



**NERVOS**

# Nervos CKB

## 加密经济的共同知识库

2019.02.26

Copyright © 2019 Nervos Foundation

Supported by Nervos Community


## 摘要

---

Nervos Network 是一个采用分层架构的加密经济网络，它将加密经济的基础设施分为两层：验证层 (Layer1) 是信任之锚，提供智能托管服务；生成层(Layer2) 提供高性能的交易以及隐私保护。 本文概述了 Nervos 的 Layer 1 —— Nervos Common Knowledge Base(CKB) 。CKB 是公有非许可链，它能创造信任，并将信任传递到上层，使整个 Nervos 网络可信。CKB 也是 Nervos 网络的价值存储层，为在网络中创建的资产、身份和其他公共知识提供公共、安全、不受审查的托管服务。

# 目录

---

1. Nervos 初衷
  2. Nervos 概述
  3. 共识机制
  4. 编程模型
    - i. 状态生成与验证
    - ii. Cell 模型
    - iii. 虚拟机
    - iv. 交易
  5. 经济模型
  6. 网络
  7. Nervos 总结
  8. 引用文献
  9. 附录
  10. 联系 Nervos
- 

# 1. Nervos 初衷

---

世界需要一个点对点加密经济网络。这个网络不仅要允许人们在上面协作，更要能激励人们协作。为了创造这样的激励，我们必须能在这个点对点网络中定义、发行、转移和拥有属于自己的资产。而区块链技术是我们得以实现这种网络的最后一块拼图。比特币是第一种公有非许可链，其设计初衷是作为点对点现金。以太坊拓宽了区块链的使用场景，创造了一个能够创建各种去中心化应用的通用可信计算平台。在比特币和以太坊网络中激增的各种应用，已经证明了加密经济这个未来的概念。然而，这些网络受制于声名狼藉的可扩展性问题，它们的交易处理能力并不会随着节点数量的增加而提高，这严重限制了它们的应用潜力。

近年来，区块链社区已经提出了多种扩容方案。通常我们可以将这些方案分为两类，链上扩容和链下扩容。链上扩容方案试图在共识层扩容。共识机制是区块链协议的核心，节点通过网络相互传递消息，并最终达成共识。然而，共识几乎是缓慢的代名词，因为在公开的网络中，消息的传递是缓慢且不确定的，节点必须等待和重试以达成共识。

为了在这一层实现扩容，我们要么提高节点的处理能力和网络带宽(但是这种方式必然会提高硬件和基础设施的成本，降低去中心化)，要么进行分片。分片技术是将节点分散到很多个小分片中，每个分片只需要处理一部分交易。现在，分片的概念已经被众多互联网巨头广泛使用，因为它们在为数百万用户服务时，面临着相同的可扩展性问题。然而，分片技术以分片协作和跨片交易的复杂性著称，即使在可信环境中，分片数量的增加也会导致分片的性能下降。

相反，链下扩容方案承认共识过程具有固有的复杂性。他们认识到不同范围的共识需要承担不同的成本，而一个公有非许可链所创造的全球共识是最昂贵的共识。虽然全球共识很难扩容，但我们可以更聪明地用它。像大多数两方或多方之间的交易，除非需要安全地固化（即作为网络中的公共知识），否则无需让网络中的每个节点都知道。使用链下扩容方案，网络会将大部分的工作转移到上层，所以没有可扩展性方面的制约。而且链下交易还能带来额外的好处，比如低延迟和高隐私性。

尽管大家都认同链下扩容的思想，但我们发现目前还没有为链下扩容量身定制的区块链项目。例如，虽然闪电网络是最早的链下扩容方案，并且经过了数年的努力已发布测试网，但其受制于比特币的底层协议，离大规模的使用还很遥远。以太坊具有强大的

可编程能力，但其受制于面向计算的经济模型，并不适合链下扩容。这是因为链下参与者要执行大部分的计算，并要求链上可以安全地托管资产，并根据计算得出的最终状态转移资产。而以太坊的面向计算的设计，使交易很难并发执行，这成了扩容的瓶颈。

此外，现有的区块链项目还需要面对经济模型上的挑战。随着越来越多的用户和应用迁移到区块链平台，储存在区块链上的数据也会急剧增加。而现有的区块链方案更多地关注共识和计算的成本，使得用户可以只付一次费用就能永远占用全节点的存储空间。此外，加密货币的价格也非常不稳定，当加密货币价格上涨时，用户可能难以承担高额的交易费。为了解决这些问题，我们提出了 Nervos CKB，一个为分层的加密经济网络而设计的公有非许可链。

## 2. Nervos 概述

---

Nervos CKB (Common Knowledge Base, 公共知识库) 是一条 layer1 的区块链, 它是为网络提供公共知识托管的去中心化的安全层。这里的公共知识是指通过全球共识验证的状态。比如, 加密资产就是一种公共知识。

在 Nervos 中, CKB 和所有的 layer2 协议共同协作, 一起为加密经济服务。CKB (或者说 layer1) 是定义和存储状态的地方, 而 layer2 (生成层, 或者说计算层, 这两个名词可互换) 则是处理大多数交易以及生成新状态的地方。Layer2 的参与者最终会在必要时, 提交一些新生成的状态到 CKB 上。如果这些状态通过了全节点的验证, 则 CKB 会将它们安全地存储到全节点中。

分层架构将状态和计算分离, 从而赋予了每一层更大的灵活性和可扩展性。例如, 在生成层(layer2)上, 不同的链可以采用不同的共识算法。CKB 是 Nervos 网络的最底层, 提供最广泛和最安全的共识。然而, 不同的应用对共识范围会有不同的偏好, 强制所有应用都使用 CKB 共识是非常低效的选择。应用可以根据它们的实际需求选择合适的状态生成方法。只有当状态需要作为

公共知识时，应用才必须提交状态给 CKB 以获得全球共识的验证。

状态生成方法，包含但不限于以下几种：

- 本地客户端的状态生成器：用户可以直接运行客户端生成新状态。开发者可以使用任何一种编程语言来实现这样的生成器。
- 网络服务：用户可能会使用传统的网络服务来生成新状态。现有的所有网络服务都可以与 CKB 协作产生状态，以获得更多的信任和更好的流动性。例如，游戏公司可以将游戏中的物品定义为 CKB 中的资产，而游戏本身提供产生游戏数据的网络服务，这些游戏数据会在 CKB 中进行验证和存储。
- 状态通道：两方或者多方用户，可能会通过点对点的通信过程来生成新状态。
- 生成链：生成链是一条可以生成并存储状态到 CKB 的区块链。生成链既可以是非许可链也可以是许可链。在每条生成链中，节点可以在更小范围内达成共识，并能提供更好的隐私和性能。



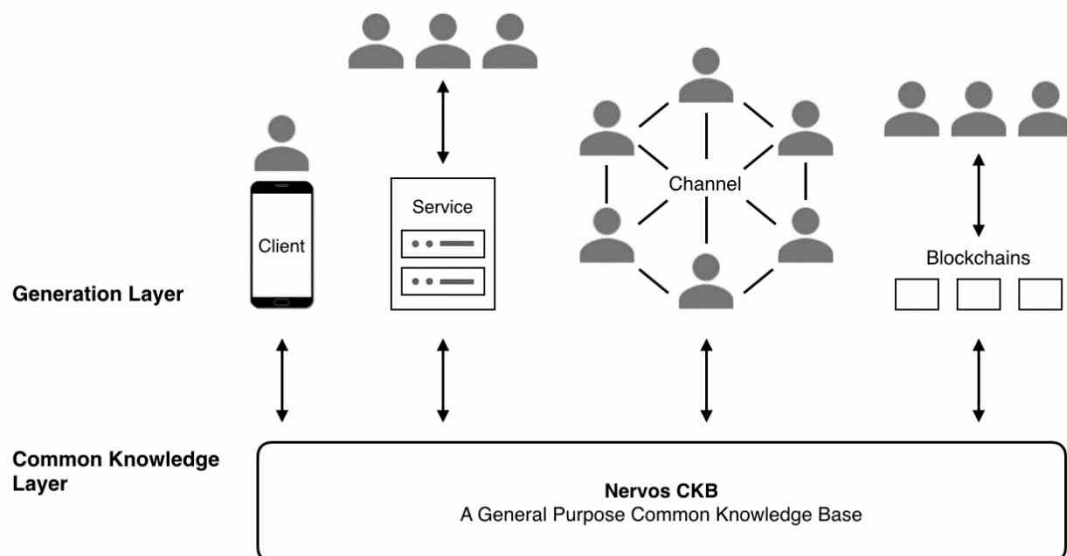


图 1. 分层网络结构

CKB 是由基于工作量证明( POW )的共识机制, 基于 RISC-V 指令系统的虚拟机, 基于 Cells 的状态模型, 面向状态的经济模型, 以及点对点网络组成。基于工作量证明的共识机制使得 CKB 能提供公共的抗审查服务。CKB 虚拟机与 Cell 模型的组合, 为开发者提供了一个带状态的图灵完备的编程模型。CKB 的经济模型是为公共知识的托管和长远可持续而设计的。CKB 的点对点网络为不同类型的节点提供了安全和最优的通信服务。

### 3. 共识机制

---

CKB 的共识机制是一个基于工作量证明的改良版中本聪共识机制，其设计目标是，在具有网络延迟和拜占庭节点的分布式环境中，达到开放性，正确性以及高性能。非许可链运行在开放的网络中，节点可以自由地加入与退出，并且没有活性假设。这些是传统的拜占庭共识算法很难解决的问题。中本聪通过引入经济激励以及概率性共识解决了这个问题。在比特币的中本聪共识中，区块扮演了投票的角色，这导致交易确认时间很长(10 分钟到 1 小时)，因此用户体验较差。

CKB 共识是一个基于中本聪共识的变种，这意味着它也允许节点能自由加入与退出。每个节点既可以通过挖矿（运行一个特定的算法来生成工作量证明）产生新区块，也可以通过验证区块的有效性参与共识。CKB 使用了对 ASIC 中立的工作量证明算法，以确保尽可能均匀地分发代币和维护网络安全。正确性包括最终一致性，可用性和公平性。最终一致性确保每个节点都能看到完全相同的状态副本。可用性确保网络可以在合理的时间内响应用户的请求。公平性确保矿工的投入可以得到公平的回报，因而愿意维护网络的安全。

高性能包括交易延迟，即从提交请求到确认执行结果的时间间隔，还有交易吞吐量，也就是每秒整个系统能够处理的交易数量。这些指标的大小依赖于区块时间（连续两个区块的平均出块时间间隔）的设置。 如果想知道更多细节请参阅 CKB 的共识白皮书。

## 4. 编程模型

CKB 提供了基于 CKB 虚拟机和 Cell 模型的带状态的图灵完备的编程模型。

表 1. 比较 Bitcoin, Ethereum 和 CKB 的编程模型

	Bitcoin	Ethereum	CKB
<b>Instruction Set</b>	Script	EVM	RISC-V
<b>Cryptographic Primitive</b>	Opcode	Precompile	Assembly
<b>Stateful</b>	No	Yes	Yes
<b>State Type</b>	Ledger	General	General
<b>State Model</b>	UTXO	Account	Cell
<b>State Verification</b>	On-chain	On-chain	On-chain
<b>State Generation</b>	Off-chain	On-chain	Off-chain

在 CKB 编程模型中包含以下三个部分

- 状态生成(链下)

- 状态验证(CKB 虚拟机)
- 状态存储(Cell 模型)

这个模型将去中心化应用的逻辑分成了两部分(生成和验证), 并且这两部分逻辑在不同的地方执行。状态生成的逻辑在链下的客户端执行; 新的状态被打包到交易中并广播到全网。CKB 交易与比特币类似, 也是基于输入/输出的结构。一笔交易输入包括对一笔交易输出的引用, 和一个能解锁该交易输出的证明。在客户端生成的新状态就是交易输出, 在 CKB 中也被称作 Cell。Cell 是 CKB 中最基本的状态存储单元, 也是用户所拥有的资产, 它必须遵循特定脚本的相关应用逻辑。CKB 虚拟机执行交易输入内的证明以验证用户有权使用被引用的 Cell; 执行 Cell 引用的脚本以验证状态转换符合特定的应用逻辑。通过这种方式, 网络中的所有节点都能验证和托管新生成的状态。

状态在 CKB 中是一等公民, 状态被包含在交易和区块中, 并且可直接在节点间同步。虽然编程模型是带状态的, 但是运行在 CKB 虚拟机中的脚本却是不带状态的, 这可以让 CKB 脚本具有确定性的执行结果, 有利于并行执行, 并且方便被其他脚本调用。

## 4.1 状态生成与验证

在 Nervos 的去中心化应用中，状态的生成和验证是分离的。由于两个过程是在不同的地方执行，因此 CKB 获得了额外的灵活性，即状态的生成和验证可以使用不同的算法实现。

使用相同的算法生成和验证状态，是处理通用问题的最直接的选择。而在我们的模型中，同样的算法可以有两套实现，一套运行在链下，执行环境因应用而定，另外一套则运行在链上的 CKB 虚拟机中。链下通过算法生成的新状态(基于先前的状态和用户输入)，被封装成一个交易，并被广播到网络中，CKB 节点会在链上执行同样的算法（不同的实现），输入同样的前置状态和用户输入，并验证计算结果是否与交易输出吻合。

状态生成与验证分离有以下几个优点：

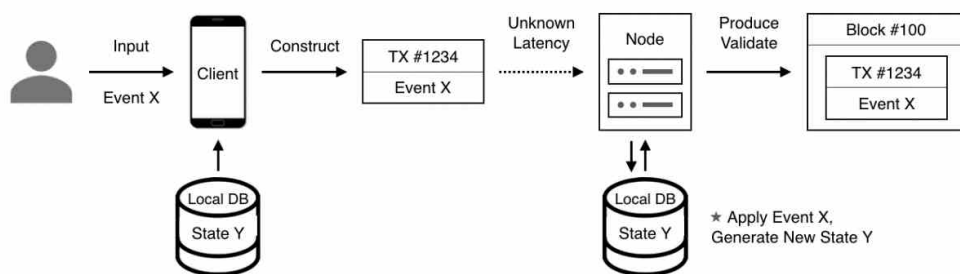
- 确定性的交易执行：交易执行的确定性是去中心化应用的一个核心要求。如果交易只包含用户输入，需要经过节点的计算才能生成新状态(如以太坊)，那么交易产生者不能确定链上的计算内容，这可能会产生意料之外的结果。而在 CKB 中，用户是在客户端生成的新状态。他们在交易被广播之前就知道新状态。无论交易是否能够通过验证，这笔交易的输出都是确定的。

- 并行处理：如果交易只包含用户输入且新状态只由节点产生，由于节点在验证过程中无法预测下一个状态，所以无法确定交易之间的依赖关系。而在 CKB 中，因为交易明确包含先前的状态和新的状态，节点可以在验证之前确定交易之间的依赖关系，从而能够并行处理交易。
- 高效的资源利用率：当应用程序的逻辑被分离并且在不同地方执行时，网络可以在节点和客户端之间更加均衡地分配计算资源，因此可以更有效率地利用系统资源。
- 灵活的状态生成：即使使用相同的算法，开发者也可以用不同的方式来实现状态的生成和验证。在客户端，开发者可以灵活地选择高性能和快速开发的编程语言。

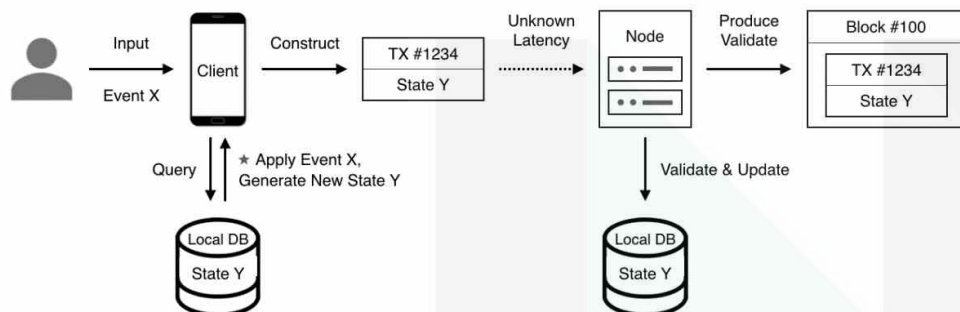
在某些场景中，状态验证可以使用不同(但相关)的算法，这种算法比状态生成算法更加高效。比特币交易可以被视为最典型的例子：构造比特币交易主要包含了一个搜索合适的 UTXO 的过程，而交易的验证就只有数字加减和简单的比较。其他有趣的案例包括排序和搜索算法：快速排序是平均情况下最好的排序算法之一，其计算复杂度是  $O(N\log(N))$ ，但验证排序结果却只需要  $O(N)$ 。用二分法搜索已排序数组中元素的索引，其计算复杂度是  $O(\log(N))$ ，但其验证算法只需要  $O(1)$ 。越是复杂的业务逻辑，就越有可能存在具有不同的计算复杂度的非对称的生成和验证算法。

利用状态生成和验证之间的非对称计算，我们还能进一步提升系统吞吐量。此外，将计算细节转移到客户端对于算法保护和隐私保护也很有价值。随着零知识证明等技术的进步，我们也许会发现更有效率的，对于通用问题的生成与验证解决方案，而 CKB 是这类解决方案最自然契合的选择。

我们将生成新状态以及创建新 Cells 的程序称为生成器。生成器在本地客户端运行(链下)。它们将用户输入和已有的 Cells 作为程序的输入，创建包含新状态的 Cells 作为输出。生成器使用的输入以及产生的输出一起组成了一笔交易。



A. Non-deterministic state generation on a node.



B. Deterministic state generation on client

图 2. 状态生成和验证的分离



## 4.2 Cell 模型

Cell 是 CKB 中最基本的状态单元，用户可以在其中包含任意的状态。一个 Cell 由以下几个字段组成：

- 容量(capacity): Cell 的大小限制。一个 Cell 的大小是指包含所有字段的总字节数。
- 数据(data): 状态数据存储在 Cell 中。它可以是空的，Cell 的总字节数必须总是小于或等于 Cell 的容量。
- 类型脚本(type): 验证状态的脚本。
- 锁定脚本(lock): 代表 Cell 的所有权的脚本。只有 Cell 的所有者才能转移 Cell。

Cell 是一个不可变的对象，自它创建后就无法修改其内容。每个 Cell 都只能被使用一次，它不能同时作为两个不同交易的输入。在 Cell 模型中，当之前的 Cells 会被标记为历史数据时，总会有总容量相同的新的 Cells 被创建出来替代它们。通过创建和发送交易，用户提供了包含新状态的 Cells，并能使存储旧状态的 Cells 失效。所有现存的(或者说活跃的) Cells 集合代表了 CKB 中最新版本的知识，而历史的(或者说废弃的) Cells 集合代表了知识的所有历史版本。

CKB 允许用户一次性转移一个 Cell 的全部容量；或者只转移

一部分容量，这会产生更多的 Cells (例如，一个容量是 10 的 Cell，可以产生两个容量是 5 的 Cells)。

CKB 虚拟机会执行两种脚本 (type 和 lock)。当检查交易输出，生成新的 Cell 时，CKB 虚拟机会执行 type 脚本，以确保 Cell 中的状态是有效的，符合特定的规则。当 Cell 作为交易输入的引用时，CKB 虚拟机会将证明作为参数执行 lock 脚本，以确保用户有适当的权限来更新或者转移 Cell。如果执行 lock 脚本返回 true，用户就可以根据 type 脚本指定的规则更新数据或者转移 Cell。

Type 和 lock 脚本的设计为各种可能性打开了方便之门，例如：

- 升级加密算法——任何人都可以部署以 C 或 C++ 语言写的密码学库，并且在 type 和 lock 脚本中使用它们。在 CKB 虚拟机中，没有硬编码的密码学原语，用户可以自由地选择任何一种密码学签名方案对交易签名。
- 多重签名——用户可以轻易创建 M-N 多签脚本或者更复杂的 lock 脚本。
- 租借——Cell 的拥有者可以将 Cells 租给其他人使用，同时保留对 Cells 的所有权。

与 UTXO 或者 Account 模型相比, Cell 模型是更通用的状态模型。UTXO 和 Account 模型都可以表达资产与拥有者的关系。UTXO 模型使用 lock 脚本定义资产的所有权, 而 Account 模型使用账户余额定义用户的资产所有权。UTXO 模型的账本历史更清晰, 但是由于缺乏通用状态, 使得其原本就表达能力不强的脚本更难使用。Account 模型非常容易理解, 并且可以非常好地支持授权以及身份管理, 但是它很难并发处理交易。拥有 type 和 lock 脚本的 Cell 模型取这两个模型之长构建了一个更通用的状态模型。

### 4.3 VM 虚拟机

CKB 虚拟机是一个用来执行 type 和 lock 脚本的, 基于 RISC-V 指令集的虚拟机。它只使用了标准的 RISC-V 指令集, 维护了一个符合标准的 RISC-V 软件实现, 因而能够获得最广泛的工业支持。运行在虚拟机上的密码学原语, 其实现和部署和普通脚本一样, 而不是作为虚拟机自定义的指令。CKB 虚拟机支持系统调用, 脚本可以通过系统调用读取 CKB 上的当前交易以及区块链上的通用信息等元数据。CKB 虚拟机定义了每条指令的周期, 在矿工执行交易验证时提供总执行周期, 以帮助矿工确定交易费。

现有的区块链项目都是将密码学原语写死在协议中。例如, 比特

币有特殊的加密操作码，比如 `OP_CHECK`，而以太坊则是使用特殊的预编译合约存放在一个特殊地址中（例如 `0001`），来支持诸如 `ecrecover` 这样的加密操作。在这些区块链项目中，为了增加新的密码学原语，我们只能采用软分叉（例如比特币复用 `opcodes` 来支持新的密码学原语）或者硬分叉的方式实施。

CKB 虚拟机是一个与密码学操作无关的虚拟机。没有任何密码学指令写死在 CKB 虚拟机中。我们总能像普通脚本那样部署和使用新的密码学原语。CKB 虚拟机作为一个符合 RISC-V 标准的软件的一个好处是，现有的以 C 语言或者其他语言实现的密码学库，都可以轻易地移植到 CKB 虚拟机上，并被 Cell 脚本所调用。CKB 在交易验证中默认的哈希算法和公钥加密算法就是用这样的方式实现的。CKB 虚拟机的与密码学操作无关的特性，可以允许 Dapp 开发者在不影响其他用户的情况下，在 Nervos 上使用任何新的密码学技术（例如 Schnorr 签名，BLS 签名，和 zkSNARKs/zkSTARKs），还可以在后量子时代继续保障 CKB 用户的资产安全。

CKB 虚拟机之所以选择一个面向硬件的指令集架构，是因为区块链是一个类似硬件的软件。尽管开发一条新链像其他软件一样容易，但更新和升级区块链却像硬件一样困难。而 RISC-V 是为

芯片设计的指令集，它非常稳定，其核心指令集在未来都不会发生变化。能够在不需要硬分叉的情况下保持与生态系统的兼容性是 CKB 虚拟机的一个关键特性。RISC-V 的简洁性也让运行时间成本的建模变得更容易，这对于计算交易费来说相当重要。如果想知道更多 CKB 虚拟机的细节请参阅 RFC 0003 。

## 4.4 交易

一笔交易就代表了一次状态转换，这会导致 Cell 的转移或更新，或者两者同时发生。在单笔交易中，用户可以更新一个或者多个 Cell，或者将他们的 Cell 转移给其他用户。在一笔交易中的状态转换是原子性的，他们要么全部成功，要么全部失败。

一个交易包含以下内容：

- 依赖(deps): 依赖的 Cell 集合，提供了验证交易所需的只读 Cells，这些 Cell 引用必须是活跃的。
- 输入(inputs): 包含 Cell 引用和证明。Cell 引用指向了在交易将被转移或者更新的活跃的 Cell。证明(例如签名)则用来验证交易创建者是否有权转移或更新被引用的 Cells。
- 输出(outputs): 在状态转换中产生的新 Cells。

CKB 的 Cell 模型和交易的设计对轻客户端非常友好。因为所有的状态都保存在区块中，同步区块的同时也完成了状态的同步。轻客户端只需要同步区块而不需要额外同步状态或计算状态转换。如果区块中只存储了事件，那么轻客户端在同步状态时就需要全节点的支持。这可能导致在大型网络中的状态同步非常困难，因为系统对同步的激励太少了。这和区块同步非常不同，矿工会尽可能广泛地传播区块以获得奖励。由于不需要额外的状态同步，CKB 协议使得轻节点和全节点更加对等，从而能够建立一个更健壮和更去中心化的系统。

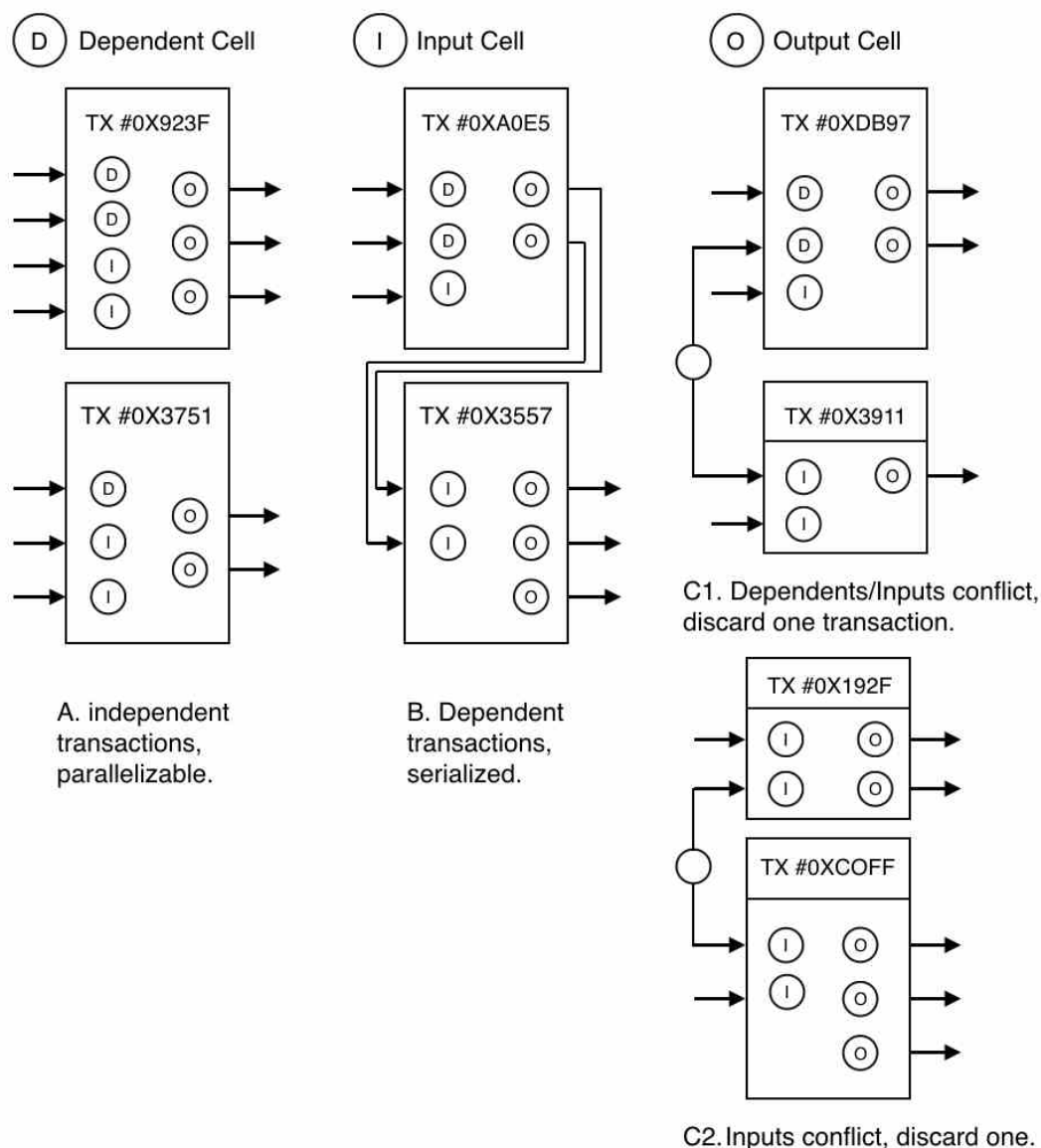


图 3. 交易并行和冲突检测 CKB 交易中的依赖和输入的设计，使节点更容易确定交易的依赖，以并行处理交易。单笔交易中可以包含不同类型的 Cells，以实现跨类型的原子操作。

## 5. 经济模型

---

一个设计良好的经济模型应该激励所有参与者做出贡献，以获得加密经济的成功以及最大化区块链的使用。CKB 经济模型的设计目标是，激励用户、开发者以及节点运营者一起为共同知识的安全托管努力。CKB 经济模型的主题是状态，而不是计算，使用 Cell 容量和交易费作为对 stakeholders 的奖励。

### 5.1 状态成本与 Cell 容量

在 CKB 上产生和存储状态需要付出成本。新状态需要被全节点验证(要付出计算成本)，而状态的存储也需要全节点持续提供磁盘空间。现有的非许可链只收取一次性的交易费，却允许状态永久地占用所有全节点的存储空间。

在 CKB 中，Cell 是最基本的状态储存单元。Cell 的拥有者可以使用 Cell 储存自己的状态，也可以借给其他人使用。因为一个 Cell 的容量在同一时间只能被一个人使用，拥有者如果自己使用这些容量，就要放弃将这些容量借出(换成 CKB 或者借给其他用户)而获利的机会。在这个机会成本下，使用者为了存储空



间所付出的成本会同时与时间和空间成正比——占用容量越大、占用时间越长，他们要付出的机会成本就越高。与预付模型(例如在以太坊社区讨论的存储租金)相比，CKB 隐含的状态成本模型可以避免预付金被用光，进而避免因系统必须回收状态导致其他依赖该状态的合约或者应用被破坏。

Cell 的元数据(capacity, type 和 lock)也是状态，也要占据用户的 Cell 容量，因而也要付出状态成本。这些元数据成本会激励用户尽可能少地产生 Cell，并提高存储效率。

## 5.2 计算成本和交易费用

更新 Cell 的数据或者转移 Cell 的所有权都会产生交易费。矿工可以基于 CKB 虚拟机在验证交易时消耗的指令周期和状态变化，去设定他们愿意接受的交易费率，市场会决定实际的交易费用。使用上述的编程模型，Cell 拥有者还可以代替他的用户支付交易费。

因为 Cell 容量是 CKB 中唯一的原生资产，因此用户用它来支付交易费是最方便的。然而，只要能被矿工接受，用户也可以使用任何自定义的资产作为交易费；在 CKB 交易中没有定死支付方式。之所以能够采用灵活的支付方式，正是因为经济模型和原

生资产的设计并非以计算为中心，而是以状态为中心。虽然 Cell 容量可以作为一种支付交易费的方式，但是它的首要功能还是确保共同知识的存储安全，也就是可以储存和长期保存状态。通过自由市场竞争支付方式并不会损害它的价值。

将交易费的支付方式限定为区块链的原生资产，是阻碍区块链大规模应用的重要障碍。这要求用户在使用任何区块链服务前都必须获得这种原生资产，这会提高了新用户的进入门槛。通过允许 Cell 拥有者代替用户支付费用，以及允许用各种自定义的资产付费，CKB 可以提供更好的用户体验，并为开发者提供更广泛的商业模式选择。

请查阅 Nervos 的 CKB 经济白皮书，了解更多关于经济模型的细节。

## 6. 网络

---

我们可以将 CKB 节点分为三类：

- 挖矿节点：他们参与 CKB 共识过程。挖矿节点收集新的交易，把他们打包到区块中，并在生成工作量证明时产生新的区块。挖矿节点不需要存储全部的交易历史，只需要存储当前活跃的 Cell 集合。
- 全节点：他们需要验证新的区块和交易，转发区块和交易，以及选择他们认可的区块链分叉。全节点是网络中的验证者。
- 轻节点：他们信任全节点，只订阅和存储和他们相关的 Cells 子集。他们只需要消耗最小的资源。当前，用户越来越多的依赖移动设备和移动 app，轻节点正是为移动设备而设计的。

现在，单一的区块链网络(在其中每个节点都扮演相同的角色并表现出相同的功能)正面临着严酷的挑战。全节点验证所有的区块和交易数据，依赖最少的外部信任，但却需要承担较高的运行成本。轻节点通过牺牲少量的信任，大幅降低了交易验证的成本，

进而带来了更好的用户体验。在一个成熟的加密经济网络中，最多的节点会是轻节点，然后才是全节点和挖矿节点。因为轻节点在获取和验证状态上依赖全节点，数量庞大的轻节点将会需要许多全节点为他们服务。在 CKB 的经济模型中，运行全节点所需要的计算和存储资源可以维持在一个合理的水平，由于运行全节点的门槛很低，会出现大量为轻节点服务的全节点，使得网络更加去中心化。

## 7. Nervos 总结

---

我们设想了一个分层的加密经济，CKB 是它的基础层。CKB 是这个加密经济的去中心化信任之锚，它保障了上层的去信任活动的安全性。CKB 是一个共同知识的托管网络，其中的状态为全球共识所验证，并且储存在高度可用的点对点网络中。CKB 是一个从零开始，为多层架构所设计的公有非许可链，它主要关注状态，而非计算。在 CKB 中，用户和开发者可以定义、发行、转移和储存加密资产，他们也可以创造数字身份，并在加密经济中使用这个身份。总之，限制 CKB 使用边界的只是我们的想象力。

## 8. 引用文献

---

1. Satoshi Nakamoto, "Bitcoin A Peer-to-Peer Electronic Cash System", 2008
2. Vitalik Buterin, "Ethereum A Next-Generation Smart Contract and Decentralized Application Platform", 2014

## 9. 附录

---

共同知识是指社区中的所有人都认同的知识。社区参与者不仅认同知识本身，还知道社区中的其他人也认同这个知识。

在过去，共同知识散落在每个人的头脑中，它的形成需要经过反复的沟通和确认。今天，随着密码学以及去中心化账本技术的进步，算法以及机器正在取代人们成为产生和存储共同知识的媒介。每一条区块链中的数据，包括数字资产和智能合约，都是一条共同知识。

区块链是共同知识库。参与区块链网络，意味着认可以及帮助验证其中所含的共同知识。区块链存储交易和证明，用户信任这些交易的有效性并且知道其他人也信任它们。

人们赖以制订计划的知识传递给他们的各种途径，对任何解释经济过程的理论来说，都是至关重要的问题。而利用起先分散在全体人民中的知识的最好途径，至少是经济政策——或设计一个有效的经济体制——的主要问题之一。

- *知识在社会中的运用*, 弗里德里希·A·哈耶克, 1945

## 10. 联系 Nervos

---



Website: <https://www.nervos.org>



Github: <https://github.com/nervosnetwork>



Blogs: <https://medium.com/nervosnetwork>



Twitter: <https://twitter.com/nervosnetwork>



Telegram: <http://t.me/nervosnetwork>



Forum: <https://talk.nervos.org>



Reddit: <https://www.reddit.com/r/NervosNetwork>