# 摘要

区块链的普及为开发人员提供了多种平台，可以根据应用程序特性和吞吐量、安全性和成本要求在其上运行智能合约。 然而，这种自由的后果是严重的分裂； 每条链都是孤立的，迫使用户孤立他们的流动性并限制在生态系统围墙之间转移流动性和状态。 本文介绍了第一个无需信任的全链互操作性协议 LayerZero，它提供了一个强大的低级通信原语，可以在此基础上构建各种跨链应用程序。 使用这个新原语，开发人员可以实现无缝的链间应用程序，如跨链 DEX 或多链收益聚合器，而无需依赖受信任的托管人或中间交易。 **简而言之，LayerZero 是第一个无需信任地实现跨所有链的直接交易的系统。** 允许交易在链之间自由流动为用户提供了整合分散的流动性的机会，同时也充分利用了不同链上的应用程序。 通过 LayerZero，我们提供了未来完全连接的 omnichain 生态系统的网络结构。

# 介绍

区块链概念的核心是去中心化、透明，不变性三大支柱。 没有单一实体控制区块链，区块链上的行为是可验证且不可逆的。 这些支柱创造了一个实体可以在不信任任何其他实体的情况下采取行动的基础。 这种信任保证是加密货币比法定货币更具吸引力的原因之一。

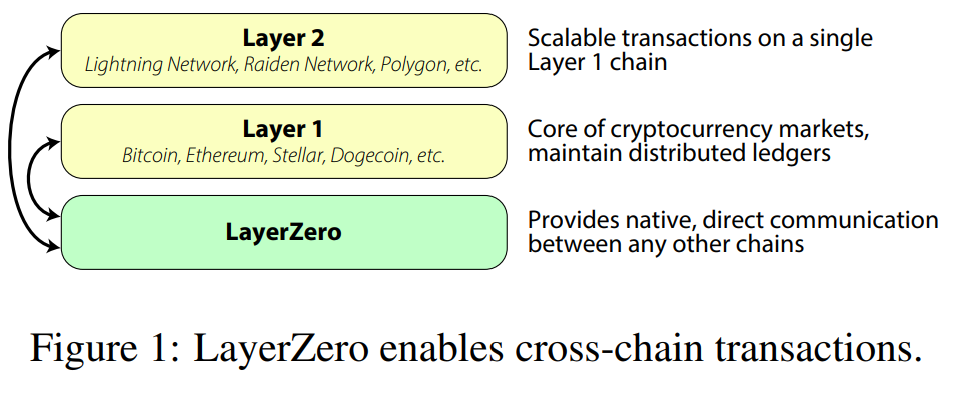
如果所有用户和所有应用程序共存于一个统一的区块链中，那么本文将到此结束。 然而，区块链的实用性导致了各种应用程序的激增，具有独特的复杂性和要求。 对多样化功能的需求刺激了专业链的发展。 这些链中的每一个都在其自己的生态系统中促进了应用程序的巨大增长，但这些生态系统之间的隔离已成为应用的关键限制。 用户和开发人员被迫在不同的链之间分配时间、资源和流动性。 所谓的第 1 层区块链数量庞大（在撰写本文 [1] 时多达 109 个）的自然结果是需要扩展上述三个支柱以同时包含跨多个链的交互。 链间按需交互的一个例子是代币的转移，我们将在本节后面讨论。

用区块链的说法，工作单元是一个交易，不可变且不可撤销。 整理成块的交易构成了区块链系统的安全基础。 但是，交易一直是单链的概念； 如下所述，跨链交互传统上需要在正常区块链密码系统之外的第三方机制。 相比之下，本文描述了第一个可以实现本地跨链交易的消息传递协议：LayerZero。

为了说明 LayerZero 提供的强大通信原语，我们看一下将代币从一条链转移到另一条链的示例。 目前，要在两条链的代币之间进行转换，用户必须利用中心化交易所或跨链去中心化交易所 (DEX)（也称为跨链桥），这两者都需要妥协。 在中心化交易所的情况下，例如 Binance.com [3]，用户必须信任跟踪存款和取款的交易所。 这种信任关系与区块链共识的基本无信任性背道而驰，缺乏链上自动化系统的安全性。 使用诸如 AnySwap [2] 或 THORChain [23] 之类的 DEX，通过在链上进行传输来缓解信任问题，但现有的 DEX 实现涉及将用户token转换为特定于协议的token，该token将其中间共识层传输到 达成交易共识。 这个中间共识层虽然通常以安全的方式实现，但确实需要用户信任侧链以促进token传输。 正如我们在本文中展示的那样，这种额外的开销是不必要的。 尽管用户需求量很大，但还没有出现既高效、直接又同时仍然保留使用区块链的核心原因的解决方案：免信任。 退一步说，LayerZero 的直接跨链交易为开发人员提供了构建它的工具。

重要的是要注意，LayerZero 和上述交易所在实现堆栈的两个不同级别上运行。 LayerZero 是一种支持各种全链应用程序的通信原语，而交易所是一个应用程序示例，可以从 LayerZero 之上的实现中重新获益。 第 2 节概述了区块链技术前景并进一步探讨了交换示例。

为了正确解释 LayerZero 的功能及其在区块链生态系统中的作用，我们首先提出了实现链间交易所需的基本通信原语的形式化，我们称之为有效交付（第 3 节）。 然后，我们将描述 LayerZero 如何以无需信任的方式提供此原语，从而保持区块链的安全承诺。 LayerZero 是第一个无需信任的全链互操作层，支持第 1 层和第 2 层链之间的直接消息传递（图 1）。



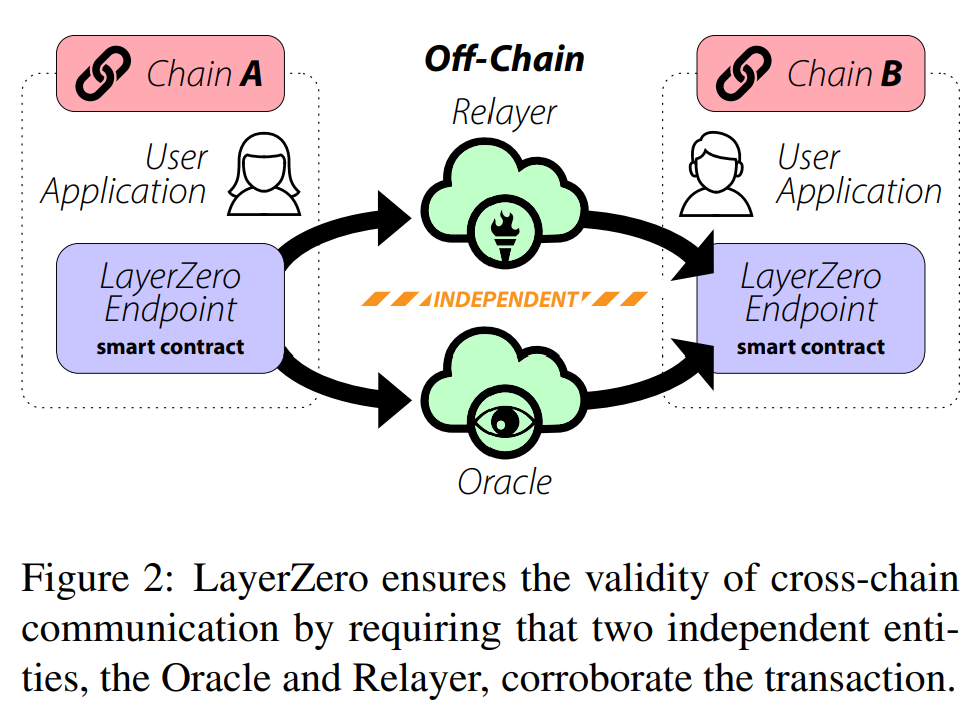
A 链和 B 链之间的跨链交易由 A 上的交易 tA、A 和 B 之间的通信协议和消息 m 组成。 有效交付表明当且仅当 tA 已提交且有效时才交付 m。 支持 LayerZero 的关键思想是，如果两个独立的实体证实交易的有效性（在本例中为 tA），那么链 B 可以确定 tA 是有效的。 图 2 在较高级别说明了这一点。 给定两个不串通的实体，如果

(1) 一个实体可以为链 A 上包含 tA 的区块生成区块头，

(2) 另一个实体可以在该区块上独立生成 tA 的证明（交易证明），并且

(3) 标头和交易证明实际上一致。

然后通信协议可以将 m 交付给链 B 上的客户端，同时保证 tA 在链 A 上稳定提交。第 4 节中描述的 LayerZero 通信协议保证 接收链上的交易将与发送链上有效的、已提交的交易配对，而不涉及任何中间链。 我们通过组合两个独立的实体来实现这一点：一个提供块头的 Oracle [7]，以及一个提供与上述交易相关的证明的中继器。



LayerZero 的接口是一个轻量级的链上客户端，我们称之为 LayerZero 端点。 每个（支持的）链上都存在一个 LayerZero 端点，任何具有 LayerZero 端点的链都可以进行涉及任何其他具有 LayerZero 端点的链的跨链交易。 从本质上讲，这创建了一个完全连接的网络，其中每个节点都直接连接到每个其他节点。 使用最小样板代码，支持任何区块链。 第 5 节通过在以太坊区块链上实施 LayerZero 的案例研究展示了这个过程。

直接与网络上的任何其他链执行跨链交易的能力为以前无法实现的一类大规模应用打开了机会，例如跨链去中心化交易所、多链收益聚合器和跨链 贷款。 第 6 节详细检查了几个这样的应用程序。 通过 LayerZero，用户可以在链间自由流动流动性，允许单一流动性池参与跨不同链和生态系统的多个去中心化金融（DeFi）应用程序，而无需通过第三方系统或中间代币。

# 背景

为了为 LayerZero 奠定基础，我们回顾了相关的现有系统，以说明它们为何不能满足新兴应用程序的需求。 讨论最终深入解释了在 LayerZero 上构建跨链交换的优势

## 2.1 相关工作

本节旨在了解跨链交互领域的重要参与者，了解他们为何未能实现无信任有效交付的理想，以及 LayerZero 如何弥补这一差距。

以太坊 [8] 是最受欢迎的通过智能合约构建的去中心化金融应用程序平台。 以太坊使用图灵完备的编程语言扩展其底层区块链，使去中心化应用程序库能够通过对开发人员友好的抽象来利用底层链的强大安全属性。 然而，底层区块链的低交易率，大约每秒 15-45 笔交易 [9]，已被证明是一个严重的可扩展性瓶颈，限制了直接在以太坊区块链上运行的应用程序的普及。 由于其编程模型和流行度，许多链间通信技术都围绕第三方链与以太坊的接口展开。 LayerZero 提供了在没有中间人的情况下直接将状态传入和传出以太坊的能力，允许用户和应用程序利用以太坊链的稳定性和可信度，而不会出现下述解决方案的成本和瓶颈。

以太坊 2.0 [22] 是一组升级提议，以解决以太坊在可扩展性、安全性和可持续性方面的缺点。 以太坊 2.0 引入了分散负载的分片链，而不是将所有交易集中在超载的以太坊主链上。 从工作量证明过渡到权益证明既消除了 51% 攻击的可能性，又减少了每笔交易的能量。 这些进步在很大程度上与 LayerZero 是正交的，除了它们肯定会提高以太坊的流行度，从而产生对方便和廉价的链间通信的更多需求。

Polygon [17]，前身为 Matic Network，是一个第 2 层网络，可解决以太坊的吞吐量和主权挑战。 尽管是最受欢迎的区块链开发平台，以太坊仍受到低吞吐量 [10] 的困扰，使其不适用于某些应用程序。 Polygon 提供特定于应用程序的、与以太坊兼容的侧链，将独立链的可扩展性和独立性与以太坊的社区和安全性相结合。 专用或吞吐量密集型应用程序在侧链上运行，并定期整合回以太坊主链。 相比之下，LayerZero 是一个较低级别的平台，可实现直接的链间通信，并可用于促进传输回以太坊链，而无需 Polygon 协议的复杂性。

Polkadot [26] 是早期开放式跨链生态系统的示例。 在 Polkadot 中，许多特定领域的平行链通过公共中继链连接，使代币和数据能够在它们之间流动。 然而，链间通信总是跨越这条中继链，从而产生额外的成本。 LayerZero 提供与 Polkadot 相同的底层通信平台，不涉及链上中间人所需的额外交易。

THORChain [23] 是一种 DEX，它使用成对的流动性池在第三方链之间转移代币。 每个流动资金池都将一种特定的第三方货币绑定到一种名为 RUNE 的 THORChain 原生代币，后者充当一种通用的交换媒介。 如果没有这种通用媒介，所有货币对都需要一个流动性池，这意味着池的数量将与货币数量的平方成正比。 不幸的是，虽然 RUNE 解决了这个可扩展性问题，但它是交易过程中的繁琐开销，使一个简单的操作变得相当复杂。 这在 THORChain 交易算法的复杂性中显而易见。 LayerZero 提供直接的链间通信，没有 THORChain 固有的可扩展性瓶颈、繁琐的中间货币或重量级协议。

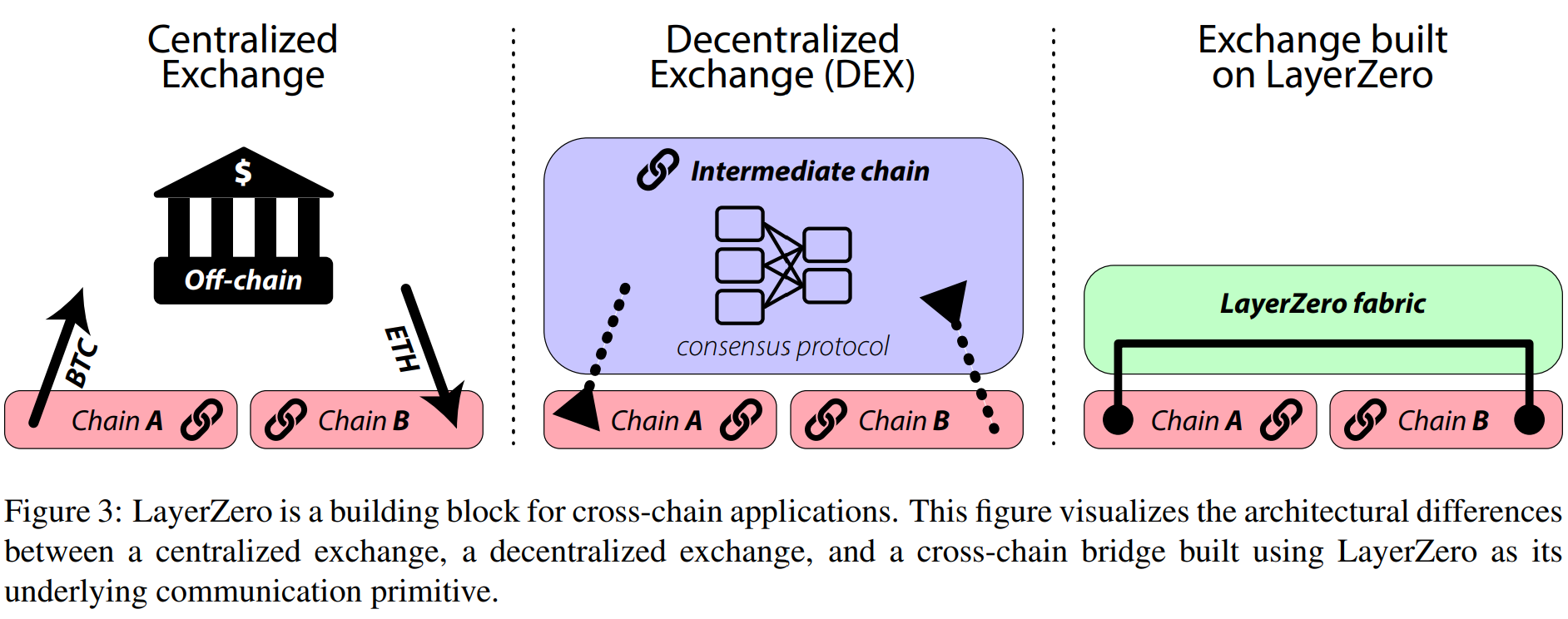
AnySwap [2] 是一种 DEX，适用于简单的成对代币交换，类似于 THORChain。 AnySwap 依赖于中间令牌 ANY，它基于 Fusion 分布式控制权限管理 [6]。 与 THORChain 一样，使用 ANY 中间代币会带来不必要的开销、延迟和额外的转账费用。

Cosmos [5] 是一种区块链网络技术，允许在支持的链之间发送任意消息。 Cosmos 包括基于 Tendermint BFT [21] 的区块链间通信（IBC [14]）协议，以促进在 Cosmos Hub 上构建的链之间的消息传递。 Cosmos 在两个关键方面与 LayerZero 不同：（1）IBC 运行一个完整的链上轻节点，以及（2）IBC 仅提供 fastfinality [24] 链之间的直接通信。 IBC 的这些局限性，加上它使用中间链来促进共识，使其类似于 Anyswap、THORChain 或 Polkadot，而不是像 LayerZero 这样的通用通信层。 Cosmos 还提供了一个与 Anyswap 或 THORChain 具有相似属性的 DEX，称为 Gravity Bridge [12]。 与 Cosmos 和 IBC 相比，LayerZero 提供无需信任的全链消息传递，并且可以扩展以在任何链上运行，包括那些提供概率最终性的链，例如以太坊和比特币。

Chainlink [7, 4] 是一个用于构建和连接去中心化预言机网络 (DON) 的框架。 虽然智能合约是防篡改的，但它们的链上性质阻碍了对其更广泛采用至关重要的基本连接：智能合约无法获取执行合约所需的链下数据，例如股票价格、物联网设备测量和 来自安全链下计算的输出。 DON 将智能合约的防篡改属性扩展到合约所依赖的数据源和外部资源，而无需信任任何中央实体。 在 DON 中，用户的智能合约向 Chainlink 接口智能合约发出链上请求，该智能合约将事件发布到许多独立的预言机节点。 每个 oracle 节点查询多个数据源以获取请求的信息，将其聚合以过滤错误或恶意来源，并可选择对数据执行信任最小化计算。 预言机节点响应 Chainlink 接口合约，执行第二级聚合以过滤错误或恶意的预言机。 这种双层过滤保证了对最终数据的信任，而无需信任任何单独的预言机或数据源。 因此，Chainlink 提供了一个强大的信息检索网络以及一个已在整个行业广泛使用的安全链下计算解决方案。 通过利用 Chainlink DON 框架，LayerZero 协议获得了确保在不同链之间可靠地传递消息的能力。

## 2.2 LayerZero 实践

开发人员可以使用 LayerZero 构建复杂的跨链应用程序，而无需牺牲免信任或引入复杂的中间链/智能合约。 图 3 说明了 LayerZero 在构建交换环境中的功能。



左侧所示的集中式交易所要求用户将代币存入中央可信机构，然后该机构会在链下跟踪该存款，并在用户请求时在其他链上授予代币。 信任这个权威违背了使用区块链的初衷，这导致了分布式交易所的出现。

中心图在较高层次上显示了典型的去中心化交易所是如何工作的——通过使用智能合约管理的共识协议来促进链 B 上token的自动铸造，去中心化交易所能够克服对中心化、可信链下的必要性 中间人。 然而，一个关键的限制是 DEX 涉及中间代币和中间链，并且只在链 B 上铸造中间代币或包装代币，而不是用户想要的实际代币。 然后，用户必须在额外的交易中用中间代币（例如 RUNE）或包装代币（例如 ANY）交换所需的代币。 这种中介/包装代币、第二笔交易以及中间链对于理想情况下应该是单一无缝交易的事物来说都是不必要的开销。 图 3 的右侧显示了建立在 LayerZero 上的交易所的样子，链 A 能够启动单个跨链交易，促进链 A 上的本地交易，并通知链 B 上的应用程序他们可以安全地授予一个 token给用户。 在此应用程序中，LayerZero 支持干净且最小的单笔交易交换，不包含任何中间代币。 实际的交换协议由跨链交易两侧的智能合约处理，LayerZero 在两者之间传递消息。 这提供了很大的灵活性，并遵循端到端原则 [18]，大部分高级交换逻辑由源链和目标链上的智能合约处理。

## 3 有效交付

在本节中，我们描述了去信任链间通信的基本属性。 为了正式描述在不同链上验证交易的问题，我们定义了有效交付的概念。 有效交付是一种通信原语，它通过提供以下保证来实现跨链令牌传输：

1. 通过网络发送的每条消息 m 都与发送方链上的事务 t 耦合。

2. 当且仅当关联交易 t 有效且已在发送方侧链上提交时，消息 m 才会传递给接收方。

中心化交易所保证有效交付，因为客户和交易所之间的协议是客户将他们的代币从一条链转移到交易所，交易所将在收到该代币后发行一些余额（非加密货币） . 然后可以从任何可用的链中提取这种非加密货币余额，这种便利可以通过交易所在每个支持的链上维护的广泛流动性池来实现。 交易所在此交易中充当中间人，用户必须相信他们会遵守交易的规定。 然而，恶意或受损的交易所可能会从客户端获取代币，发放余额，然后拒绝允许从另一个链中提取该余额，从而有效地从用户那里窃取代币。 即使用户愿意信任交易所，近年来也有许多成功的黑客攻击或破坏加密货币交易所的尝试 [15]，因此用户可以更好地使用不需要任何可信中间人的解决方案。 在更高的层面上，加密货币的核心租户之一是它们独立于银行等中心化实体，因此依赖中心化交易所违背了它们的目的。

使用中心化交易所的替代方案是去中心化交易所，例如 THORChain [23] 或 AnySwap [2]。 所有现有的 DEX 都使用中间令牌，例如 THORChain 中的 RUNE 或 AnySwap 中的 ANY。 由于这些中间令牌由每个 DEX 各自的协议管理，因此 DEX 能够保证有效交付，因为恶意用户不可能伪造中间令牌。 现有的 DEX 解决方案并不理想，因为它们涉及两个中间交易——一个将发送方的代币转换为中间代币，另一个将中间代币转换为接收链上所需的“真实”代币。 除此之外，用户有必要完全信任在源链上确认交易并将铸造令牌的意图传达给目标链的中间共识层。 虽然现有交易所确实支持跨链代币转移，但它们是以不必要的复杂性和成本为代价的。 这样做的缺点很明显，因为跨链应用程序尚未得到广泛采用。 链间交易问题的理想解决方案是在链之间使用单个 oneswap 交易，而不涉及任何受信任的中间实体——换句话说，无需信任的有效交付。 在我们的工作中，我们实现了一个通用消息传递协议，该协议提供任意用户数据的无信任有效传递，而不仅仅是令牌。 分布式交易所或其他 DeFi 应用程序将使用我们的消息传递原语来实现以提供跨链交易，而低级消息传递协议提供的灵活性程度使更高级别的应用程序能够实现以前不可能实现的范围广泛的功能。

## 设计

LayerZero 的核心是一种提供无信任有效交付的通信协议。 我们的协议建立在第 4.1 节中介绍的一系列组件之上。 我们在第 4.2 节讨论传输协议的通信流程，在第 4.3 节描述 LayerZero 如何在不涉及可信中介服务的情况下实现有效交付，并在第 4.4 节中展示我们针对基于低成本智能合约的轻客户端端点的新颖设计。

### ·4.1 LayerZero 模块

LayerZero 端点是 LayerZero 面向用户的界面。 LayerZero 网络中的每条链都有一个 LayerZero 端点，其实现为一系列智能合约。 端点的目的是允许用户使用 LayerZero 协议后端发送消息，以保证有效传递。

LayerZero 端点分为四个模块：Communicator、Validator、Network 和 Libraries。 Communicator、Validator 和 Network 模块构成端点的核心功能（图 4），而 LayerZero 支持的每个新链都作为附加库添加。 这种设计允许我们在不修改三个核心模块的情况下添加对新链的支持。 我们在 4.4 节中解释了每个模块的功能。

Oracle 是一种第三方服务，它提供一种机制，独立于其他 LayerZero 组件，从一条链读取区块头并将其发送到另一条链。 理论上，这个 Oracle 可以是任何提供这种机制的第三方服务，但在实践中我们希望使用 Chainlink [7, 4]，它是目前去中心化 oracle 网络的行业领导者。

Relayer 是一种链下服务，在功能上类似于 Oracle，但它不是获取块头，而是获取指定交易的证明。

为确保有效传递，唯一的要求是对于使用 LayerZero 协议发送的任何给定消息，Oracle 和 Relayer 必须相互独立。 该协议本身不需要任何特定的 Relayer 实现，理论上 LayerZero 的用户甚至可以实现自己的 Relayer 服务。 这种设计允许用户确保 Relayer 不会与 Oracle 串通，这种独立性使我们能够实现无信任的验证交付，如第 4.3 节所示。 实际上，LayerZero 提供 Relayer 服务，而 Oracle 则由 Chainlink 的去中心化预言机网络和相关共识机制处理。

### 4.2 LayerZero 协议

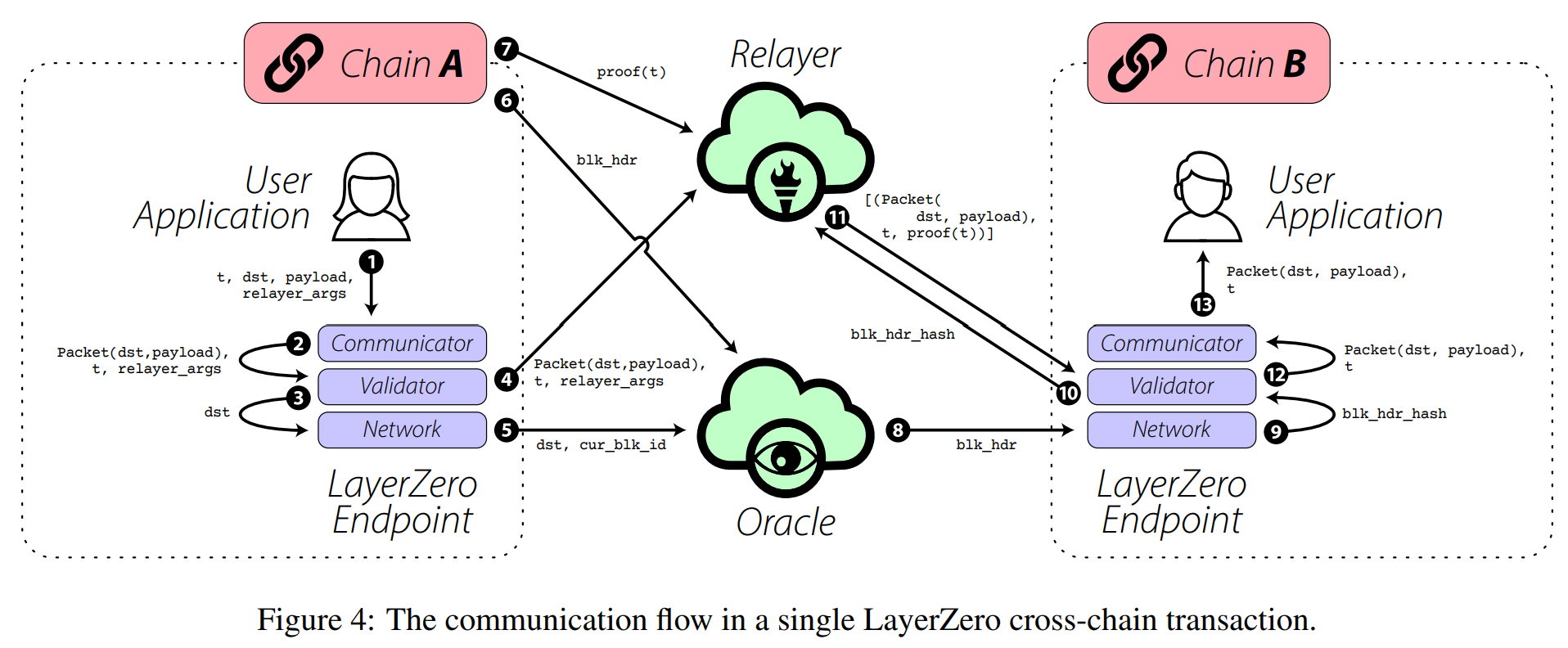


图 4 说明了有效传递单个 LayerZero 消息所涉及的步骤。 图中每个带圈的数字代表协议的一个步骤，对应本节的一个段落。 本节将介绍链 A 上的用户应用程序通过 LayerZero 向链 B 上的用户应用程序发送单个消息的示例。 在第 5 节中，我们描述了在两个以太坊虚拟机之间发送消息的情况下如何实现各种组件和协议步骤。

第 1 步：链 A 上的用户应用程序（App A）执行一系列操作作为交易 T 的一部分。我们通过交易标识符 t 唯一标识交易 T——此标识符的格式可能因链 A 的类型而异。 事务 T 中包含的一个步骤是通过 LayerZero 传输消息，并以 T 为条件进行有效传递。出于说明目的，并且不失一般性，我们假设在这种情况下应用程序 A 使用我们的参考中继器。 App A 向 LayerZero Communicator 发送请求，其中包含以下信息：

* t：T 的唯一交易标识符。
* dst：指向链 B 上智能合约的全局标识符。
* payload：App A 希望发送给 App B 的任何数据。
* 中继器参数：在 App A 希望使用参考中继器的情况下描述支付信息的参数。

第 2 步：Communicator 构造一个包含 dst 和 payload 的 LayerZero 数据包，称为 Packet(dst, payload)，并将其与 t 和 relayer args 一起发送给 Validator。

第 3 步：Validator 将 t 和 dst 发送到 Network。 这一步通知Network，链A上当前区块的区块头需要发送到链B。

Step 4: Validator将Packet(dst, payload), t, relayer args转发给Relayer，通知Relayer该笔交易 T 的证明需要预取并最终发送到链 B。这与第 3 步同时发生。

第 5 步：网络将 dst 和当前交易的块 ID（cur blk id）发送到 Oracle。 这通知 Oracle 获取链 A 上当前区块的区块头并将其发送到链 B。如果同一区块中发生多个 LayerZero 交易，则步骤 5 仅执行一次。

第六步：Oracle从链A读取区块头（blk hdr）。

第七步：Relayer从链A读取与交易T关联的交易证明（proof(t)）（可能会在链下存储）。 步骤 6 和 7 彼此异步发生。

第 8 步：Oracle 确认 blk hdr 对应的块在链 A 上稳定提交，然后将 blk hdr 发送到链 B 上的网络。确定何时发生这种情况的机制因链而异，但通常会涉及等待一定数量的 块确认。

第 9 步：网络将指定为 blk hdr 哈希的块哈希发送给验证器。

步骤 10：Validator 将 blk hdr hash 转发给 Relayer。

步骤 11：在收到 blk hdr 哈希后，Relayer 发送与当前块匹配的任何 Packet(dst, payload), t, proof(t) 元组的列表。 如果多个用户同时在相同端点之间发送消息，则同一块内可能有多个数据包和相关联的交易证明。

第 12 步：Validator 使用接收到的交易证明结合网络存储的区块头来验证关联交易 T 是否有效并已提交。 如果区块头和交易证明不匹配，则消息将被丢弃。 如果它们匹配，则将 Packet(dst, payload) 发送到 Communicator。

第 13 步：Communicator 向 App B 发送 Packet(dst, payload)。

### 4.3 实现无信任有效交付

**无需信任**：LayerZero 设计的关键在于用户无需信任 LayerZero 的组件。 不需要信任，这是一个强条件，我们只需要 Oracle 和 Relayer 之间的独立性这个较弱的条件。 这种独立而不是信任的要求是使 LayerZero 高效和轻量级的一个方面。 只要 Oracle 和 Relayer 之间没有恶意串通，LayerZero 就可以保证有效交付。

**有效传递**：通过第 4.2 节所示的 LayerZero 协议，当且仅当与 m 关联的交易 t 的交易证明可以在步骤 12 中得到验证时，Communicator 将消息 m 传递给用户应用程序。此验证步骤当且仅当区块头和交易证明匹配时成功，只会在以下两种情况下发生：

1. Oracle 提供的区块头和 Relayer 提供的交易证明都有效。

2. Oracle提供的区块头和Relayer提供的交易证明均无效，但仍然匹配。

场景 2 只有在 Oracle 和 Relayer 串通的情况下才会发生，因为在统计上不可能在不知道特定块头的情况下发送可以针对块头进行验证的交易证明，反之亦然。 但是，LayerZero 的设计消除了串通的可能性，如第 1 节所述。因此，如果消息被传递到接收方的用户应用程序，则可以保证满足有效传递的属性。

如第 3 节所述，能够保证无信任有效交付的通信协议，即无需信任中间实体或代币的有效交付，是实现跨链交易的理想解决方案。 LayerZero 是第一个也是唯一一个证明此属性的系统。 这一事实将推动用户采用 LayerZero 作为跨链消息传递的首选方法

### 4.4 LayerZero 端点

LayerZero 端点目前作为 LayerZero 网络中包含的每条链上的一系列智能合约实施。 LayerZero 端点的核心功能封装在三个模块中：通信、验证和网络。 这些模块以类似于网络堆栈的方式运行，消息在发送方的堆栈中向下发送——通信器到验证器再到网络——然后在接收方的堆栈中向上发送。

除了核心模块之外，LayerZero Endpoint 还可以通过库进行扩展，这些库是辅助智能合约，定义了如何处理特定链的通信。 LayerZero 网络中的每条链都有一个关联的库，每个端点都包含每个库的副本。 这种模块化设计使 LayerZero 网络可以快速轻松地扩展，以包含按需添加的新链。 此外，两条链之间的通信只需要它们各自的库出现在两端，使 LayerZero 成为一个完全连接的网络，能够在任何一对节点之间编排交易

### 4.5 LayerZero 端点成本可扩展性

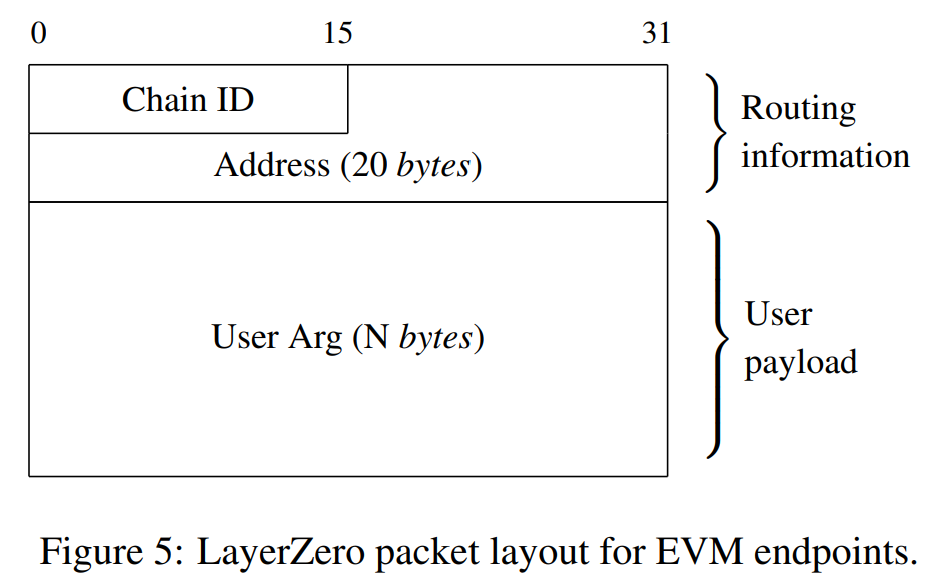
正如许多读者可能会指出的那样，在第 1 层链上运行智能合约可能成本过高，尤其是随着存储数据量的增加。 为了使 LayerZero Endpoint 实用，我们有必要设计尽可能轻量级的客户端。 以前通过交叉链状态机复制 (SMR) 进行无信任跨链验证的工作，例如 Golden Gate [11]，每天可能要花费数百万美元才能在以太坊等流行的第 1 层链上运行。

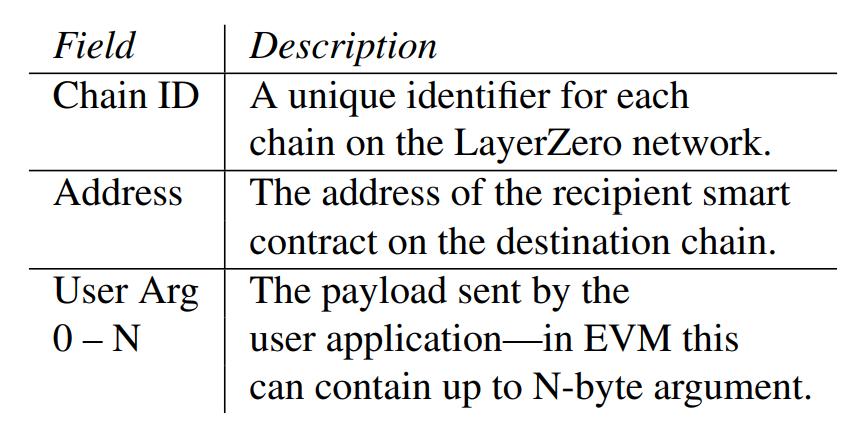
为了解决这个问题，我们着手设计尽可能轻量级的客户端。 我们的主要观察是，没有必要在客户端复制和存储块头。 相反，我们将获取必要的跨链标头和交易证明的任务委托给链下实体：Oracle 和 Relayer。 这导致 LayerZero 端点非常轻巧，即使在像以太坊这样以昂贵着称的 [20] 链上，它们也具有成本效益。

## 5 案例研究：EVM 上的 LayerZero

在本节中，我们简要描述了我们如何实现对在以太坊虚拟机 (EVM) [13] 上运行 LayerZero 的支持的细节。 为简洁起见，我们关注系统的各个方面，其实现可能因链而异，并强调我们的实现如何处理以太坊链的特定要求。 如4.1节所述，当前版本的LayerZero依赖Chainlink提供Oracle服务，希望用户使用我们提供的Relayer服务。

LayerZero 数据包：LayerZero 数据包的格式将根据源链和目标链而有所不同。 我们在图 5 中说明了 EVM 端点 [19] 的 LayerZero 数据包的精确布局。每个字段的功能如下：





**发送方链交易稳定性**：为了确保消息交易在源链上稳定，我们依赖于去中心化预言机网络的固有特性——预言机只会在听到一定数量的消息后通知特定区块头的目的链。 块确认，在以太坊的情况下是 15。准确地说，LayerZero 协议的第 8 步（第 4.2 节）只会在 Oracle 听到链 A 上的 15 个块确认后执行。

**LayerZero 端点**：我们将 LayerZero 端点实现为一系列智能合约，由我们在第 4.4 节中描述的四个主要模块组成。 对于大多数现有的区块链，包括以太坊区块链，我们能够将通信器、验证器和网络分别作为单独的智能合约来实现。 但是，这种设计并不排除在具有不同要求的（未来）链上实施 LayerZero Endpoint。

LayerZero 端点的库组件是本案例研究中为以太坊区块链提供支持的关键。 我们实现了一个库来处理图 5 中所示的 EVM 特定 LayerZero 数据包的构造，并处理 EVM 智能合约地址信息的编码和解码。

库的另一项职责是处理验证交易证明所涉及的实际计算。 我们的 EVM 库处理 EVM 块上交易的 Merkle-Patricia 树验证 [16]，我们基于 Golden Gate [11] 的开源实现。

## 6 LayerZero上的应用

**跨链去中心化交易所**：如第 2.2 节所述，LayerZero 支持专门处理原生资产的跨链 DEX（跨链桥）。 与发行包装代币或通过中间侧链的现有 DEX 设计相反，可以构建使用 LayerZero 在链间发送消息的 DEX，这样两条链上都存在流动性池，用户可以简单地将其本机资产存入一个池中并提取 来自另一个的本地资产。 LayerZero 的消息传递原语足够强大，可以实现直接桥接（1:1 定价）、自动做市（ab = k 定价）和任何其他衍生（例如类似于 Curve DAO 定价 [25]）。 LayerZero 提供的有效交付保证使广泛的去中心化交易所应用成为可能。

**多链收益聚合器**：当前的收益聚合器通常在单链生态系统的范围内运行，Yearn Finance [27] 等项目使用单链策略实现收益聚合。 这些单链收益聚合系统的一个主要弱点是它们无法利用当前生态系统之外的任何收益机会，可能会错失许多最佳收益。 使用 LayerZero 进行跨链交易的收益聚合器将允许制定战略，利用所有生态系统中的最佳机会，增加获得高收益机会的机会，并使用户能够利用市场低效率。 多链收益聚合器将严格优于单链收益聚合器，因为在最坏的情况下，该策略将退化为仅利用一条链上的机会，而在最好的情况下，它将有更多的机会可供选择。

**多链借贷**：今天，用户没有简单的方法来利用他们不持有资产的链上的机会。 例如，假设资产整合在 ETH 中的用户想要利用 Polygon [17] 上的机会。 他们的选择是 (1) 将他们的整个资产基础转移到另一个链并将其转换为所需的货币，或者 (2) 在以太坊上借出他们的资产，借入所需的资产，然后将该资产桥接到目标链。 LayerZero 支持一种借贷协议，该协议允许用户将其整个资产基础保留在以太坊上，借出，然后直接在 Polygon 上的 MATIC 中借入。 这消除了中介成本，例如过桥费和掉期费。

这三个示例仅代表 LayerZero 实现的众多可能性中的一小部分。 通过利用 LayerZero，开发人员将能够编写他们的应用程序而不必担心链间和链内交易之间的不同语义，并且用户将能够自由地跨链移动流动性。 鉴于无信任跨链交易的力量，我们期待着社区将开发出创造性的新应用。

## 7 结论

本文介绍了第一个不涉及任何中间交易的去信任全链互操作平台 LayerZero 的设计和实现。 我们表明，通过利用两个独立的、不受信任的链下实体，即 Oracle 和 Relayer，LayerZero 能够实现有效交付，而无需昂贵的跨链状态机复制或中间代币。 我们的协议以不排除使用任意中继服务的方式设计，这确保了中继和 Oracle 之间没有勾结。 LayerZero 协议支持支持链之间的本地交易，而新颖的 LayerZero 端点设计可以轻松扩展以支持任何链。 除此之外，我们的端点设计足够轻便，可以在以太坊等昂贵的第 1 层链上运行，而不会产生高昂的成本。 我们展示了一个案例研究，说明如何在 LayerZero 中实现对基于 EVM 的链的支持，使用参考 Relayer 实现与 Chainlink 的去中心化预言机网络相结合，以通过 LayerZero 实现跨链交易。

LayerZero 是连接各种不相交的区块链生态系统并允许链和社区之间流动性、数据和想法的无摩擦移动的主干。

**References**

[1] All layer 1 blockchain protocols. https://blockchaincomparison.com/blockchain-protocols/. Accessed: 2021-5-13.

[2] Anyswap dex user guide. https://anyswapfaq.readthedocs.io/en/latest/index.html. Accessed: 2021-5-13.

[3] Binance.com. https://www.binance.com/. Accessed: 2021- 5-14.

[4] BREIDENBACH, L., CACHIN, C., CHAN, B., COVENTRY, A., ELLIS, S., JUELS, A., KOUSHANFAR, F., MILLER, A., MAGAURAN, B., MOROZ, D., NAZAROV, S., TOPLICEANU, A., TRAMER` , F., AND ZHANG, F. Chainlink 2.0: Next steps in the evolution of decentralized oracle networks. White paper, ChainLink, 2021.

[5] What is cosmos? https://v1.cosmos.network/intro. Accessed: 2021-5-15.

[6] Dcrm - fusion.org. https://www.fusion.org/tech/dcrm. Accessed: 2021-5-13.

[7] ELLIS, S., JUELS, A., AND NAZAROV, S. Chainlink: A decentralized oracle network. White paper, ChainLink, 2017.

[8] Ethereum. https://ethereum.org/en/. Accessed: 2021-5- 13.

[9] Ethereum 2.0 (eth2) vision. https://ethereum.org/en/ eth2/vision/. Accessed: 2021-5-13.

[10] GALAL, H. S., ELSHEIKH, M., AND YOUSSEF, A. M. An efficient micropayment channel on ethereum. In Data Privacy Management, Cryptocurrencies and Blockchain Technology. Springer, 2019, pp. 211–218.

[11] Golden gate – trustless-bridging ethereum (evm) blockchains – part 1: Basics. https://loredanacirstea.medium.com/ golden-gate-trustless-bridging-ethereum-evmblockchains-part-1-basics-d016300ea0dd. Accessed: 2021-5-14.

[12] Announcing the gravity bridge. https://blog.althea.net/ gravity-bridge/. Accessed: 2021-5-15.

[13] HILDENBRANDT, E., SAXENA, M., RODRIGUES, N., ZHU, X., DAIAN, P., GUTH, D., MOORE, B., PARK, D., ZHANG, Y., STEFANESCU, A., ET AL. Kevm: A complete formal semantics of the ethereum virtual machine. In 2018 IEEE 31st Computer Security Foundations Symposium (CSF) (2018), IEEE, pp. 204– 217.

[14] Ibc overview — cosmos sdk. https://docs.cosmos.network/ master/ibc/overview.html. Accessed: 2021-5-15.

[15] LAZARENKO, A., AND AVDOSHIN, S. Financial risks of the blockchain industry: A survey of cyberattacks. In Proceedings of the Future Technologies Conference (2018), Springer, pp. 368– 384.

[16] LU, Z., WANG, Q., QU, G., ZHANG, H., AND LIU, Z. A blockchain-based privacy-preserving authentication scheme for vanets. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 27, 12 (2019), 2792–2801.

[17] Polygon: Ethereum’s internet of blockchains. https:// polygon.technology/lightpaper-polygon.pdf.

[18] SALTZER, J. H., REED, D. P., AND CLARK, D. D. End-to-end arguments in system design. ACM Transactions on Computer Systems (TOCS) 2, 4 (1984), 277–288.

[19] Solidity types. https://docs.soliditylang.org/en/ v0.5.3/types.html. Accessed: 2021-5-14.

[20] SPAIN, M., FOLEY, S., AND GRAMOLI, V. The Impact of Ethereum Throughput and Fees on Transaction Latency During ICOs. In International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2019) (Dagstuhl, Germany, 2020), V. Danos, M. Herlihy, M. Potop-Butucaru, J. Prat, and S. Tucci-Piergiovanni, Eds., vol. 71 of OpenAccess Series in Informatics (OASIcs), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 9:1–9:15.

[21] Tendermint. https://tendermint.com/. Accessed: 2021-5- 15.

[22] The eth2 upgrades. https://ethereum.org/en/eth2/. Accessed: 2021-5-13.

[23] Thorchain. https://thorchain.org. Accessed: 2021-5-13.

[24] VIRIYASITAVAT, W., DA XU, L., BI, Z., AND SAPSOMBOON, A. New blockchain-based architecture for service interoperations in internet of things. IEEE Transactions on Computational Social Systems 6, 4 (2019), 739–748.

[25] WARREN, W., AND BANDEALI, A. 0x: An open protocol for decentralized exchange on the ethereum blockchain. URl: https://github. com/0xProject/whitepaper (2017), 04–18.

[26] WOOD, G. Polkadot: Vision for a heterogeneous multi-chain framework. White paper, Polkadot, 2016. [27] yearn.finance. https://yearn.finance/. Accessed: 2021-5- 14.