IRIS 网络白皮书

用于构建可信分布式商业应用的跨链服务基础设施及协议

Harriet Cao harriet@bianjie.ai
Haifeng Xi haifeng@bianjie.ai

NOTE:如果你可以在 GitHub 上阅读，那么我们仍然在积极完善这个文档。 请定期检查更新!

目录

免责声明

本白皮书及其相关文档用于接下来对 IRIS 网络的开发和应用。仅做信息传播之用并可能更改。

本文介绍了一个开发中的项目

本文基于 IRIS 基金会有限公司（IRIS Foundation Limited）提供的已知信息及假设，包含了符合该基金会理念的前瞻性声明。

本文所设想的 IRIS 网络尚在开发中，并将不断更新，这些更新包括但不限于关键治理和关键技术。 开发使用 IRIS 通证或与之相关的测试平台（软件）以及技术，可能无法实现或无法完全实现本白皮书所述的目标。

如果 IRIS 网络得以完成，可能与本文所述有所不同。本文不对未来的任何计划、预测或前景的成功性或者合理性做出陈述或保证，本文的任何内容都不应被视为对未来的承诺或陈述。

并非监管类产品的要约

IRIS 通证不代表任何一种有价证券或者任何其它司法监管下的产品。

本文不构成对任何有价证券或其他受监管产品的出价要约或询价邀请，也不构成以投资为目的的促销、邀请或询价。其购买条款并非提供金融服务的文件或任何类型的招股说明书。

IRIS 通证不代表任何平台或软件系统、或 IRIS 基金会有限公司以及其他任何公司的股权、股份、单位、资本权益或版权使用费、利润、回报或收入，不代表任何与平台有关的公有或私有企业、公司、基金会以及其他受监管实体的知识产权。

并非建议

本白皮书不构成任何对 IRIS 通证的购买建议。请不要依赖本白皮书去达成任何合同或购买的决策。

风险警告

购买 IRIS 通证并参与 IRIS 网络伴随着极大的风险。

在购买 IRIS 通证之前，您应该仔细评估并考虑风险，包括在 https://www.irisnet.org/上和其他任何文档中列出的风险。

仅代表 IRIS 基金会有限公司的意见

本白皮书所表达的观点为 IRIS 基金会有限公司的观点，并不一定反映任何政府、准政府、当局或公共机构(包括但不限于任何管辖范围内的任何管辖机构)的官方政策或立场。

本白皮书中的信息基于 IRIS 基金会认为可靠的来源，但并不能保证其准确性或完整性。

英文是本白皮书的授权语言

本白皮书和相关材料仅以英文发行。任何翻译并未经 IRIS 基金会有限公司或其他人员认证，其准确性和完整性无法保证，仅供参考。如果翻译与本白皮书的英文版本之间有任何不一致之处，则以英文版本为准。

非第三方从属关联或背书

本白皮书中提及的特定公司和平台仅供参考。使用任何公司和/或平台名称和商标并不意味着与其有任何的从属关联或背书。

您必须获得所有必要的专业建议

您有必要在决定是否购买 IRIS 通证或参与 IRIS 网络项目之前，必须咨询律师、会计师和/或税务专业人员，以及其他专业顾问。

IRIS 概览

IRIS 网络是以希腊女神 Iris 的名字命名的，她是彩虹的化身，是在天堂和人类之间忠诚传递消息的信使。

契约关系是人类社会的基本组成部分，区块链技术的重要性在于提供一种非常有效和低成本的方式来实现可靠的契约关系：第一次出现了多方参与复杂的业务交互时不再需要（本来非常昂贵的）信任。 也就是说区块链技术为分布式商业提供了最重要的元素：以极低的交易成本提升网络效益。越来越多的人认识到区块链作为新的价值互联网的影响力，并将逐步把当前的商业模式转变为更高效的分布式网络。 特别是内置于大多数现代区块链中的通证机制，强调每个网络参与者的权利，并将革新商业的现有模式[1]。

不过，区块链技术仍处于早期阶段。与其它新技术一样也存在缺点，包括有限的性能和还没有发展起来的治理机制。目前，这些缺点使区块链难以支持真实的分布式商业协作。 诸如 Hyperledger Fabric 和 R3 Corda，以及以太坊企业联盟（Ethereum Enterprise Alliance）等组织都在试图通过联盟链（consortium chains）解决这些性能和治理的问题，使区块链技术更适用于企业。然而，如今的联盟链由大型企业公司主导的，他们封闭式的链上链下治理模式非常低效。联盟链可能因为缺乏公有链的通证经济模型及其开放性和激励性而缺乏活力。

我们希望发展当前的区块链技术，让成千上万的中小企业(Small Medium Businesses，SMBs)，甚至是个体自由职业者，可以在一个开放的网络中提供他们的服务并享受回报。为了实现这一目标，我们确定了以下挑战以及随之而来的技术创新机会：

并非所有的运算都可以或应该以诸如智能合约这样的形式在区块链上实现

以太坊提供了图灵完备的虚拟机 [2]运行智能合约，带给人们开发分布式应用的诸多希望。 然而，智能合约只能处理确定性逻辑（因此每个节点在处理完同一交易和块后都能达到相同的状态），而大量现存的业务逻辑是不确定的，在不同时间和不同环境参数下可能会发生变化。 特别是现在，业务系统越来越依赖于计算机的算法进行决策优化，包括自然语言处理(Natural Language Processing，NLP），机器学习和操作研究算法。我们经常会故意在这些算法中添加一些随机性，以使决策不仅仅是局部最优状态，同时试图找到一个更好的次优结果。

另一方面，一些真实世界的业务逻辑应该在链下运行，不应该作为诸如可重复运算的智能合约这种类型来执行。利用分布式账本集成和协同链下的服务和资源，是进一步推动区块链技术在更多真实场景中应用的关键。

如何利用现有的区块链资源，包括公有链和联盟链

使用一个公有链来处理所有用例是不可行的。每天都有不同的区块链上线，各自专注于解决问题的一个方面，比如分布式存储、资产所有权或市场预测等。据 coinmarketcap.com 显示，目前有超过 1000 种加密货币在不同的交易平台上活跃。

构建业务应用程序时涉及处理存储以及不同数据源的来源，我们的另一个工作动机是如何通过重用一些现有的工作，比如存储(IPFS, SIA, Storj.io 等等)、数据发送（Augur，Gnosis，Oraclize 等）和物联网（IOTA 等）提供的这些专用的区块链，而不是“重新发明轮子”。

此外，有很多（近）实时业务交易确实需要更密切的联盟链/许可链/私有链来处理性能问题、安全问题和业务治理要求。因此，我们对分布式商业基础设施的愿景是要具备在多种异构链，包括公共链/联盟链/许可链/私有链之间具备互操作的能力。

跨链技术是满足这一需求非常自然的解决方案。 然而目前为止，现有的跨链技术主要是为了在已有区块链中提供互操作性，并专注于通证的价值转移。 如何使用不同区块链提供的资源，这一问题仍然没有答案。

比较现有的跨链技术如 Cosmos [3] 和 Polkadot[4]，提出的跨链技术，我们发现 Cosmos 为互操作性和可扩展性提供了更成熟的基础。 尤其我们发现 Cosmos 的多枢纽多分区（ "many hubs and many zones" ）和每个分区都是独立的区块链，拥有独立的治理模型（ "each zones are independent blockchains having independent governance models" 的设计，提供了一种非常合适的体系架构，可以用 SOC（Seperation of Concern，SOC）的方式对现实世界的复杂性进行建模。 为了最好地重用现有框架，我们提出了 IRIS 网络（IRIS Network），它是由一个枢纽和众多分区构成的去中心化的跨链网络，基于 Cosmos/Tendermint [5]实现，具有更为完善的通证使用。

鉴于 IRIS 网络是基于 Cosmos/Tendermint 设计的，我们将首先讨论 Cosmos/Tendermint，总结我们从 Cosmos/Tendermint 继承的特性和独特的创新。

Cosmos 和 Tendermint

Cosmos[3]想要建立“区块链的互联网”。这是由许多被称为分区“Zone”的独立区块链构成的互联网络，每个分区都由经典的拜占庭容错（Byzantine fault-tolerant，BFT）共识协议（如 Tendermint）提供支持。

Tendermint 提供了一个高性能、一致的、安全的 BFT 共识引擎，严格的分叉问责保证能够控制作恶者的行为。Tendermint 非常适合用于扩展异构区块链，包括公有链以及注重的性能的许可链/联盟链，像 Ethermint [6]就是一次对 Ethereum 以太坊 POS 机制的快速实现。 使用 Tendermint 在许可/联盟链域中的成功案例包括 Oracle [7]，CITA [8] 和 Hyperledger Burrow [9]。

Tendermint 作为共识协议用于在 Cosmos Hub 上构建第一个分区。Cosmos Hub 可以连接到许多不同类型的分区，并且通过一种相当于区块链之间的虚拟 UDP 或 TCP 的 IBC 协议（ Inter-blockchain Communication，IBC）实现跨链通信。通证可以安全地通过 Cosmos Hub 从一个分区转移到另一个分区，而不需要在分区之间的交易所或受信任的第三方。

为了使用 Cosmos Hub 开发强大的可互操作区块链和区块链应用，Cosmos SDK 提供了区块链常用模块的开发“入门套件”，而不是限制可实现的用户故事，从而为应用定制提供了最大的灵活性。

IRIS 技术创新

IRIS 网络的目标是为构建分布式商业应用提供技术基础设施，它超越了主要用于数字资产的现有区块链系统。

我们打算通过 IRIS 网络解决的关键挑战在于两个方面：

- 利用分布式账本支持链下运算资源的集成和协同
- 服务跨异构区块链的互操作性

我们通过将面向服务的基础架构融入 Cosmos / Tendermint 来应对这些挑战。

我们的设计继承了多年来面向服务架构(Service-oriented Architecture，SOA)实践的思维模式。 SOA 是一种架构方法，用于创建由自治服务构建的系统，这些系统具有明确的边界、共享模式和契约[13]。早期的 SOA 实践侧重于实施企业服务总线（Enterprise Service Bus，ESB），通过服务提供者和服务消费者之间的各种点对点连接组成公用总线，实现服务间的通信。但是，通过 ESB 集中管理服务可能会触发单点故障，也会增加服务部署的依赖性。最近大量的微服务架

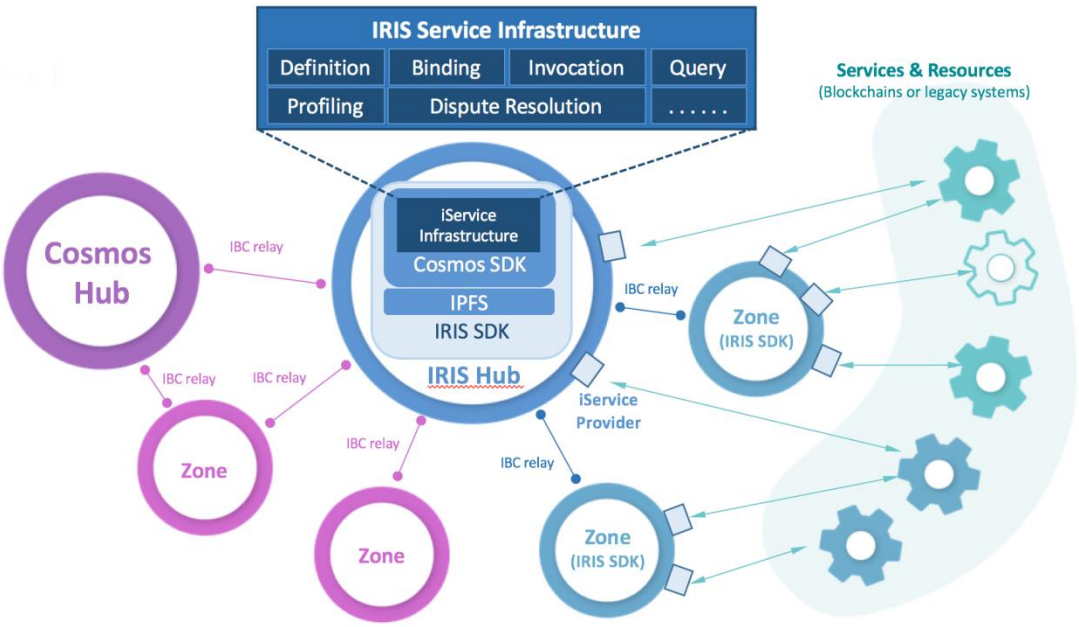
构可以看作是 SOA 的发展，不再关注 ESB 而是使用轻量级的消息队列进行服务间通信。在 IRIS 网络中，服务之间的通信旨在通过实施区块链作为信任机器来协同实现商业协作，使它在服务提供者和服务消费者之间很难建立信任的前提下也能运行。

IRIS 网络使用 Tendermint 协议作为高性能的共识引擎。利用 Tendermint 的区块链应用接口（Application BlockChain Interface，ABCI）提供的灵活性，我们定义了一组服务的基础交易类型，包括：服务提供，服务消费和服务治理。如前所述，大多数业务逻辑不适合作为区块链上确定的智能合约来实施，我们正在使用这个服务层将业务应用的特定逻辑和事务处理移出区块链，仅使用区块链对这些服务产生的结果达成共识。这一想法也受到区块链社区已有成果的启发，将一些复杂计算从主链上移除以解决性能问题，例如 Lightning Network 的离线状态通道[10]以及 Plasma 的防欺诈侧链[11]。尽管我们没有实施侧链，但是我们将传统业务逻辑计算从区块链中剥离出来，并将其用作复杂业务协作的可信中介总线。

对于跨链通信，Cosmos IBC [12]定义了一个协议用于将价值从一条链上的某个帐户转移到另一条链上的某个帐户的协议。而 IRIS 网络设计了新的语义，以允许利用 IBC 调用跨链计算资源。我们认为这种能力在构建可扩展的业务应用程序时非常重要。更详细的潜在用例将会在后面描述。

IRIS 网络旨在提供服务基础设施，以处理和协同链上的交易处理与链下的数据处理和业务逻辑执行。必要时，扩展的 IBC 功能允许那些链下的处理被跨链调用。按目前的设想，IRIS 网络还将包含客户端工具，一个支持跨链多资产存储的智能钱包以及服务消费方和提供方使用的 iServices。 我们计划提供 SDK 以轻松构建 iServices。 例如，对于特定的服务定义，客户端 Client SDK 将生成服务提供方的框架以及服务消费方的模块。

IRIS 网络设计



如上图所示，IRIS 网络在设计上与 Cosmos 网络具有相同的拓扑结构。 我们计划将 IRIS Hub 作为 Cosmos 众多分区和区域型 Hub 之一与 Cosmos Hub 连接起来。IRIS SDK 本身就是计划对 Cosmos SDK 的扩展，由 IRIS SDK 开发的 IRIS 全节点旨在提供一个服务的基础设施，并在内部集成了分布式文件系统 IPFS（InterPlanetary File System, IPFS）。

IRIS Services（又名“iServices”）旨在对链下服务从定义、绑定（服务提供方注册）、调用到治理（分析和争端解决）的全生命周期传递，来跨越区块链世界和传统业务应用世界之间的鸿沟。 IRIS SDK 通过增强的 IBC 处理逻辑来支持服务语义，以允许分布式商业服务在区块链互联网上可用。

尽管 IRIS 网络专注于为分布式业务应用提供创新解决方案，但它仍是 Cosmos 网络的一部分。 连接到 IRIS Hub 的所有分区都可以通过标准 IBC 协议与 Cosmos 网络中的任何其他分区进行交互。此外，我们相信通过引入服务语义层可以实现全新的业务场景，创新的 IRIS 网络将增加 Cosmos 网络的规模和多样性。

网络参与者

1. **服务消费者** 服务消费者是通过向网络发送请求并接收响应来使用链下服务的用户。
2. **服务提供者** 服务提供者是那些可能提供一个或多个 iService 定义的用户，并且通常是其它公有链和联盟链甚至传统企业应用系统中链下服务和资源之间的适配器。服务提供者监听和处理传入的请求，并将结果发送回网络。一个服务提供者可以向其它服务提供者发送请求而同时成为服务消费者。服务提提供者可以按计划为他们提供的任何服务收取费用，默认情况下服务费使用 IRIS 网络的原生费用通证“IRIS”定价；也可以在适当的时候考虑使用 Cosmos 白名单中的其他费用通证对服务定价。

3. **分析员** 分析员是一种特殊用户，代表了发起建立 IRIS 网络的 IRIS 基金会有限公司（IRIS Foundation Limited），这是一家注册在香港的股份有限公司。分析员是在分析模式中调用 iServices 的唯一授权用户，旨在帮助创建和维护服务提供者的概要文件，通过这些客观的概要文件服务消费者可以选择合适的服务提供者。

IRIS 服务

在本节中，我们列出了在 IRS 网络上部署 iService 时预计使用的技术参数。

服务定义

Method 包括 :

- ID (int): iService 中该方法的唯一标识
- Name (string): iService 中该方法的唯一名称
- Description (string): 对该方法的描述
- Input (string): 对输入参数的结构化定义
- Output (string): 对输出结果的机构化定义
- Error (string): 对可能出现的错误条件的结构化定义
- OutputPrivacy (enum): 设置此方法是非隐私的还是公钥加密的，可选值 NoPrivacy/PubKeyEncryption

ServiceDefinition 包括:

- Name (string): 该 iService 服务的名称
- Description (string): 对此 iService 服务的描述
- Tags (string): 此 iService 服务以逗号分隔的关键字
- Creator (string): 对此 iService 服务创建者的描述. *可选*
- ChainID (string): 最初定义此 iService 服务的区块链标识
- Messaging (enum): 设置此服务消息是单播还是多播，可选值 Unicast/Multicast
- Methods ([]Method): 定义此 iService 服务中可用的方法

CreateServiceDefinitionTx 交易包括:

- Definition (ServiceDefinition): 创建的服务定义

服务绑定:

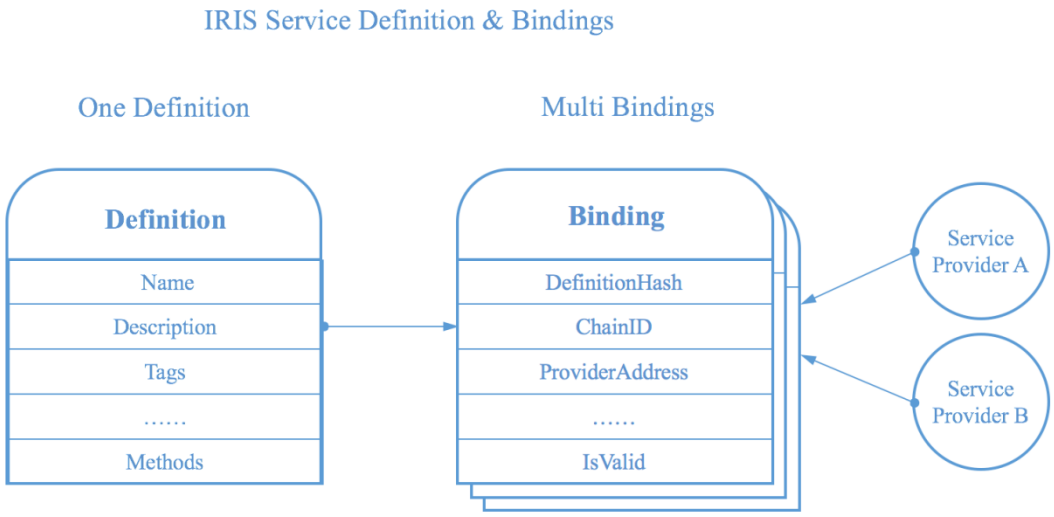
CreateServiceBindingTx 交易包括:

- DefinitionHash ([]byte): 服务定义的哈希值，由服务提供者绑定
- ChainID (string): 服务提供者所接入的区块链标识

- ProviderAddress ([]byte): 服务提供者的区块链地址
- BindingType (enum): 对服务是本地还是全局的设置，可选值 Local/Global；其中 Global 选项将绑定暴露到全局，反之则使用 Local
- ProviderDeposit (int64): 服务提供者的保证金。服务提供者必须提供大于系统参数 MinProviderDeposit 所设（以 IRIS 通证计数）的保证金，才能创建有效的绑定，押金越高意味着更值得信任
- ServicePricing (string): 服务定价。基于每个方法对服务价格模型的结构化定义，包括每次的调用成本、批量折扣、促销条款等；服务费默认使用 IRIS 通证也可以是白名单列出的其它费用通证
- ServiceLevel (string): 服务水平。对服务提供者自己认可所绑定服务水平的结构化定义，包括响应时间、可用性等。
- BindingExpiration (int64): 此绑定过期的区块高度，采用负数即表示“永不过期”

UpdateServiceBindingTx 交易包括:

- DefinitionHash ([]byte): 服务定义的哈希值，由服务提供者绑定
- ChainID (string): 服务提供者接入区块链标识
- ProviderAddress ([]byte): 服务提供者的区块链地址
- ChangeSet (string): 更改集，由前面三个字段确定的现有绑定所需更改内容的结构化定义



服务提供者可以在任何时间更新 ServicePricing，ServiceLevel 和 BindingExpiration，但他们的少量保证金将随后续（分别由 ServiceLevelUpdateSlash 和 BindingExpirationUpdateSlash 规定的）两种情况减少。当 BindingExpiration 设置的高度低于当前区块高度，将立即被解释为无效的绑定。

更新 ProviderDeposit 将始终被视为 *adding to*，即增加当前保证金余额。当保证金低于 MinProviderDeposit 时，绑定将失效，直到服务提供者增加余额高于阈值方可恢复。绑定过期或者失效的时候，保证金余额将自动返还给服务提供者。

BindingType 可用从 Local 更改为 Global，但反之不行。要把一个绑定从 Global 降到 Local，服务提供者只能先使绑定的问题失效，然后重新创建一个 Local 型的绑定。

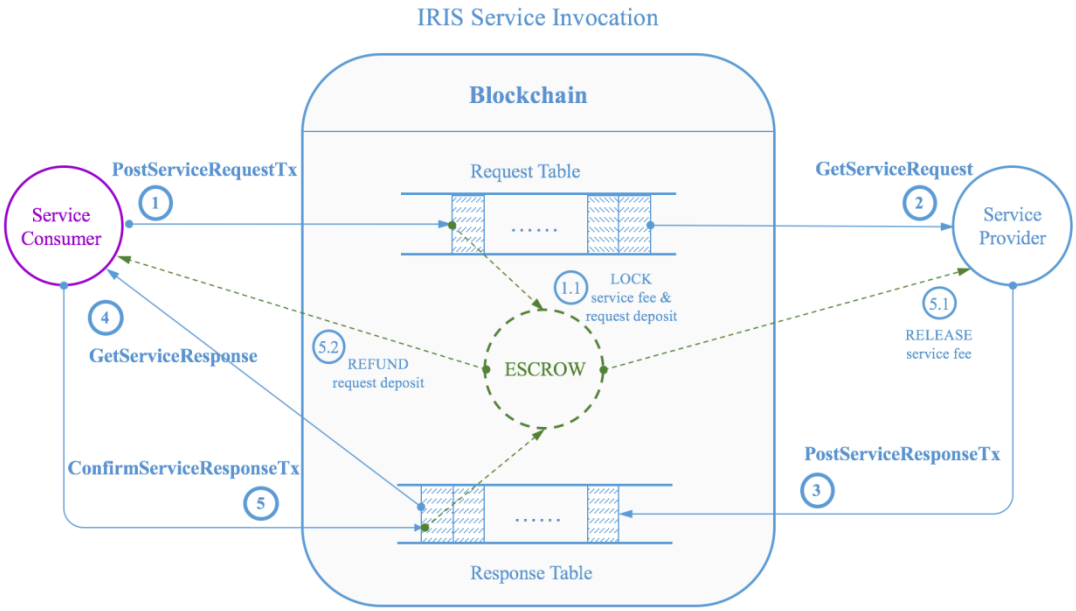
如果服务提供者出于某种原因需要将绑定移到一个新地址，是不允许直接更新 ProviderAddress 的；必须先使当前绑定失效，再使用所需的 ProviderAddress 创建另一个绑定.

一个 ServiceBinding 的对象将在应用程序成功的执行这两个交易时（例如，在 IRIS SDK 种的 iService 业务逻辑）被创建或更新.

ServiceBinding 包括:

- DefinitionHash ([]byte)
- ChainID (string)
- ProviderAddress ([]byte)
- ServiceLevel (string)
- ServicePricing (string)
- BindingExpiration (int64)
- IsValid (enum): 可选项 True/False

服务调用



服务消费者和服务提供者被建议通过 *端点（endpoints）* 交互。有两种服务端点 —— *请求表（request table）* 和 *响应表（response table）* （见上图）。服务请求被添加到请求表，感兴趣的服务提供者监控这些请求表并接收和处理发送给他们的请求； 服务结果（或错误）被返回由相应的服务消费者反过来监控的响应表中。

Multicast 多播服务的所有绑定共享一个请求表;而 Unicast 单播服务为每个绑定创建和维护单独的请求表。 至于另一个方向（的服务）将为每个服务消费者创建和管理专用的响应表。

ServiceRequest 交易包括:

- ChainID (string): 服务消费者所接入的区块链标识 ID
- ConsumerAddress ([]byte): 服务消费者所的区块链地址
- DefinitionHash ([]byte): 服务定义的哈希值
- MethodID (int): 被调用的方法 ID
- InputValue (string): 输入值的结构化表示
- BindingHash ([]byte):在'Unicast'单播服务情况下目标绑定的哈希值。*可选*
- MaxServiceFee (int64): 服务消费者愿意为一个'Multicast'多播请求支付的最大服务费金额 。*可选*
- Timeout (int): 服务消费者愿意等待响应返回的最大区块数

PostServiceRequestTx 交易包括:

- Requests ([]ServiceRequest): 被发送的服务请求
- RequestDeposits ([]int64): 服务消费者必须为每个请求者(使用 IRIS 计数)支付大于 MinRequestDeposit 的押金。此押金目的是为了激励消费者及时确认收到的服务响应（参考 ConfirmServiceResponseTx).

应用程序将验证用户与'ChainID'和'ConsumerAddress'一致的每一个请求、目标绑定的有效性、请求押金充足，服务消费者帐户余额足够支付请求押金和服务费用，并且请求的总数小于 MaxRequestPostBatch。

当一个已验证请求被追加到请求表中时，相关服务费用（对'Multicast'多播方式是'MaxServiceFee'）将从服务消费者的账户中扣除并锁定到第三方托管。

GetServiceRequest 查询包括:

- DefinitionHash ([]byte): 服务定义的哈希值
- BindingHash ([]byte): 服务提供者对此问题绑定服务的哈希值； 应用程序将验证绑定有效，并且调用者具有匹配绑定的区块链标识 ChainID 和服务提供者地址 ProviderAddress

- BeginHeight (uint64): 应用程序应当从此区块高度开始检索对服务提供者的请求，一般是最大请求数'BatchSize'和'MaxRequestGetBatch'中较小的一个
- BatchSize (int): 返回的最大请求数

ServiceResponse 查询包括:

- RequestHash ([]byte): 所匹配请求的哈希值
- BindingHash ([]byte): 服务提供者绑定服务的哈希值
- OutputValue ([]byte): 输出结果的结构化（可能加密）表示。*可选*
- ErrorMsg (string): 错误信息的结构化表示。*可选*

PostServiceResponseTx 交易包含:

- Responses ([]ServiceResponse): 服务回复内容

应用程序将验证服务提供者的每个响应是否带有匹配的区块链标识 ChainID 和地址 ProviderAddress，并且该交易中的响应数少于 MaxResponsePostBatch。经过验证的请求将附加到目标消费者的响应表中。

GetServiceResponse 查询包括:

- RequestHash ([]byte): 原始请求的哈希值，应用程序将校验调用者是否有匹配的区块链标识'ChainID'和服务消费者地址'ConsumerAddress'
- BeginHeight (uint64): 应用程序应当从此区块高度开始检索服务提供者的响应，一般是最大响应数 BatchSize 和 MaxResponseGetBatch 中的较小的一个
- BatchSize (int): 返回的最大响应数

ConfirmServiceResponseTx 交易包含:

- ResponseHash ([][]byte): 待确认响应的哈希值

应用程序将验证每个待确认响应是否确实是由调用者发起的请求，并且此交易中的响应数量小于 MaxResponseConfirmBatch。

从 Timeout 超时窗口中退出的响应(对于'Multicast'多播服务，当 MaxServiceFee 随响应返回增加而消耗光时)将不会被应用程序接受。服务消费者一收到'Unicast'单播响应就立即开始处理。然而，对于 Multicast 多播响应，必须等待 Timeout 超时窗口结束，然后才开始处理可能收到的全部响应。

当一个 Unicast 单播响应被用户确认，相关服务费用将从托管账户中释放到匹配的服务提供者账户 ——其中扣除少量（由'ServiceFeeTaxRate'定义的）税金并添加到*系统准备金（system reserve）* 中； 同时，相关请求的押金也将退还给服务消费者。

在 Multicast 多播请求的情况有点复杂：每个响应确认时，只有部分请求押金被退还给服务消费者，即此响应相关服务费用占 MaxServiceFee 的比例； 在所有的响应被确认之后，此请求剩余的托管余额将会退还给服务消费者。

如果请求超时而没有看到任何响应，则应用程序将托管中的相关余额以及请求押金全额退还给用户。但是，如果用户没有（在 ResponseConfirmTimeout+响应区块数的高度之前）及时确认回复，请求押金将被执行一个（由 ResponseConfirmDelayPenaltyRate 定义的）小惩罚再退还给消费者，与此同时，相关服务费将照常释放给服务提供者。

争议解决

在任何情况下如果服务消费者对服务响应不满意，就应该存在一种机制允许服务消费者发出投诉，从而获得可接受的解决方案，而不必求助于诸如法律系统等集中的权威。同时，这种机制应避免引入可能会被任一方滥用的主观评价。

解决出现在 IRIS 网络上的争议，其过程类似于服务调用，不仅服务消费者向服务提供者发送 Complaint（服务），而且服务提供者以一个 Resolution（服务）进行响应。这些交互是通过一对称为 *投诉表（complaint table）* 和 *解决方案表（resolution table）* 的全局端点来实现的。

在 IRIS 网络目前的设计中，提交投诉时需要一份消费押金。如果服务消费者不及时确认某个解决方案，将从该押金中扣除罚款。同样地，如果服务提供者不及时响应投诉，他的保证金将被部分削减。

Complaint 包含:

- ResponseHash ([]byte): 对争议响应的哈希
- Problem (string): 对服务响应的问题描述
- PreferredDisposal (enum): 可以是 Refund 或 Redo 选项

Resolution 包括:

- ComplaintHash ([]byte): 对所匹配投诉的哈希
- Disposal (enum): 可以是 Refund 或 Redo 选项
- Refund (uint64): 服务费退款. *可选*
- OutputValue ([]byte):输出结果的结构化（可能加密）表示。 *可选*

如上所述，我们预期的争议解决流程可能不具有法律约束力。尽管如此，我们认为这将为解决 IRIS 网络上的常见争议提供有效手段。

服务分析

引入 iService 生态系统带来一些挑战。一个主要的挑战是找到一种方法，让服务消费者更容易发现合适的服务提供者——服务消费者需要性能指标来评估和选择服务提供商，但如果没有服务消费者的使用，性能指标也就不可用。

为了试图解决这个循环问题，我们计划引入一种分析机制，在这个机制中，一个享有特权的系统用户即分析员，在常规的调度下调用所有活动的服务。这将使网络中的客观性能数据（例如响应时间、可用性、投诉处理等）对实际消费者有用。

服务分析调用将免除服务费用和消费押金，但会产生网络交易费用。这些调用来自那些被应用程序识别和认可的保留地址。

对于新服务，分析活动将保持相对稳定的水平，随着它们开始吸引真正的服务消费者调用并预计生成更多的性能数据，分析活动将逐渐减少单个服务的使用。

分析过程中发生的交易费用默认情况下从系统准备金（system reserve）中支付，必要时基金会准备金（Foundation reserve）将会介入。

查询

上述所有与服务生命周期相关的对象都可以使用 ABCI 'Query'接口[[3]]查询到。这些查询将通过'Query'连接执行，并不参与共识过程。我们已经看到了如何得到的 GetServiceRequest 和得到 GetServiceResponse 查询作为服务调用过程的一部分。上面描述的所有与服务相关的生命周期对象都可以使用 ABCI 查询接口 3 来查询。

以下是我们目前计划的查询的一个非详尽的摘要：

服务对象

Object	Commonly Used Filters	Authorization
Service Definition	Name, keywords, source (chain ID), messaging type, with active bindings...	Anyone can query
Service Binding (for a given definition)	Location (local or remote), pricing, service level, expiration...	Anyone can query
Service Request	Service definition and binding, blockchain height, batch size	Only matched provider(s)
Service Response	Service request, blockchain height, batch size	Only matched consumer

性能指标

Area	Metrics	Authorization

Area	Metrics	Authorization
Provider (address)	Number of services provided (ever and active), response time (min, max and average), requests served (local and remote), requests missed, complaints received, complaints ignored, ...	Anyone can query
Provider (binding)	Active time, response time (min, max and average), requests served (local and remote), requests missed, complaints received, complaints ignored, ...	Anyone can query
Consumer	Number of services ever used, requests made, requests confirmed (in time and missed), complaints made, resolutions confirmed, ...	Anyone can query
Consumer (service, binding)	Requests made, requests confirmed (in time and missed), complaints made, resolutions confirmed, ...	Anyone can query

IBC 改进

在 Cosmos 上建立自己的服务基础架构有一个独特优势：*服务可以部署一次，并通过区块链互联网随时随地进行调用*。具体来说，我们的计划是完成以下内容：

1. 服务定义广播到 IRIS 网络中的每个分区；
2. 全局服务绑定广播到 IRIS 网络中的每个分区；
3. 针对远程提供者的服务请求或投诉被路由到提供者所连接的区块链；
4. 针对远程消费者的服务响应或决议被路由回消费者所连接的区块链。

在处理一个 CreateServiceDefinitionTx 交易时，应用程序被设计为首先在本地验证和存储 ServiceDefinition 对象，然后创建一个包含了对每个相邻链定义的 IBCPacket。

每个相邻链最终从相应的中继过程中接收包含该数据包的 IBCPacketTx；如果此定义在接收链中尚不存在，则后者将通过为他的每个相邻链（除了最初接收数据包的源链之外）创建一个 IBCPacket 来传递此定义；如果该定义已经存在，接收链将停止传递此定义。

同样，当 ServiceBinding 创建或更新它的 BindingType 集合或更新为 Global 时，将为每个相邻链创建一个包含绑定的 IBCPacket 数据包，并在整个 IRIS 网络中进行广播。

上述 IBCPacket 由以下部分组成：

- Header（IBCPacketHeader）：数据包的头部

- Payload（ServiceDefinition 或 ServiceBinding）：服务定义或绑定的字节数

前面提到的 IBCPacketHeader 由以下部分组成：

- SrcChainID（string）：创建此数据包的区块链标识 ID
- DstChainID（string）：此数据包将派往的相邻区块链标识 ID
- Number（int）：数据包对应的唯一编号
- Status（enum）：'NoAck'
- Type (string)：“iris-service-definition”或者是“iris-service-binding”

现在让我们来看看如何通过 IBC 发生跨链服务调用。当请求一个 Unicast 单播服务时，应用程序检查目标绑定是否是 Local 本地的，如果是 Local 则将 ServiceRequest 追加到相应的请求表中，如 2.2 所述；否则，将会创建一个包含 ServiceRequest 的 IBCPacket。

包含一个'ServiceRequest'的 IBCPacket 由以下部分组成：

- Header（IBCPacketHeader）：数据包的头部
- Payload（ServiceRequest）：服务请求的字节数

前面提到的 IBCPacketHeader 由以下部分组成：

- SrcChainID（string）：创建此数据包的区块链标识 ID
- DstChainID（string）：远程服务提供者所在的区块链标识 ID，例如'ServiceRequest'、'ServiceBinding'、'ChainID'
- Number（int）：数据包对应的唯一编号
- Status（enum）：'AckPending'
- Type（string）：“iris-service-request”
- MaxHeight（int）：当前高度加上'ServiceRequest.Timeout'

当远程请求最终到达目标链时，应用程序将其追加到相应的端点（请求表）中，以便进行目标绑定。对此远程请求的响应将被包装在一个收据 IBCPacket 中，该收据被一直路由回到源链，并将其追加到原始消费者的远程端点（响应表）中。

对远程的 Multicast 多播服务的请求以相同的方式处理，只不过可以在源链中生成多个 IBCPacket。

远程投诉和解决的工作方式与请求和响应的方式相同，因此在此不做详细阐述。

下面是应用程序依赖 IBCPacket 类型的完整列表：

类型	iService 对象
"iris-service-definition"	ServiceDefinition

类型	iService 对象
"iris-service-binding"	ServiceBinding
"iris-service-request"	ServiceRequest
"iris-service-response"	ServiceResponse
"iris-complaint"	Complaint
"iris-resolution"	Resolution

用例

在本节中，我们为 IRIS 网络列出了一些可能的用例。

分布式人工智能用于隐私保护下的数据分析

这个用例的服务基础架构已由位于上海的初创公司边界智能进行了原型设计，并将其应用到联盟链产品 BEAN (BlockchainEdge Analytics Network)中，用于解决长期已来为运行分析模型获取数据的挑战。尽管同态加密是允许通过加密数据实现计算的关键方法之一，但由于性能低下，实际上无法解决现实世界的机器学习问题。因此，BEAN 的创建提供了另一种解决方案--利用传统的分布式人工智能研究[14]中的模型并行性和 SOA 设计模式，作为区块链的附加层开发分布式分析服务。

为了保护数据存取，运行在数据端的（部分）模型需要开放给客户端，并在服务定义中说明。由于只有部分模型开放给客户端，模型开发人员不必担心有人窃取他们的想法；同样，数据拥有者永远不需要担心失去对其数据使用的控制，因为数据不会离开他们的数据源。

其他潜在的好处可能包括以下几点：

1. 仅在链上交换少量参数数据，这可以帮助提高性能。
2. 一种更实用的数据使用审计方法，这在医疗保健领域经常被用到。

医疗健康数据隐私度高，涉及众多安全需求。这就为医疗健康数据用于跨组织协作的目的提出了挑战（例如用于辅助诊断的跨医院会诊记录搜索，新药临床试验的患者身份，健康保险自动理赔等）。最小化可行产品（Minimum Viable Product,MVP）服务层的实现是建立在 Ethermint 的基础之上，试图连接众多医院、保险公司和分析服务提供者，以提供具有隐私保护的医疗健康数据分析能力。

支持链上服务注册和调用的智能合约已经实现。链下数据处理的一个例子是支持相关诊断组（Diagnosis Related Group,DRG）的分组分析服务。更具体地说，当一个医院用户调用 DRG 服务时，原始医疗记录将在链下进行处理，使用服务提供者提供的客户端 NLP(由 SQL 和 Python 实现)代码存根来提取通过区块链接收 DRGS 服务传来的结构化数据，而不传递高度机密的原始医疗记录。

BEAN 场景阐述了一个更复杂的服务使用案例，包括实现分布式分析、连接服务提供者和服务消费者、利用区块链提供可审计交易平台以及可信任的分布式计算基础。

数据和分析电子市场

通过对几个 AI+区块链项目的研究，发现似乎大部分项目都旨在提供数据交换市场和分析 API 市场。在建议的 IRIS 基础架构中，通过使用 IRIS 服务提供者 SDK 来发布数据作为数据服务和包装分析 API 作为分析服务，从而轻松地构建这些网络。

分布式电子商务

将建议的 IRIS 基础架构与传统系统（例如 ERP）集成以获取库存信息，或对可信数据源进行链间查询以获取交通和天气数据等信息，此方法与许多企业应用程序开发人员已经熟悉的方法非常相似。通过集成这些服务来支持分布式电子商务应用程序，就有可能提供与中心化系统（例如 Amazon 亚马逊或 Alibaba 阿里巴巴）相近的用户体验。

公有链和联盟链的结合

对于许多业务场景而言，采用混合了公有链和联盟链优良特性的混合架构，从而可以提供有益的结果，特别是在性能、安全性和经济激励方面。

例如，医院和保险公司可以组成联盟链以支持高性能的医疗保险交易，同时识别其他信息，例如关于某些疾病的全球服务的统计数据，这些信息可以从其他公有链中调用。从公有链接收到的通证可以返回给联盟链中的信息提供者，从而激励系统参与者改善和提高服务质量。利用 IRIS 提供的这种基础架构，可以在满足严格的性能和安全要求的前提下实现大规模的自发协作。

IRIS 服务基础架构可以支持许多用例，例如更高效的基于资产的安全系统、分布式监管技术（如严格评估，互助市场等）。IRIS 项目计划之一还包括与此类应用程序项目团队展开密切合作，以支持并使他们能够拥有所需的区块链基础架构，让他们专注于更高效地交付预期的业务价值。

通证经济

与 Cosmos 网络相似，IRIS 网络也被设计为支持多通证模型。这些通证被不同的分区所拥有，同时可以通过 IRIS 枢纽从一个分区转移到另一个分区。我们构建了两种通证类型来支持 IRIS 网络上的操作：

- 权益通证
- 费用通证

权益通证

与 Cosmos 网络 [15]15 采用同样的权益机制设计，IRIS 枢纽也拥有自己特殊的原生通证来表示权益。通证命名为 IRIS。关于 IRIS 通证，我们设计了一些具体功能，其包括：

- 通过验证人和代理人系统，将 IRIS 通证整合到 IRIS 网络的共识引擎验证人中；
- 代表投票权，参与 IRIS 网络的治理

费用通证

在 IRIS 网络中有两种费用通证：

- **网络费用** 用来进行垃圾请求防范和支付验证人进行账本维护的报酬；
- **服务费用** 用来支付部署 iServices 的服务提供者的报酬。默认支付服务的通证为 IRIS 通证

IRIS 网络旨在支持所有来自 Cosmos 网络白名单中的费用通证，例如 Photon 光子，以及 IRIS 通证。

支持各种白名单的中的费用通证是我们计划从 Cosmos 中采用的一个特性。它可以为网络的参与者提供增强的体验。在 Cosmos 中，对于网络费用通证 network fee token，每一个验证人都拥有配置文件来定义他自己对每一个费用通证的价值评估。验证人可以运行一个独立的定时器，根据实时市场数据更新配置信息，或者使用默认的配置数据。

对于服务费用通证 service fee token 的设计，同样支持多通证模型。因此，在 IRIS 网络上的服务提供者，可以自由选择白名单中出现的通证为其服务收费。

为了帮助 IRIS 网络的参与者降低通证价格的波动性，基金会希望发展全球性的 iService 以从不同的交易所、或者从 oracle 预言机的潜在信息中获取市场价格数据。

权益通证和费用通证都会进一步细化以确保符合法律和监管规定的义务。

初始通证分配

最开始，系统预计发放 2000000000 个 IRIS 通证。IRIS 通证的分布计划如下：

- **私募**：20%
- **核心开发团队**：20%（自 IRIS Hub 主网上线起的 4 年成熟期，每月释放 1/48。）
- **IRIS 基金会**：15%（保留用作基金会的日常建设和运营）
- **生态建设**：45%（在链接到 IRIS Hub 的分区中进行通证交换；激励潜在用户；奖励的合作伙伴）

如果 IRIS 网络完全开发完毕，IRIS 通证每年的通胀速率会根据账户进行调整，因为事实上，很大一部分流通中的 IRIS 通证都将被参与者作为权益证明主动投入到共识引擎中。

首要说明的是，私募的 IRIS 通证将用于开发 IRIS 网络。这部分通证的使用计划如下：

- **运营**：10%（包括服务提供者和承建商的费用，例如，审计、顾问、法律和其他第三方费用，以及其它管理费））
- **软件开发**：50%（包括直接用于开发上线所需的成本、费用和开支）
- **开发者支持**：10%（包括黑客马拉松、志愿者奖励和培训项目）
- **研发赞助**：10%（包括会议、研究项目和大学合作）
- **市场拓展**：20%（包括业务发展，社区发展和推广，以及出差、交流、出版、发行和其他费用等）

路线图

预期的 IRIS 项目路线图如下。这里我们重申一下，路线图只是一个大概方向，将来根据项目实施会有变化。

- **盘古**（2018 年 1 月~2018 年 7 月）：IRIS 项目的第一阶段，将集中在构建和运行 IRIS Hub 以及与 Cosmos Hub 的交互。同时，我们将发布一个初始版本的 IRIS 网络移动客户端。
- **女娲**　（2018 年 7 月~2018 年 11 月）：第二阶段主要集中在构建 IRIS Service Layer。这将涉及启用服务定义、绑定、调用和查询。在这个阶段，我们也打算为开发者准备一个 beta 版的 IRIS SDK。
- **夸父**　（2018 年 12 月~2019 年 5 月）：第三阶段主要专注于 IRIS 网络的增量升级，以支持我们计划中先进的 IRIS Service 治理特性。
- **后裔**　（2019 年 6 月之后）：第四阶段专注在 IRIS 网络、IRIS SDK 和移动客户端的技术创新，以及开发者的参与。

团队

Tendermint（该团队开发了 Tendermint 共识引擎，当前正在开发 Cosmos），**Wancloud 万云**（ 万向区块链 子公司）和 **边界智能** 将共同构建 IRIS 网络的基础设施。

Tendermint 将在扩展 Tendermint ABCI 和 Cosmos IBC 技术方面，为 IRIS 项目团队提供技术咨询和开发支持。Wancloud 万云 将是 Cosmos 和 IRIS 生态系统的关键战略合作伙伴，它将积极参与 Cosmos 和 IRIS 在亚太地区的开发和发展。

边界智能将是 IRIS 网络的核心开发团队，IRIS 充分借助团队创建分布式智能分析服务的经验，实现医疗数据分析与交换。边界智能是一家于 2016 年在上海成立的初创公司，专注于利用先进的区块链技术和人工智能技术为医疗和金融行业开发创新产品和提供解决方案。边界智能中的一个核心产品 BEAN（区块链智能信息边缘分析网络 Blockchain Edge Analytics Network）是一条许可链，它利用自然语言分析及机器学习技术为隐私保护、医疗健康数据的分析和交换提供分布式智能分析服务。边界智能正不断扩展 BEAN 功能来构建 IRIS 网络，同时，**边界智能**也是 Cosmos 网络在中国的运营和服务合作伙伴。

核心成员

曹恒

曹恒 是边界智能的创始人，该公司是一家成立于上海的初创公司。边界智能专注于利用分布式 AI 技术为区块链提供智能化服务，支持可信高效的行业协作。曹恒是曾获得过 2010 年美国运筹学和管理学研究协会（INFORMS）颁发的数据分析和人工智能技术领域的“Daniel H. Wagner”大奖。在创立边界智能之前，曹恒在 IBM 研究院工作 16 年，曾担任 IBM 研究院上海研究院院长，是前 IBM 全球研究院大数据分析技术带头人。曹恒拥有卡耐基梅隆大学机器人硕士学位和清华大学自动化控制学士学位。

奚海峰

奚海峰 是高级技术专家和企业家。自 2009 年从美国归国以来，他曾在三家公司担任首席技术官，其中一家是纳斯达克上市公司。他还在上海联合创立过两家初创公司，并在那里担当技术和工程学的领导角色。奚海峰拥有马里兰大学电子与计算机工程硕士学位，以及清华大学自动化控制硕士和学士学位。

Jae Kwon

2005 年，Jae 毕业于康奈尔大学计算机科学学士学位后，曾在硅谷担任软件开发工程师，起先在亚马逊从事 Alexa 开发工作，后来在 Yelp 带领移动应用开发团队。Jae 在认识到区块链问题后，开发了 Tendermint BFT 共识算法和 Tendermint Core 共识引擎。Tendermint 是第一个可信的 PoS 算法。 除了开发 Tendermint 之外，Jae 还是 Cosmos 的创始人之一。

陶曲明

陶曲明自 2016 年 8 月加入万向以来，便负责万向区块链集团的咨询服务，万云 BaaS 平台，以及 ChainBase 加速器和孵化器服务。在加入万向前，他曾在多家

全球领先企业里积累了超过 18 年的丰富服务管理和业务管理的实践经验。Tom 率先将云服务，物联网数据服务平台和加速器技术引入中国市场。自 2013 年起，Tom 一直在跟踪区块链，云计算，物联网和智能制造行业的发展趋势。Tom 拥有复旦大学物理学硕士学位和南开大学电气工程学士学位。

顾问

Jim Yang

Jim Yang 是 Tendermint 首席运营官。 他是 ChatX 移动信息工作室的创始人兼首席执行官。 ChatX 开发了各种移动消息/社交应用程序。 他还共同创立了 Identyx 并一直担任 CEO 至被 Red Hat 收购。 Identyx 开发一个开源企业身份管理软件。

Zaki Manian

Zaki Manian 是可信物联网联盟执行董事，也是区块链和加密货币技术发展的丰富贡献者。 Zaki 在密码学和分布式共识系统方面拥有深厚的专业积累。 他还是 Cosmos 项目以及其创业项目的顾问。

Adrian Brink

Adrian Brink, 是 Tendermint / Cosmos 网络的核心开发者和社区负责人。

Michael Yuan

Dr. Michael Yuan 博士是 CyberMiles Foundation 基金会的负责人。Michael 在德克萨斯大学奥斯汀分校获得了天体物理学博士学位。 他编写的 5 本关于软件开发的书已由 Prentice Hall，Addison-Wesley 和 O'Reilly 出版。 Michael 是 Firefox，Fedora，JBoss 等大型开源项目中的活跃代码提交者。 他是企业应用软件和移动软件方面的专家，也是参与了多个由美国政府资助的研究项目。

参考文献

##

##########################

- 1 Wanxiang Blochchain Inc., Distributed Business Value Research Institute,"Blockchain and Distributed Business Whitepaper", September 2017.
- 2 Ethereum Foundation, "Ethereum Homestead Documentation",http://ethdocs.org/en/latest/

- 3 Jae Kwon, Ethan Buchman,"Cosmos, A Network of DistributedLedgers", https://cosmos.network/whitepaper
- 4 Gavin Wood, "Polkadot: Vision For a Heterogeneous Muilti-chainFramework", https://polkadot.io/
- 5 Tendermint, https://tendermint.readthedocs.io/en/master/
- 6 Ethermint, https://ethermint.zone/
- 7 Oracle International Corporation, "Accountability and Trust inDistributed Ledger Systems", USA Patent Application 20170236120, August17, 2017, http://www.freepatentsonline.com/y2017/0236120.html
- 8 Jan Xie, "CITA Technical Whitepaper",https://github.com/cryptape/cita-whitepaper/blob/master/en/technical-whitepaper.md
- 9 Hyperledger Burrow, https://github.com/hyperledger/burrow
- 10 Joseph Poon, Thaddeus Dryja, "The Bitcoin Lightning Network:Scalable Off-Chain Instant Payments", January 14, 2016,https://lightning.network/lightning-network-paper.pdf
- 11 Joseph Poon, Vitalik Buterin, "Plasma: Scalable Autonomous SmartContracts", August 11, 2017, https://www.plasma.io/plasma.pdf
- 12 Ethan Frey, "Cosmos IBC Specification", Sep. 29, 2017,https://github.com/cosmos/ibc/blob/master/README.md
- 13 Thomas Erl, "SOA: Principles of Service Design", Prentice Hall;1st edition (July 28, 2007)
- 14 Dean, J., Corrado, G.S., Monga, R., et al, Ng, A. Y. "Large ScaleDistributed Deep Networks". In Proceedings of the Neural InformationProcessing Systems (NIPS'12) (Lake Tahoe, Nevada, United States,December 3--6, 2012). Curran Associates, Inc, 57 Morehouse Lane, RedHook, NY, 2013, 1223-1232.
- 15 Tendermint Blog, "Cosmos Validator Economics -- Revenue Streams",January 2018, https://medium.com/@tendermint/b5b2c682a292
- 16 Sunny Aggarwal, "Cosmos Token Model", December 2017,https://drive.google.com/file/d/1jtyYtx7t1xy9gxEi2T5lXFNd8xUY7bhJ/view