Hyperledger 白皮书

摘要

本文介绍了一些区块链原理性的使用案例，并概括了其基本要求及高层架构。区块链技术在不断演变中，Hyperledger 的设计正运用了这项技术构架。在相同网络的企业竞争合作中，Hyperledger 作为商对商（B2B）、商对客（B2C）的一种交易协议，既符合规章制度，又能够支持各类要求的实现。其设计（下面会详述）的核心元素是智能合约（又称链上代码）、数字资产、记录储存库、中心化共识网络、加密安全。此外，区块链主要产品还涉及行业性能要求、身份识别、私下机密交易、以及便携式共识模型等。

更多关于 Hyperledger 术语的问题，请查阅*术语表*.

背景

区块链是一种新兴技术，它能够彻底改变融资、供应链、及其他交易网络，为创新与发展带去新的机遇，同时还能减少运营成本，降低风险。自 2009 年来比特币在交易领域迅速崛起，许多企业及行业投入了大量资源来研究其底层技术，从而传播这广受欢迎但又颇具争议的加密币。

区块链是一种点对点分布式账本技术，它是第一个在金融领域获得动力的技术，因为它能够有效而安全地发行、交易、管理、服务资产。在记录系统（SoR）范围内，生态系统中的每一位成员都需要维护好自己的账本系统，并审核与其他成员的交易进展，这个过程既低效，又昂贵，而且没有内部组织操作流标准。而分布式账本则完全不同，它能够减少成本，使业务网络的建立更加容易，并且不再需要提供中心点控制。

由于共享账本概念在商业领域越来越受关注，区块链智能合约也因此引起了人们的注意。智能合约是各种商业规则的集合，它在区块链上运行，由一组利益相关方共有并进行验证。智能合约在商业自动化进程中非常管用，而且诚信可靠，它能够使利益相关方以团体形式处理并验证合约条款。Hyperledger 就是采用链上代码（chaincode）执行智能合约的。

比特币及其他加密币的开发就是用来对抗任何形式的审查的——任何人都能参与，而且不需要建立身份，只要贡献一点时间来完成运算周期就行。在比特币区块链模型中，没有中心机构来发放许可，因为这些网络是非授权的。他们要进行无数工作量证明运算，成本非常高。

Hyperledger 对传统区块链模型进行了革新，其中包括管理参与者的访问许可权。换句话说，Hyperledger 是有权限的共享账本。Hyperledger 为身份识别、审核及隐私提供了一个安全、健康的模型，从而缩短运算周期，实现有效扩展，应对业内各种运用要求。

进入 2016 年，关于区块链的认识越来越多，研究出一个适用且可扩展的解决方案变得日益紧迫。

为什么是一个新构架

作为一项刚开发的技术，区块链还不能满足商业交易中各类复杂要求。其可扩展性还存在问题，也不能支持私下机密交易，这些限制都使其难以在众多以业务为主的应用程序中运行自如。为了应对各种各样的市场需求，Hyperledger 的设计以产业应用为重点，解决了现有缺点，扩展了业内先行者原有的功能。

我们的设想

我们设想了区块链技术会为现代商业交易带来的革新与改变，分析了业内的使用案例及密钥要求，设计并建立了一套系统来推动区块链技术的广泛运用。

注：建议您先回顾下术语表再继续阅读以下材料。

一个涵盖众多网络的世界

我们希望能够出现众多区块链网络，使每个网络账本都能执行不同的业务。基于这一期望，我们开发了 Hyperledger。现在单一普遍的通用网络或许已经出现，但是要想网络账本核心功能的运行依靠其他网络来完成，还尚未有实例。Hyperledger 除了能够实现这种网络独立运行之外，它还有一个寻址系统，能够通过一个账本的交易，发现并利用另一账本中的正当交易与智能合约（链上代码 chaincode）。

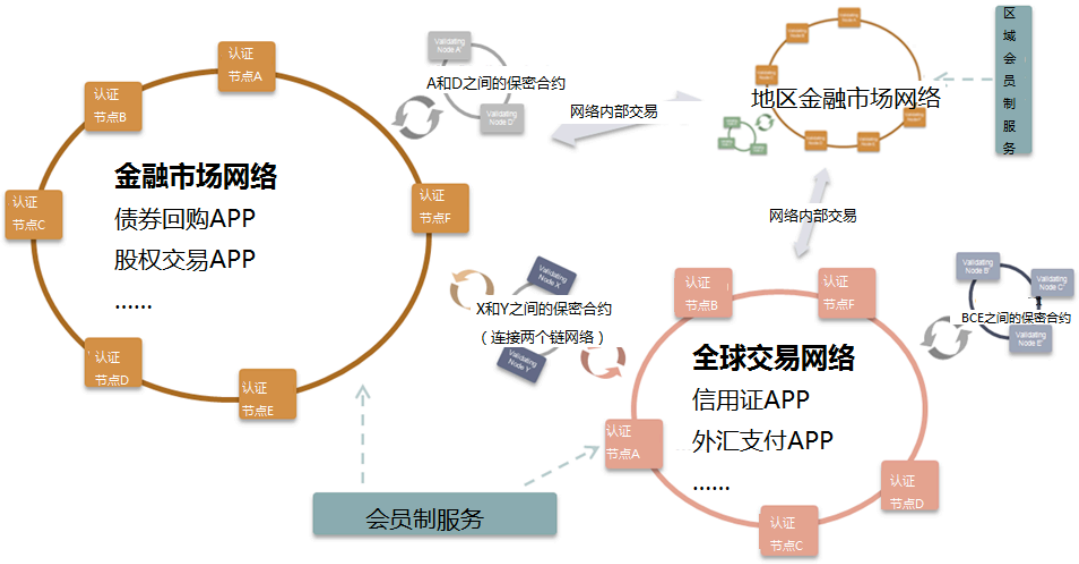


图 1：一个涵盖众多网络的世界

对网络权限的需求日益增加

有权限的网络是通过已知白名单组织来运行验证与非验证节点的，由网络发行机关来授予交易者一个识别身份。发行机关根据网络的目的来确定适当的访问权限，而这种权限需要进行身份识别，然后才能在网络进行交易。这一网络能够

公开运行，并且很容易就能融入到移动应用。它也能完全私下运行，受邀参与者只有通过验证，才能获悉其情况。因为 Hyperledger 构架的设计符合各种目的，能够允许各网络之间互相访问，而且其协议也能够满足各类使用要求，并且还设有不同级别的权限。

隐私及机密的重要性

我们相信任何区块链构架都有一个基本要求，那就是网络上任何一方的身份及行为模式，都不能允许未经授权的人通过账本来打探情况。我们也希望能够保证区块链用户的业务逻辑及其他交易参数的机密性，做到除了合约涉及的利益相关方或交易中的资产，没有人能够访问这些数据。

行业用例

我们编写了一套区块链初步要求，这些对接下来介绍的使用案例来说至关重要。

（注：下述用例能推动指导架构及测试驱动的开发。这些案例虽然还在发展中，但是所有参与者都应该认同其内容及技术层次的优先顺序。如果您觉得哪里不尽人意，就可以提出进行改变。请不要超过四个抽象用例，最好是三个。）

商务合同

商务合同能够通过编码，使双方或更多参与者自动执行合约条款，并且保证诚信。虽然区块链上的信息本质上是公开的，但是商对商合约要求具备隐私保护的机制，来保护敏感的商业信息，防止信息泄露给同样能够访问账本的外部人员。

虽然说保密协议对商业案例来说很重要，但是也有很多场合需要也应该做到合约公开，从而保证账本上所有人员都能轻松获取消息：比如，用来确定报价以供竞标的账本。这种合约就需要标准化，这样竞标人就可以很快获取信息。

资产存管

金融证券之类的资产必须在区块链网络上实现去中心化，这样所有同种资产的利益相关方就能直接访问每一资产，从而发起交易，获取相关信息，而不再需要通过层层中间人来进行了。交易基本实现实时结算，而且也必须保证所有利益相关方能够实时掌握资产情况。对任何资产种类，利益相关方都应有权增加商务规则，这样也能通过自动化逻辑应用来减少运营成本。创造资产的人必须像用例保证的那样，实现资产及相关交易规则保密或者公开。

供应链

区块链框架必须保证每一位供应链网络中的参与者都能够：输入并追踪原材料来源；记录部件生产的遥测数据；追踪航运商品的出处；保证包括成品生产、贮存、销售及后续事宜在内的所有记录都不可改变。除了之前描述过的**商务合约**及**资产存管**模式，供应链这一用例更多强调的是其深度搜索性，保证可以在过去层层交易中追溯所需记录。其核心是为每一个从其他组件品中制造出来的商品创建出处。

如果您想了解这些用例如何嵌入区块链系统的，或者想了解更多关于这些用例的要求，请点击*此处.*

特色要求

下面提到的特色要求以健全产业的用例为基础，推动了 Hyperledger 构架的开发。这些要求包括身份识别与可审核性、私下交易、保密合约、模块共识、性能绩效、可扩展性、链上代码（chaincode）以及智能合约。

身份识别与可审核性

交易的隐私性固然重要，但是区块链商业用途也要遵守一定的规章制度，使监管方能够访问调查交易记录。事后（有时得好几年后）交易一方必须提供身份识别以及资产所有权，在没有机制的情况下，通过身份识别来确定当事人身份以及其在账本上的活动。

于是 Hyperledger 协议顺势而生，它在注册机关登记了一个加密证书，从而锁定用户的机密数据。注册机关能够发布并撤销网络参与者的身份验证。针对每一个身份，协议都会生成一个秘钥，供成员在网络上进行交易，而且不会透露交易方的身份，保障网络隐私。

关于身份识别及可审核性还有疑问的话，请参考问答环节身份管理部分。

私下交易及保密合约

如果交易模式能够进行公开观察与分析，那么共享账本就会泄露商业关系的细节，而这些细节本不应透露给竞争者的。供应方或买方的圈子本来就小，支持双方贸易的系统就更不应该泄露任何一方的交易情况。因此，运用于商业活动的区块链就必须提供一个机制，来保证未授权的第三方不能获悉有关身份、交易模式、以及保密合约等条款。

Hyperledger 可以通过加密交易来保证其内容的机密性，只有利益相关方能够对其进行解密并执行。此外，业务逻辑（通过智能合约来实现）也做了加密处理来确保安全（如果利益相关方要求机密的话），并且只有在运行的时候才能加载、破解。关于这一点，会在之后架构部分详细阐明。

其他关于机密性的问题请参考问答环节的机密性部分。

模块共识

由于不同行业及领域有各自的网络要运行，所以不同的网络也要配置不同的共识算法来满足不同的使用要求。Hyperledger 协议下的共识必须运用可插入式算法，使用户自行在配置中自行选择共识算法。Hyperledger 协议的首次发行将提供拜占庭容错算法（BFT），这种算法采用的是实用拜占庭容错算法（PBFT）协议。我们期待今后社区中能有更多人分享其他共识算法模型。

*更多关于共识及 Hyperledger 预包装共识措施的问题请参考问答环节的共识*部分。

逻辑=链上代码=智能合约

区块链逻辑，或者我们常说的“**智能合约**”，是各方之间自动执行的协议，其中所有相关条款都由代码编成，能够进行自动结算，今后通过签名或其他触发事件来执行。在 Hyperledger 项目中，我们把它叫做“链上代码”（chaincode），通过链上代码来建立并区分区块链逻辑及其书面合约。（这个术语还在检测中，可能会有所改变。）

链上代码（chaincode）概念比智能合约还要再宽泛一些，智能合约这一概念是由 Nick Szabo 提出来的。链上代码能够通过任何主流编程语言进行编写，并且在 Hyperledger 的上下文背景层内执行。链上代码能够定义智能合约模块语言（类似 Velocity 或者 Jade），限制环境函数的执行以及其运算灵活度，从而满足法律合约的要求。

更多相关问题请参考问答环节链上代码部分。

性能绩效以及可扩展性

如果从经济角度来认识区块链框架的网络应用，那么在设计过程中就必须把长期的业绩考虑进去。一个账本或一套账本必须能够持续运作 100 多年，并且在用户可获取的时效内，允许客户使用其发掘、搜索、身份识别及其他功能。不过随着时间的推移，指定网络的节点数量及交易人员也会越来越多，因此，其框架还必须在保证性能表现的前提下，解决这些问题。

更多相关问题请参考问答环节用法部分。

架构

下面图 2 将为您展示 Hyperledger 所参考的架构，包括三大类：会员制、区块链、链上代码。这些都是逻辑结构，而不是对独立步骤、地址空间或者（虚拟）机器中分区组件的物理描述。

其中有些部分会从头建起，有的会采用现有的开源码，还有一些则与现有服务结合从而实现所需功能。

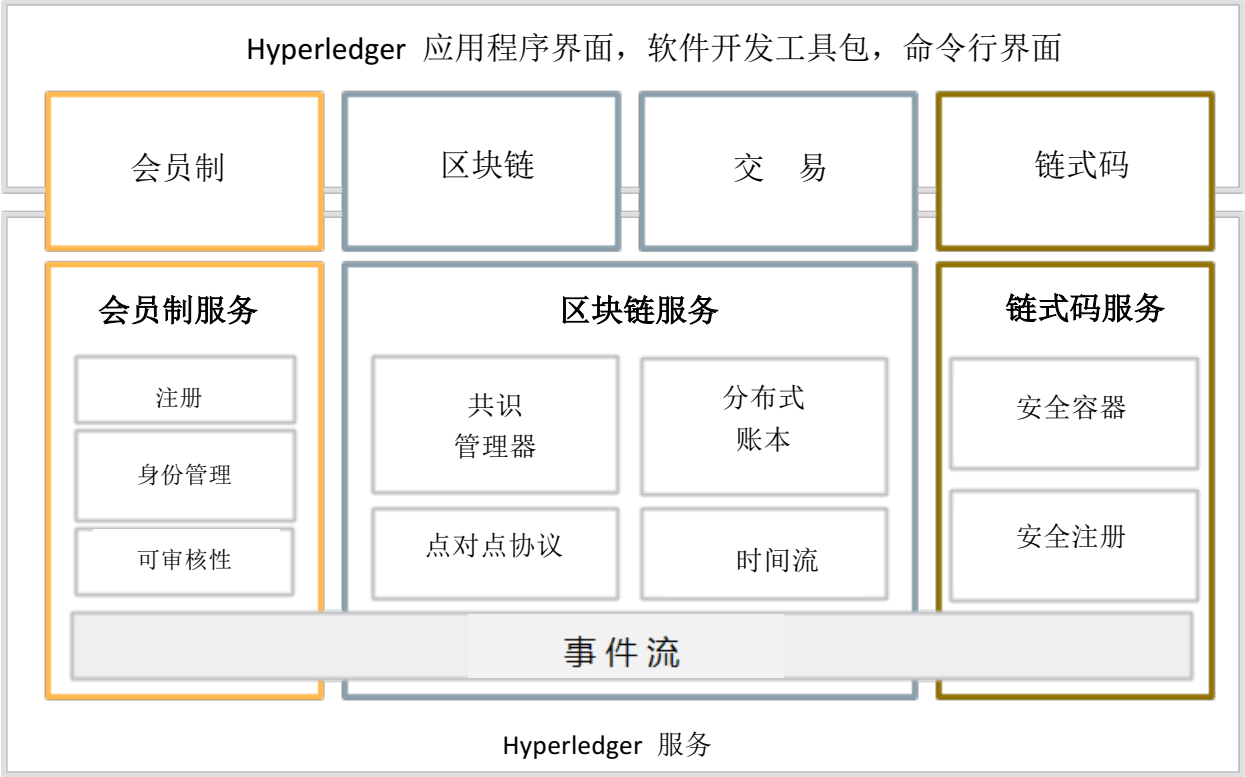


图 2：Hyperledger 参考架构

会员制服务负责管理的是网络上的身份识别、隐私与机密。参与者通过注册来获取身份，然后属性授权机构才能发放密钥来进行交易。声誉管理器能够使审计人员浏览某参与者的交易情况，如果审计人员已经获得参与者授予的适当访问权限的话。

区块链服务负责管理分布式账本，通过在超文本传输协议 2.0 建立的点对点协议进行。数据结构经过优化能够有效维护众参与者重复的整体状态。不同的共识算法或将嵌入每一个配置中，以保证高度一致性（通过拜占庭容错算法来处理错误，通过崩溃容忍来处理延误与中断，或借助工作量证明方案来应对审查）。

链上代码（Chaincode）服务负责提供安全又轻便的沙盒装载路径，供链上代码执行验证节点。整个环境是一个封锁且安全的容器，内含一个签署过的图片

库，包括安全的操作系统及链上代码语言，以及 Golang（准备期）、Java（计划期）、Node.js（计划期）等软件开发工具包组图与执行环境。如果有需要的话，也能添加其他程序语言。

会员制

Hyperledger 是一个私人认证网络协议，因此所有个体都需要注册会员服务来获取身份，从而进行网络访问及交易。在网络设置过程中，验证器可以鉴定交易所需权限级别。网络设置过程也会设置权限，实现轻松访问、高速高效应用、或严格控制网络环境。

注册服务能够控制并管理 Hyperledger 参与者的权限。

身份管理服务能够提供管理保障，根据 Hyperledger 参与者的身份与角色来授予信息披露权限。

可审核服务能够帮助授权个体将个人或团体（具体根据附属机构或所扮演角色来定）之间的交易连接起来，并且访问系统中某一用户的活动情况，或其本身的运行情况。

区块链

区块链服务由三个关键部分组成：点对点（P2P）协议，分布式账本以及共识管理器。

点对点协议运用的是 Google RPC 协议（谷歌远程过程调用协议，在超文本传输协议 2.0 标准下执行），其功能包括双向流、流控制、在单一连接下执行多路复用要求等。最重要的是，它能与现有网络基础设施结合，包括防火墙、代理服务器以及安全保护等。这一组合能够为对等节点采用的信息提供由点对点到多路传送的定义。

分布式账本用于管理区块链及整体状态，主要有三大属性：

- 高效计算每一区块完整数据集的加密散列值。
- 当一个对等点脱离同步并试图“赶上大部队”时，能够及时将这些最小的变化传送到数据集中。
- 减少每个对等点运行时所需的存储数据量。

分布式账本运用 RocksDB 来长久保存数据集，并且为了符合其三大属性，它还建立了一个内部数据结构来反映状态。大型文件（文档等）存储在区块链账本之外的贮存库中，而其散列值则能够作为交易的一部分存放在链内，这是为了保证文件的完整性。

Hyperledger 能够支持两种交易：代码部署交易以及代码调用交易。代码部署交易能够执行一条链上代码的提交、更新或者终止等命令，其验证节点必须保护代码及其执行环境的真实性与完整性。相比之下，代码调用交易则是一种用来执行链上代码函数的应用程序界面，其过程类似于运用统一资源标识符来调用 JEE 的小型服务程序。值得注意的是，其中每一个链上代码都负责维护自己的状态，而且函数调用是改变链上代码状态的一个常见办法。

共识管理器是一个抽象概念，它定义了共识算法与其他 Hyperledger 组件之间的接口。共识管理器接收交易，根据算法来决定如何组织以及何时执行这些交易。交易成功执行后就会在账本上反映变化。

Hyperledger 执行的是拜占庭协议，其在容错与扩展性方面都有强大优势。

Event Hub 是一个去中心化网络，它的属性很复杂，因为同一事件可能会多次出现，而且每次都会覆盖到每个对等节点。回调函数能够终止接收同一事件的多次调用，这样一来，对等节点（最好是非验证的本机节点）就能够管理应用程序相关的发布或订阅事件了。只要条件满足，对等节点就发动事件，顺序不分先后。这些事件不会永远存在，所以应用程序如果有需要就得及时捕捉。

链上代码（chaincode）

正如之前几个部分定义的，链上代码是一个去中心化的交易程序，在验证节点上执行。

链上代码服务使用 Docker 来存放链上代码，而不需要依靠特定虚拟机器或电脑语言。Docker 为沙盒链上代码的执行提供了一个安全、轻便的方法。其环境是一个封锁又安全的容器，还有一套签署过的图片库，包括安全的操作系统及链上代码语言，以及 Golang 的 SDK 组图与执行环境。如果有需要的话，也能添加其他程序语言。

安全注册服务能够使 Docker 安全注册 Hyperledger 图片库，以及包括链上代码在内的自定义图片库。

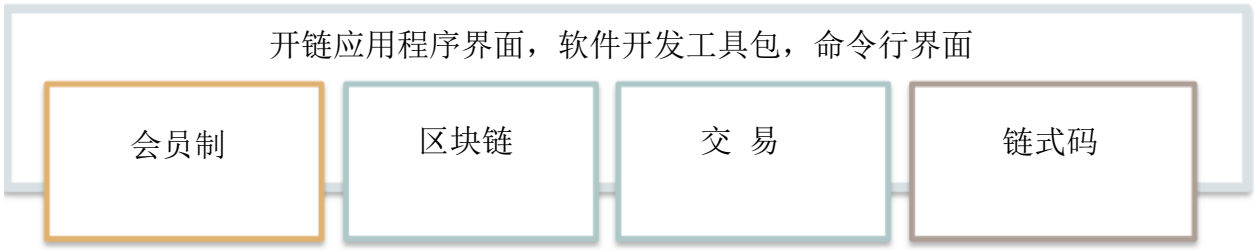
整个环境的状态代表了每一链上代码的状态。每一个链上代码都要采用“密钥-值”的格式来存储数据，并记录自己的状态，其中密钥和值都是任意字节数组。整个环境的状态还包括与其相连的区块的数量。

链上代码交易是有时间限定的，并且在链上代码装置过程中是按特定形式装配的，这和数据库调用或网站服务调用类似。如果超过交易时限，那么就按出错处理，不会给账本状态带去任何改变。如果调用的函数有相同保密范围限制的话，一个链上代码函数就能够调用另一个链上代码函数——也就是说，如果两个机密链上代码之间，有相同的一组验证器，那么它们之间就能互相调用。

新区块进行交易时，会维持区块链上最后区块的整体状态差量。如果当前区块达到共识，那么变量就会提交到数据库，之后整体状态的区块数量就会增加 1 个。如果节点没有达到共识，那么差量就会不会算入，数据库也不会进行修改。

应用编程界面

Hyperledger 包括了 REST、JSON RPC 应用程序界面、事件以及连接网络的应用程序软件开发工具包。典型的一点是，应用程序与一个对等节点进行交互需要通过某种形式来验证该个体的正当权限，其中客户的信息是有其身份识别签名的，而且要经过节点的核实。



Hyperledger 有一套命令行界面（CLI）来管理运行网络。这套命令行界面也可以用于链上代码测试过程中。REST 应用程序界面以及软件开发工具包（SDK）是建立在 JSON-RPC 应用程序界面上的，这也是现在最完整的应用程序界面层。软件开发工具包则能够适用于 Golang，JavaScript，Java 等语言中，如果有需要，也可以添加其他编程语言。

这一应用程序界面能够分成以下几类：

- 身份识别——通过登记来获得或撤销认证
- 寻址——定位并追踪交易来源
- 交易——账本上的执行单元
- 链上代码——在区块链上运行的程序
- 区块链——账本的内容
- 网络——区块链网络的信息
- 存储库——文件或文档的外部储存
- 事件——区块链上的订阅或发行事件

应用模型

Hyperledger 应用程序遵照 MVC-B 架构——模型、视图、控制器、区块链：

- 视图逻辑——手机或网络用户界面与控件逻辑进行交互。
- 控件逻辑——在用户界面、数据模型以及 Hyperledger 应用程序界面之间相互协调，从而驱动转换并执行链上代码。
- 数据模型——应用程序数据模型——运行链下数据，包括文档及大型文件。
- 区块链逻辑——区块链逻辑是区块链领域控件逻辑及数据模型的扩展。控件逻辑通过链上代码进行加强，而数据模型则由区块链交易来进行强化。

比如，使用 Node.js 环境的 PaaS 应用程序可能有一个网页前端用户界面，或者本地移动应用程序会有一个后端模型（由数据管理服务器提供）。逻辑语言或能与一个或多个链上代码进行交互从而执行区块链上的交易。

网络技术

有三种潜在配置模型：云服务器托管的单一网络、云服务器托管的多个网络，以及参与者托管的内联网。

云服务器托管的单一网络是最简易、最高效的拓扑结构，其中每一位参与者都有一组对等节点，包括验证节点。尽管网络在云环境下运行并且托管给物理硬件厂商，参与者还是能够根据合约来控制运算资源，从而在中心化环境中实现去中心化配置。

云服务器托管的多个网络能够使参与者通过云供应商管理对等节点，如果这些节点能够在超文本传输协议（HyperledgerTTP）下互相连接的话。

参与者托管的内联网通过超文本传输协议来使用参与者所有的网络。

结论

Hyperledger 的任务是将区块链技术引入大众市场。回顾了可行的区块链解决方案，也了解了业界领先者及技术推广者给出的相关用例后，我们相信区块链将会成为至关重要的技术模型，推动众多工业与企业进行革新。

我们注意到，业内目前急需一套为企业打造的区块链框架，做到既高效，又可扩展，并且能够为隐私与机密相关的需求提供企业级的支持。我们也发现了各种不同的用例，而每一用例可能需要不同的区块链底层实现。

为了全面发掘区块链技术的潜能，并且开创一套能满足各种使用需求的标准，我们设计了灵活且可延伸的 Hyperledger 构架。此外，我们还在引导 Hyperledger 协议的默认实现，其中涉及了各种计算机科学学科的先进成果。

为了增进您对 Hyperledger 协议的理解，您可以阅读我们的协议使用说明，它能够帮助您运用 Hyperledger 创建您的应用程序，并推动项目发展。