节点主要负责检验交易的合法性，并更新和维护区块链数据和账本状态。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。

14 Fabric主要组件

Fabric 的组件包括**客户端（Client），网络节点（Peer），CA（Certificate Authority）节点和排序节点（Orderer）**。

- **CA 节点：**主要给Fabric网络中的成员提供基于数字证书的身份信息，可以生成或取消成员的身份证书（certificate）。在成员身份明确的基础上，Fabric可以实现权限控制的管理。

- **排序节点（Orderer）：**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，发给各个节点把交易持久化到区块链的账本中。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）。

15 Fabric的SDK

在一个部署好节点的Fabric网络中，没有可用的智能合约，也没有默认的可用的区块链账本存在，一切都需要经由SDK来进行后续的部署工作。**超级账本Fabric的SDK为Fabric分布式网络提供了通道创建、智能合约安装、交易调用等一系列的api接口。**只有创建了通道并安装了智能合约，Fabric网络才是一个有效的区块链网络。

因此，SDK是Fabric体系中不可或缺的一部分。经由SDK和Fabric网络节点之间进行信息交互，我们才能够完成交易流程，以及获取区块信息、交易记录等相关信息。

SDK为Fabric区块链提供了丰富的api接口。主要包括四个方面的接口：channel配置、chaincode部署、chaincode调用、systemchaincode调用。

16 Fabric的SDK

- **通道（channel）操作：**有channel创建、更新channel配置、channel加入peer节点操作等。
- **智能合约(chaincode)操作：**主要包括chaincode安装、chaincode实例化、chaincode升级操作。
- **智能合约(chaincode)调用：**通过SDK可以进行交易的提交及查询操作，可以配置丰富的chaincode内部逻辑实现并提供丰富的api接口供SDK调用。
- **BlockEvent(区块事件)的监听：**当一笔交易形成区块并在peer节点落账后，这笔交易才算是真正的完成。当Peer区块落账完成后，会发出BlockEvent事件，可以通过SDK提供的接口对该事件进行监听，用以判断指定交易是否落账成功，并及时反馈给客户端。

17 Fabric的SDK

SDK的交易流程

第一步：客户端通过SDK，将交易打包为交易提案(proposal)，这个提案是一个调用chaincode函数的请求。SDK将交易提案打包成正确结构格式并用用户的秘钥证书产生一个用于这笔交易的唯一签名。SDK将签名提案发送到需要进行背书的peer节点。

第二步：背书peer节点校验交易提案格式有效性、是否被提交过、签名信息有效性、发送该交易的Client端是否是channel上有效的授权用户。校验信息正确的话，peer就会将交易提案作为chaincode函数的输入参数进行调用。经chaincode执行会产生一个交易结果包含了返回值、读集、写集。这时产生的交易结果不会对数据库账本本身产生变化，该交易结果会以提案响应的形式返回给SDK。

18 Fabric的SDK

SDK的交易流程

第三步： SDK搜集所有的响应结果，验证背书peer节点的签名有效性、比较提案响应是否一致。若SDK仅仅是需要用于账本查询，那么应用程序会检查返回结果，并不会向orderer节点提交交易。若Client程序打算提交这笔交易，则会将背书并整理后的交易集进行签名，再发送到orderer节点进行排序，orderer服务会根据时间先后顺序为每一个channel上的交易进行排序处理并形成区块(区块中可以有很多交易，根据区块产生配置规则确定)，最后派发到相关的提交Peer节点。

19 Fabric的SDK

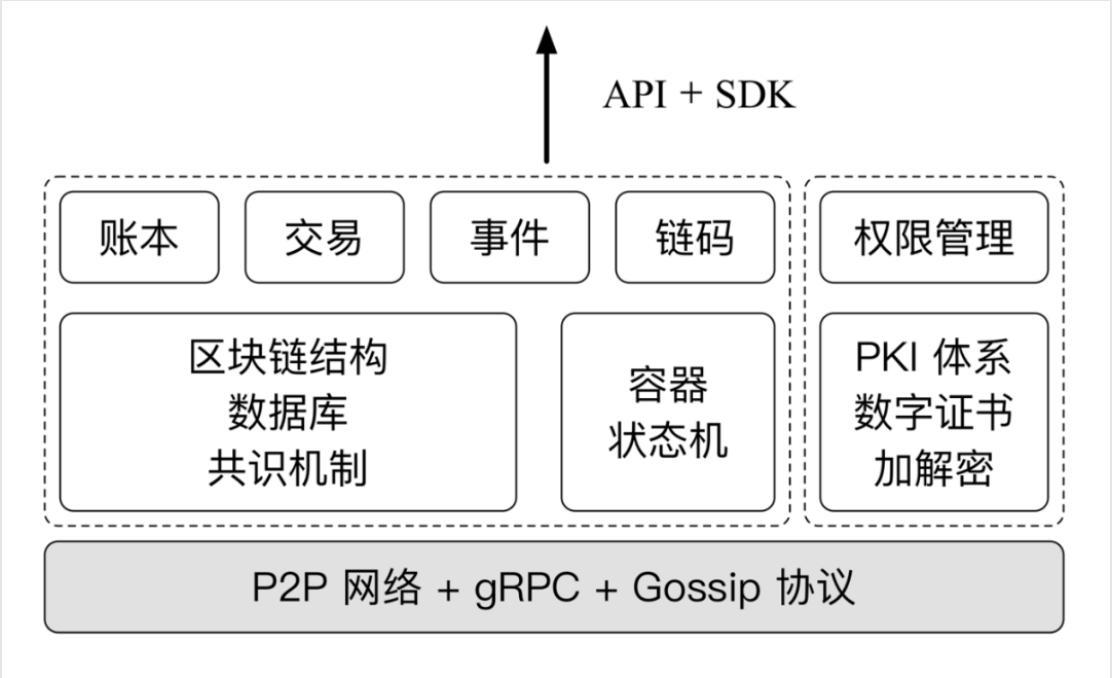
SDK的交易流程

第四步： SDK交易集区块交付到该指定channel的Peer节点上，在区块中的交易需要确认背书策略有效，比较读写集和当前账本状态等来确定该交易在区块中是否有效。当区块在peer节点完成落账后，会发送一个BlockEvent(区块事件)给客户端，由对应的SDK接口进行区块事件的接收。
SDK接收到该区块事件后，对事件进行解析，判断哪些交易执行成功或失败，将执行结果反馈给客户端，至此，一个完整的交易流程结束。

根据交易信息内指定的chaincode信息找到对应的chiancode并将交易发送给该chaincode，chaincode接收到peer发送过来的签名交易后，执行内部智能合约的逻辑。执行完成后，peer会搜集包括交易体、交易结果、操作的读写集等信息打包为一个背书。

20 Fabric的总体架构

Fabric为应用提供了gRPC API（解决了不同语言间通信的复杂性以及环境的不同），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。



21 Fabric的总体架构

Fabric的总体架构中，

账本：是最核心的结构，记录应用信息。

应用：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（Gossip协议是一个最终一致性协议，它是由种子节点发起，当一个种子节点有状态需要更新到网络中的其他节点时，它会随机的选择周围几个节点散播消息，收到消息的节点也会重复该过程，直至最终网络中所有的节点都收到了消息。）