青岛理工大学信息与控制工程学院

毕业设计外文翻译

Hyperledger Fabric Docs
超级账本官方文档

设计题目：基于区块链技术的票据操作系统的设计与实现

专业班级：软件173

学生姓名：赵帅

学生学号：201722240

指导教师：周 炜

完成日期：2021年3月19日

Introduction

In general terms, a blockchain is an immutable transaction ledger, maintained within a distributed network of peer nodes. These nodes each maintain a copy of the ledger by applying transactions that have been validated by a consensus protocol, grouped into blocks that include a hash that bind each block to the preceding block.

The first and most widely recognized application of blockchain is the Bitcoin cryptocurrency, though others have followed in its footsteps. Ethereum, an alternative cryptocurrency, took a different approach, integrating many of the same characteristics as Bitcoin but adding smart contracts to create a platform for distributed applications. 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.

As the popularity of Bitcoin, Ethereum and a few other derivative technologies grew, interest in applying the underlying technology of the blockchain, distributed ledger and distributed application platform to more innovative enterprise use cases also grew. However, many enterprise use cases require performance characteristics that the permissionless blockchain technologies are unable (presently) to deliver. In addition, in many use cases, the identity of the participants is a hard requirement, such as in the case of financial transactions where Know-Your-Customer (KYC) and Anti-Money Laundering (AML) regulations must be followed.

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

Participants must be identified/identifiable

Networks need to be permissioned

High transaction throughput performance

Low latency of transaction confirmation

Privacy and confidentiality of transactions and data pertaining to business transactions

While many early blockchain platforms are currently being adapted for enterprise use, Hyperledger Fabric has been designed for enterprise use from the outset. The following sections describe how Hyperledger Fabric (Fabric) differentiates itself from other blockchain platforms and describes some of the motivation for its architectural decisions.

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.

One key point of differentiation is that Hyperledger was established under the Linux Foundation, which itself has a long and very successful history of nurturing open source projects under open governance that grow strong sustaining communities and thriving ecosystems. Hyperledger is governed by a diverse technical steering committee, and the 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.

Fabric has a highly modular and configurable architecture, enabling innovation, versatility and optimization for a broad range of industry use cases including banking, finance, insurance, healthcare, human resources, supply chain and even digital music delivery.

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). This means that most enterprises already have the skill set needed to develop smart contracts, and no additional training to learn a new language or DSL is needed.

The Fabric platform is also permissioned, meaning that, unlike with a public permissionless network, the participants are known to each other, rather than anonymous and therefore fully untrusted. This means that while the participants may not fully trust one another (they may, for example, be competitors in the same industry), a network can be operated under a governance model that is built off of what trust does exist between participants, such as a legal agreement or framework for handling disputes.

One of the most important of the platform's differentiators is its support for pluggable consensus protocols that enable the platform to be more effectively customized to fit particular use cases and trust models. For instance, when deployed within a single enterprise, or operated by a trusted authority, fully byzantine fault tolerant consensus might be considered unnecessary and an excessive drag on performance and throughput. In situations such as that, acrash fault-tolerant (CFT) consensus protocol might be more than adequate whereas, in a multi-party, decentralized use case, a more traditional byzantine fault tolerant (BFT) consensus protocol might be required.

Fabric can leverage consensus protocols that do not require a native cryptocurrency to incent costly mining or to fuel smart contract execution. Avoidance of a cryptocurrency reduces some significant risk/attack vectors, and absence of cryptographic mining operations means that the platform can be deployed with roughly the same operational cost as any other distributed system.

The combination of these differentiating design features makes Fabric one of the better performing platforms available today both in terms of transaction processing and transaction confirmation latency, and it enables privacy and confidentiality of transactions and the smart contracts (what Fabric calls “chaincode”) that implement them.

Let’s explore these differentiating features in more detail.

Modularity

Hyperledger Fabric has been specifically architected to have a modular architecture. Whether it is pluggable consensus, pluggable identity management protocols such as LDAP or OpenID Connect, key management protocols or cryptographic libraries, the platform has been designed at its core to be configured to meet the diversity of enterprise use case requirements.

At a high level, Fabric is comprised of the following modular components:

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

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

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

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.

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

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

There is fair agreement in the industry that there is no “one blockchain to rule them all”. Hyperledger Fabric can be configured in multiple ways to satisfy the diverse solution requirements for multiple industry use cases.

Permissioned vs Permissionless Blockchains

In a permissionless blockchain, virtually anyone can participate, and every participant is anonymous. In such a context, there can be no trust other than that the state of the blockchain, prior to a certain depth, is immutable. 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).

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. A permissioned blockchain provides a way to secure the interactions among a group of entities that have a common goal but which may not fully trust each other. By relying on the identities of the participants, a permissioned blockchain can use more traditional crash fault tolerant (CFT) or byzantine fault tolerant (BFT) consensus protocols that do not require costly mining.

Additionally, in such a permissioned context, the risk of a participant intentionally introducing malicious code through a smart contract is diminished. First, the participants are known to one another and all actions, whether 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. Rather than being completely anonymous, the guilty party can be easily identified and the incident handled in accordance with the terms of the governance model.

Smart Contracts

A smart contract, or what Fabric calls “chaincode”, functions as a trusted distributed application that gains its security/trust from the blockchain and the underlying consensus among the peers. It is the business logic of a blockchain application.

There are three key points that apply to smart contracts, especially when applied to a platform:

many smart contracts run concurrently in the network,

they may be deployed dynamically (in many cases by anyone), and

application code should be treated as untrusted, potentially even malicious.

Most existing smart-contract capable blockchain platforms follow an order executearchitecture in which the consensus protocol:

validates and orders transactions then propagates them to all peer nodes,

each peer then executes the transactions sequentially.

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.

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

Further, since all transactions are executed sequentially by all nodes, performance and scale is limited. The fact that the smart contract code executes on every node in the system demands that complex measures be taken to protect the overall system from potentially malicious contracts in order to ensure resiliency of the overall system.

A New Approach

Fabric introduces a new architecture for transactions that we call execute-order-validate. It addresses the resiliency, flexibility, scalability, performance and confidentiality challenges faced by the order-execute model by separating the transaction flow into three steps:

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

order transactions via a (pluggable) consensus protocol, and

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

This design departs radically from the order-execute paradigm in that Fabric executes transactions before reaching final agreement on their order.

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. Thus, each transaction need only be executed (endorsed) by the subset of the peer nodes necessary to satisfy the transaction’s endorsement policy. This allows for parallel execution increasing overall performance and scale of the system. This first phase also eliminates any non-determinism, as inconsistent results can be filtered out before ordering.

Because we have eliminated non-determinism, Fabric is the first blockchain technology that enables use of standard programming languages.

Privacy and Confidentiality

As we have discussed, in a public, permissionless blockchain network that leverages PoW for its consensus model, transactions are executed on every node. This means that neither can there be confidentiality of the contracts themselves, nor of the transaction data that they process. Every transaction, and the code that implements it, is visible to every node in the network. In this case, we have traded confidentiality of contract and data for byzantine fault tolerant consensus delivered by PoW.

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

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.

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. All have their trade-offs.

Encrypting data is one approach to providing confidentiality; however, in a permissionless network leveraging PoW for its consensus, the encrypted data is sitting on every node. Given enough time and computational resource, the encryption could be broken. For many enterprise use cases, the risk that their information could become compromised is unacceptable.

Zero knowledge proofs (ZKP) are another area of research being explored to address this problem, the trade-off here being that, presently, computing a ZKP requires considerable time and computational resources. Hence, the trade-off in this case is performance for confidentiality.

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.

Hyperledger Fabric, being a permissioned platform, enables confidentiality through its channel architecture and private data feature. In channels, participants on a Fabric network establish a sub-network where every member has visibility to a particular set of transactions. 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. Private data allows collections between members on a channel, allowing much of the same protection as channels without the maintenance overhead of creating and maintaining a separate channel.

Pluggable Consensus

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. Specifically, the ordering service. Since consensus is modular, its implementation can be tailored to the trust assumption of a particular deployment or solution. This modular architecture allows the platform to rely on well-established toolkits for CFT (crash fault-tolerant) or BFT (byzantine fault-tolerant) ordering.

Fabric currently offers a CFT ordering service implementation based on the etcd library of the Raft protocol. For information about currently available ordering services, check out our conceptual documentation about ordering.

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

Performance and Scalability

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. The Hyperledger Fabric Performance and Scale working group currently works on a benchmarking framework called Hyperledger Caliper.

Several research papers have been published studying and testing the performance capabilities of Hyperledger Fabric. The latest scaled Fabric to 20,000 transactions per second.

Conclusion

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

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

Hyperledger Fabric is the most active of the Hyperledger projects. The community building around the platform is growing steadily, and the innovation delivered with each successive release far out-paces any of the other enterprise blockchain platforms.

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.

A blockchain ledger is often described as decentralized because it is replicated across many network participants, each of whom collaborate in its maintenance. We ’ ll see that decentralization and collaboration are powerful attributes that mirror the way businesses exchange goods and services in the real world.

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

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.

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.

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

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.

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

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

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

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. 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.

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.

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? 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?

That business network would look more like this:

This is a blockchain network, wherein every participant has their own replicated copy of the ledger. In addition to ledger information being shared, the processes which update the ledger are also shared. 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.

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.

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.

What is Hyperledger Fabric?

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

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.

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

Hyperledger Fabric also offers several pluggable options. Ledger data can be stored in multiple formats, consensus mechanisms can be swapped in and out, and different MSPs are supported.

Hyperledger Fabric also offers the ability to create channels, allowing a group of participants to create a separate ledger of transactions. 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. If two participants form a channel, then those participants — and no others — have copies of the ledger for that channel.

Shared Ledger

Hyperledger Fabric has a ledger subsystem comprising two components: the world state and the transaction log. Each participant has a copy of the ledger to every Hyperledger Fabric network they belong to.

The world state component describes the state of the ledger at a given point in time. It’s the database of the ledger. 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. The ledger, then, is a combination of the world state database and the transaction log history.

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

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. 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.

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

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

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

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. 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.

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

Hyperledger Fabric has been designed to allow network starters to choose a consensus mechanism that best represents the relationships that exist between participants. As with privacy, there is a spectrum of needs; from networks that are highly structured in their relationships to those that are more peer-to-peer.

介绍

一般来说，区块链是一个由分布式网络中的节点维护的不可篡改的账本。这些节点通过执行被共识协议验证过的交易来各自维护一个账本的副本，账本以区块的形式存在，每个区块通过哈希和之前的区块相连。

第一个被广为人知的区块链应用是加密货币比特币，而其他应用都是从它衍生出来的。以太坊是另一种加密货币，它采用了不同方法，整合了许多类似比特币的特征，但是新增了智能合约为分布式应用创建了一个平台。比特币和以太坊属于同一类区块链，我们将其归类为公共非许可（Public Permissionless）区块链技术。这些基本上都是公共网络，允许任何人在上面匿名互动。

随着比特币、以太坊和其他一些衍生技术的普及，越来越多的人想要将区块链基础技术、分布式账本和分布式应用平台用到企业业务中去。但是，许多企业业务对性能要求较高，目前非许可区块链技术无法达到。此外，在许多业务中，对参与者身份要求比较严格，如在金融交易业务中，必须遵循“了解客户（Know-Your-Customer，KYC）”和“反洗钱（Anti-Money Laundering，AML）”的相关法规。

对于企业应用，我们需要考虑以下要求：

参与者必须是已认证的或者可识别的

网络需要获得许可

高交易吞吐量性能

交易确认低延迟

与商业交易有关的交易和数据的隐私和机密性

当前许多早期的区块链平台正在为企业应用做调整，而 Hyperledger Fabric 从一开始就设计为企业用途。下面的部分描述了 Hyperledger Fabric（Fabric）与其他区块链平台的不同，并讲解了其架构设计的一些理念。

Hyperledger Fabric

Hyperledger Fabric 是一个开源的企业级许可分布式账本技术（Distributed Ledger Technology，DLT）平台，专为在企业环境中使用而设计，与其他流行的分布式账本或区块链平台相比，它有一些主要的区别。

一个主要区别是 Hyperledger 是在 Linux 基金会下建立的，该基金会本身在开放式治理的模式下培育开源项目的历史悠久且非常成功，发展了强大的可持

续社区和繁荣的生态系统。Hyperledger 由多元化的技术指导委员会进行管理，Hyperledger Fabric 项目由多个组织的不同的维护人员管理。从第一次提交以来，它的开发社区已经发展到超过 35 个组织和近 200 个开发人员。

Fabric 具有高度模块化和可配置的架构，可为各行各业的业务提供创新性、多样性和优化，其中包括银行、金融、保险、医疗保健、人力资源、供应链甚至数字音乐分发。

Fabric 是第一个支持通用编程语言编写智能合约（如 Java、Go 和 Node.js）的分布式账本平台，不受限于特定领域语言（Domain-Specific Languages，DSL）。这意味着大多数企业已经拥有开发智能合约所需的技能，并且不需要额外的培训来学习新的语言或特定领域语言。

Fabric 平台也是许可的，这意味着它与公共非许可网络不同，参与者彼此了解而不是匿名的或完全不信任的。也就是说，尽管参与者可能不会完全信任彼此（例如，同行业中的竞争对手），但网络可以在一个治理模式下运行，这个治理模式是建立在参与者之间确实存在的信任之上的，如处理纠纷的法律协议或框架。

该平台最重要的区别之一是它支持可插拔的共识协议，使得平台能够更有效地进行定制，以适应特定的业务场景和信任模型。例如，当部署在单个企业内或由可信任的权威机构管理时，完全拜占庭容错的共识可能是不必要的，并且大大降低了性能和吞吐量。在这种的情况下，崩溃容错（Crash Fault-Tolerant，CFT）共识协议可能就够了，而在去中心化的场景中，可能需要更传统的拜占庭容错（Byzantine Fault Tolerant，BFT）共识协议。

Fabric 可以利用不需要原生加密货币的共识协议来激励昂贵的挖矿或推动智能合约执行。不使用加密货币会降低系统的风险，并且没有挖矿操作意味着可以使用与任何其他分布式系统大致相同的运营成本来部署平台。

这些差异化设计特性的结合使 Fabric 成为当今交易处理和交易确认延迟方面性能较好的平台之一，并且它实现了交易的隐私和保密以及智能合约（Fabric 称之为“链码”）。

让我们更详细地探索这些区别。

模块化

Hyperledger Fabric 被专门设计为模块化架构。无论是可插拔的共识、可插

拔的身份管理协议（如 LDAP 或 OpenID Connect）、密钥管理协议还是加密库，该平台的核心设计旨在满足企业业务需求的多样性。

总体来看，Fabric 由以下模块化的组件组成：

可插拔的排序服务对交易顺序建立共识，然后向节点广播区块；

可插拔的成员服务提供者负责将网络中的实体与加密身份相关联；

可选的 P2P gossip 服务通过排序服务将区块发送到其他节点；

智能合约（“链码”）隔离运行在容器环境（例如 Docker）中。它们可以用标准编程语言编写，但不能直接访问账本状态；

账本可以通过配置支持多种 DBMS；

可插拔的背书和验证策略，每个应用程序可以独立配置。

业界一致公认，没有“可以一统天下的链（one blockchain to rule them all）”。Hyperledger Fabric 可以通过多种方式进行配置，以满足不同行业应用的需求。

许可和非许可区块链

在一个非许可区块链中，几乎任何人都可以参与，每个参与者都是匿名的。在这样的情况下，区块链状态达到不可变的区块深度前不存在信任。为了弥补这种信任的缺失，非许可区块链通常采用“挖矿”或交易费来提供经济激励，以抵消参与基于“工作量证明（PoW）”的拜占庭容错共识形式的特殊成本。

另一方面，许可区块链在一组已知的、已识别的且经常经过审查的参与者中操作区块链，这些参与者在产生一定程度信任的治理模型下运作。许可区块链提供了一种方法来保护具有共同目标，但可能彼此不完全信任的一组实体之间的交互。通过依赖参与者的身份，许可区块链可以使用更传统的崩溃容错（CFT）或拜占庭容错（BFT）共识协议，而不需要昂贵的挖掘。

另外，在许可的情况下，降低了参与者故意通过智能合约引入恶意代码的风险。首先，参与者彼此了解对方以及所有的操作，无论是提交交易、修改网络配置还是部署智能合约都根据网络中已经确定的背书策略和相关交易类型被记录在区块链上。与完全匿名相比，可以很容易地识别犯罪方，并根据治理模式的条款进行处理。

智能合约

智能合约，在 Fabric 中称之为“链码”，作为受信任的分布式应用程序，

从区块链中获得信任，在节点中达成基本共识。它是区块链应用的业务逻辑。

有三个关键点适用于智能合约，尤其是应用于平台时：

多个智能合约在网络中同时运行，

它们可以动态部署（很多情况下任何人都可以部署），

应用代码应视为不被信任的，甚至可能是恶意的。

大多数现有的具有智能合约能力的区块链平台遵循顺序执行架构，其中共识协议：

- 验证并将交易排序，然后将它们传播到所有的节点，
- 每个节点按顺序执行交易。

几乎所有现有的区块链系统都可以找到顺序执行架构，从非许可平台，如 Ethereum（基于 PoW 共识）到许可平台，如 Tendermint、Chain 和 Quorum 。

采用顺序执行架构的区块链执行智能合约的结果一定是确定的，否则，可能永远不会达成共识。为了解决非确定性问题，许多平台要求智能合约以非标准或特定领域的语言（例如 Solidity）编写，以便消除非确定性操作。这阻碍了平台的广泛采用，因为它要求开发人员学习新语言来编写智能合约，而且可能会编写错误的程序。

此外，由于所有节点都按顺序执行所有交易，性能和规模被限制。事实上系统要求智能合约代码要在每个节点上都执行，这就需要采取复杂措施来保护整个系统免受恶意合约的影响，以确保整个系统的弹性。

一种新方法

针对交易 Fabric 引入了一种新的架构，我们称为执行-排序-验证。为了解决顺序执行模型面临的弹性、灵活性、可伸缩性、性能和机密性问题，它将交易流分为三个步骤：

- 执行一个交易并检查其正确性，从而给它背书，
- 通过（可插拔的）共识协议将交易排序，
- 提交交易到账本前先根据特定应用程序的背书策略验证交易

这种设计与顺序执行模式完全不同，因为 Fabric 在交易顺序达成最终一致前执行交易。

在 Fabric 中，特定应用程序的背书策略可以指定需要哪些节点或多少节点来保证给定的智能合约正确执行。因此，每个交易只需要由满足交易的背书策略

所必需的节点的子集来执行（背书）。这样可以并行执行，从而提高系统的整体性能和规模。第一阶段也消除了任何非确定性，因为在排序之前可以过滤掉不一致的结果。

因为我们已经消除了非确定性，Fabric 是第一个能使用标准编程语言的区块链技术。

隐私和保密性

正如我们所讨论的，在一个公共的、非许可的区块链网络中，利用 PoW 作为其共识模型，交易在每个节点上执行。这意味着合约本身和他们处理的交易数据都不保密。每个交易以及实现它的代码，对于网络中的每个节点都是可见的。在这种情况下，我们得到了基于 PoW 的拜占庭容错共识却牺牲了合约和数据的保密性。

对于许多商业业务而言，缺乏保密性就会有问题。例如，在供应链合作伙伴组成的网络中，作为巩固关系或促进额外销售的手段，某些消费者可能会获得优惠利率。如果每个参与者都可以看到每个合约和交易，在一个完全透明的网络中就不可能维持这种商业关系，因为每个消费者都会想要优惠利率。

第二个例子考虑到证券行业，无论一个交易者建仓（或出仓）都会不希望她的竞争对手知道，否则他们将会试图入局，进而影响交易者的策略。

为了解决缺乏隐私和机密性的问题来满足企业业务需求，区块链平台采用了多种方法。所有方法都需要权衡利弊。

加密数据是提供保密性的一种方法；然而，在利用 PoW 达成共识的非许可网络中，加密数据位于每个节点上。如果有足够的时间和计算资源，加密可能会被破解。对于许多企业业务而言，不能接受信息可能受损的风险。

零知识证明（Zero Knowledge Proofs，ZKP）是正在探索解决该问题的另一个研究领域。目前这里的权衡是计算 ZKP 需要相当多的时间和计算资源。因此，在这种情况下需要权衡资源消耗与保密性能。

如果可以使用其他共识，或许可以探索将机密信息限制于授权节点内。

Hyperledger Fabric 是一个许可平台，通过其通道架构和 私有数据特性实现保密。在通道方面，Fabric 网络中的成员组建了一个子网络，在子网络中的成员可以看到其所参与到的交易。因此，参与到通道的节点才有权访问智能合约（链码）和交易数据，以此保证了隐私性和保密性。私有数据通过在通道中的成员间

使用集合，实现了和通道相同的隐私能力并且不用创建和维护独立的通道。

可插拔共识

交易的排序被委托给模块化组件以达成共识，该组件在逻辑上与执行交易和维护帐本的节点解耦。具体来说，就是排序服务。由于共识是模块化的，可以根据特定部署或解决方案的信任假设来定制其实现。这种模块化架构允许平台依赖完善的工具包进行 CFT（崩溃容错）或 BFT（拜占庭容错）的排序。

Fabric 目前提供了一种基于 etcd 库 中 Raft 协议 的 CFT 排序服务的实现。更多当前可用的排序服务请查阅排序服务概念文档。另外，请注意，这些并不相互排斥。一个 Fabric 网络中可以有多种排序服务以支持不同的应用或应用需求。

性能和可扩展性

一个区块链平台的性能可能会受到许多因素的影响，例如交易大小、区块大小、网络大小以及硬件限制等。Hyperledger Fabric 性能和规模工作组 正在开发一个叫 Hyperledger Caliper 的基准测试框架。

已经发表了一些研究和测试 Hyperledger Fabric 性能的文章。最新的一篇是将 Fabric 扩展到 20000 笔交易每秒（Scaled Fabric to 20,000 transactions per second）。

结论

任何对区块链平台严谨的评估都应该在其名单中包含 Hyperledger Fabric。

而且，Fabric 的这些特性使其成为一个高度可扩展的系统，该平台是支持灵活的信任假设的许可区块链，因此能够支持从政府、金融、供应链物流到医疗保健等各种的行业应用。

Hyperledger Fabric 是 Hyperledger 中最活跃的项目。围绕平台的社区建设正在稳步增长，每一个连续发布的版本所带来的创新都远远超过其他任何一个企业区块链平台。

什么是区块链?

一个分布式账本

区块链网络的核心是一个分布式账本，记录网络上发生的所有交易。

区块链账本通常被描述为 去中心化的，因为它会被复制到许多网络参与者

中，每个参与者都在 协作 维护账本。我们将看到去中心化和协作是强大的属性，反映了企业在现实世界中交换商品和服务的方式。

除了分散和协作之外，信息仅能以附加的方式记录到区块链上，并使用加密技术保证一旦将交易添加到账本就无法修改。这种“不可修改”的属性简化了信息的溯源，因为参与者可以确定信息在记录后没有改变过。这就是为什么区块链有时被描述为 证明系统 。

智能合约

为了支持以同样的方式更新信息，并实现一整套账本功能（交易，查询等），区块链使用 智能合约 来提供对账本的受控访问。

智能合约不仅是在网络中封装和简化信息的关键机制，它还可以被编写成自动执行参与者的特定交易的合约。

例如，可以编写智能合约以规定运输物品的成本，其中运费根据物品到达的速度而变化。根据双方同意并写入账本的条款，当收到物品时，相应的资金会自动转手。

共识

保持账本在整个网络中同步的过程称为 共识 。该过程确保账本仅在交易被相应参与者批准时更新，并且当账本更新时，它们以相同的顺序更新相同的交易。稍后您将学习更多关于账本，智能合约和共识的知识。目前，将区块链视为共享的复制交易系统就足够了，该系统通过智能合约进行更新，并通过称为共识的协作流程来保持一致。

为什么区块链有用？

现在的记录系统

现在的交易网络只不过是已存在的业务记录保存网络的升级版本。 业务网络 中的成员彼此交易，但他们分别维护各自的交易记录。他们所交易的东西，无论是 16 世纪的佛兰芒挂毯还是今天的证券，必须在每次出售时确定其来源，以确保出售物品的企业拥有的所有权。

现代技术已经从石碑和纸质文件夹演变为硬盘驱动器和云平台，但底层结构是相同的。因为没有管理网络参与者身份的统一系统，因而溯源非常费力，需要数天才能清理证券交易（其世界交易量以数万亿美元计算），合同必须手动签署和执行，而且系统中的每个数据库的信息都是孤立的，这也意味着单点故障。

即使可见性和信任的需求很明确，但在如今支离破碎的信息和流程共享方法下，不可能构建一个跨业务网络的记录系统。

区块链的不同

如果业务网络不是由“现代”交易系统代表的效率低下的老鼠窝（译者注：老鼠窝，指乱七八糟的系统），而是有一套在网络上建立身份，执行交易和存储数据的标准方法，那会怎么样？如果资产来源可以通过查看交易列表来确定，此列表一旦写入，无法更改，因此可信任，那会怎么样？

这就是一个区块链网络，其中每个参与者都有自己的账本副本。除了共享账本信息之外，还共享更新账本的过程。与今天使用参与者的 私人 程序更新其 私人 账本的系统不同，区块链系统具有 共享 程序来更新 共享 账本。

利用共享账本协调其业务网络的能力，区块链网络可以减少与处理私人信息相关的时间、成本和风险，同时提高信任和可见性。

你现在已经知道区块链是什么，以及为什么它有用。还有许多其他重要的细节，但它们都与信息和流程共享的这些基本思想有关。

什么是 Hyperledger Fabric?

Linux 基金会于 2015 年创建了 Hyperledger（超级账本）项目，以推进跨行业的区块链技术。它不是用来宣布一个区块链标准，而是鼓励通过社区流程开发区块链技术的协作方法，其中包括鼓励开放式开发、和随着时间的推移采用关键标准的知识产权。

Hyperledger Fabric 是 Hyperledger 中的区块链项目之一。与其他区块链技术一样，它有一个账本，使用智能合约，是一个参与者管理交易的系统。

Hyperledger Fabric 与其他区块链系统不同的地方是 私有 和 许可 。与允许未知身份参与网络的开放式非许可系统（需要诸如“工作量证明”之类的协议来验证交易并保护网络）不同，Hyperledger Fabric 网络的成员需要从可信赖的 成员服务提供者（MSP） 注册。

Hyperledger Fabric 还提供多种可插拔选项。账本数据可以以多种格式存储，共识机制可以交换替换，并且支持不同的 MSP。

Hyperledger Fabric 还提供创建 通道 的功能，允许一组参与者创建各自的交易账本。对于某些网络而言，这是一个特别重要的选择。这些网络中，一些参与者可能是竞争对手，并且不希望他们做出的每笔交易都被每个参与者知晓，例如，

他们只向某些参与者提供的特殊价格，而其他人不是。如果两个参与者组成一个通道，那么只有这两个参与者拥有该通道的账本副本，而其他参与者没有。

共享账本

Hyperledger Fabric 有一个账本子系统，包括两个组件： 世界状态 和 交易日志 。每个参与者都拥有他们所属的每个 Hyperledger Fabric 网络的账本副本。

世界状态组件描述了在给定时间点的账本的状态。它是账本的数据库。交易日志组件记录产生世界状态中当前值的所有交易；这是世界状态的更新历史。然后，账本包括世界状态数据库和交易日志历史记录。

账本中世界状态的数据存储是可替换的。默认情况下，这是 LevelDB 键值存储数据库。交易日志不需要是可插拔的。它只记录区块链网络使用账本数据库前后的值。

智能合约

Hyperledger Fabric 智能合约用 链码 编写，当该应用程序需要与账本交互时，由区块链外部的应用程序调用。在大多数情况下，链码只与账本的数据库、世界状态（例如，查询）交互，而不与交易日志交互。

链码可以用几种编程语言实现。目前支持 Go、Node.js 和 Java 链码。

隐私

根据网络的需求，企业对企业（B2B）网络中的参与者可能对他们共享的信息量非常敏感。对于其他网络，隐私不是最受关注的问题。

Hyperledger Fabric 支持私有网络（使用通道）是很重要的，因为网络是相对开放的。

共识

交易必须按照发生的顺序写入账本，即使它们可能位于网络中不同的参与者集合之中。为此，必须建立交易的顺序，且必须采用一种方法来拒绝错误（或恶意）插入到账本中的非法交易。

这是一个彻底的计算机科学研究领域，且有很多方法可以实现它，每个方法都有不同的权衡。例如，PBFT（实用拜占庭容错算法）可以为文件副本提供一种机制，使其能够保持各个副本的一致性，即使在发生损坏的情况下也是如此。或者，在比特币中，通过称为挖矿的过程进行排序，其中竞争计算机竞相解决加密难题，该难题定义所有过程随后构建的顺序。

Hyperledger Fabric 被设计为允许网络启动者选择最能代表参与者间存在的关系的共识机制。与隐私一样，有一系列需求；从他们的关系高度结构化的网络，到更加点对点的网络。