目录

[1 介绍 introduction 4](#_Toc16785)

[1.1 Hyperledger Fabric 4](#_Toc18591)

[1.2 Modularity 模块化 5](#_Toc32515)

[1.3 permissioned 许可 5](#_Toc7201)

[1.4 smart contracts 智能合约 6](#_Toc23315)

[1.5 A New Approach 新方式 7](#_Toc5578)

[1.6 Privacy and Confidentiality 隐私和机密性 7](#_Toc4015)

[1.7 Pluggable Consensus 可插拔性共识 9](#_Toc12752)

[1.8 Performance and Scalability 性能和可扩展性 9](#_Toc12848)

[1.9 Conclusion 结论 10](#_Toc31293)

[1.10 Acknowledgement 致谢 11](#_Toc3642)

[2 使用 getting started 11](#_Toc30402)

[2.1 prerequisites 预备 11](#_Toc21723)

[2.1.1 install cURL 12](#_Toc9197)

[2.1.2 Docker and Docker Compose 12](#_Toc10581)

[2.1.3 Go Programming Language 13](#_Toc30006)

[2.1.4 Node.js Runtime and NPM 14](#_Toc8562)

[2.1.5 Windows extras 15](#_Toc16532)

[2.2 install sample,Binries and Docker images 初始化简单的二进制Docker镜像 17](#_Toc15708)

[2.3 Hyperledger Fabric SDKs 20](#_Toc10088)

[2.4 Hyperledger Fabric CA 20](#_Toc15179)

[3 主要概念 key concepts 21](#_Toc28668)

[3.1 introduction 介绍 21](#_Toc3464)

[3.2 Hyperledger Fabric Functionalities 功能 21](#_Toc7162)

[3.3 Hyperledger Fabric Model 模型 21](#_Toc11665)

[3.4 Hyperledger Fabric Network 网络 21](#_Toc11130)

[3.5 Identity 身份 21](#_Toc7526)

[3.6 Membership 成员 21](#_Toc14744)

[3.7 Peers 节点 21](#_Toc12366)

[3.8 Private data 私有数据 22](#_Toc18104)

[3.9 Ledger 账本 22](#_Toc16008)

[3.10 Use Cases 用例 22](#_Toc26076)

[4 教程 tutorials 22](#_Toc17352)

[4.1 Writing Your First Application 第一个应用 22](#_Toc28088)

[4.2 Building Your First Network 第一个网络 22](#_Toc9850)

[4.3 Adding an Org to a Channel 子链中增加一个组织 22](#_Toc4975)

[4.4 Upgrading Your Network Components 升级网络组件 23](#_Toc5649)

[4.5 Using Private Data in Fabric 使用链码私有数据 23](#_Toc17430)

[4.6 Chaincode Tutorials 链码实例 23](#_Toc8439)

[4.7 Chaincode for Developers 开发链码 23](#_Toc31602)

[4.8 Chaincode for Operators 运营链码 23](#_Toc26330)

[4.9 System Chaincode Plugins 系统链码插件 23](#_Toc2446)

[4.10 Using CouchDB 使用Videos 23](#_Toc12893)

[4.11 Videos 视频 23](#_Toc1564)

[5 运营指导 operation guides 24](#_Toc23944)

[5.1 Upgrading to the Newest Version of Fabric 24](#_Toc26557)

[5.2 Updating a Channel Configuration 24](#_Toc15967)

[5.3 Membership Service Providers (MSP) 24](#_Toc9892)

[5.4 Channel Configuration (configtx) 24](#_Toc1418)

[5.5 Endorsement policies 24](#_Toc19756)

[5.6 Pluggable transaction endorsement and validation 24](#_Toc1786)

[5.7 Access Control Lists (ACL) 24](#_Toc11873)

[5.8 Error handling 24](#_Toc25833)

[5.9 Logging Control 24](#_Toc31114)

[5.10 Securing Communication With Transport Layer Security (TLS) 25](#_Toc6058)

[5.11 Bringing up a Kafka-based Ordering Service 25](#_Toc23344)

[6 相关命令 commands reference 25](#_Toc24864)

[6.1 peer 25](#_Toc7993)

[6.2 peer chaincode 25](#_Toc28893)

[6.3 peer channel 25](#_Toc9494)

[6.4 peer version 25](#_Toc26159)

[6.5 peer logging 25](#_Toc15885)

[6.6 peer node 25](#_Toc1960)

[6.7 configtxgen 26](#_Toc3878)

[6.8 configtxlator 26](#_Toc1003)

[6.9 cryptogen 26](#_Toc6992)

[6.10 Service Discovery Command Line Interface (discover) 26](#_Toc28089)

[6.11 Fabric-CA Commands 26](#_Toc32677)

[7 相关架构 Architcture reference 26](#_Toc30104)

[7.1 Architecture Explained 26](#_Toc31497)

[7.2 Transaction Flow 26](#_Toc13709)

[7.3 Hyperledger Fabric CA's User Guide 26](#_Toc25453)

[7.4 Hyperledger Fabric SDKs 27](#_Toc18361)

[7.5 Service Discovery 27](#_Toc32381)

[7.6 Channels 27](#_Toc26339)

[7.7 Capability Requirements 27](#_Toc8335)

[7.8 CouchDB as the State Database 27](#_Toc4135)

[7.9 Peer channel-based event services 27](#_Toc19447)

[7.10 Private Data 27](#_Toc6000)

[7.11 Read-Write set semantics 27](#_Toc7228)

[7.12 Gossip data dissemination protocol 27](#_Toc27745)

[8 常见问题 frequently asked questions 28](#_Toc25541)

[9 代码贡献 contributions welcome 28](#_Toc31700)

[10 术语 glossary 28](#_Toc22502)

[11 其他 28](#_Toc14883)

[11.1 术语 glossary 28](#_Toc18051)

[11.2 提问 still have questions 38](#_Toc1229)

[11.3 各版本简介 39](#_Toc27042)

[11.3.1 1.0 版本介绍 39](#_Toc15930)

[11.3.2 1.1 版本介绍 39](#_Toc20636)

[11.3.3 1.2 版本介绍 39](#_Toc23722)

[11.3.4 1.3 版本介绍 40](#_Toc21457)

[11.4 Fabric CA 40](#_Toc11670)

[词汇 41](#_Toc28429)

<https://hyperledger-fabric.readthedocs.io/en/release-1.2/whatis.html>

# 1 介绍 introduction

The first and most widely recognized application of blockchain is the [Bitcoin](https://en.wikipedia.org/wiki/Bitcoin) cryptocurrency.Bitcoin and Ethereum fall into a class of blockchain that we would classify as public permissionless blockchain technology. Basically, these are public networks, open to anyone, where participants interact anonymously.

For enterprise use, we need to consider the following requirements:

1 Participants must be identified/identifiable （安全）

2 Networks need to be permissioned （安全）

3 High transaction throughput performance （性能）

4 Low latency of transaction confirmation （性能）

5 Privacy and confidentiality of transactions and data pertaining to business transactions （安全）

## 1.1 Hyperledger Fabric

Hyperledger Fabric is an open source enterprise-grade permissioned distributed ledger technology (DLT) platform, designed for use in enterprise contexts, that delivers some key differentiating capabilities over other popular distributed ledger or blockchain platforms.

--针对企业应用的分布式账本技术

Hyperledger Fabric project by a diverse set of maintainers from multiple organizations. It has a development community that has grown to over 35 organizations and nearly 200 developers since its earliest commits.

--目前有35个组织和200多名开发人员提交代码

Fabric has a highly modular and configurable architecture

--高度模块化可配置化

Fabric is the first distributed ledger platform to support smart contracts authored in general-purpose programming languages such as Java, Go and Node.js, rather than constrained domain-specific languages (DSL).

--第一个用通用开发语言支持的合约，而不需特定领域开发语言

its support for pluggable consensus protocols that enable the platform to be more effectively customized to fit particular use cases and trust models.

--支持可插拔的协议 更有效的客户定制化

Fabric can leverage consensus protocols that do not require a native cryptocurrency to incent costly mining or to fuel smart contract execution.

--fabric无需加密货币及合约运行的燃料

## 1.2 Modularity 模块化

the platform has been designed at its core to be configured to meet the diversity of enterprise use case requirements.

--面向企业工业需求的可配置化设计

Hyperledger Fabric can be configured in multiple ways to satisfy the diverse solution requirements for multiple industry use cases.、

--Fabric能够通过配置满足不同的工业需求

Fabric is comprised of the following modular components:

--fabric包括了一系列模块化组件

1 A pluggable ordering service establishes consensus on the order of transactions and then broadcasts blocks to peers.

插拔式的排序服务确保共识有序并广播其他peer节点

2 A pluggable membership service provider is responsible for associating entities in the network with cryptographic identities.

插拔式的用户管理服务通过加密身份在网络下不同企业联系在一起

3 An optional peer-to-peer gossip service disseminates the blocks output by ordering service to other peers.

可选的gossip点对点的通讯服务 用于节点之间的信息通信

4 Smart contracts (“chaincode”) run within a container environment (e.g. Docker) for isolation. They can be written in standard programming languages but do not have direct access to the ledger state.

合约单独运行在一个docker容器中，相互之间隔离。可能使用标准的编程语言，但不能访问区块状态

5 The ledger can be configured to support a variety of DBMSs.

账本存储支持不同的数据存储管理系统

6 A pluggable endorsement and validation policy enforcement that can be independently configured per application.

一个可插拔的背书策略和策略确认可以在每个应用独立配置

## 1.3 permissioned 许可

In a permissionless blockchain, every participant is anonymous.

In order to mitigate this absence of trust, permissionless blockchains typically employ a “mined” native cryptocurrency or transaction fees to provide economic incentive to offset the extraordinary costs of participating in a form of byzantine fault tolerant consensus based on “proof of work” (PoW).

为了缓解在匿名情况下的信任问题，有了挖矿和gas ，为了抵消拜占庭容错问题有了激励

Permissioned blockchains, on the other hand, operate a blockchain amongst a set of known, identified and often vetted participants operating under a governance model that yields a certain degree of trust.

许可的区块链存在另外的问题，需要运维一系列资料、提供证书并验证参与者的合法性

Additionally, in such a permissioned context, the risk of a participant intentionally introducing malicious code through a smart contract is diminished

另外 在授信环境下，通过合约引入恶意代码的风险减小。

First, the participants are known to one another and all actions

首先参与者都知道对方及行为

submitting application transactions, modifying the configuration of the network or deploying a smart contract are recorded on the blockchain following an endorsement policy that was established for the network and relevant transaction type.

其次 不论提交事务 还是部署合约、修改配置，都要通过网络经过背书策略，

## 1.4 smart contracts 智能合约

A smart contract, or what Fabric calls “chaincode”，。as a trusted distributed application that gains its security/trust from the blockchain

小的智能合约，fabric称为链码。作为一个可信任的分布式应用。

The order-execute architecture can be found in virtually all existing blockchain systems, ranging from public/permissionless platforms such as Ethereum (with PoW-based consensus) to permissioned platforms such as Tendermint, Chain, and Quorum.

共识服务存在于已有的区块链系统中，不论是公开的平台 （如以太坊） 还有私有的平台 如Tendermint、Chain、Quorum

Smart contracts executing in a blockchain that operates with the order-execute architecture must be deterministic，otherwise, consensus might never be reached;

合约的执行在链码上执行必须是没有二义性，否则合约可以不执行

many platforms require that the smart contracts be written in a non-standard, or domain-specific language (such as Solidity) so that non-deterministic operations can be eliminated. This hinders wide-spread adoption because it requires developers writing smart contracts to learn a new language and may lead to programming errors.

合约不需要学习一门新语言，比如solidity，也不会因此引入语法错误。

since all transactions are executed sequentially by all nodes, performance and scale is limited.

所有事务在所有节点必须是有序的，性能和规模因此被限制。

## 1.5 A New Approach 新方式

the transaction flow into three steps:

一个交易遵从以下三步

1 execute a transaction and check its correctness, thereby endorsing it,

执行一个交易 并且确认他的正确性 然后背书他

2 order transactions via a (pluggable) consensus protocol, and

通过共识服务，进行排序交易

3 validate transactions against an application-specific endorsement policy before committing them to the ledger

提交到账本之前，先验证特殊应用的背书策略。

In Fabric, an application-specific endorsement policy specifies which peer nodes, or how many of them, need to vouch for the correct execution of a given smart contract.

在fabric中，不论是全部节点还是特定的指定节点，应用的特定背书策略都要确保执行的正确性。

Thus, each transaction need only be executed (endorsed) by the subset of the peer nodes necessary to satisfy the transaction’s endorsement policy.

因此，每个事务需要仅仅系列认可的背书节点中安全的执行。

This allows for parallel execution increasing overall performance and scale of the system.

他允许并行执行增加了整体性能和系统扩展性。

This first phase also eliminates any non-determinism, as inconsistent results can be filtered out before ordering.

第一个阶段同样消除了任何不确定性，同样不一致性结果在排序之前也被过滤掉。

Because we have eliminated non-determinism, Fabric is the first blockchain technology that enables use of standard programming languages. In the 1.1.0 release, smart contracts can be written in either Go or Node.js, while there are plans to support other popular languages including Java in subsequent releases

因为我们消除了不确定性，Fabric是第一个支持标准编程语言的区块链技术，在1.1.0 release版本中，智能合约可以用Go或node.js编写，有计划支持更多流行的编程语言支持在后续版本中。

## 1.6 Privacy and Confidentiality 隐私和机密性

As we have discussed, in a public, permissionless blockchain network that leverages PoW for its consensus model, transactions are executed on every node.

正如我们所讨论的，在一个公共场所，没有许可的区块链账本利用PoW实现共识，交易在每个节点上执行。

This means that neither can there be confidentiality of the contracts themselves, nor of the transaction data that they process.

这意味着既不能给合约本身加密，也不能给交易数据加密

Every transaction, and the code that implements it, is visible to every node in the network.

针对每笔交易，网络中的每个节点都是可见的。

In this case, we have traded confidentiality of contract and data for byzantine fault tolerant consensus delivered by PoW.

在这种情况下，我们用合约和数据的安全性换取了使用PoW容错替代拜占庭的容错性。

This lack of confidentiality can be problematic for many business/enterprise use cases.

对很多业务/企业来说，缺乏保密性是一个大问题

For example, in a network of supply-chain partners, some consumers might be given preferred rates as a means of either solidifying a relationship, or promoting additional sales.

比如，在一个供应链网络中，一些客户可能需要给予优惠价格，作为巩固关系或促进额外销售。

If every participant can see every contract and transaction, it becomes impossible to maintain such business relationships in a completely transparent network – everyone will want the preferred rates!

如果每个合作伙伴都能看到合约和交易，在一个透明的网络下，大家都想这种折扣，这类关系的维护就变得不可能。

As a second example, consider the securities industry, where a trader building a position (or disposing of one) would not want her competitors to know of this, or else they will seek to get in on the game, weakening the trader’s gambit.

在举一个例子，在证券行业，交易员构建一个头寸买卖测策略，不希望竞争对手知道，否则他们会想办法参与进来，最终会削弱交易结果。

In order to address the lack of privacy and confidentiality for purposes of delivering on enterprise use case requirements, blockchain platforms have adopted a variety of approaches.

为了解决为交付企业用例需求而缺乏隐私和机密性问题，区块链平台采用了多种方法

All have their trade-offs.

所有人都有他们的取舍。

Encrypting data is one approach to providing confidentiality;

加密数据是提供机密性的一种方法

however, in a permissionless network leveraging PoW for its consensus, the encrypted data is sitting on every node.

然后，在没有信任的网络环境中，采用PoW达成共识，每个节点存储加密数据。

Given enough time and computational resource, the encryption could be broken.

只要时间和资源下，加密可能被破解。

For many enterprise use cases, the risk that their information could become compromised is unacceptable.

在很多企业，信息被泄露的风险是不能接受的。

In a permissioned context that can leverage alternate forms of consensus, one might explore approaches that restrict the distribution of confidential information exclusively to authorized nodes.

在一个授权的环境下，可以利用其它形式的协商一致意见，可以探索将机密信息完全分配给授权节点的方法。

In a permissioned context that can leverage alternate forms of consensus, one might explore approaches that restrict the distribution of confidential information exclusively to authorized nodes.

在一个授权的环境下，可以利用其它形式的协商一致意见，可以探索将机密信息完全分配给授权节点的方法。

## 1.7 Pluggable Consensus 可插拔性共识

The ordering of transactions is delegated to a modular component for consensus that is logically decoupled from the peers that execute transactions and maintain the ledger.

交易排序工作委托给了共识组件，该组件把事务执行和账本顺序维护进行逻辑分离。

Specifically, the ordering service. Since consensus is modular, its implementation can be tailored to the trust assumption of a particular deployment or solution.

具体而言，共识服务因为是模块化的，可能根据部署和解决方案在信任场景假设情况来进行定制化。

This modular architecture allows the platform to rely on well-established toolkits for CFT (crash fault-tolerant) or BFT (byzantine fault-tolerant) ordering.

这种模块化架构允许平台依赖成熟的工具包进行CFT(崩溃容错)或BFT(拜占庭容错)的排序

In the currently available releases, Fabric offers a CFT ordering service implemented with Kafka and Zookeeper.

在当前已经发布的版本中，fabric提供了用kafka和zookeepr实现的CFT交易排序服务。

In subsequent releases, Fabric will deliver a Raft consensus ordering service implemented with etcd/Raft and a fully decentralized BFT ordering service.

在接下来的版本中，fabric将发布基于Raft的排序服务使用etcd/Raft算法 和 完全分散的BFT排序服务

Note also that these are not mutually exclusive. A Fabric network can have multiple ordering services supporting different applications or application requirements.

值得注意的是这些排序服务之间不是相互排斥的，一个fabric网络支持不同的排序服务针对不同的业务应用

## 1.8 Performance and Scalability 性能和可扩展性

Performance of a blockchain platform can be affected by many variables such as transaction size, block size, network size, as well as limits of the hardware, etc.

区块链平台的性能受到很多因素的影响，比如事务内容大小、区块大小、磁盘的限制等。

The Hyperledger community is currently developing a draft set of measures within the Performance and Scale working group, along with a corresponding implementation of a benchmarking framework called Hyperledger Caliper.

Hyperledger社区工作小组研发了一系列测量性能和扩展性的草案，以及称之为Hyperledger分类卡的基准实现。

While that work continues to be developed and should be seen as a definitive measure of blockchain platform performance and scale characteristics,

虽然这项研发工作还在进行，不过应该被作为区块链衡量性能和扩展性的标准。

a team from IBM Research has published a peer reviewed paper that evaluated the architecture and performance of Hyperledger Fabric.

IBM的研究团队发布了篇关于区块链节点性能和扩展性评估方面的论文，

The paper offers an in-depth discussion of the architecture of Fabric and then reports on the team’s performance evaluation of the platform using a preliminary release of Hyperledger Fabric v1.1.

该论文深度讨论了fabric体系结构，然后团队对fabric1.1所做了一个初步的评测报告。

The benchmarking efforts that the research team did yielded a significant number of performance improvements for the Fabric v1.1.0 release that more than doubled the overall performance of the platform from the v1.0.0 release levels.

通过基准测试，研究团队已经促从V1.0.0 性能到v1.1.0 整体性能提升了一倍。

## 1.9 Conclusion 结论

Any serious evaluation of blockchain platforms should include Hyperledger Fabric in its short list.

任何有价值的功能都应该包括在区块链短功能列表清单中。

Combined, the differentiating capabilities of Fabric make it a highly scalable system for permissioned blockchains supporting flexible trust assumptions

总而言之，区块链不同的能力使得fabric称为一个高可扩展的持久化区块链系统。

that enable the platform to support a wide range of industry use cases ranging from government, to finance, to supply-chain logistics, to healthcare and so much more.

使得平台更大范围的支持包括政府、金融、供应链物流、医疗保健等

More importantly, Hyperledger Fabric is the most active of the (currently) ten Hyperledger projects. The community building around the platform is growing steadily,

更重要的，fabric 有最多的活跃活跃项目，围绕平台的社区人员稳定增长，

and the innovation delivered with each successive release far out-paces any of the other enterprise blockchain platforms.

并且每次版本的发版 创新元素远远超过其他企业级区块链平台。

Hyperledger Fabric, being a permissioned platform, enables confidentiality through its channel architecture.

Hyperledger Fabric 作为一个授信平台，通过子链架构来确保其机密性。

Basically, participants on a Fabric network can establish a “channel” between the subset of participants that should be granted visibility to a particular set of transactions.

基本上，Fabric网络上的参与者可以在参与者之间建立一个"通道"，通过通道授予事务可见性。

Think of this as a network overlay.

这种授权是整个网络覆盖的。

Thus, only those nodes that participate in a channel have access to the smart contract (chaincode) and data transacted, preserving the privacy and confidentiality of both.

因此，只有那些参与通道的节点才能访问智能契约(链码)和数据事务，从而保护两者的隐私和机密性。

To improve upon its privacy and confidentiality capabilities, Fabric has added support for private data and is working on zero knowledge proofs (ZKP) available in the future. More on this as it becomes available.

为了改进隐私和保密能力，fabric将来的版本中会增加ZKP(零知识证明)的支持。这一点将变得更可用。

## 1.10 Acknowledgement 致谢

“Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains”

https://arxiv.org/abs/1801.10228v1

# 2 使用 getting started

Before we begin, if you haven’t already done so, you may wish to check that you have all the Prerequisites installed on the platform(s) on which you’ll be developing blockchain applications and/or operating Hyperledger Fabric.

在我们开始之间，需要准备开发或运用区块链的平台上,初始化准备工作。

Once you have the prerequisites installed, you are ready to download and install HyperLedger Fabric.

一旦安装了先决条件，就可以下载安装HyperLedger Fabric。

While we work on developing real installers for the Fabric binaries, we provide a script that will Install Samples, Binaries and Docker Images to your system.

为了Fabric二进制文件开发的安装，我们提供了一个初始化脚本来构建简单二进制和docker镜像系统，

The script also will download the Docker images to your local registry.

这个脚本还将下载docker镜像到你的本地注册表

## 2.1 prerequisites 预备

Before we begin, if you haven’t already done so, you may wish to check that you have all the Prerequisites installed on the platform(s) on which you’ll be developing blockchain applications and/or operating Hyperledger Fabric.

在我们开始之间，需要准备开发或运用区块链的平台上,初始化准备工作。

### 2.1.1 install cURL

Download the latest version of the cURL tool if it is not already installed or if you get errors running the curl commands from the documentation.

如果还没有下载环境或者通过命令做了错误的下载，使用cURL 工具下载最新版本

cURL 下载地址

https://curl.haxx.se/download.html

note:注意

If you’re on Windows please see the specific note on Windows extras below.

如果是windos环境，请查看下面windows extras说明

### 2.1.2 Docker and Docker Compose

You will need the following installed on the platform on which you will be operating, or developing on (or for), Hyperledger Fabric:

你需要按下列操作在Fabric区块链开发或运营平台上进行安装

1 MacOSX, \*nix, or Windows 10: Docker Docker version 17.06.2-ce or greater is required.

mac操作系统 或 win10 Docker需要在17.06.2-ce 以上版本

2 Older versions of Windows: Docker Toolbox - again, Docker version Docker 17.06.2-ce or greater is required.

windows更早的版本 ：Docker 工具版本一样 Docker需要在17.06.2-ce 以上版本

You can check the version of Docker you have installed with the following command from a terminal prompt:

可以通过下列终端提示命令，确认docker的版本号

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

docker --version

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Note：注

Installing Docker for Mac or Windows, or Docker Toolbox will also install Docker Compose.

在Mac或Windows安装Docker，Docker Toolbox 一样可以安装Docker Compose。

If you already had Docker installed, you should check that you have Docker Compose version 1.14.0 or greater installed.

如果已经安装了Docker,你应该确认Docker compass的版本应该在1.14.0以上。

If not, we recommend that you install a more recent version of Docker.

如果没有，我们建议您安装Docker的最新版本。

You can check the version of Docker Compose you have installed with the following command from a terminal prompt:

使用下面的终端命令确认Docker Compose版本

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

docker-compose --version

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

### 2.1.3 Go Programming Language

Hyperledger Fabric uses the Go Programming Language for many of its components.

Fabric的很多组件都使用的Go语言

Go version 1.10.x is required.

--Go版本需要在1.10.x

Given that we will be writing chaincode programs in Go, there are two environment variables you will need to set properly;

基于我们要使用Go语言编写链码，有两个环节变量需要设置属性。

you can make these settings permanent by placing them in the appropriate startup file, such as your personal ~/.bashrc file if you are using the bash shell under Linux.

可以通过设置放在适启动位置永久保留，如果使用的linux系统，在 ~/.bashrc 文件中设置。

First, you must set the environment variable GOPATH to point at the Go workspace containing the downloaded Fabric code base, with something like:

首先，你必须设置环境变量GOPATH，用来作为下载fabric代码基础地址 ，比如：

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

export GOPATH=$HOME/go

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Note 注

You must set the GOPATH variable

你必须设置GOPATH变量

Even though, in Linux, Go’s GOPATH variable can be a colon-separated list of directories, and will use a default value of $HOME/go if it is unset,

即使在Linux环境下，Go变量可以采用系列分号分割，如果没有设置并将使用默认变量$HOME/go。

the current Fabric build framework still requires you to set and export that variable, and it must contain only the single directory name for your Go workspace.

当前Fabric构建框架仍需要设置并导入该变量，而且他必须包括Go工作区的单个目录名。

(This restriction might be removed in a future release.)

这个限制可能在将来版本中删除。

Second, you should (again, in the appropriate startup file) extend your command search path to include the Go bin directory, such as the following example for bash under Linux:

其次，您应该扩展命令搜索路径，以包括Go bin 目录，linux采用下面例子

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

export PATH=$PATH:$GOPATH/bin

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

While this directory may not exist in a new Go workspace installation, it is populated later by the Fabric build system with a small number of Go executables used by other parts of the build system.

虽然在新安装Go工作空间这个目录可能不存在，稍后Fabric构建系统将放入一些Go可执行文件，构建系统的其他部分将使用这些文件。

So even if you currently have no such directory yet, extend your shell search path as above.

因此，即使您目前还没有这样的目录，也可以像上面那样扩展shell搜索路径。

If you will be developing applications for Hyperledger Fabric leveraging the Hyperledger Fabric SDK for Node.js, you will need to have version 8.9.x of Node.js installed.

如果要使用Farbric开发应用程序，使用Node.js作为fabric sdk,需要安装node.js 8.9.x以上版本。

### 2.1.4 Node.js Runtime and NPM

If you will be developing applications for Hyperledger Fabric leveraging the Hyperledger Fabric SDK for Node.js, you will need to have version 8.9.x of Node.js installed.

如果要使用Farbric开发应用程序，使用Node.js作为fabric sdk,需要安装node.js 8.9.x以上版本。

note:注

Node.js version 9.x is not supported at this time.

Node.js 9.x版本目前不支持。

Node.js - version 8.9.x or greater

采用Node.js 8.9.x 或更早

note:注

Installing Node.js will also install NPM, however it is recommended that you confirm the version of NPM installed. You can upgrade the npm tool with the following command:

安装node.js还需要安装NPM,需确认一下NPM安装的版本，可以通过下面命令更新npm 工具包

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

npm install npm@5.6.0 -g

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

#### 2.1.4.1 python

note 注

The following applies to Ubuntu 16.04 users only.

下列应用仅在Ubuntu 16.04 版本中使用

By default Ubuntu 16.04 comes with Python 3.5.1 installed as the python3 binary. The Fabric Node.js SDK requires an iteration of Python 2.7 in order for npm install operations to complete successfully.

默认情况下，Ubuntu 16.04 自带python3.5.1二进制安装。Fabric node.js sdk需要Python2.7的迭代才能成功完成npm安装操作。

Retrieve the 2.7 version with the following command:

使用以下命令检索2.7版本

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

sudo apt-get install python

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Check your version(s):

确认你的版本

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

python --version

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

### 2.1.5 Windows extras

If you are developing on Windows 7, you will want to work within the Docker Quickstart Terminal which uses Git Bash and provides a better alternative to the built-in Windows shell.

如果你在win7下开发，如果你希望在Docker Quickstart终端中运行，该终端使用Git Bash，并提供比内置Windows shell更好选择。

However experience has shown this to be a poor development environment with limited functionality.

然而，经验表明这是一个功能有限的开发环境。

It is suitable to run Docker based scenarios, such as Getting Started, but you may have difficulties with operations involving the make and docker commands.

它适合运行基于Docker的场景。例如在开始使用的时候，在make和制作docker命令遇到各种困难。

On Windows 10 you should use the native Docker distribution and you may use the Windows PowerShell.

如果再window10上运行，你需要使用本机Docker发行版，并且可以使用windows PowerShell.

However, for the binaries command to succeed you will still need to have the uname command available.

然而，要使二进制命令成功运行，还需要使用uname命令。

You can get it as part of Git but beware that only the 64bit version is supported.

你可以将其作为Git的一部分获得，但是注意只支持64位版本。

Before running any git clone commands, run the following commands:

在运行git命令以前，需运行以下命令

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

git config --global core.autocrlf false

git config --global core.longpaths true

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

You can check the setting of these parameters with the following commands:

可以使用下列命令确认设置的参数

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

git config --get core.autocrlf

git config --get core.longpaths

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

These need to be false and true respectively.

需要分别分会false和true返回值。

Note：注

If you have questions not addressed by this documentation, or run into issues with any of the tutorials, please visit the Still Have Questions? page for some tips on where to find additional help.

如果在文档中没有提到的问题，或者教程中任何问题，请到下面地址查看，里面有一些额外的提示。

The curl command that comes with Git and Docker Toolbox is old and does not handle properly the redirect used in Getting Started.

Git和Docker工具箱附带的curl命令很旧，不能正确地处理开始时使用的重定向。

Make sure you install and use a newer version from the cURL downloads page

确保安装并使用cURL下载页面的新版本。

For Node.js you also need the necessary Visual Studio C++ Build Tools which are freely available and can be installed with the following command:

为Node.js安装还需要安装Visual Studio C++构建，这些工具都是免费的，可以通过下面命令安装

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

npm install --global windows-build-tools

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

See the NPM windows-build-tools page for more details.

可以从NPM页面查看更多构建细节

https://www.npmjs.com/package/windows-build-tools

Once this is done, you should also install the NPM GRPC module with the following command:

一旦执行完，通过以下命令确认安装NPG GRPC模块

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

npm install --global grpc

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Your environment should now be ready to go through the Getting Started samples and tutorials.

你现在环境现在应该已经准备好进行入门示例和教程。

## 2.2 install sample,Binries and Docker images 初始化简单的二进制Docker镜像

While we work on developing real installers for the Hyperledger Fabric binaries, we provide a script that will download and install samples and binaries to your system.

当下开始fabric二进制文件安装的的话，我们提供了一个脚本用来下载和安装二进制包。

We think that you’ll find the sample applications installed useful to learn more about the capabilities and operations of Hyperledger Fabric.

我们认为你通过简单的运用可以学习更多关于fabric的能力和操作。

|  |
| --- |
| **Note 注** |
| If you are running on Windows you will want to make use of the Docker Quickstart Terminal for the upcoming terminal commands. Please visit the Prerequisites if you haven’t previously installed it.  如果运行在windos上，你想使用Docker Quickstart 终端来执行终端命令，如果之前还没有安装，请先访问先决条件。  If you are running on Windows you will want to make use of the Docker Quickstart Terminal for the upcoming terminal commands. Please visit the Prerequisites if you haven’t previously installed it.  如果运行在windos上，你想使用Docker Quickstart 终端来执行终端命令，如果之前还没有安装，请先访问先决条件。  If you are using Docker Toolbox on Windows 7 or macOS, you will need to use a location under C:\Users (Windows 7) or /Users (macOS) when installing and running the samples.  如果你在win7或macOs下使用docker,你将需要在c:\Users 或 /Users 目录下运行示例  If you are using Docker for Mac, you will need to use a location under /Users, /Volumes, /private, or /tmp. To use a different location, please consult the Docker documentation for file sharing.  如果要在Mac下使用Docker，你需要使用以下目录之一 /Users, /Volumes, /private, or /tmp。如果需要使用不同的目录，请参阅Docker文档共享文件。  If you are using Docker for Windows, please consult the Docker documentation for shared drives and use a location under one of the shared drives.  如果想在windos下使用docker,请参阅Docker文档共享驱动并在共享驱动下使用一个本地目录。 |

Determine a location on your machine where you want to place the fabric-samples repository and enter that directory in a terminal window.

确定机器上要放置fabric-samples存储库的位置，并在终端窗口中输入该目录。

The command that follows will perform the following steps:

下面命令将执行以下步骤：

1 If needed, clone the hyperledger/fabric-samples repository

如果需要，克隆fabric-samples到仓库

2 Checkout the appropriate version tag

检出指定版本

3 Install the Hyperledger Fabric platform-specific binaries and config files for the version specified into the /bin and /config directories of fabric-samples

在fabric-samples的/bin/和/config目录中安装指定版本的超级账本Fabric平台专用二进制文件和配置文件

4 Download the Hyperledger Fabric docker images for the version specified

下载指定版本的fabric镜像

Once you are ready, and in the directory into which you will install the Fabric Samples and binaries, go ahead and execute the following command:

准备完成之后，在准备安装Fabric 实例、二进制、Go环境的目录下，先执行下面命令

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

curl -sSL http://bit.ly/2ysbOFE | bash -s 1.2.1

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

|  |
| --- |
| **Note 注** |
| If you get an error running the above curl command, you may have too old a version of curl that does not handle redirects or an unsupported environment.  如果crul命令将运行错误，你可能是因为curl版本太老没有处理重定向或一个不支持的环境。  Please visit the Prerequisites page for additional information on where to find the latest version of curl and get the right environment.  请访问准备的额外信息页，那里有最新的crul版本并获取正确的环境变量  Alternately, you can substitute the un-shortened URL:  或者，你可以用没有缩写的地址替换  https://raw.githubusercontent.com/hyperledger/fabric/master/scripts/bootstrap.sh |
| You can use the command above for any published version of Hyperledger Fabric. Simply replace 1.2.1 with the version identifier of the version you wish to install.  你可以使用以上命令发行已发布的fabric所有版本，简单把1.2.1 版本号标识替换自己想要的即可。 |

The command above downloads and executes a bash script that will download and extract all of the platform-specific binaries you will need to set up your network and place them into the cloned repo you created above.

上面的命令将下载和执行脚本，脚本会下载平台所需的二进制包，这些二进制文件来构建你的fabric网络，克隆下面文件放入repo中。

It retrieves the following platform-specific binaries:

它检索下列二进制平台文件

configtxgen,

configtxlator,

cryptogen,

idemixgen

orderer,

peer, and

fabric-ca-client

and places them in the bin sub-directory of the current working directory.

并且替换他们在当前目录的子目录下。

You may want to add that to your PATH environment variable so that these can be picked up without fully qualifying the path to each binary. e.g.:

你如果将路径增加到环境变量中，这样就可以不指定路径的情况下获取这些二进制功能。例如

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

export PATH=<path to download location>/bin:$PATH

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Finally, the script will download the Hyperledger Fabric docker images from Docker Hub into your local Docker registry and tag them as ‘latest’.

最后，脚本将从Docker Hub下载fabric镜像到本地注册中心，并标记为'latest'

The script lists out the Docker images installed upon conclusion.

脚本列出了再结束安装的Docker映像。

Look at the names for each image;

看看每个镜像的名称

these are the components that will ultimately comprise our Hyperledger Fabric network.

上面这些组件将组成fabric网络的基础

You will also notice that you have two instances of the same image ID - one tagged as “amd64-1.x.x” and one tagged as “latest”.

你将可以注意到你有两个相同镜像实例，一个标记为“amd64-1.x.x” ，另一个标记为“amd64-1.x.x”

Prior to 1.2.0, the image being downloaded was determined by uname -m and showed as “x86\_64-1.x.x”.

在1.2.0之前，已下载的镜像使用uname -m确定，显示为“x86\_64-1.x.x”.

|  |
| --- |
| note注 |
| On different architectures, the x86\_64/amd64 would be replaced with the string identifying your architecture.  在不同的架构下，x86\_64/amd64 将被替换为标识体系结构的字符串。 |
| If you have questions not addressed by this documentation, or run into issues with any of the tutorials, please visit the Still Have Questions? page for some tips on where to find additional help.  如果遇到任何文档中没有提到的问题，或者运行实例中遇到的问题，请访问遗留问题网，一些tips在一些额外帮助页找到。 |

## 2.3 Hyperledger Fabric SDKs

Hyperledger Fabric offers a number of SDKs to support various programming languages. There are two officially released SDKs for Node.js and Java:

Hyperledger Fabric提供了很多SDK支持各种编程语言，下面提供了二种sdk node.js 和java

1 Hyperledger Fabric Node SDK and Node SDK documentation.

<https://github.com/hyperledger/fabric-sdk-node>

2 Hyperledger Fabric Java SDK.

<https://github.com/hyperledger/fabric-sdk-java>

In addition, there are three more SDKs that have not yet been officially released (for Python, Go and REST), but they are still available for downloading and testing:

另外，还有三种SDK(python\Go\rest)没有发布,不过也可以下载测试

1 Hyperledger Fabric Python SDK.

<https://github.com/hyperledger/fabric-sdk-py>

2 Hyperledger Fabric Go SDK.

<https://github.com/hyperledger/fabric-sdk-go>

3 Hyperledger Fabric REST SDK.

<https://github.com/hyperledger/fabric-sdk-rest>

## 2.4 Hyperledger Fabric CA

Hyperledger Fabric provides an optional certificate authority service that you may choose to use to generate the certificates and key material to configure and manage identity in your blockchain network.

Hyperledger Fabric提供了一个可选的认证授权服务，可以用来生成证书和关键材料用来管理区块链网络中的身份认证。

However, any CA that can generate ECDSA certificates may be used.

然而，任何CA都可以ECDSA 证书

# 3 主要概念 key concepts

## 3.1 introduction 介绍

Hyperledger Fabric is a platform for distributed ledger solutions underpinned by a modular architecture delivering high degrees of confidentiality, resiliency, flexibility, and scalability.

fabric是一个分布式账本解决方案，基础是高度的机密性、弹性、灵活性和可伸缩性的模块化架构。

It is designed to support pluggable implementations of different components and accommodate the complexity and intricacies that exist across the economic ecosystem.

他被设计成不同组件间支持可插拔实现，以适应经济环境的错综复杂性。

We recommend first-time users begin by going through the rest of the introduction below in order to gain familiarity with how blockchains work and with the specific features and components of Hyperledger Fabric.

为了熟悉区块链是如何工作、特征和组件，我们建议从下面的文档阅读开始。

Once comfortable — or if you’re already familiar with blockchain and Hyperledger Fabric — go to Getting Started and from there explore the demos, technical specifications, APIs, etc.

一旦熟悉，或者已经对Fabric和区块链有了解，就可以从那些demo,技术特征、API等方面开始了。

#### What is a Blockchain?

##### A Distributed Ledger

At the heart of a blockchain network is a distributed ledger that records all the transactions that take place on the network.

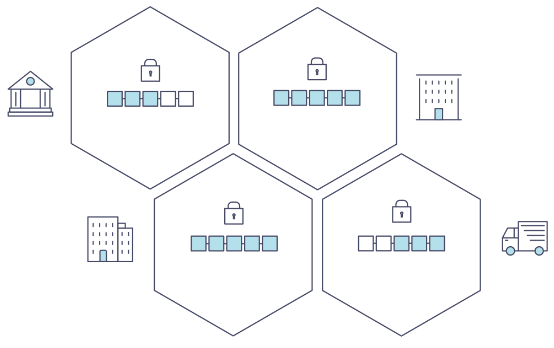
区块链网络的核心是一个分布式账本，账本上记录了所有在网络上的事务。

A blockchain ledger is often described as decentralized because it is replicated across many network participants, each of whom collaborate in its maintenance.

一个区块链环境常常被描述为部署在网络节点上是分散的，维护是集中化的。

We’ll see that decentralization and collaboration are powerful attributes that mirror the way businesses exchange goods and services in the real world.

我们看到在真实商品和服务交换的商业世界，分散和协作是强大的属性。



In addition to being decentralized and collaborative, the information recorded to a blockchain is append-only, using cryptographic techniques that guarantee that once a transaction has been added to the ledger it cannot be modified.

为了达到分散和集群，信息记录只允许追加，使用加密技术确保一旦事务提交就不能被修改。

This property of “immutability” makes it simple to determine the provenance of information because participants can be sure information has not been changed after the fact.

这个不变属性使得确认信息出入变得简单起来，因为参与者可以确信信息没有被修改过。

It’s why blockchains are sometimes described as systems of proof.

这是为什么区块链可以用来证明系统的原因。

##### Smart Contracts

To support the consistent update of information — and to enable a whole host of ledger functions (transacting, querying, etc) — a blockchain network uses smart contracts to provide controlled access to the ledger.

支持信息共识 -- 为了可以使用整个账本功能(交易、查询等) --区块网络使用智能合约去控制访问账本。

[图 略]

Smart contracts are not only a key mechanism for encapsulating information and keeping it simple across the network, they can also be written to allow participants to execute certain aspects of transactions automatically.

智能合约不仅是一种封装信息和保持简单通信的关键机制，他们还可以被允许参与自动执行事务的某些方面。

A smart contract can, for example, be written to stipulate the cost of shipping an item where the shipping charge changes depending on how quickly the item arrives.

例如，一份精明的合约，可以规定运输的费用依据达到的速度。

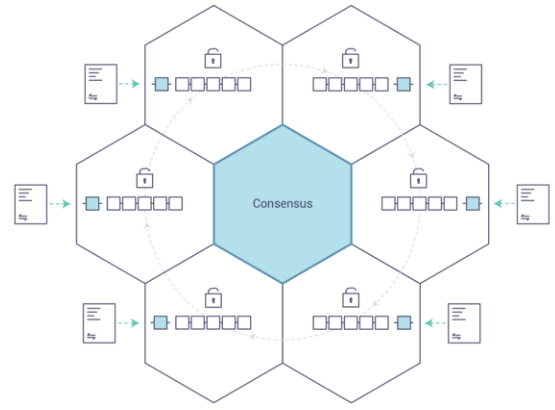
With the terms agreed to by both parties and written to the ledger, the appropriate funds change hands automatically when the item is received.

在双方同意的条款和账本记录的前提下，当商品达到之时自动进行资金的结算。

##### Consensus

The process of keeping the ledger transactions synchronized across the network — to ensure that ledgers update only when transactions are approved by the appropriate participants, and that when ledgers do update, they update with the same transactions in the same order — is called consensus.

跨网络保持账本事务同步的过程--确保账本更新仅仅是事务在被提供者认可的情况下。账本更新具有相同的事务、相同的顺序称为共识。



You’ll learn a lot more about ledgers, smart contracts and consensus later.

你将学习大量的关于账本、智能合约和共识的内容。

For now, it’s enough to think of a blockchain as a shared, replicated transaction system which is updated via smart contracts and kept consistently synchronized through a collaborative process called consensus.

现在，只要认识到区块是一个共享的、可复制的事务系统。通过一个称之为共识的协作过程来更新合约、保持同步一致。

#### Why is a Blockchain useful?

##### Today’s Systems of Record

The transactional networks of today are little more than slightly updated versions of networks that have existed since business records have been kept.

今天的网络事务只是现有商业记录的的略微更新。

The members of a business network transact with each other, but they maintain separate records of their transactions.

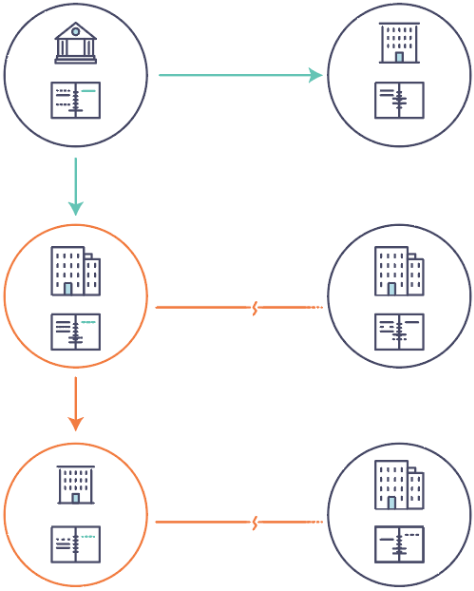
网络交易会员相互进行交易，但是他们彼此独立维护记录。

And the things they’re transacting — whether it’s Flemish tapestries in the 16th century or the securities of today — must have their provenance established each time they’re sold to ensure that the business selling an item possesses a chain of title verifying their ownership of it.

他们交易的事情 --无论是16世界的佛兰德挂毯，还是今天的证券交易 -- 他们每次交易的时候都必须经过一系列验证确认他们的所有权。

What you’re left with is a business network that looks like this

你所剩下就是这样一个商业网络



Modern technology has taken this process from stone tablets and paper folders to hard drives and cloud platforms, but the underlying structure is the same.

现代技术把这个过程从石板、纸质文件转移到了硬盘和云平台，但是这些底层的机构是相同的。

Unified systems for managing the identity of network participants do not exist, establishing provenance is so laborious it takes days to clear securities transactions (the world volume of which is numbered in the many trillions of dollars), contracts must be signed and executed manually, and every database in the system contains unique information and therefore represents a single point of failure.

统一管理网络身份认证系统是不存在的，确认身份来确保交易安全是需要耗费如此大的劳力(世界需要耗资数以万亿美金)，合约必须手动签名和执行，每个数据库在系统中包含唯一的标识，因此还存在单点故障问题。

It’s impossible with today’s fractured approach to information and process sharing to build a system of record that spans a business network, even though the needs of visibility and trust are clear.

今天在信息在支离破碎和流程共享的情况下，构建一个跨越行业的网络记录是不可能的。尽管可见和信任的需求是那么清晰。

##### The Blockchain Difference

What if, instead of the rat’s nest of inefficiencies represented by the “modern” system of transactions, business networks had standard methods for establishing identity on the network, executing transactions, and storing data?

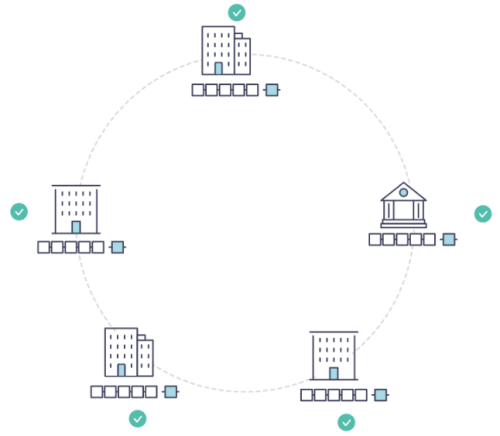
如果业务网络拥有建立身份、执行事务、存储数据能力，而不是现代老鼠窝的低效事务处理，那结果会怎样呢？

What if establishing the provenance of an asset could be determined by looking through a list of transactions that, once written, cannot be changed, and can therefore be trusted?

如果可以通过查看列表来确认资产来源，一旦写入就不能被修改，可以被信任，那又会如何呢？

That business network would look more like this:

商业网络可能看起来像这样



This is a blockchain network, wherein every participant has their own replicated copy of the ledger.

这是一个区块链网络，每个参与者都有他们自己分类账本的副本。

In addition to ledger information being shared, the processes which update the ledger are also shared.

除了账本信息被共享，账本更新过程也被共享。

Unlike today’s systems, where a participant’s private programs are used to update their private ledgers, a blockchain system has shared programs to update shared ledgers.

跟今天的系统不一样，今天系统私有程序去维护更新私有账本，区块链系统采用共享数据更新共享账本。

With the ability to coordinate their business network through a shared ledger, blockchain networks can reduce the time, cost, and risk associated with private information and processing while improving trust and visibility.

通过共享账本可以协调他们业务网络，区块链网络可以压缩时间、成本、风险，同时提高可见性和信任度处理能力。

You now know what blockchain is and why it’s useful. There are a lot of other details that are important, but they all relate to these fundamental ideas of the sharing of information and processes.

你现在知道了区块链是什么和为什么用。这里还有大量的重要细节跟信息共享和处理过程相关。

#### What is Hyperledger Fabric?

The Linux Foundation founded the Hyperledger project in 2015 to advance cross-industry blockchain technologies.

Linux基金会2015年成立fabric工程，用于推进跨行业区块链技术。

Rather than declaring a single blockchain standard, it encourages a collaborative approach to developing blockchain technologies via a community process, with intellectual property rights that encourage open development and the adoption of key standards over time.

不是宣布一个简单的区块链标准，而是鼓励通过社区的方式开发一个区块链技术，鼓励开放开发及随着时间的推移的标准知识产权。

Hyperledger Fabric is one of the blockchain projects within Hyperledger. Like other blockchain technologies, it has a ledger, uses smart contracts, and is a system by which participants manage their transactions.

Fabric是超级账本下面一个区块链项目，跟其他区块链技术一样，有账本存储、有智能合约、还有一个参与者管理其事务的系统。

Where Hyperledger Fabric breaks from some other blockchain systems is that it is private and permissioned.

fabric同其他区块链系统不同之处在于这是私有的及需要授权的。

Rather than an open permissionless system that allows unknown identities to participate in the network (requiring protocols like “proof of work” to validate transactions and secure the network), the members of a Hyperledger Fabric network enroll through a trusted Membership Service Provider (MSP).

超级账本网络成员的登记进入通过会员服务(MSP)注册，而不是开放无许可允许未知授权成员系统(需要工作量证明来验证事务和确保网络)。

Hyperledger Fabric also offers several pluggable options.

Fabric同样提供了几个可插拔的选项。

Ledger data can be stored in multiple formats, consensus mechanisms can be swapped in and out, and different MSPs are supported.

账本数据可以存储在多种形式下，共识机制可以进行交换，支持不同的MSP。

Hyperledger Fabric also offers the ability to create channels, allowing a group of participants to create a separate ledger of transactions.

fabric还提供了创建子链的能力，允许一组参与者创建独立事务账本。

This is an especially important option for networks where some participants might be competitors and not want every transaction they make — a special price they’re offering to some participants and not others, for example — known to every participant.

在网络上这是一个非常重要的功能，有些参与者是竞争对手，不想让对方看到一些内容。比如 有一些合作伙伴之间有一些特别的价格，而另外一些没有，不想每个参与者都参与。

If two participants form a channel, then those participants — and no others — have copies of the ledger for that channel.

如果两个合作伙伴组成一个子链，其他伙伴能参与，这二者之间有分类账副本，而其他参与者没有。

##### Shared Ledger

Hyperledger Fabric has a ledger subsystem comprising two components: the world state and the transaction log.

fabric有一个账本子系统，由世界状态和事务日志两大组件组成。

Each participant has a copy of the ledger to every Hyperledger Fabric network they belong to.

fabric网络中每个参与者都有一个账本副本。

The world state component describes the state of the ledger at a given point in time. It’s the database of the ledger.

世界状态描述的是账本当前节点的状态。这是账本数据库。

The transaction log component records all transactions which have resulted in the current value of the world state; it’s the update history for the world state.

事务日志组件记录了导致当前世界状态的所有历史记录。这是世界状态的更新历史。

The ledger, then, is a combination of the world state database and the transaction log history.

因此，分类账本是世界状态数据和事务历史日志的一个组合

The ledger has a replaceable data store for the world state. By default, this is a LevelDB key-value store database.

账本有一个可替换的世界状态数据存储，默认情绪下，是levelDB key-value进行数据存储。

The transaction log does not need to be pluggable. It simply records the before and after values of the ledger database being used by the blockchain network.

事务日志可插拔需求。就是一个区块网络下区块数据的修改前、修改后的值简单记录。

##### Smart Contracts

Hyperledger Fabric smart contracts are written in chaincode and are invoked by an application external to the blockchain when that application needs to interact with the ledger.

fabric共识规则写在合约代码中，当外部应用需要与账本进行交互时，由外部应用程序调用。

In most cases, chaincode interacts only with the database component of the ledger, the world state (querying it, for example), and not the transaction log.

在大多数情况下，合约代码仅仅与账本世界状态组件进行交互(比如 查询)，而不与事务日志交互。

Chaincode can be implemented in several programming languages. Currently, Go and Node are supported.

合约可以被几种编程语言实现。当前支持Go和Node.

##### Privacy

Depending on the needs of a network, participants in a Business-to-Business (B2B) network might be extremely sensitive about how much information they share.

根据网络的需要，在B2B的网络参与者看呢个对他们共享的数据非常敏感。

For other networks, privacy will not be a top concern.

而对于其他网络，隐私不是首要问题。

Hyperledger Fabric supports networks where privacy (using channels) is a key operational requirement as well as networks that are comparatively open.

跟其他比较开放的网络一样，fabric认为隐私是首要操作需求。

##### Consensus

Transactions must be written to the ledger in the order in which they occur, even though they might be between different sets of participants within the network.

交易事务必须按他们发生的顺序写入账本，即使他们在网络中不同的参与者之间发生。

For this to happen, the order of transactions must be established and a method for rejecting bad transactions that have been inserted into the ledger in error (or maliciously) must be put into place.

为了达到这一点，交易事务必须确认是按顺序的，并且必须建立拒绝错误插入分类账本的不良交易的方法。

This is a thoroughly researched area of computer science, and there are many ways to achieve it, each with different trade-offs.

这是计算机领域一个深入研究的领域，有很多方法去实现它，每种方法都是不同的权衡。

For example, PBFT (Practical Byzantine Fault Tolerance) can provide a mechanism for file replicas to communicate with each other to keep each copy consistent, even in the event of corruption.

例如，PBFT(实用的拜占庭式容错)可以提供一种机制，让文件副本彼此通信，以保持每个副本的一致性，即使在出现损坏的情况下也是如此。

Alternatively, in Bitcoin, ordering happens through a process called mining where competing computers race to solve a cryptographic puzzle which defines the order that all processes subsequently build upon.

或者，在比特币中，排序是通过一个称为挖掘的过程进行的，在这个过程中，相互竞争的计算机竞相解决一个密码谜题，这个谜题定义了所有进程随后构建的顺序。

Hyperledger Fabric has been designed to allow network starters to choose a consensus mechanism that best represents the relationships that exist between participants.

超分类结构被设计成允许网络启动者选择最能代表参与者之间存在的关系的一致机制。

As with privacy, there is a spectrum of needs;

与隐私一样，人们也有各种各样的需求;

from networks that are highly structured in their relationships to those that are more peer-to-peer.

从高度结构化的人际关系网络到更加对等的人际关系网络。

We’ll learn more about the Hyperledger Fabric consensus mechanisms, which currently include SOLO and Kafka.

我们将学到更多关于共识机制，当前包括SOLO和kafka

#### Where can I learn more?

Identity (conceptual documentation)

身份认证(概念性文档)

A conceptual doc that will take you through the critical role identities play in a Fabric network (using an established PKI structure and x.509 certificates).

概念性文档，将带你了解身份在fabric网络中扮演的关键角色(使用PKI结构和x.509证书)

Membership (conceptual documentation)

会员关系(概念文档)

Talks through the role of a Membership Service Provider (MSP), which converts identities into roles in a Fabric network.

通过成员服务提供者(MSP)将角色转换为Fabric网络中的角色。

Peers (conceptual documentation)

对等节点

Peers — owned by organizations — host the ledger and smart contracts and make up the physical structure of a Fabric network.

对等节点--组织结构所拥有 --托管分类账本和智能合约，并构成fabric网络的物理结构。

Building Your First Network (tutorial)

构建你的第一个网络(教程)

Learn how to download Fabric binaries and bootstrap your own sample network with a sample script. Then tear down the network and learn how it was constructed one step at a time.

学习如何下载fabric二进制，使用简单脚本引导构建你的简单网络，然后拆开网络，学习如何一步步构建起来。

Writing Your First Application (tutorial)

编写你的第一个应用(教程)

Deploys a very simple network — even simpler than Build Your First Network — to use with a simple smart contract and application.

部署一个非常简单的网络--甚至比构建你的第一个网络还简单--去使用一个简单的智能合约和应用。

Transaction Flow

交易流

A high level look at a sample transaction flow.

更高角度查看简单示例事务流。

Hyperledger Fabric Model

Fabric 模型

A high level look at some of components and concepts brought up in this introduction as well as a few others and describes how they work together in a sample transaction flow.

从更高层次角度介绍组件和概念，描述一个简单的事务流是如何一起工作的。

## 3.2 Hyperledger Fabric Functionalities 功能

Hyperledger Fabric is an implementation of distributed ledger technology (DLT) that delivers enterprise-ready network security, scalability, confidentiality and performance, in a modular blockchain architecture.

fabric是分布式账本技术(DLT)的一种实现，它在模块化架构方式提供了企业级网络安全、可伸缩性、保密性和性能。

Hyperledger Fabric delivers the following blockchain network functionalities:

fabric提供了下列网络功能：

#### Identity management

To enable permissioned networks, Hyperledger Fabric provides a membership identity service that manages user IDs and authenticates all participants on the network.

为了启用许可的网络，fabric提供了一个成员管理服务，该服务管理用户ID和所有节点的授权。

Access control lists can be used to provide additional layers of permission through authorization of specific network operations.

访问控制列表可以增加额外的控制，通过特定网络授权操作。

For example, a specific user ID could be permitted to invoke a chaincode application, but be blocked from deploying new chaincode.

例如，一个特别的用户ID可以授权区块应用，但不能用来部署新的链代码。

#### Privacy and confidentiality

Hyperledger Fabric enables competing business interests, and any groups that require private, confidential transactions, to coexist on the same permissioned network.

fabric促使相互竞争商业利益，已经任何需求私人、机密交易的团体在相同的私有网络上共存。

Private channels are restricted messaging paths that can be used to provide transaction privacy and confidentiality for specific subsets of network members.

私有通道是被限制的消息传递路径，用来提供网络成员子集提供事务隐私和机密性。

All data, including transaction, member and channel information, on a channel are invisible and inaccessible to any network members not explicitly granted access to that channel.

所有数据，包括事务、会员、通道信息，对未经授权的通道会员是不可见和不可访问的。

#### Efficient processing

Hyperledger Fabric assigns network roles by node type.

fabric根据网络类型分配网络角色。

To provide concurrency and parallelism to the network, transaction execution is separated from transaction ordering and commitment.

为了提高网络并行性和并发性，事务执行从事务排序和提交中分离出来。

Executing transactions prior to ordering them enables each peer node to process multiple transactions simultaneously.

在排序服务前预先执行事务，为了可以使每个对等节点同时执行多个事务。

This concurrent execution increases processing efficiency on each peer and accelerates delivery of transactions to the ordering service.

并行执行增加每个节点的处理效率，并加速事务向排序服务的加速分发。

In addition to enabling parallel processing, the division of labor unburdens ordering nodes from the demands of transaction execution and ledger maintenance, while peer nodes are freed from ordering (consensus) workloads.

为了支持并行处理，劳动分工把排序服务对事务执行和账本维护，对等节点将从共识服务中解放出来。

This bifurcation of roles also limits the processing required for authorization and authentication;

这种分工不同，同样对授权和验证的处理不同。

all peer nodes do not have to trust all ordering nodes, and vice versa, so processes on one can run independently of verification by the other.

所有对等节点不必信任所有共识节点，反之亦然，一个节点的处理可以独立验证其他节点。

#### Chaincode functionality

Chaincode applications encode logic that is invoked by specific types of transactions on the channel.

合约应用编码逻辑是在通道上被特定类型事务调用。

Chaincode that defines parameters for a change of asset ownership, for example, ensures that all transactions that transfer ownership are subject to the same rules and requirements.

例如，定义资产变更的参数，确保所有的事务交易关系遵循相同规则和要求。

System chaincode is distinguished as chaincode that defines operating parameters for the entire channel.

系统链码与定义整个通道操作参数的链码区分开来。

Lifecycle and configuration system chaincode defines the rules for the channel;

生命周期和配置的系统链码定义了通道的规则。

endorsement and validation system chaincode defines the requirements for endorsing and validating transactions.

背书和验证链码定义了背书需求和事务验证需求。

#### Modular design

Hyperledger Fabric implements a modular architecture to provide functional choice to network designers.

fabric实现了模块化架构，为网络提供者提供了可选择性功能。

Specific algorithms for identity, ordering (consensus) and encryption, for example, can be plugged in to any Hyperledger Fabric network.

身份、共识、加密等等特定的算法，可以被插拔进任何fabri网络中。

The result is a universal blockchain architecture that any industry or public domain can adopt, with the assurance that its networks will be interoperable across market, regulatory and geographic boundaries.

其结果是一个通用的区块链架构可以适用于任何产业或可以适配的公共领域，确保其可以跨市场、规则、地理边界进行互操作。

## 3.3 Hyperledger Fabric Model 模型

This section outlines the key design features woven into Hyperledger Fabric that fulfill its promise of a comprehensive, yet customizable, enterprise blockchain solution:

本章概述了编入fabric的关键特征，该特征做出全面、可定制、企业级区块链的承诺：

1 Assets — Asset definitions enable the exchange of almost anything with monetary value over the network, from whole foods to antique cars to currency futures.

资产-资产定义了几乎所有有价值的都系都可以在网上通过货币进行交互，从食物到古董车到期货。

2 Chaincode — Chaincode execution is partitioned from transaction ordering, limiting the required levels of trust and verification across node types, and optimizing network scalability and performance.

链码-链码执行是从事务顺序划分、请求信任等级的限制和节点类型的验证、提升网络可扩展性和性能。

3 Ledger Features — The immutable, shared ledger encodes the entire transaction history for each channel, and includes SQL-like query capability for efficient auditing and dispute resolution.

账本功能 - 不变的、可共享的账本编码为每个通道编码了每个子链的整个交易历史，其中为高效审计和解决冲突提供了类似SQL的查询能力

4 Privacy — Channels and private data collections enable private and confidential multi-lateral transactions that are usually required by competing businesses and regulated industries that exchange assets on a common network.

隐私 - 子链和私有技术集使得私人和多边贸易称为可能。在一个普通网络上交易通常需要接受监管。

5 Security & Membership Services — Permissioned membership provides a trusted blockchain network, where participants know that all transactions can be detected and traced by authorized regulators and auditors.

安全与会员许可 - 获得许可的会员提供一个受信任的区块网络 ，参与者知道所有的交易，可以被审计人员检查和跟踪。

6 Consensus — A unique approach to consensus enables the flexibility and scalability needed for the enterprise.

协商一致 - 一个独特方法为企业提供了所需的灵活性和可伸缩性。

#### Assets

Assets can range from the tangible (real estate and hardware) to the intangible (contracts and intellectual property).

资产可以是有形的(房地产和硬件)，也可以是无形的(合同和知识产权)。

Hyperledger Fabric provides the ability to modify assets using chaincode transactions.

超级账本提供了使用链码事务修改资产的能力。

Assets are represented in Hyperledger Fabric as a collection of key-value pairs, with state changes recorded as transactions on a Channel ledger.

资产在超级账本结构中表示为键-值对的集合，状态变更记录作为子链上的事务。

Assets can be represented in binary and/or JSON form.

资产可以被描写为二进制或者JSON格式。

You can easily define and use assets in your Hyperledger Fabric applications using the Hyperledger Composer tool.

使用hyperledger比对工具，你可以很容易定义和使用资产在你的fabric应用。

#### Chaincode

Chaincode is software defining an asset or assets, and the transaction instructions for modifying the asset(s);

链码时软件定义了一个或多个资产，及修改资产的事务结构。

in other words, it’s the business logic.

换句话说，这是业务逻辑。

Chaincode enforces the rules for reading or altering key-value pairs or other state database information.

链码强制读取或者变更键值对或者另外的状态数据信息。

Chaincode functions execute against the ledger’s current state database and are initiated through a transaction proposal.

Chaincode函数执行当前账本状态数据库，并执行一个事务提议。

Chaincode execution results in a set of key-value writes (write set) that can be submitted to the network and applied to the ledger on all peers.

链代码的执行会产生一组键值集(写入集)，可以将其提交到网络并应用于所有对等节点中。

#### Ledger Features

The ledger is the sequenced, tamper-resistant record of all state transitions in the fabric.

分类账本是有序、有序的事务状态记录。

State transitions are a result of chaincode invocations (‘transactions’) submitted by participating parties.

事务状态是链码执行的结果。

Each transaction results in a set of asset key-value pairs that are committed to the ledger as creates, updates, or deletes.

每个事务结果都是一系列键值对，包括创建、更新、删除。

The ledger is comprised of a blockchain (‘chain’) to store the immutable, sequenced record in blocks, as well as a state database to maintain current fabric state.

账本有一系列不变的、有序的账本记录组成，以及一个维护当前fabric状态的数据库。

There is one ledger per channel.

每个通道是一个分类账本。

Each peer maintains a copy of the ledger for each channel of which they are a member.

每个对等节点维护一个账本副本为各自的会员所属的子链。

Some features of a Fabric ledger:

fabric 账本的一些特征

1 Query and update ledger using key-based lookups, range queries, and composite key queries

查询，基于键值对的更新，范围查询和符合查询

2 Read-only queries using a rich query language (if using CouchDB as state database)

只读查询支持富查询。(如果使用couchDb作为数据存储)

3 Read-only history queries — Query ledger history for a key, enabling data provenance scenarios

只读历史查询。--支持关键字的历史账本查询，支持数据来源场景。

4 Transactions consist of the versions of keys/values that were read in chaincode (read set) and keys/values that were written in chaincode (write set)

事务包括区块链键值对读的版本和区块键值对写的版本。

5 Transactions contain signatures of every endorsing peer and are submitted to ordering service

事务包括了每次背书的签名和每次共识的提交。

6 Transactions are ordered into blocks and are “delivered” from an ordering service to peers on a channel

事务是按顺序给区块的，在子链里从共识服务传递给对等节点。

7 Peers validate transactions against endorsement policies and enforce the policies

peer根据背书策略验证事务并执行事务

8 Prior to appending a block, a versioning check is performed to ensure that states for assets that were read have not 、changed since chaincode execution time

添加区块之前，执行版本控制检查，确保资产读取状态没有被改变。

9 There is immutability once a transaction is validated and committed

事务一旦验证和提交将不可改变。

10 A channel’s ledger contains a configuration block defining policies, access control lists, and other pertinent information

通道的分类账本包括区块定义策略，访问控制列表，和其他相关项。

11 Channels contain Membership Service Provider instances allowing for crypto materials to be derived from different certificate authorities

通道包含成员服务提供者实例，允许从不同的证书颁发机构获取加密材料。

See the Ledger topic for a deeper dive on the databases, storage structure, and “query-ability.”

深入有关数据库、存储结构、查询能力，请参阅超级账本专题

#### Privacy

Hyperledger Fabric employs an immutable ledger on a per-channel basis, as well as chaincode that can manipulate and modify the current state of assets (i.e. update key-value pairs).

Fabric在每个通道上使用不可变账本，以及可以操纵和修改资产当前状态的链码。

A ledger exists in the scope of a channel — it can be shared across the entire network (assuming every participant is operating on one common channel) — or it can be privatized to include only a specific set of participants.

分类账本存在一个子链范围内，--可以被跨网络共享访问（假设所有参与者在一个公共通道上运行），--或者他可以私有化，只包含一组特定参与者。

In the latter scenario, these participants would create a separate channel and thereby isolate/segregate their transactions and ledger.

在后续方案中，特定参与者可以创建一个单独的通道，从而隔离他们事务和账本。

In order to solve scenarios that want to bridge the gap between total transparency and privacy, chaincode can be installed only on peers that need to access the asset states to perform reads and writes (in other words, if a chaincode is not installed on a peer, it will not be able to properly interface with the ledger).

为了解决场景之间差距总透明和隐私性，链码只能安装在背书节点中。需要访问执行读取和写入的资产状态。（换句话说，如果链码不是安装在子链上，将无法正确与总账接口对接）

When a subset of organizations on that channel need to keep their transaction data confidential, a private data collection (collection) is used to segregate this data in a private database, logically separate from the channel ledger, accessible only to the authorized subset of organizations.

当一个子链上的组织子集保持事务数据的机密性时，私有数据需要从私有数据库中分离。

Thus, channels keep transactions private from the broader network whereas collections keep data private between subsets of organizations on the channel.

因此，通道使事务在不同子链更广泛的网络中数据保持私有。

To further obfuscate the data, values within chaincode can be encrypted (in part or in total) using common cryptographic algorithms such as AES before sending transactions to the ordering service and appending blocks to the ledger.

为了进一步混淆数据，链码可以使用加密算法进行混淆(如AES),在向排序服务和传递给背书节点之前.

Once encrypted data has been written to the ledger, it can be decrypted only by a user in possession of the corresponding key that was used to generate the cipher text.

一旦加密数据写入账本中，只有拥对应密钥的的用户才能解密这些数据。

See the Private Data topic for more details on how to achieve privacy on your blockchain network.

关于在区块架构在网络隐私上更多的细节，请参阅私有数据主题。

#### Security & Membership Services

Hyperledger Fabric underpins a transactional network where all participants have known identities.

fabric支持一个所有参与者都知道身份的交易网络。

Public Key Infrastructure is used to generate cryptographic certificates which are tied to organizations, network components, and end users or client applications.

公钥设备基础用来生成与组织、网络组件、终端用户和客户端应用相关联的加密证书。

As a result, data access control can be manipulated and governed on the broader network and on channel levels.

因此，数据访问控制可以更广泛的网络和通道级别上操作和管理数据访问控制。

This “permissioned” notion of Hyperledger Fabric, coupled with the existence and capabilities of channels, helps address scenarios where privacy and confidentiality are paramount concerns.

超级账本许可概念，加上通道存在和功能，有助于提供隐私性和机密性的重要场景。

See the Membership Service Providers (MSP) topic to better understand cryptographic implementations, and the sign, verify, authenticate approach used in Hyperledger Fabric.

为了更好的理解fabric中密码实现，验证和授权验证，请参阅MSP主题。

注：安全性、私密性的实现依赖 基于PKI的授权体系和子链隔离。

#### Consensus

In distributed ledger technology, consensus has recently become synonymous with a specific algorithm, within a single function.

在分布式账本技术中，合约已经变成了特定算法的同义词。

However, consensus encompasses more than simply agreeing upon the order of transactions, and this differentiation is highlighted in Hyperledger Fabric through its fundamental role in the entire transaction flow, from proposal and endorsement, to ordering, validation and commitment.

然而，共识不仅仅是简单交易顺序维护，在整个事务流、从背书到共识、验证和提交中扮演重要角色。

In a nutshell, consensus is defined as the full-circle verification of the correctness of a set of transactions comprising a block.

简而言之，共识的定义是对包含块的一组事务的正确性进行全循环验证。

Consensus is achieved ultimately when the order and results of a block’s transactions have met the explicit policy criteria checks.

当块事务和排序满足显示策略检查时，最终达成一致意见。

These checks and balances take place during the lifecycle of a transaction, and include the usage of endorsement policies to dictate which specific members must endorse a certain transaction class, as well as system chaincodes to ensure that these policies are enforced and upheld.

当检查和平衡在事务生命周期中，包括背书策略来规定哪些特定的成员必须背书某个事务类，以及使用系统链码来确保这些策略得到执行和维护。

Prior to commitment, the peers will employ these system chaincodes to make sure that enough endorsements are present, and that they were derived from the appropriate entities.

在作出承诺之前，对等节点都将使用哪些系统链码有足够的背书，并确保背书来自适当的实体。

Moreover, a versioning check will take place during which the current state of the ledger is agreed or consented upon, before any blocks containing transactions are appended to the ledger.

此外，还将进行版本控制检查，在此期间，将任何包含事务的快添加到分类账本之前，对分类账的当前状态达成一致性同意。

This final check provides protection against double spend operations and other threats that might compromise data integrity, and allows for functions to be executed against non-static variables.

最后检查提供了可能危及数据完整性的双花费用或其他威胁，允许对非静态变量执行函数。

In addition to the multitude of endorsement, validity and versioning checks that take place, there are also ongoing identity verifications happening in all directions of the transaction flow.

除了进行大量背书，有效性和版本控制之外，事务流各个方向要进行身份认证。

Access control lists are implemented on hierarchical layers of the network (ordering service down to channels), and payloads are repeatedly signed, verified and authenticated as a transaction proposal passes through the different architectural components.

访问控制是基于网络分层的(订阅服务到子链)上实现的，当事务提案通过不同的体系结构组件时，负载将被反复签名、验证、验证。

To conclude, consensus is not merely limited to the agreed upon order of a batch of transactions;

综上所述，共识不仅仅局限于达成共识的一批事务记录。

rather, it is an overarching characterization that is achieved as a byproduct of the ongoing verifications that take place during a transaction’s journey from proposal to commitment.

相反，它是事务从建议到承诺的过程进行验证的副产品。

## 3.4 Hyperledger Fabric Network 网络

#### What is a Fabric Network?

#### Who should read this?

#### The business requirements -- Example

#### Components of a Network

#### Creating the Network

#### Defining a Consortium

#### Creating a channel for a consortium

#### Peers and Channels

#### Applications and Smart Contracts

#### Growing the network

#### Simplifying the visual vocabulary

#### Adding another consortium definition

#### Adding a new channel

#### Adding another peer

#### Joining a peer to multiple channels

#### Network fully formed

## 3.5 Identity 身份

#### What is an Identity

#### A Simple Scenario

#### What are PKIs

#### Digital Certificates

#### Authentication, Public keys, and Private Keys

#### Certificate Authorities

#### Certificate Revocation Lists

## 3.6 Membership 成员

If you’ve read through the documentation on identity you’ve seen how a PKI can provide verifiable identities through

a chain of trust. Now let’s see how these identities can be used to represent the trusted members of a blockchain

network.

如果你已经阅读了身份认证的文档，你将清楚身份认证在区块链网络中如果通过PKI提供身份验证。现在，让我们看看在区块链网络的成员管理中，数字证书为什么可以确定是可信任的。

This is where a Membership Service Provider (MSP) comes into play — it identifies which Root CAs and Intermediate CAs are trusted to define the members of a trust domain, e.g., an organization, either by listing the

identities of their members, or by identifying which CAs are authorized to issue valid identities for their members, or

— as will usually be the case — through a combination of both.

这是MSP发挥作用的地方，根CA和中间CA证书定义可信域的成员。例如，一个组织，要么通过列出成员的身份，要么通过授权验证CA的有效性，或者两者的结合。

The power of an MSP goes beyond simply listing who is a network participant or member of a channel. An MSP

can identify specific roles an actor might play either within the scope of the organization the MSP represents (e.g.,

admins, or as members of a sub-organization group), and sets the basis for defining access privileges in the context of

a network and channel (e.g., channel admins, readers, writers).

MSP的功能远不是简单的列出网络参与者或子链会员。MSP可以确定参与者在MSP所代表的组织范围内可能扮演的特定角色(例如 管理员、子组织人员)，并为上下文中定义的访问权限(例如，通道管理员、只读用户、可写用户)。

The configuration of an MSP is advertised to all the channels where members of the corresponding organization

participate (in the form of a channel MSP). In addition to the channel MSP, peers, orderers, and clients also maintain

a local MSP to authenticate member messages outside the context of a channel and to define the permissions over a

particular component (who has the ability to install chaincode on a peer, for example).

MSP的配置告诉所有的子链在组织中有哪些参与者。另外，在子链的MSP，peer，orderer和client同样维护了本地MSP授权会员，用于子链外的消息授权和定义组件权限(比如，谁有能力去初始化子链在peer节点上)。

In addition, an MSP can allow for the identification of a list of identities that have been revoked — as discussed in the

Identity documentation — but we will talk about how that process also extends to an MSP.

另外，正如身份文档中讨论的一样，MSP同样可以用来鉴定已撤销的身份列表。但我们将讨论这个过程如何扩展到MSP中的。

We’ll talk more about local and channel MSPs in a moment. For now let’s see what MSPs do in general.

一会我们将讨论本地MSP和子链中的MSP，现在让我们查看MSP是如何工作的。

#### Mapping MSPs to Organizations

An organization is a managed group of members. This can be something as big as a multinational corporation or a

small as a flower shop. What’s most important about organizations (or orgs) is that they manage their members under

a single MSP. Note that this is different from the organization concept defined in an X.509 certificate, which we’ll talk

about later.

组织是一组收管理的成员组成。组织可以是跨国公司一样的大公司也可以花店一样的小店，组织最重要的是在MSP下管理他们的成员。值得注意的是，组织的概念同X.509证书中定义的不一样，稍后我们讨论这个概念。

The exclusive relationship between an organization and its MSP makes it sensible to name the MSP after the organization, a convention you’ll find adopted in most policy configurations. For example, organization ORG1 would likely have an MSP called something like ORG1-MSP. In some cases an organization may require multiple membership groups — for example, where channels are used to perform very different business functions between organizations. In these cases it makes sense to have multiple MSPs and name them accordingly, e.g., ORG2-MSP-NATIONAL and ORG2-MSP-GOVERNMENT, reflecting the different membership roots of trust within ORG2 in the NATIONAL sales channel compared to the GOVERNMENT regulatory channel.

组织和MSP的排斥关系使得命名是敏感，这个约定被大多数配置策略所接受。例如，组织ORG1 在MSP中被命名为ORG1-MSP.在另外的一些情况，组织可能需要多个会员关系组，例如，子链被用来组织之间执行非常不同的业务功能。在这种情况下，多MPS及相互适配是有意义的。例如，ORG2-MSP-NATIONAL 和 ORG2-MSP-GOVERNMENT，在国际销售渠道和政府销售渠道拥有不同的意义。

Two different MSP configurations for an organization. The first configuration shows the typical relationship between an MSP and an organization — a single MSP defines the list of members of an organization. In the second configuration, different MSPs are used to represent different organizational groups with national, international, and governmental affiliation.

组织的两种不同的MSP配置，第一种配置展示了MSP和组织典型关系，一个MSP定义组织的用户列表。第二种配置，不同MSP代表不同的而组织关系：国际、国内、政府。

Organizational Units and MSPs

An organization is often divided up into multiple organizational units (OUs), each of which has a certain set of responsibilities. For example, the ORG1 organization might have both ORG1-MANUFACTURING and ORG1-DISTRIBUTION OUs to reflect these separate lines of business. When a CA issues X.509 certificates, the OU field in the certificate specifies the line of business to which the identity belongs.

一个组织通常划分成多个组织单元，每个组织单元都拥有一组特定的职责。例如，ORG1组织有ORG1-MANUFACTURING 和ORG1-DISTRIBUTION两个组织单元，反应不同的业务线。当CA发行X.509证书时，证书的OU字段指定身份所属的业务部门。

We’ll see later how OUs can be helpful to control the parts of an organization who are considered to be the members of a blockchain network. For example, only identities from the ORG1-MANUFACTURING OU might be able to access a channel, whereas ORG1-DISTRIBUTION cannot.

稍后我们将看到，OUs如何有助于控制管理组织中区块链网络成员的部分。例如，只有来自组织ORG1-MANUFACTURING的证书单元可能访问子链，而ORG1-DISTRIBUTION不能。

Finally, though this is a slight misuse of OUs, they can sometimes be used by different organizations in a consortium to distinguish each other. In such cases, the different organizations use the same Root CAs and Intermediate CAs for their chain of trust, but assign the OU field to identify members of each organization. We’ll also see how to configure MSPs to achieve this later.

最后，这是OUs虽然有些滥用，有时候被用来区分同一联盟中的不同组织。在这种情况下，不同的组织使用相同的根CA和中间CA作为他们的信任链，但是分配OU字段来识别每个组织的成员。我们将同样看到配置MSP如何实现这一点。

#### Local and Channel MSPs

MSPs appear in two places in a blockchain network: channel configuration (channel MSPs), and locally on an actor’s premise (local MSP). Local MSPs are defined for clients (users) and for nodes (peers and orderers). Node local MSPs define the permissions for that node (who the peer admins are, for example). The local MSPs of the users allow the user side to authenticate itself in its transactions as a member of a channel (e.g. in chaincode transactions), or as the owner of a specific role into the system (an org admin, for example, in configuration transactions).

MSP使用在区块链网络中二个位置:通道配置(channel MSPs)和本地MSP(local MSP)。本地MSP被定义为客户端(client)和节点(peer和orderers)定义。本地MSPs定义了节点权限(例如，谁是peer管理者)。用户本地MSPs允许用户端在事务中作为通道成员进行身份验证，或者作为进入系统的特定角色(例如 配置事务中作为组织管理员)。

Every node and user must have a local MSP defined, as it defines who has administrative or participatory rights at that level (peer admins will not necessarily be channel admins, and vice versa).

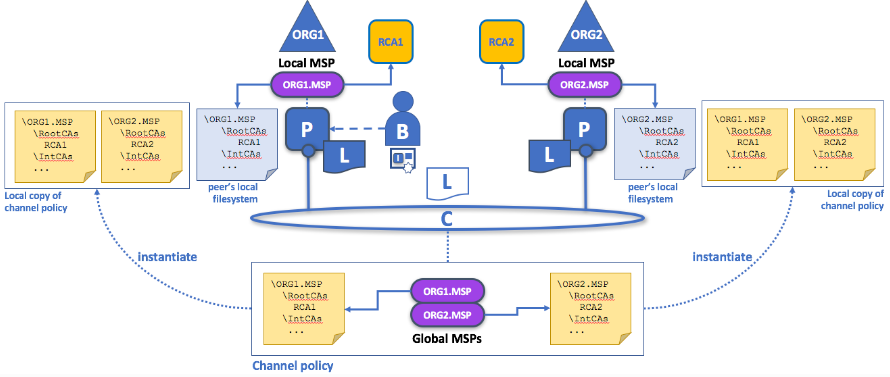
每个节点和用户必须有一个本地MSP定义，作为谁是管理或参与权(peer管理员不一定事通道管理员)。

In contrast, channel MSPs define administrative and participatory rights at the channel level. Every organization participating in a channel must have an MSP defined for it. Peers and orderers on a channel will all share the same view of channel MSPs, and will therefore be able to correctly authenticate the channel participants. This means that if an organization wishes to join the channel, an MSP incorporating the chain of trust for the organization’s members would need to be included in the channel configuration. Otherwise transactions originating from this organization’s identities will be rejected.

相比之下，渠道MSP定义了子链级别的管理者和参与者权限。链中每个组织参与者必须有一个MSP定义。通道上的对等节点和共识节点将共享相同的链MSPs视图，因此能够正确地验证通道参与者。这意味着如果组织希望加入该通道，则需要在通道配置中包括一个信任链的MSP。否则，源自组织事务的标识将被拒绝。

The key difference here between local and channel MSPs is not how they function — both turn identities into roles — but their scope.

本地和子链的MSPs关键不同，不在于如何工作的原理(如何将身份转换为角色)，而是他们范围。



Local and channel MSPs. The trust domain (e.g., the organization) of each peer is defined by the peer’s local MSP, e.g., ORG1 or ORG2. Representation of an organization on a channel is achieved by adding the organization’s MSP to the channel configuration. For example, the channel of this figure is managed by both ORG1 and ORG2. Similar principles apply for the network, orderers, and users, but these are not shown here for simplicity.

本地和链MSP.每个节点可信域在peer的本地MSP中定义，比如 ORG1或ORG2.通过组织的MSP增加到子链中的配置来实现对通道组织的表示。例如，子链中ORG1和ORG2两个组织通过数字化管理，相似的原因应用于网络、共识、和用户，为了简单起见，这里没有显示。

You may find it helpful to see how local and channel MSPs are used by seeing what happens when a blockchain administrator installs and instantiates a smart contract, as shown in the diagram above.

在上面图表中，有助于了解本地和子链MSP当链码管理安装和初始化智能合约的时候是如何使用的。

An administrator B connects to the peer with an identity issued by RCA1 and stored in their local MSP. When B tries to install a smart contract on the peer, the peer checks its local MSP, ORG1-MSP, to verify that the identity of B is indeed a member of ORG1. A successful verification will allow the install command to complete successfully. Subsequently, B wishes to instantiate the smart contract on the channel. Because this is a channel operation, all organizations on the channel must agree to it. Therefore, the peer must check the MSPs of the channel before it can successfully commit this command. (Other things must happen too, but concentrate on the above for now.)

管理员B通过数字证书连接到peer节点，通过RCA1进行发布，并储存在本地MSP中。当企业B试图在节点中安装智能合约，对等节点peer本地MSP，验证节点B的数字证书ORG1用户。验证完成后，安装命令将被允许执行。随后，B希望在子链上进行初始化安装。因为这是一个子链操作，链中所有的组织参与者必须都同意。因此，在事务命令成功提交，对等节点必须确认子链的MSPs。(另外的事也必须发生，但现在只集中的上面的)

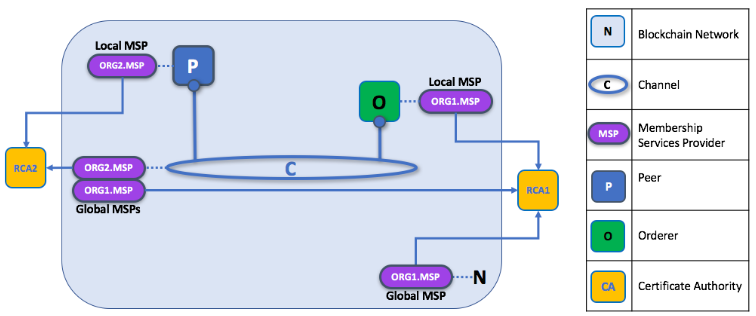
Local MSPs are only defined on the file system of the node or user to which they apply. Therefore, physically and logically there is only one local MSP per node or user. However, as channel MSPs are available to all nodes in the channel, they are logically defined once in the channel configuration. However, a channel MSP is also instantiated on the file system of every node in the channel and kept synchronized via consensus. So while there is a copy of each channel MSP on the local file system of every node, logically a channel MSP resides on and is maintained by the channel or the network.

本地MSP仅在定义在文件系统节点或应用。因此，在物理和逻辑上，只有一个本地MSP节点或用户。逻辑上在通道上一旦定义，通道MSPs对通道中所有节点是可用的。然而，通道MSP也在通道中每个节点的文件系统上实例化，通过公式保持同步。因此，每个节点的本地文件都有一个通道MSP副本。通道MSP驻留及维护在子链或网络上。

#### MSP Levels

The split between channel and local MSPs reflects the needs of organizations to administer their local resources, such as a peer or orderer nodes, and their channel resources, such as ledgers, smart contracts, and consortia, which operate at the channel or network level. It’s helpful to think of these MSPs as being at different levels, with MSPs at a higher level relating to network administration concerns while MSPs at a lower level handle identity for the administration of private resources. MSPs are mandatory at every level of administration — they must be defined for the network, channel, peer, orderer, and users.

子链和本地MSP的划分反映了组织资源(对等节点或共识节点)和渠道资源需求(例如账本、智能合约和组织)，那些在子链和网络级别的操作。将MSP看做不同级别是非常有用的，较高级别的MSP涉及网络管理问题，而较低级别管理私有资源资源的管理。MSP的每个级别的管理都是强制的，他们必须定义网络、子链、共识和用户。



MSP Levels. The MSPs for the peer and orderer are local, whereas the MSPs for a channel (including the network configuration channel) are shared across all participants of that channel. In this figure, the network configuration channel is administered by ORG1, but another application channel can be managed by ORG1 and ORG2. The peer is a member of and managed by ORG2, whereas ORG1 manages the orderer of the figure. ORG1 trusts identities from RCA1, whereas ORG2 trusts identities from RCA2. Note that these are administration identities, reflecting who can administer these components. So while ORG1 administers the network, ORG2.MSP does exist in the network definition.

MSP层级。对等节点和排序服务MSP是本地的，而访问所有子链参与者是共享的(包括链网络配置)。在这个图中，网络配置通道通过ORG1来管理，另外应用子链能被ORG1和ORG2管理。对等节点被ORG2进行管理，然而ORG1 管理图表中的排序，然而ORG1管理图中排序，ORG1信任RCA1中证书，然而ORG2信任RCA2的信任证书。注意，这些是管理标识，反映谁可以管理这些组件。所以，当ORG1管理网络时，ORG2的MSP就不会存在网络中。

**Network** **MSP:**

The configuration of a network defines who are the members in the network — by defining the MSPs of the participant organizations — as well as which of these members are authorized to perform administrative tasks (e.g., creating a channel).

网络配置定义了谁是网络成员--通过定义MPS组织参与者--以及这些成员被授权执行管理任务(例如，创建通道)。

**Channel MSP:**

It is important for a channel to maintain the MSPs of its members separately. A channel provides private communications between a particular set of organizations which in turn have administrative control over it. Channel policies interpreted in the context of that channel’s MSPs define who has ability to participate in certain action on the channel, e.g., adding organizations, or instantiating chaincodes. Note that there is no necessary relationship between the permission to administrate a channel and the ability to administrate the network configuration channel (or any other channel). Administrative rights exist within the scope of what is being administrated (unless the rules have been written otherwise — see the discussion of the ROLE attribute below).

对通道而言，单独维护其成员MSP非常重要的。通道提供特定组织之间的私有通信，这些组织反过来又对通道进行管理控制。通道策略在通道上下文解释谁有能力参与该通道的某些操作。例如，添加组织，或者初始化合约。注意，需要管理通道的能力和管理通道配置的能力没有必然联系(或者任何其他通道)。(除非规则另有规定，请参与下面关于角色属性讨论)

**Peer MSP:**

This local MSP is defined on the file system of each peer and there is a single MSP instance for each peer. Conceptually, it performs exactly the same function as channel MSPs with the restriction that it only applies to the peer where it is defined. An example of an action whose authorization is evaluated using the peer’s local MSP is the installation of a chaincode on the peer.

本地MSP在每个文件系统中被定义，每个对等节点都是一个MSP实例。从概念上讲，它执行通道MSP完全相同的功能，只适合定义它的对等节点。使用对等方的本地MSP评估其授权的操作的一个示例是在对等节点上安装链代码。

**Orderer MSP:**

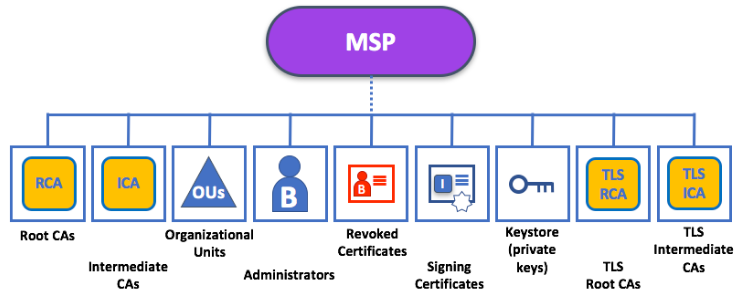
Like a peer MSP, an orderer local MSP is also defined on the file system of the node and only applies to that node. Like peer nodes, orderers are also owned by a single organization and therefore have a single MSP to list the actors or nodes it trusts.

同对等节点的MSP，共识本地MSP同样定义在文件系统节点中，仅仅应用在该节点。同peer节点相似，共识同样拥有组织，因此，只有一个MSP来列出它信任的参与者或节点。

#### MSP Structure

So far, you’ve seen that the most important element of an MSP are the specification of the root or intermediate CAs that are used to establish an actor’s or node’s membership in the respective organization. There are, however, more elements that are used in conjunction with these two to assist with membership functions.

到目前为止，您已经看到MSP最重要的元素是根CA或中间CA的规范，这些CA用于各自的组织中建立参与者或节点的成员关系。但是，更多的元素与这两种元素结合用来协助成员功能。



The figure above shows how a local MSP is stored on a local filesystem. Even though channel MSPs are not physically structured in exactly this way, it’s still a helpful way to think about them.

上面图表显示本地MSP是如何储存在本地文件系统中的。尽管通道MSP的物理结构并非完全如此，但仍然是很有帮助的思考方式。

As you can see, there are nine elements to an MSP. It’s easiest to think of these elements in a directory structure, where the MSP name is the root folder name with each subfolder representing different elements of an MSP configuration.

正如你所看到的，MSP有9个元素。目录结构中最容易想到这些元素，其中MSP名称是根文件夹名称，每个子文件夹标识MSP位置的不同元素。

**Intermediate CAs:**

This folder contains a list of X.509 certificates of the Intermediate CAs trusted by this organization. Each certificate must be signed by one of the Root CAs in the MSP or by an Intermediate CA whose issuing CA chain ultimately leads back to a trusted Root CA.

此文件夹包含该组织信任的由中间CA发行的一系列X.509证书。每个证书必须被一个根CA或由根CA发行的中间CA签名。

An intermediate CA may represent a different subdivision of the organization (like ORG1-MANUFACTURING and ORG1-DISTRIBUTION do for ORG1), or the organization itself (as may be the case if a commercial CA is leveraged for the organization’s identity management). In the latter case intermediate CAs can be used to represent organization subdivisions. Here you may find more information on best practices for MSP configuration. Notice, that it is possible to have a functioning network that does not have an Intermediate CA, in which case this folder would be empty.

中间CA可能代表组织的不同细分(比如 组织ORG1的ORG1-MANUFACTURING 和 ORG1-DISTRIBUTION)，或者代表组织自身(如果使用商业CA用于证书管理也可能这种情况)。后一种情况，中间CA可以用来代表组织的细分支。在这你可以找到更多MSP配置最佳实践的内容。请注意，这是一个没有中间CA的功能网络，在这种情况下，该文件夹可能是空的。

Like the Root CA folder, this folder defines the CAs from which certificates must be issued to be considered members of the organization.

与根CA文件夹一样，该文件夹定义了由组织发行的一系列证书。

**Organizational Units (OUs):**

These are listed in the $FABRIC\_CFG\_PATH/msp/config.yaml file and contain a list of organizational units, whose members are considered to be part of the organization represented by this MSP. This is particularly useful when you want to restrict the members of an organization to the ones holding an identity (signed by one of MSP designated CAs) with a specific OU in it.

这些都包含在 $FABRIC\_CFG\_PATH/msp/config.yaml 文件中，并包含一系列组织单元，其成员被认为是这个MSP所代表组织的一部分。当您希望将组织的成员限制为具有标识(指定MSP ca之一签名)并包含特定OU的成员时，这一点非常有用。

Specifying OUs is optional. If no OUs are listed, all the identities that are part of an MSP — as identified by the Root CA and Intermediate CA folders — will be considered members of the organization.

指定OUs是可选的。如果没有列出OUs，那么MSP(由根CA和中间CA标识)所有标识将被视为组织成员。

**Administrators**:

This folder contains a list of identities that define the actors who have the role of administrators for this organization. For the standard MSP type, there should be one or more X.509 certificates in this list.

文件夹包含了一系列定义管理角色的证书。使用标准MSP类型，列表中应该包含一个或多个X.509证书。

It’s worth noting that just because a actor has the role of an administrator it doesn’t mean that they can administer particular resources! The actual power a given identity has with respect to administering the system is determined by the policies that manage system resources. For example, a channel policy might specify that ORG1-MANUFACTURING administrators have the rights to add new organizations to the channel, whereas the ORG1-DISTRIBUTION administrators have no such rights.

值得注意的是，仅仅因为参与者具有管理者角色，并不意味着可以管理特定资源。给定身份的实际权力是由管理系统资源的策略决定的。例如，通道策略可能指定组织ORG1-MANUFACTURING 管理者，有权向通道中添加新组织，而ORG1-DISTRIBUTION管理员没有这种权限。

Even though an X.509 certificate has a ROLE attribute (specifying, for example, that a actor is an admin), this refers to a actor’s role within its organization rather than on the blockchain network. This is similar to the purpose of the OU attribute, which — if it has been defined — refers to a actor’s place in the organization.

即使X.509证书有角色属性(例如 指定者是管理员)，指定在组织中的角色而不是区块网络中的角色。与OU组织单元目标相似，如果定义了OU属性,涉及组织中角色的位置。

The ROLE attribute can be used to confer administrative rights at the channel level if the policy for that channel has been written to allow any administrator from an organization (or certain organizations) permission to perform certain channel functions (such as instantiating chaincode). In this way, an organizational role can confer a network role.

在通道级别角色属性可以用来提供管理者权限，在策略中定义定义允许组织管理者执行某些通道功能。通过这种方式，组织角色可以赋予网络角色。

**Revoked Certificates:**

If the identity of an actor has been revoked, identifying information about the identity — not the identity itself — is held in this folder. For X.509-based identities, these identifiers are pairs of strings known as Subject Key Identifier (SKI) and Authority Access Identifier (AKI), and are checked whenever the X.509 certificate is being used to make sure the certificate has not been revoked.

如果参与者标识被撤销，关于证书的证书信息，不是证书本身，被存储在文件夹中。对于基于X.509的标识，这些标识对被称之为SKI(Subject Key Identifier)和AKI(Authority Access Identifier)。X.509证书的使用为了确认证书没有被撤销。

This list is conceptually the same as a CA’s Certificate Revocation List (CRL), but it also relates to revocation of membership from the organization. As a result, the administrator of an MSP, local or channel, can quickly revoke a actor or node from an organization by advertising the updated CRL of the CA the revoked certificate as issued by. This “list of lists” is optional. It will only become populated as certificates are revoked.

这个列表概念上类似于CA的撤回列表(CRL)，但同样涉及成员组织中资格的撤销。因此，作为(本地或通道)MSP管理员，可以迅速撤销角色或节点。这给“列表的列表”是可选的，仅仅当证书被调度时被填充。

**Node Identity:**

This folder contains the identity of the node, i.e., cryptographic material that — in combination to the content of KeyStore — would allow the node to authenticate itself in the messages that is sends to other participants of its channels and network. For X.509 based identities, this folder contains an X.509 certificate. This is the certificate a peer places in a transaction proposal response, for example, to indicate that the peer has endorsed it — which can subsequently be checked against the resulting transaction’s endorsement policy at validation time.

This folder is mandatory for local MSPs, and there must be exactly one X.509 certificate for the node. It is not used for channel MSPs.

**KeyStore for Private Key:**

This folder is defined for the local MSP of a peer or orderer node (or in an client’s local MSP), and contains the node’s signing key. This key matches cryptographically the node’s identity included in Node Identity folder and is used to sign data — for example to sign a transaction proposal response, as part of the endorsement phase.

This folder is mandatory for local MSPs, and must contain exactly one private key. Obviously, access to this folder must be limited only to the identities of users who have administrative responsibility on the peer.

Configuration of a channel MSPs does not include this folder, as channel MSPs solely aim to offer identity validation functionalities and not signing abilities.

**TLS Root** **CA:**

This folder contains a list of self-signed X.509 certificates of the Root CAs trusted by this organization for TLS communications. An example of a TLS communication would be when a peer needs to connect to an orderer so that it can receive ledger updates.

MSP TLS information relates to the nodes inside the network — the peers and the orderers, in other words, rather than the applications and administrations that consume the network.

There must be at least one TLS Root CA X.509 certificate in this folder.

**TLS Intermediate CA:**

This folder contains a list intermediate CA certificates CAs trusted by the organization represented by this MSP for TLS communications. This folder is specifically useful when commercial CAs are used for TLS certificates of an organization. Similar to membership intermediate CAs, specifying intermediate TLS CAs is optional.

For more information about TLS, click here.

## 3.7 Peers 节点

#### A word on terminology

#### Ledgers and Chaincode

#### Applications and Peers

#### Peers and Channels

#### Peers and Organizations

#### Peers and Identity

#### Peers and Orderers

## 3.8 Private data 私有数据

What is private data?

What is a private data collection?

Transaction flow with private data

How a private data collection is defined

Purging data

## 3.9 Ledger 账本

#### What is a Ledger?

#### A Blockchain Ledger

#### World State

#### Blockchain

#### Blocks

#### Transactions

#### World State database options

#### Example Ledger: fabcar

#### More information

## 3.10 Use Cases 用例

The Hyperledger Requirements WG is documenting a number of blockchain use cases and maintaining an inventory here.

超级账本需要WG是一个记录区块的文档，这里用来记录和维护库存。

# 4 教程 tutorials

## 4.1 Writing Your First Application 第一个应用

## 4.2 Building Your First Network 第一个网络

## 4.3 Adding an Org to a Channel 子链中增加一个组织

## 4.4 Upgrading Your Network Components 升级网络组件

## 4.5 Using Private Data in Fabric 使用链码私有数据

## 4.6 Chaincode Tutorials 链码实例

## 4.7 Chaincode for Developers 开发链码

## 4.8 Chaincode for Operators 运营链码

## 4.9 System Chaincode Plugins 系统链码插件

## 4.10 Using CouchDB 使用Videos

## 4.11 Videos 视频

# 5 运营指导 operation guides

## 5.1 Upgrading to the Newest Version of Fabric

## 5.2 Updating a Channel Configuration

## 5.3 Membership Service Providers (MSP)

## 5.4 Channel Configuration (configtx)

## 5.5 Endorsement policies

## 5.6 Pluggable transaction endorsement and validation

## 5.7 Access Control Lists (ACL)

## 5.8 Error handling

## 5.9 Logging Control

## 5.10 Securing Communication With Transport Layer Security (TLS)

## 5.11 Bringing up a Kafka-based Ordering Service

# 6 相关命令 commands reference

## 6.1 peer

## 6.2 peer chaincode

## 6.3 peer channel

## 6.4 peer version

## 6.5 peer logging

## 6.6 peer node

## 6.7 configtxgen

## 6.8 configtxlator

## 6.9 cryptogen

## 6.10 Service Discovery Command Line Interface (discover)

## 6.11 Fabric-CA Commands

# 7 相关架构 Architcture reference

## 7.1 Architecture Explained

## 7.2 Transaction Flow

## 7.3 Hyperledger Fabric CA's User Guide

## 7.4 Hyperledger Fabric SDKs

## 7.5 Service Discovery

## 7.6 Channels

## 7.7 Capability Requirements

## 7.8 CouchDB as the State Database

## 7.9 Peer channel-based event services

## 7.10 Private Data

## 7.11 Read-Write set semantics

## 7.12 Gossip data dissemination protocol

# 8 常见问题 frequently asked questions

# 9 代码贡献 contributions welcome

见fabric 贡献文档

# 10 术语 glossary

# 11 其他

## 11.1 术语 glossary

|  |  |
| --- | --- |
| 术语 | 简单解释 |
| Anchor peer |  |
| ACL |  |
| Block |  |
| Chain |  |
| Chaincode |  |
| Channel |  |
| Commitment |  |
| Concurrency Control Version Check |  |
| Configuration Block |  |
| Consensus |  |
| Consortium |  |
| Current State |  |
| Dynamic Membership |  |
| Endorsement |  |
| Endorsement Policy |  |
| Hyperledger Fabric CA |  |
| Genesis Block |  |
| Gossip Protocol |  |
| Initialize |  |
| Install |  |
| Instantiate |  |
| Invoke |  |
| Leading Peer |  |
| Ledger |  |
| Member |  |
| Membership Service Provider |  |
| Membership services |  |
| Ordering Service |  |
| Organization |  |
| Peer |  |
| Policy |  |
| Private Data |  |
| Private Data Collection |  |
| Proposal |  |
| Query |  |
| Software Development Kit(SDK) |  |
| Smart Contract |  |
| State Database |  |
| System Chain |  |
| Transaction |  |
| World State |  |

Terminology is important, so that all Hyperledger Fabric users and developers agree on what we mean by each specific term.

术语非常重要，所有Fabric用户和开发认可每个特定术语的描述。

What is a smart contract for example. The documentation will reference the glossary as needed, but feel free to read the entire thing in one sitting if you like;

比如什么叫智能合约,文档根据需要引用术语表，如果您喜欢可以一次性阅读完毕。

it’s pretty enlightening!

这是相当于有启发

##### Anchor peer

Used to initiate gossip communication between peers from different organizations.

为了不同组织的节点间建立通信。

The anchor peer serves as the entry point for another organization’s peer on the same channel to communicate with each of the peers in the anchor peer’s organization.

锚定节点是作为相同子链不同组织节点间通信的入口服务。

Cross-organization gossip is scoped to channels.

跨组织通信智能在子链之间。

In order for cross-org gossip to work, peers from one organization need to know the address of at least one peer from another organization in the channel.

为了实现跨组织通信，每个组织节点至少配置一个peer节点用于通信。

Each organization added to a channel should identify at least one of its peers as an anchor peer (there can be more than one).

每个组织加入一个子链，至少为其添加一个锚定节点便于不同节点间通讯身份校验(可以配置多个)。

The anchor peer address is stored in the configuration block of the channel.

锚定节点地址存储在子链的配置中。

##### ACL

An ACL, or Access Control List, associates access to specific peer resources (such as system chaincode APIs or event services) to a Policy (which specifies how many and what types of organizations or roles are required).

ACL(访问控制列表)，将对等节点资源(比如链码api和事件服务)和关联策略(哪些特定资源和角色允许)起来。

The ACL is part of a channel’s configuration.

ACL 是子链配置的一部分。

It is therefore persisted in the channel’s configuration blocks, and can be updated using the standard configuration update mechanism.

这些被持久化在链码配置块中，可以通过标准的更新机制进行更新。

An ACL is formatted as a list of key-value pairs, where the key identifies the resource whose access we wish to control, and the value identifies the channel policy (group) that is allowed to access it.

ACL被格式化为一系列键值对，那些键资源是系统控制资源，那些值标识控制允许访问策略。

For example lscc/GetDeploymentSpec: /Channel/Application/Readers defines that the access to the life cycle chaincode GetDeploymentSpec API (the resource) is accessible by identities which satisfy the /Channel/Application/Readers policy.

例如 lscc/GetDeploymentSpec: /Channel/Application/Readers 定义生命周期的链码开发的通过./Channel/Application/Readers安全策略的API

A set of default ACLs is provided in the configtx.yaml file which is used by configtxgen to build channel configurations.

configtx.yaml定义了一组默认的ACLs，configtx.xml使用yaml文件来构建通道配置。

The defaults can be set in the top level “Application” section of configtx.yaml or overridden on a per profile basis in the “Profiles” section.

默认值可以在“Application”部分顶级设置，也可以在每个配置中“Profiles”中重写。

##### Block

An ordered set of transactions that is cryptographically linked to the preceding block(s) on a channel.

在一个子链中，一些有序事务通过加密方式连接到前一个快。

##### Chain

The ledger’s chain is a transaction log structured as hash-linked blocks of transactions.

分布式链是个的事务日志，该结构以hash链接区块链事务。

Peers receive blocks of transactions from the ordering service, mark the block’s transactions as valid or invalid based on endorsement policies and concurrency violations, and append the block to the hash chain on the peer’s file system.

Peer端从排序服务接收事务区块，根据背书策略或当前验证结果，决定区块是否有效，并将区块HASH链接到peer文件系统中。

##### Chaincode

See Smart-Contract. 见 See Smart-Contract.

##### Channel

A channel is a private blockchain overlay which allows for data isolation and confidentiality.

子链是一个对区块数据隔离性和机密性的私有覆盖层

A channel-specific ledger is shared across the peers in the channel, and transacting parties must be properly authenticated to a channel in order to interact with it.

特定子链账本在子链中不同节点之间共享，为了与通道进行交流，相互之间必须进行身份验证。

Channels are defined by a Configuration-Block.

子链在区块配置中定义。

##### Commitment

Commitment 承诺 保障

Each Peer on a channel validates ordered blocks of transactions and then commits (writes/appends) the blocks to its replica of the channel Ledger.

通道上的每个对等点验证有序的事务块，然后将这些块提交(写/追加)到链账本的副本上。

Peers also mark each transaction in each block as valid or invalid.

Peer节点同样标记区块链中事务有效或无效。

##### Concurrency Control Version Check

Concurrency Control Version Check is a method of keeping state in sync across peers on a channel.

并发版本控制是一种子链中不同节点保持状态同步的方法。

Peers execute transactions in parallel, and before commitment to the ledger, peers check that the data read at execution time has not changed.

Peer节点并行执行事务，在事务提交前，各节点确认获取到的数据没有被变更。

If the data read for the transaction has changed between execution time and commitment time, then a Concurrency Control Version Check violation has occurred, and the transaction is marked as invalid on the ledger and values are not updated in the state database.

如果数据在执行时和提交时之间被修改了，就触发了并发控制检查违规，账本标记为无效，数据值不会更新。

##### **Configuration Block**

Contains the configuration data defining members and policies for a system chain (ordering service) or channel.

包括了成员配置数据和系统链(排序服务)或子链策略。

Any configuration modifications to a channel or overall network (e.g. a member leaving or joining) will result in a new configuration block being appended to the appropriate chain.

对通道或整个网络的任何配置修改都会生成一个新的配置区块并存放在适当的链中。

This block will contain the contents of the genesis block, plus the delta.

这个需求包括了网络区块的内容，加到了delta中。

##### Consensus

A broader term overarching the entire transactional flow, which serves to generate an agreement on the order and to confirm the correctness of the set of transactions constituting a block.

整个事务流一个更广泛的术语，用于生成排序协议并确认一个块事务集的正确性。

##### Consortium

A consortium is a collection of non-orderer organizations on the blockchain network.

联盟是一个在区块网络中没有排序机构集合。

These are the organizations that form and join channels and that own peers.

组织来自或加入子链，与那些节点同行。

While a blockchain network can have multiple consortia, most blockchain networks have a single consortium.

一个区块链网络可以有多个联盟，但是大多数链网络只有一个联盟。

At channel creation time, all organizations added to the channel must be part of a consortium.

在子链创建时，所有组织的添加成员必须是联盟成员。

However, an organization that is not defined in a consortium may be added to an existing channel.

然而，没有在联盟中定义的组织可以添加进子链中。

##### Current State

See World-State. 见World-State.

##### Dynamic Membership

Hyperledger Fabric supports the addition/removal of members, peers, and ordering service nodes, without compromising the operationality of the overall network.

fabric 支持增加或删除会员、对等节点、排序服务节点，不会影响整个网络的运营。

Dynamic membership is critical when business relationships adjust and entities need to be added/removed for various reasons.

当业务关系调整或者各种原因需要增加或删除机构，动态成员就变得非常重要。

##### Endorsement

Refers to the process where specific peer nodes execute a chaincode transaction and return a proposal response to the client application.

特点peer节点执行链码事务并返回一个提案响应给客户端的过程。

The proposal response includes the chaincode execution response message, results (read set and write set), and events, as well as a signature to serve as proof of the peer’s chaincode execution.

这个提案响应包括链码的执行返回报文，结果(读取、写入集)，事件，作为对等端链代码执行的证明的签名。

Chaincode applications have corresponding endorsement policies, in which the endorsing peers are specified.

链码应用具有相应的背书策略，其中指定了背书对等节点。

##### Endorsement Policy

Defines the peer nodes on a channel that must execute transactions attached to a specific chaincode application, and the required combination of responses (endorsements).

定义通道上的对等节点，该节点必须执行附加在链码应用上的系列事务及所需的响应组合。

A policy could require that a transaction be endorsed by a minimum number of endorsing peers, a minimum percentage of endorsing peers, or by all endorsing peers that are assigned to a specific chaincode application.

策略可以要求事务最少数量背书节点、最少背书节点百分比或所有合约指定的背书节点认可。

Policies can be curated based on the application and the desired level of resilience against misbehavior (deliberate or not) by the endorsing peers.

策略可以根据应用和背书节点期望弹性情况来制定，

A transaction that is submitted must satisfy the endorsement policy before being marked as valid by committing peers.

事务必须满足背书策略才能被提交节点标记为有效。

A distinct endorsement policy for install and instantiate transactions is also required.

安装和初始化事务同样也需要一个特定的背书策略。

##### Hyperledger Fabric CA

Hyperledger Fabric CA is the default Certificate Authority component, which issues PKI-based certificates to network member organizations and their users.

Hyperledger Fabric CA 是一个默认的证书授权组件，向其他网络成员和组织颁发PKI证书。

The CA issues one root certificate (rootCert) to each member and one enrollment certificate (ECert) to each authorized user.

CA颁发给每个成员颁发一个根证书，给每一个授权用户颁发一个背书证书。

##### Genesis Block

The configuration block that initializes the ordering service, or serves as the first block on a chain.

排序服务的第一个区块或者链中第一个区块

##### Gossip Protocol

The gossip data dissemination protocol performs three functions:

gossip数据传播协议执行三个功能：

1) manages peer discovery and channel membership;

管理节点发现和子链会员。

2) disseminates ledger data across all peers on the channel;

将账本数据分布到链上所有节点上。

3) syncs ledger state across all peers on the channel.

子链上所有节点同步账本状态。

Refer to the Gossip topic for more details.

请参与gossip主体获取更多细节。

A method to initialize a chaincode application.

初始化链码应用的方法

##### Initialize

A method to initialize a chaincode application.

初始化链码应用的方法

##### Install

The process of placing a chaincode on a peer’s file system.

对等节点的文件系统中放置链码的过程。

##### Instantiate

The process of starting and initializing a chaincode application on a specific channel.

在特定子链上合约的启动和初始化的过程。

After instantiation, peers that have the chaincode installed can accept chaincode invocations.

实例化之后，那些安装了合约的peer节点可以接受合约的调度。

##### Invoke

Used to call chaincode functions.

用于调用链码函数。

A client application invokes chaincode by sending a transaction proposal to a peer.

客户端通过向节点发送事务提案来调度合约。

The peer will execute the chaincode and return an endorsed proposal response to the client application.

对等节点将执行合约并返回客户端应用的背书响应。

The client application will gather enough proposal responses to satisfy an endorsement policy, and will then submit the transaction results for ordering, validation, and commit.

客户端应用将收集足够多提案响应来满足背书策略，然后提交事务结果进行共识、验证和提交。

The client application may choose not to submit the transaction results.

客户端应用也可以选择不提交事务结果。

For example if the invoke only queried the ledger, the client application typically would not submit the read-only transaction, unless there is desire to log the read on the ledger for audit purpose.

例如，调用只查询分类账本，客户端应用通常不会提交只读事务，除非需要在分类账本上记录读操作进行审计。

The invoke includes a channel identifier, the chaincode function to invoke, and an array of arguments.

调度包括子链标识、子链功能和参数数组。

##### Leading Peer

Each Organization can own multiple peers on each channel that they subscribe to.

每个组织可以在他们订阅的子链上拥有多个对等节点，

One or more of these peers should serve as the leading peer for the channel, in order to communicate with the network ordering service on behalf of the organization.

一个或多个peer节点作为子链的主要对等节点，代表组织与同网络共识服务进行通信。

The ordering service delivers blocks to the leading peer(s) on a channel, who then distribute them to other peers within the same organization.

子链上订阅服务交付区块给领导节点，然后领导节点分发给相同组织中的其他对等节点。

##### Ledger

THIS REQUIRES UPDATING

这些需要更新

A ledger consists of two distinct, though related, parts – a “blockchain” and the “state database”, also known as “world state”.

一个账本宝行两个不同又相关的部分，一个是区块链链，一个是数据状态，我们所所熟知的"world state"

Unlike other ledgers, blockchains are immutable – that is, once a block has been added to the chain, it cannot be changed.

不像其他账本，区块链是不可变的--一旦区块链被添加到链中，将不能被改变。

In contrast, the “world state” is a database containing the current value of the set of key-value pairs that have been added, modified or deleted by the set of validated and committed transactions in the blockchain.

相反，“world state”是一个包含当前系列key-value键值对值的数据库，这些键值对值在一系列验证和事务提交中进行增加、修改、删除。

It’s helpful to think of there being one logical ledger for each channel in the network.

这是在网络中每个子链中的形成逻辑账本非常有帮助的，

In reality, each peer in a channel maintains its own copy of the ledger – which is kept consistent with every other peer’s copy through a process called consensus.

实际上，链上每个节点都维护他字节拥有的账本副本--通过一个叫共识的过程同每个其他节点保持一致。

The term Distributed Ledger Technology (DLT) is often associated with this kind of ledger – one that is logically singular, but has many identical copies distributed across a set of network nodes (peers and the ordering service).

(分布式账本技术)DLT术语是提供与分类账本相关联 -- 在逻辑上是单一的，但有很多完全相同的分布在网络系列节点中。

##### Member

See Organization. 见 Organization

##### Membership Service Provider

The Membership Service Provider (MSP) refers to an abstract component of the system that provides credentials to clients, and peers for them to participate in a Hyperledger Fabric network.

会员关系服务(MSP)指的是系统的一个抽象组件，给客户端提供支持，让peer节点参与进到区块链网络中来。

Clients use these credentials to authenticate their transactions, and peers use these credentials to authenticate transaction processing results (endorsements).

客户端使用这些证书来验证他们的事务，而peer节点使用这些授权凭证来处理事务结果(背书)。

While strongly connected to the transaction processing components of the systems, this interface aims to have membership services components defined, in such a way that alternate implementations of this can be smoothly plugged in without modifying the core of transaction processing components of the system.

这些接口虽然与系统事务处理紧密相连，但接口的目标是定义成员服务组件，如此就可以在不修改系统事务处理核心组件的基础上，比较顺畅的接入进来。

##### Membership services

Membership Services authenticates, authorizes, and manages identities on a permissioned blockchain network.

成员关系服务对已许可的区块网络进行验证、授权和身份管理。

The membership services code that runs in peers and orderers both authenticates and authorizes blockchain operations.

成员服务代码运行在peer和排序节点上进行鉴权和授权操作。

It is a PKI-based implementation of the Membership Services Provider (MSP) abstraction.

这是基于PKI的会员提供者(MSP)的抽象实现。

##### Ordering Service

A defined collective of nodes that orders transactions into a block.

将一系列节点事务集排序进入一个区块中。

The ordering service exists independent of the peer processes and orders transactions on a first-come-first-serve basis for all channel’s on the network.

排序服务依赖于所有对等节点的处理，网络子链中事务处理依据先进先出原则。

The ordering service is designed to support pluggable implementations beyond the out-of-the-box SOLO and Kafka varieties.

排序服务除支持开箱即用的SOLO和Kafka外，还支持可插入实现。

The ordering service is a common binding for the overall network;

排序服务是整个网络的通用绑定

it contains the cryptographic identity material tied to each Member.

排序服务包括每个成员的密码标识内容。

##### Organization

Also known as “members”, organizations are invited to join the blockchain network by a blockchain service provider.

也被称为“成员”，组织被区块链提供者邀请加入区块链链网络。

An organization is joined to a network by adding its Membership Service Provider (MSP) to the network.

组织通过会员关系提供者(MSP)加入组织。

The MSP defines how other members of the network may verify that signatures (such as those over transactions) were generated by a valid identity, issued by that organization.

MSP中定义了如何验证网络中的会员签名(例如事务上的签名)，确认是否为该组织发出去的证书。

The particular access rights of identities within an MSP are governed by policies which are are also agreed upon when the organization is joined to the network.

在MSP中，身份的特定访问权限受MSP策略控制，该策略同样要加入网络中的组织的同意。

An organization can be as large as a multi-national corporation or as small as an individual.

组织可以像多跨国公司一样大，也可以像个人一样小。

The transaction endpoint of an organization is a Peer.

一个组织的事务端是一个peer节点。

A collection of organizations form a Consortium.

一个组织的集合形成一个联盟

While all of the organizations on a network are members, not every organization will be part of a consortium.

虽然网络上所有的组织都是会员，但不是每个组织都是联盟中一员。

##### **Peer**

A network entity that maintains a ledger and runs chaincode containers in order to perform read/write operations to the ledger.

是一个维护账本和运营在账本上执行读、写操作合约的网络实体。

Peers are owned and maintained by members.

对等节点由成员用户和维护。

##### Policy

Policies are expressions composed of properties of digital identities, for example: Org1.Peer OR Org2.Peer.

策略是一系列数组证书组成的表达式，例如 Org1.Peer 或 Org2.Peer.

They are used to restrict access to resources on a blockchain network.

在区块链网络中，这些用来限制访问资源。

For instance, they dictate who can read from or write to a channel, or who can use a specific chaincode API via an ACL.

例如，这些可以子链中读写的指令，或谁可以对链码进行ACL访问

Policies may be defined in configtx.yaml prior to bootstrapping an ordering service or creating a channel, or they can be specified when instantiating chaincode on a channel.

策略可以在configtx.yaml预先定义，然后传播到共识服务中，或者他们也可以在子链中初始链码时指定。

A default set of policies ship in the sample configtx.yaml which will be appropriate for most networks.

在简单configtx.yaml中提供了一系列策略，这些策略适合于大多数网络。

##### Private Data

Confidential data that is stored in a private database on each authorized peer, logically separate from the channel ledger data.

私密数据存储在每个授权节点的私有数据库中，与子链账本逻辑上分离。

Access to this data is restricted to one or more organizations on a channel via a private data collection definition.

私有数据的访问是被限制的，在预先设置的前提下只允许链上一个或某几个组织进行访问。

Unauthorized organizations will have a hash of the private data on the channel ledger as evidence of the transaction data.

子链中未经授权的组织将获取一个私有数据的hash,作为事务数据的一个存在证明。

Also, for further privacy, hashes of the private data go through the Ordering-Service, not the private data itself, so this keeps private data confidential from Orderer.

同样，作为进一步隐私，通过私有数据的hash值进行排序服务，而非私有数据本身，这样对排序服务保障了私有数据的私密性。

##### Private Data Collection

Used to manage confidential data that two or more organizations on a channel want to keep private from other organizations on that channel.

用来管理子链上两个或多个组织上之间的私密性数据，

The collection definition describes a subset of organizations on a channel entitled to store a set of private data, which by extension implies that only these organizations can transact with the private data.

子链上的collection定义了一个组织的子集，用于存储一系列私有数据，进一步来说，只有这些组织才能处理这些数据。

##### Proposal

A request for endorsement that is aimed at specific peers on a channel.

通道上特定对等节点进行背书请求。

Each proposal is either an instantiate or an invoke (read/write) request.

每个提案要么是实例化，要么调用(读、写)请求。

##### Query

A query is a chaincode invocation which reads the ledger current state but does not write to the ledger.

query是区块链上个中调度方式，该调度智能读账本当前状态而不能写账本。

The chaincode function may query certain keys on the ledger, or may query for a set of keys on the ledger.

合约功能可以查询账本上某些keys,也可以查询账本上的一组键。

Since queries do not change ledger state, the client application will typically not submit these read-only transactions for ordering, validation, and commit.

由于查询不能改变账本状态，典型性的，客户端应用不能提交那些只读事务作为共识、校验、提交。

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.

作为非典型性情况，客户端应用可以选择性提交只读事务作为排序、校验、提交。例如，客户端想在账本上做审计，证明谁某一个时间点访问了某个状态。

##### Software Development Kit(SDK)

The Hyperledger Fabric client SDK provides a structured environment of libraries for developers to write and test chaincode applications.

Fabric 客户端SDK提供了Librarie结构性库环境，用于写和测试合约应用。

The SDK is fully configurable and extensible through a standard interface.

SDK是通过标准接口是完全配置化和可扩展的。

Components, including cryptographic algorithms for signatures, logging frameworks and state stores, are easily swapped in and out of the SDK.

组件包括了签名加密算法，日志框架，状态存储，SDK可以轻松的与外界交互。

The SDK provides APIs for transaction processing, membership services, node traversal and event handling.

SDK提供了API作为事务处理、成员服务、节点遍历、事务处理。

Currently, the two officially supported SDKs are for Node.js and Java, while three more – Python, Go and REST – are not yet official but can still be downloaded and tested.

目前提供Node.js和Java两种SDKs,虽然更多如Python、Go、REST 目前还没有正式发布，但可以下载和测试。

##### Smart Contract

A smart contract is code – invoked by a client application external to the blockchain network – that manages access and modifications to a set of key-value pairs in the World State.

智能合约是一段代码，在区块链网络中被客户端调用，在world state中管理一系列键值对的访问和修改。

In Hyperledger Fabric, smart contracts are referred to as chaincode.

在Fabric中，智能合约被称为链码。

Smart contract chaincode is installed onto peer nodes and instantiated to one or more channels.

智能合约安装在peer节点上，并实例化一到多个子链中。

##### State Database

Current state data is stored in a state database for efficient reads and queries from chaincode.

当前状态数据存储在状态数据库中，为了链码更高效的读取和查询。

Supported databases include levelDB and couchDB.

支持的数据库包括levelDB和couchDB

##### System Chain

Contains a configuration block defining the network at a system level.

包含在定义了一个系统级别的网络配置区块链中，。

The system chain lives within the ordering service, and similar to a channel, has an initial configuration containing information such as: MSP information, policies, and configuration details.

系统在运行在共识服务中，与子链相似，拥有一个包括MSP信息、策略和配置明细的初始化配置。

Any change to the overall network (e.g. a new org joining or a new ordering node being added) will result in a new configuration block being added to the system chain.

整体网络的任何变更(比如 新组织的加入、新共识节点的加入)，将会有新的配置区块加入系统链中。

The system chain can be thought of as the common binding for a channel or group of channels.

系统链可以看做子链或子链组的一种公共绑定。

For instance, a collection of financial institutions may form a consortium (represented through the system chain), and then proceed to create channels relative to their aligned and varying business agendas.

例如，一组金融机构可以形成一个(通过系统链来描述)联盟，然后开始创建与其利益一致的不同业务议程的通道。

##### Transaction

Invoke or instantiate results that are submitted for ordering, validation, and commit.

共识、校验、提交的最终提交结果是调度或实例化。

Invokes are requests to read/write data from the ledger.

调度是账本上数据的读写。

Instantiate is a request to start and initialize a chaincode on a channel.

实例化在子链上安装合约或开始请求。

Application clients gather invoke or instantiate responses from endorsing peers and package the results and endorsements into a transaction that is submitted for ordering, validation, and commit.

应用客户端要么来此背书节点的调度或者实例化响应，并将结果和背书打包打包到提交排序、验证和提交的事务中。

##### World State

Also known as the “current state”, the world state is a component of the HyperLedger Fabric Ledger.

也被称为“当前状态”，世界状态是超级账本的组成部分。

The world state represents the latest values for all keys included in the chain transaction log.

世界状态代表包括在链码状态下所有最新的值。

Chaincode executes transaction proposals against world state data because the world state provides direct access to the latest value of these keys rather than having to calculate them by traversing the entire transaction log.

链码执行事务提案而不是世界状态数据，因为世界状态提供了直接访问这些键值得直接访问而不是遍历整个事务日志计算他们。

The world state will change every time the value of a key changes (for example, when the ownership of a car – the “key” – is transferred from one owner to another – the “value”) or when a new key is added (a car is created).

世界状态随着每次值得变化都发生改变(例如，汽车的拥有者从一个换成另外时)，或者新增一个值(创建一辆汽车).

As a result, the world state is critical to a transaction flow, since the current state of a key-value pair must be known before it can be changed.

因此，世界状态对于事务流而言非常关键，因为键值对被修改前当前状态必须被知晓。

Peers commit the latest values to the ledger world state for each valid transaction included in a processed block.

peer节点提交的账本中的最新值每一个有效事务都包括在事务块中。

## 11.2 提问 still have questions

## 11.3 各版本简介

### 11.3.1 1.0 版本介绍

### 11.3.2 1.1 版本介绍

### 11.3.3 1.2 版本介绍

New major features

主要新特征

1 Private Data Collections: 私人数据收集

A way to keep certain data/transactions confidential among a subset of channel members. We also have an architecture document on this topic which can be found here.

channel成员子集之间保持一种可信的数据/事务通信。

2 Service Discovery: 服务发现

Discover network services dynamically, including orderers, peers, chaincode, and endorsement policies, to simplify client applications.

动态的网络服务发现，包括排序节点、peer节点、合约、背书策略、以及简单客户端应用。

3 Access control: 访问控制

How to configure which client identities can interact with peer functions on a per channel basis.

通过配置可以达到客户端身份与每个通道上的对等函数交互。

4 Pluggable endorsement and validation:可插拔的背书和验证

Utilize pluggable endorsement and validation logic per chaincode.

对每个链代码使用可插入背书和验证逻辑

New tutorials 新教程

1 Upgrade to version v1.2:升级到v1.2

Leverages the BYFN network to show how an upgrade flow should work. Includes both a script (which can serve as a template for upgrades), as well as the individual commands.

利用BYFN网络展示了一个升级流程，包括了一个脚本（作为升级模板），作为单个命令。

2 CouchDB: CouchDB

How to set up a CouchDB data store (which allows for rich queries).

如何设置CouchDB作为存储（丰富的查询支持）。

3 Private data: 私有数据

Shows how to set up a collection using BYFN.

演示了如何使用BYFN设置集合

4 Query certificates based on various filter criteria (Fabric CA): 基于过滤标准(Fabric CA)的查询证书

Describes how to use fabric-ca-client to manage certificates.

描述了如何使用fabric-ca-client去管理证书

Other new documentation 另外新文档

Service Discovery CLI:服务发现CLI

Configuring the discovery service using the CLI.

使用CLI配置服务发现

Release notes 发布说明

For more information, including FAB numbers for the issues and code reviews that made up these changes (in addition to other hygiene/performance/bug fixes we did not explicitly document), check out the:

更多信息,包括问题的FAB编号和构成这些更改的代码审查,请参与如下地址

1 Fabric release notes.

https://github.com/hyperledger/fabric/releases/tag/v1.2.0

2 Fabric CA release notes.

https://github.com/hyperledger/fabric-ca/releases/tag/v1.2.0

### 11.3.4 1.3 版本介绍

## 11.4 Fabric CA

见 fabric ca文档

# 词汇

consensus protocol 一致性协议

Cryptocurrency 加密货币

Permission 许可

underlying technology 底层技术

innovative enterprise 创新型企业

performance characteristics 性能特征

hard requirement 硬需求

Anti-Money Laundering (AML) 反洗钱

Know-Your-Customer (KYC) 了解客户

motivation  动机

distributed ledger technology (DLT)  分布式账本技术

domain-specific languages (DSL) 领域特定语言

anonymous  匿名

[crash fault-tolerant](https://en.wikipedia.org/wiki/Fault_tolerance) (CFT) 故障容错

byzantine fault tolerant (BFT) 拜占庭容错

“proof of work” (PoW). 工作量证明

malicious 恶意的

resiliency 弹性

resilienpotentially malicious contracts 潜在恶意代码

non-determinism 不确定性

The benchmarking efforts 基准测试

Zero knowledge proofs (ZKP) 零知识证明

Membership Service Provider (MSP) 会员关系服务提供者