

# MULTIVERSUM

HERE TO STAY

白皮书 版本 1.0.5

商业 | 技术

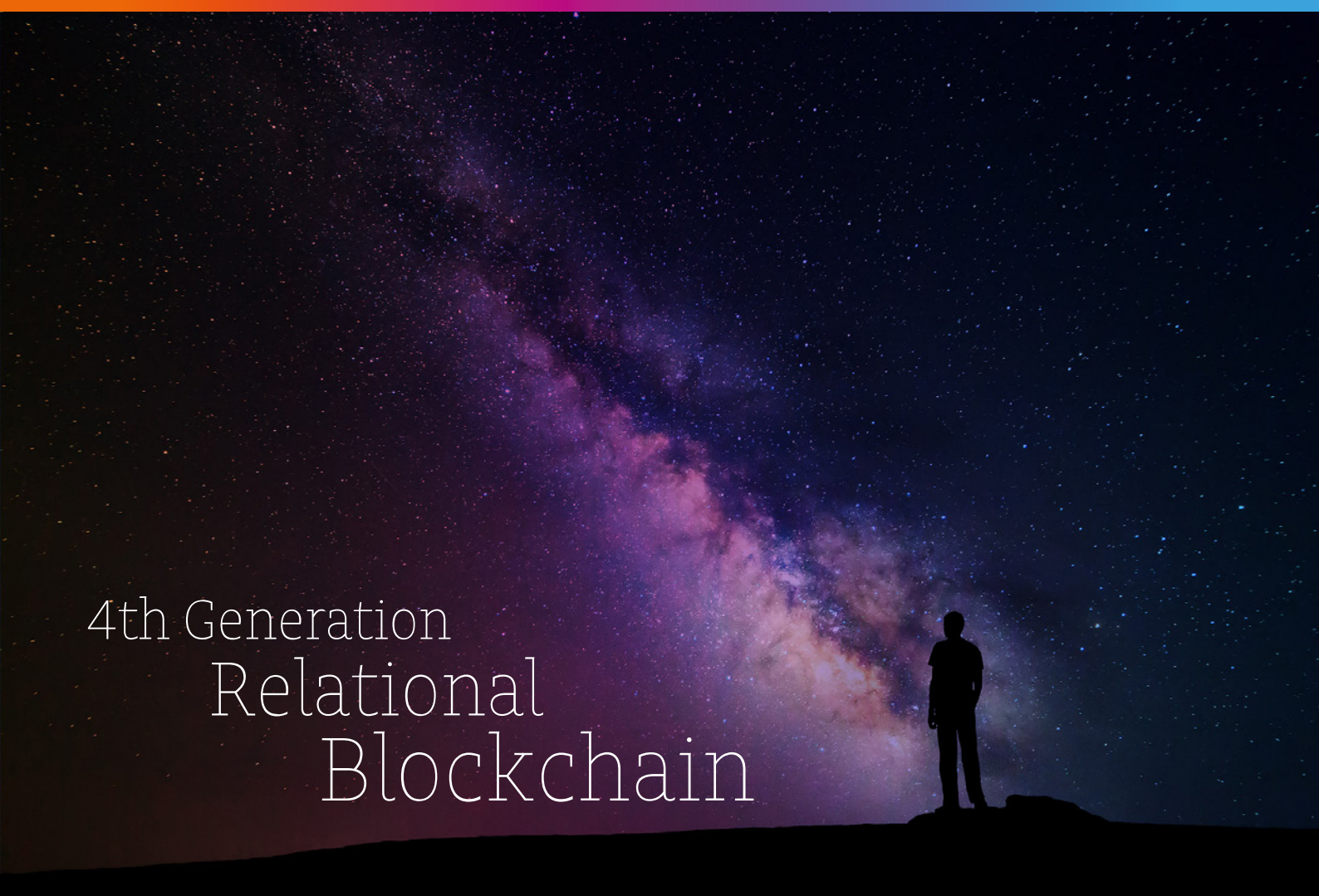
英语（原始文档）

中文翻译：魏俊

2018.02.06,

作者：Multiversum团队

[www.multiversum.io](http://www.multiversum.io)



4th Generation  
Relational  
Blockchain



## 声明

本人仅对所提供的原始文档按照“原样”进行翻译，仅对翻译的文本内容正确性负责。如有疑问或者问题可以联系译者或是参考原始文档。

Haslett5236  
2018年2月19日

# Multiversum 的身份和使命

加密货币的先锋比特币，以及基于工作量证明算法进行交易验证的所有各种克隆和分支都被视为第一代区块链。

以太坊领导智能合约功能区块链的第二代产品则更具异构性，可轻松标记资产。

两种架构的每区块都具有极低的能效和中低的块验证速度和交易。解决可扩展性，速度和能源消耗问题是第三代区块链解决方案的目标，它使用不同的方法和技术，例如证明权益验证算法，离线路径，图-链以及完全或部分的集中化。

第四代远远超出了这一点，实现了更快，更具可扩展性的解决方案，同时尝试从商业角度提高竞争力；简单的数据链不够灵活，无法满足企业环境需求，因为复杂的数据结构需要在表格中组织（如关系数据库一样）。

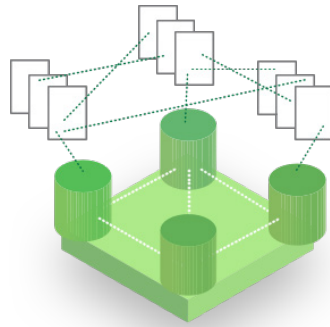
同时，这些结构需要通过基于区块链的技术进行验证并使得结构不可变，从而提高可追溯性和安全性。换句话说，第四代区块链将这项技术引入到一个完整的主要生产应用程序中，并在数据存储，应用程序分散，审计，安全性和可靠性等方面扩展当前以业务为导向的服务。Multiversum提供复杂的数据组织，而不是数据排序，链拆分和重新连接以实现更高的可伸缩性和并行性，以及完整性验证验证（即服务器代码的密码验证）而不是现有工作证明或证明权益解决方案的概念。

此外，Multiversum将以ERC20 / ERC23集成为特色，允许来自其他解决方案的硬币和代币托管在我们的连锁店，反之亦然，公证服务作为外部确认方法。同时，伴随着这些创新，我们当然会利用我们的同事已经实施的一些好的解决方案。

# Multiversum

## 第四代关系区块链

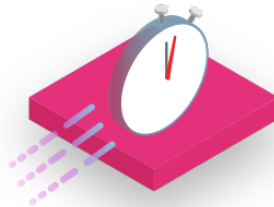
为什么选择Multiversum 4.0区块链??



### 关系区块链

一个全新的区块链，它具有不同类型的数据，与多维结构相关。

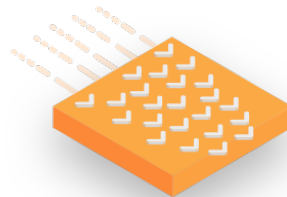
< 0,2 sec



### 交易速度

在不到0.2秒的时间内，资金将通过钱包转移，包括对交易的安全验证，是世界上最快的其一。

64000 tps → ∞



### 交易吞吐量

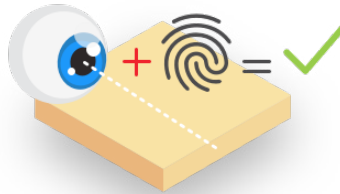
无与伦比的可扩展性：64核心服务器上高达64,000 Tps（1000 Tps /核心）。

POI



## 完整性证明

PoS（权益证明）将被PoI（完整性证明）取代。



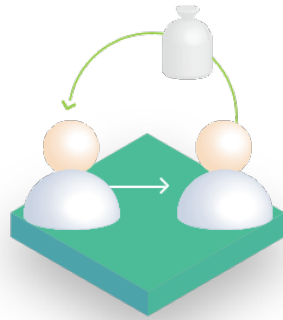
## 下一代钱包t

采用生物识别技术输入获取和资金转移的尖端安全性。



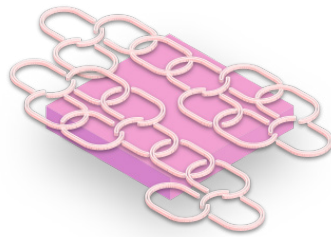
## 环境友好

Multiversum交易将具有微不足道的成本，并且几乎没有环境影响。



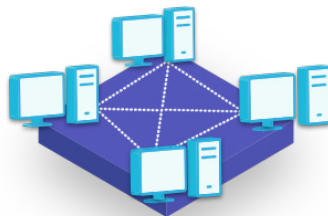
## 回滚

可选回滚可以在Multiversum托管的令牌上激活



## 可分割链

针对链路可分性的节点间资源优化。



## 恢复节点分配

MTV节点散布世界各地的可靠性和全球灾难恢复。

# 公开演示

## 当前的区块链状态

前沿的区块链现象都有一个共同的特征：出色的安全性和可靠性。在我们以巨大的处理能力，不可接受的污染，高昂的交易成本和缓慢为代价支付了这些难以代表当前的技术进步标准的同时，我们对现代金融和商业用例给出了合理的技术答案。

这种缓慢是由于缺乏横向可伸缩性引起的，即仅通过增加处理器而不是用更快的版本替换它们而获得的计算能力的增加。这种缓慢的另一个原因是其目前的区块链安全机制所固有的，旨在通过在计算能力和/或成本方面实现非常高的成本来防止任何人接管大部分集群（工作证明和权益证明）。

此外，当前区块链是单个数据实体状态变化的简单连接；重构这些实体的实际状态意味着整个链扫描，这会导致更大的系统缓慢和资源使用。

这种简化使得区块链不适合科学和工业用途，因为数据结构方面的要求可能变得非常复杂。此外，安全措施在数据层面停止，因为它们不能确保用户安全，因此即使位于链中或者阻止恶意账户，也无法恢复丢失或被盗的硬币和令牌。最后，另一个问题是加密货币之间的碎片化和不均匀性，这些加密货币无法相互通信并生活在无关的宇宙中。



## Multiversum和 区块链的全球应用

Multiversum技术通过自我验证和有组织的数据实体的分布式结构来增强数据层，从而通过符号链接将传统区块链相互连接在一起。

该技术为分布式和分布式自我验证交易系统奠定了基础：Multiversum区块链。

Multiversum允许创建一个关系加密数据库（一种先进的和有组织的数据存储解决方案），而不是现有的区块链简单数据模型，该数据库不仅可以处理单个数据类型，而且还可以处理复杂数据图形中的一系列数据结构相互关联。现在的关系是区块链的头等公民，并通过密码方法来保证实现。

它们中的每一个在请求状态改变时都会有自己的子链从原来的分支分裂出来，这些分支将在操作之后重新加入，以便被验证。

因此Multiversum是一种不断发展的区块链技术，提供独特的功能以克服先前分析的不便之处，并针对各种环境设置了一套加密验证和分配技术：行政，工业，金融和管理。

Multiversum的主要目标之一是在任何时候向市场提供最先进的产品：这将有可能采用敏捷软件开发方法。

敏捷方法意味着初始项目设计参与程度大幅度降低，有利于项目开发过程中遇到的经验增值，这些经验表明机会和威胁事先难以预测，奖励最佳做法并留下不足之处。

敏捷是一个既定的软件开发标准，并敦促开发商，产品所有者和投资者将项目范围视为灵活且容易适应市场需求。此外，在软件这样一个快速发展的领域，经过六个月的研究和一年的实施，发布一款产品，当它被设想为符合十八个月前的市场需求时，意味着提供一个过时的产品，以回答过时的问题，这可能已经被竞争对手解决，并且对刚刚产生的挑战缺乏反应。

相反，敏捷提供了在交付时向市场提供最具创新性的产品的机会。

## 速度及技术

这项技术的优势之一的确是速度，这要归功于它能够并行运行不同的交易以及我们区块链的拆分重新加入机制。这些功能可实现更大的水平可扩展性，并增加处理事务处理能力，为现有处理增加额外的计算能力，从而使每个节点数量更高，性能更佳。



## 水平可伸缩性

Multiversum 受益于两个特定功能，以最大限度地提高系统效率：

1-主链能够根据请求的资源 and 数据流，在多个子链中自动分割，并行化跨多个线程和节点的工作，从而优化其结构。 这个分拆过程一直执行到工作负荷正常化时，仍然自主地，链条将再次变成整体。 所有这些都是可能的，因为这种技术允许链的每个模块验证来自两个不同输入链路的两个不同子链。

2-2-数据分片，即允许在多个节点之间分配数据的技术。 给定一个ABC数据系列和三个群集节点，我们将有一个数据分布如下：

AB

BC

CA

此细分允许更高的事务处理速度，因为数据查询仅会影响子链节点，从而优化每个步骤。 我们技术的另一个极其重要的特性是高可用性7：即使在网络中的某些节点关闭的情况下，依赖于确保服务连续性的群集类型的机会。 使用前面的例子（A，B和C节点），如果C离线，则A节点和B节点仍然保持完全可操作，从而允许服务的连续性而没有任何类型的数据丢失，只要50%+ 1的节点保持操作。 这样，在发生多节点故障的情况下，集群将自动重新组织与每个节点通信的数据分配，直到完成操作恢复。

## 环境

Multiversum 也是环保型的：我们的主要目标之一是降低密码验证所需的计算能力，从而避免挖掘（工作量证明），这是巨大的能源和资源浪费。 我们正在实施完整性证明（Proof of Integrity），该协议通过检查解决每个交易持续性的软件的真实性和完整性来执行加密验证。

## 数据管理

Multiversum 及其加密关系数据库可以轻松构建它，而不会受到数据链接的限制。 每个钱包都会有一系列状态，并且会链接到一个人（用户），并且新的钱包状态更改将包括两个数据字段：以前的状态，以检查验证。 一个到最后一个交易（或最后一个主链链接）的链接，以便知道新状态链接的来源。 更改后，交易修改将被添加，其修改后的状态链接将重新加入主链。 因此，新事务将继承两个哈希值：一个来自状态链接，一个来自前一个事务，并且这样所有操作都将验证与事务本身相关的以前的事务。

这种先进的解决方案能够管理复杂的数据场景，使人们能够在我们的技术上实施任何类型的应用程序，确保全球机构，政府，金融和工业扩散，使整个区块链领域向前迈进了一步。

# MULTIVERSUM

HERE TO STAY

## Unique Features !

### **Crypto relational DB**

Autovalidating Complex  
Data structures

### **Proof of Integrity**

(Protocol Innovation)

### **Divisible/Re-joinable chains**

(Parallel Work)

### **Biometric Data integration as Electronic Signature seed**

(User Security)

### **Sharding data**

(Parallel Work)

### **Double Access Lock**

(Structural Security)

### **Minimal ecological footprint**

### **Reverse Access Denial**

(Structural Security)

### **Reciprocal chain confirmation**

(Interoperability with other BC)

### **Rollback**

(User Security)

### **Advanced API offer**

### **Native off-chain adapter for own ERC20**

(Interoperability with other BC)

### **Self managing Crypto-Cluster**

### **Java, Spring and Javascript**

(Libraries for Integration)

### **Native on chain adapter for own ERC20**

(Interoperability with other BC)

### **Freezable wallets**

(User Security)

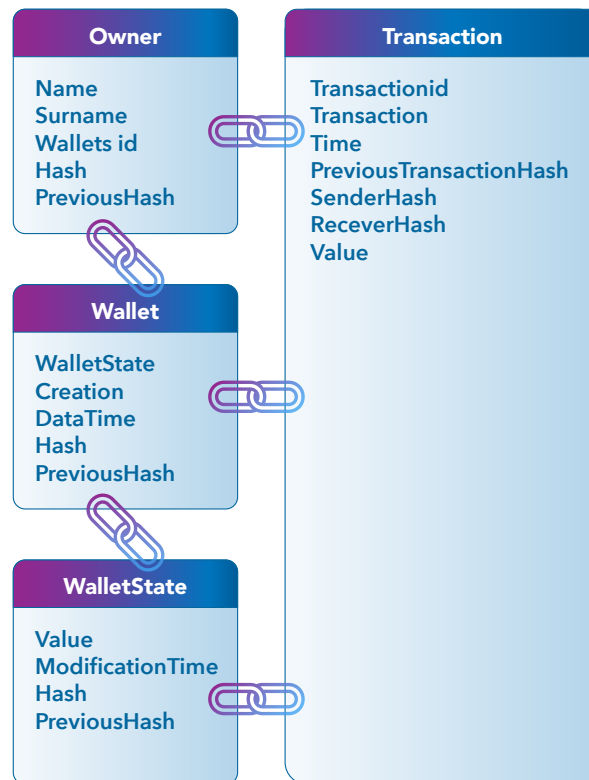
### **ERC23**

(Interoperability with other BC)

# Multiversum使命

Multiversum的目标是在区块链世界中实现代际升级，作为独有特点，我们提出以下目标：

1. 通过自我验证的复杂数据结构实现加密关系数据库
2. 基于当前系统工作负载（并行工作）的可分割/可重新连接的链
3. 数据分片（并行工作）
4. 高级API产品
5. 回滚（用户 安全性）
6. 可冷冻钱包（用户安全）
7. 将生物识别数据集成为电子签名的种子
8. ERC23接口（与其他区块链的互操作性）
9. 用于其自身ERC20 / ERC23的本机离线适配器（与其他区块链的互操作性）
10. 适用于ERC20 / ERC23客户端的原生离线适配器（与其他区块链的互操作性）
11. 完整性证明（协议创新）
12. 双重访问锁（结构安全）
13. 反向访问拒绝（结构安全）
14. 相互链确认（与其他区块链的互操作性）
15. Java, Spring和Javascript集成
16. ACID模型
17. 事务模型
18. 与SQL类似的语言



### 1. 实现具有自我验证复杂数据结构的加密关系数据库

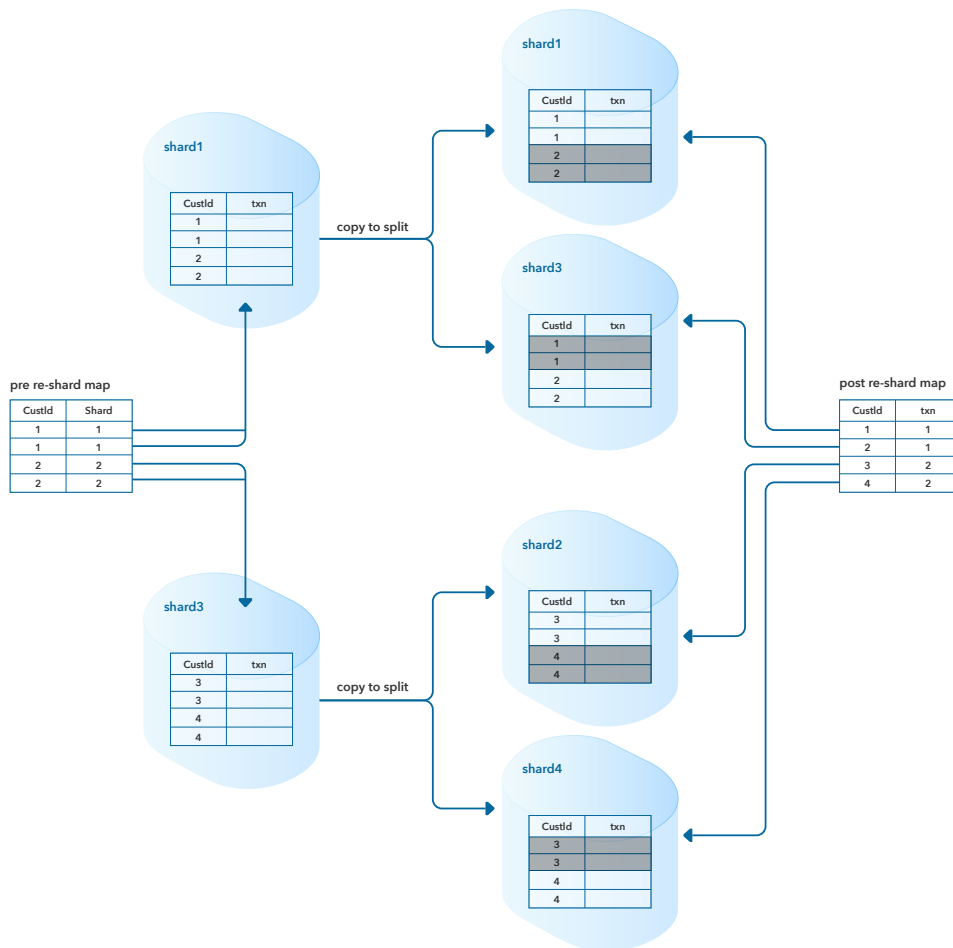
Multiversum对于工业和机构使用有强烈的使命，我们拥有复杂结构的数据的背景下，不可能以简单的链条以高效且规范化的方式来表示。我们的目标是成为市场上的第一个关系密码关系数据库，如果需要，可以分散或简单分布。这种能力源自可链式实体概念化：在我们的技术中，主链能够分成次链，包含不同的实体和记录集。这些实体将在其最后持续状态下重新加入，并在经过必要的修改后再次重新加入到主链的最后一个环节，从而再次成为一个整体。“可链接”接口预设了一种包含以前记录的两个或多个哈希的记录，不仅验证了一个但更多的子链。在Verstive硬币使用的Multiversum标准实现中，链上共存的可链接实体将属于四个表：用户，电子钱包，电子钱包状态，交易，相互关联以及相互确认它们自己。

### 2. 基于当前系统工作负载的可分割/可重新连接链（并行工作）

从一个给定的链路获得多条链路并将其加入的能力允许该技术使用工作负载分析器，这将指示集群需要在两个辅助链中分裂主链（并且可能无限期地再次分裂）的交易执行发生。一旦工作负载再次下降，允许多个预先存在的子链接回并验证。该机制允许并行工作，同时保持交易记录的安全性。

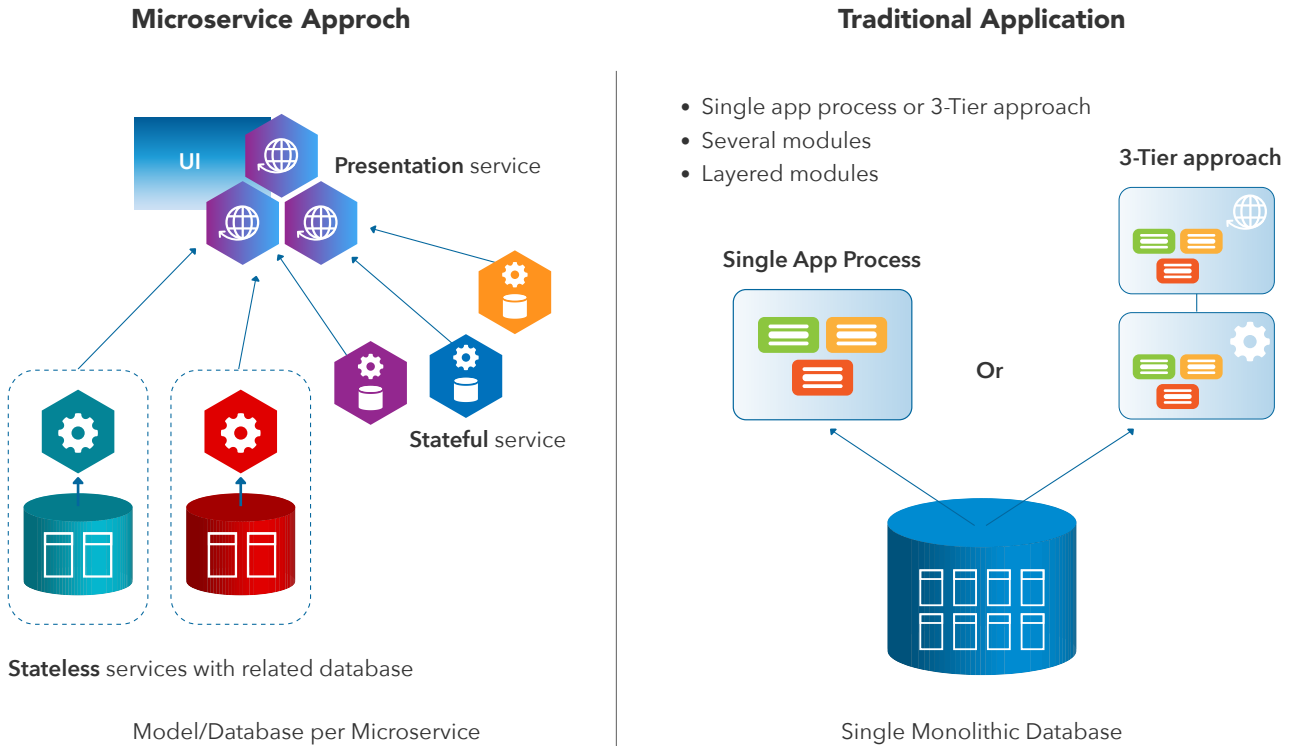
### 3. 数据分片（并行工作）

每个节点将包含整个链式数据或仅包含链条的一部分。 当需要数据分片时，协调器节点将设置特定的数据分区模式，以根据当前工作负载优化其自己的分配。 根据高可用性技术，即使在部分集群突然丢失的情况下，始终可以确保可靠性和持久性，因为至少有50%+ 1的节点能够存活。 这些节点在部分集群崩溃后，将能够重新分配和重新组织数据结构，以便能够尽快应对另一个部分集群崩溃。 通过技术2和3，Multiversum区块链将提高并行工作和数据分片容量，这意味着横向扩展能力，增强的安全性，高可用性，系统弹性，没有单点故障8和自我灾难恢复。



#### 4. 微服务结构和高级API提供

在基于微服务器和无服务器模式的平台上开发，Multiversum将能够提供先进的安全和现代API功能，并适应两种结构。

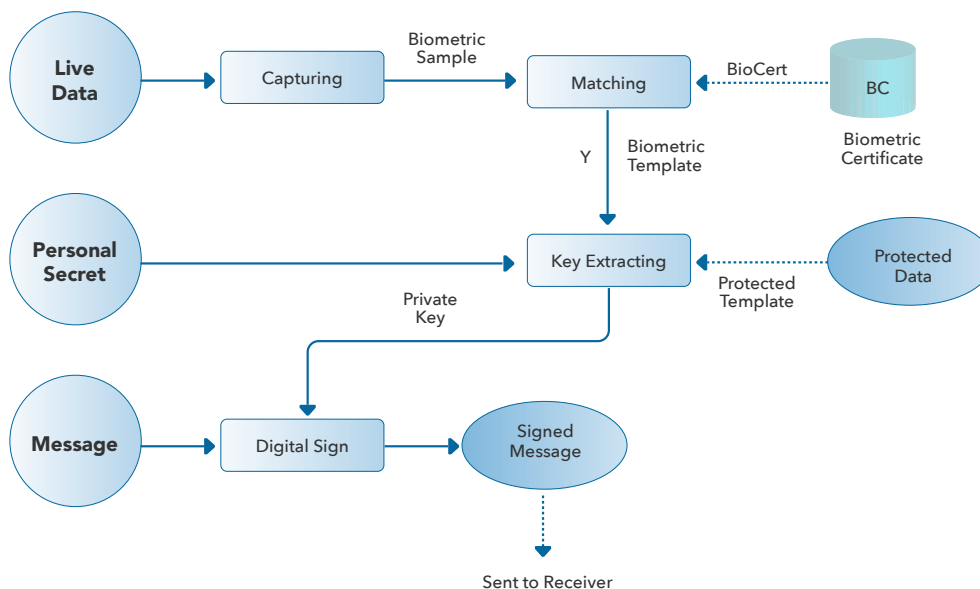


#### 5. 回滚（用户安全）

我们的技术在事务环境中将允许回滚不需要的操作，即通过实现一组事务恢复状态来恢复早期状态而不破坏链验证的可信度。可以启用此功能，可选地，在Multiversum区块链上托管的所有令牌和应用程序上。

### 6. 可冻结钱包（用户安全）

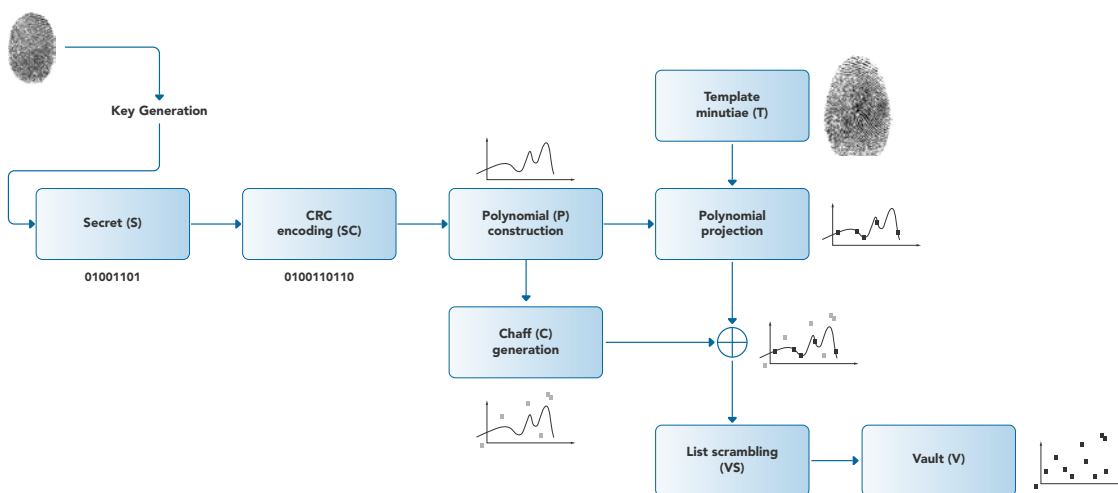
在业务逻辑方面研究其可行性后，将实施包含防止钱包冻结功能以防非法或可疑活动的机会。如果需要，基于Multiversum区块链的专有应用程序可以选择实现此功能。



Biometric Digital Key Generation Framework

### 7. 将生物识别数据整合为电子签名的种子

从Je-Gyeong Jo, Jong-Won Seo和Hyung-Woo Lee的作品11的研究开始，Multiversum团队将评估指纹，视网膜扫描和图形等生物特征数据的可行性。签名作为非对称加密密钥的来源，以保证签名者身份的真实性。加密数据的安全性及其在法律论证中的使用将被评估。此外，生物识别数据将用于Android，IOS和其他平台应用程序来管理用户安全。



Fuzzy Vault Scheme for Biometric Digital Key Protection



## 8. ERC23接口（与其他区块链的互操作性）

Versium硬币将开发实施ERC23接口，向后兼容ERC20，以确保与其他链的互操作性。

```
int totalSupply();
int balanceOf(String walletId);
boolean transfer(String receiverWalletId, int value);
boolean transferFrom(String senderWalletId, String receiverWalletId, int value);
boolean approve(String spenderWalletId, int _value);
int allowance(String walletId, String spenderWalletId);
boolean Transfer(String senderWalletId, String receiverWalletId, int value);
boolean Approval(String walletId, String spenderWalletId, int _value);
```

## 9. 用于专有ERC20 / ERC23的本地离线适配器（与其他区块链的互操作性）

Multiversum将开发本地适配器，以允许其自有硬币和令牌的进站和出站流向非专有链。

## 10. 用于外部ERC20 / ERC23的原生离线适配器（与其他区块链的互操作性）

Multiversum将开发本地适配器，以允许自己的链上来自非专有链的硬币和令牌的进站和出站流。



# Integrity

## 11. 完整性证明（协议创新）

Multiversum提出了一种算法，它能够验证编译节点的密码有效性以及来自一个已编译节点的响应的一致性，作为取代工作量证明和权益证明和其各式各样的变体的解决方案大部分节点。

该验证是针对随机种子挑战进行的，与通过软件自身的外部组件（受保护免于逆向工程，并通过加密通道与节点软件进行通信）以及事务数据计算的哈希结合进行验证。

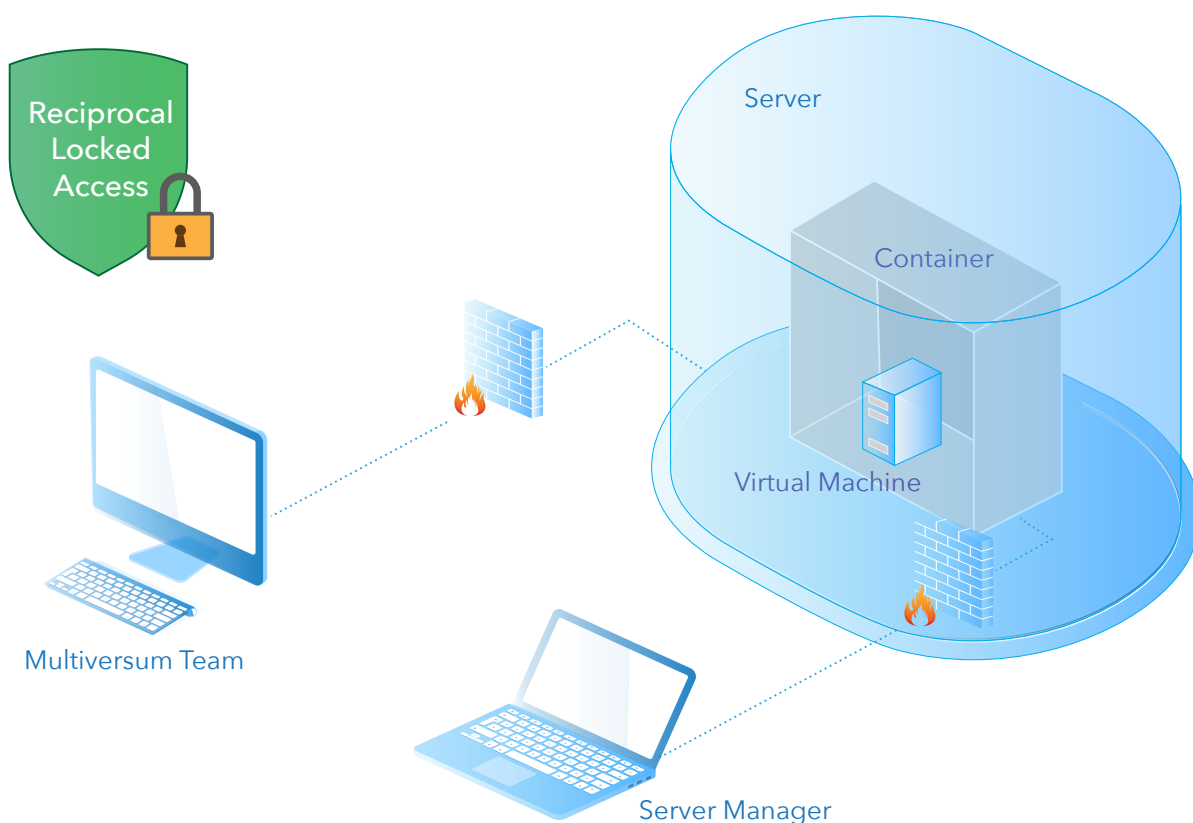
为了验证事务，对于每个节点上的特定事务，此计算的结果必须相同。此过程需要显著降低的计算能力，防止其他块验证解决方案（PoW, PoS, DpoS）典型的计算能力浪费，提供结构安全性，既不基于统计模型也不基于拜占庭共识模型，这在小型群集中非常脆弱。



## Access Denied

### 12. 双重访问锁（结构安全）

节点将分布在安全的虚拟容器中，主机操作员无法使用凭据，从而妨碍访问；因此安全性参考Linux Security14最佳实践，例如SELinux和/或其他软件包。同时，如果某人有访客机器凭证，他仍然无法访问它，无法访问运行该节点的主机。事实上，该节点通过双重访问锁来保护。



### 13. 反向访问拒绝（结构安全）

在第12点描述的访问锁定需要相互排除对主机操作员和最终拥有节点凭证的人的节点访问；这确保了不是由Multiversum直接管理的每个节点都是真实的并且是任何人无法访问的，基本上是自主的并且与外部人类干预隔离。除了操作系统和安全系统之外，还将在容器内分发三个基本组件：Multiversum Server编译代码，带有非对称密钥的证书以向Multiversum集群

进行身份验证，第11点中已介绍了一个组件，负责基于服务器代码散列，证书，挑战种子和交易数据。可能会实施其他可选安全技术，例如在编译阶段使用随机密码自动更新容器访问凭证，以防止任何人访问。这种机制可能被用于集群访问证书。

#### 14. 相互链确认（与其他区块链的互操作性）

Multiversum将研究外部链集成组件的可行性，能够存储其他区块链的状态（最终以标记交换），从而提供额外的验证和信任。也可以使用相同的技术让Multiversum将其自己的状态验证分享给其他区块链，即“外包”验证。将为此功能提供特定的接口，这也将需要在现有和将来的区块链实现中进行推广。这种功能将依赖无服务器组件，在容器编译后也可以访问该组件，以允许将适配器包含在其他链中。

#### 15. 与Java，Spring和Javascript集成

Multiversum将为Java，Javascript和其他主流语言提供分组在功能库中的高端接口，从而使我们的技术在企业和机构层面更容易被采用。带有Spring15等框架的集成模块也将被开发。这种类型的图书馆将促进Multiversum在专有解决方案中的整合，无论是在私人连锁店还是官方MainNet。



#### 16. ACID模型

Multiversum将满足ACID16范例；这个首字母缩略词表达了交易所需的逻辑属性。为了确保安全的事务模型，所实现的技术需要满足以下属性：原子性：事务在其执行过程中不可分割，其执行必须完整或为空，且不允许部分执行。一致性：任何事务都会将数据库从一个有效状态转移到另一个有效状态。持久数据必须根据所有定义的规则有效。隔离：每一笔交易都必须以独立的方式执行：交易的最终失败不会影响其他同时发生的交易。耐用性：也称为持久性，强制一旦事务被提交，结果不会由于任何原因而丢失（崩溃，错误，电力损失）。

#### 17. 交易模型

Multiversum将交易数据保存在交易18模型中，确保所涉及的多个子链上的所有数据或全部

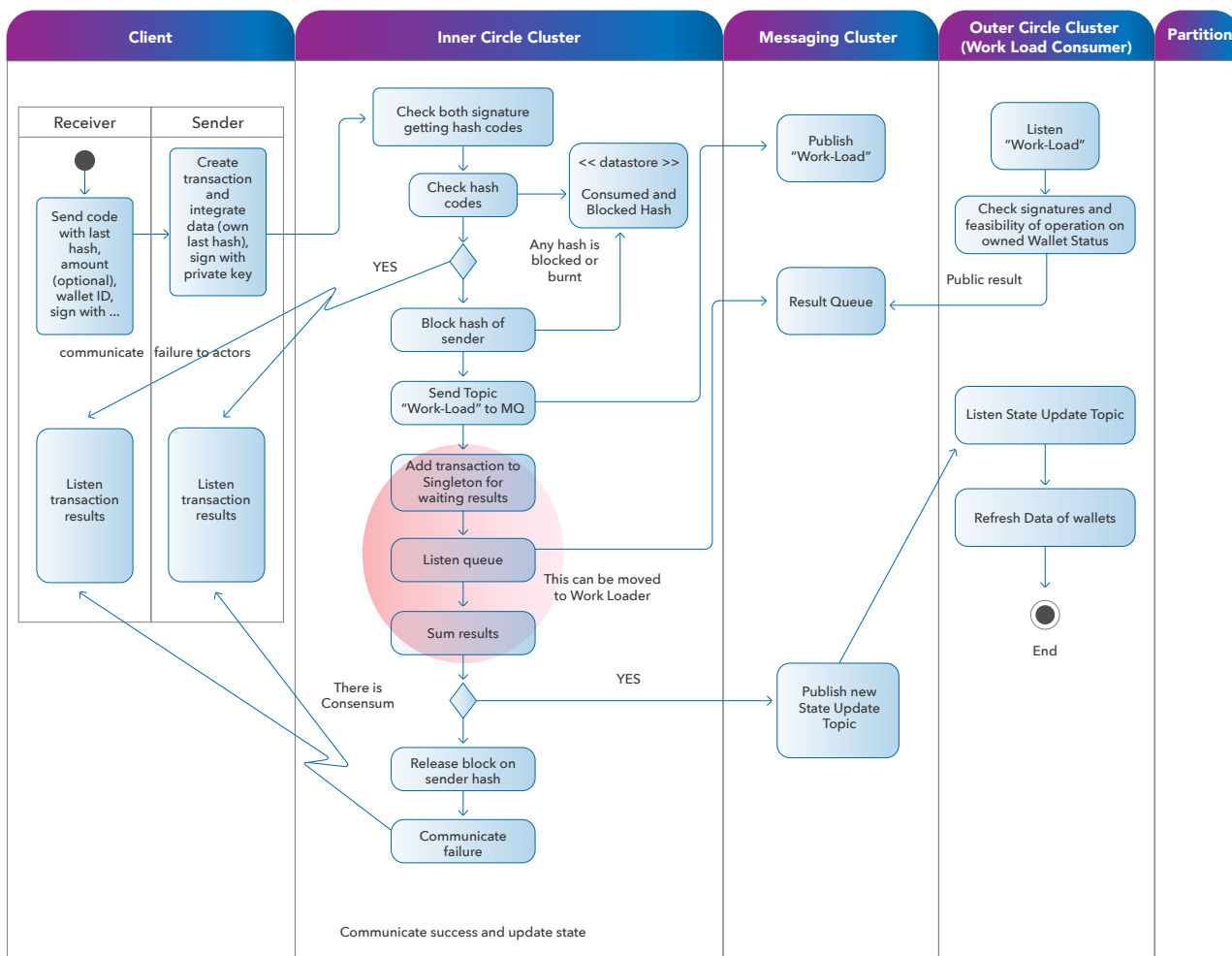
数据都不会被持久化，从而强化每个已执行交易和数据完整性的一致性。

### 18. 类似SQL

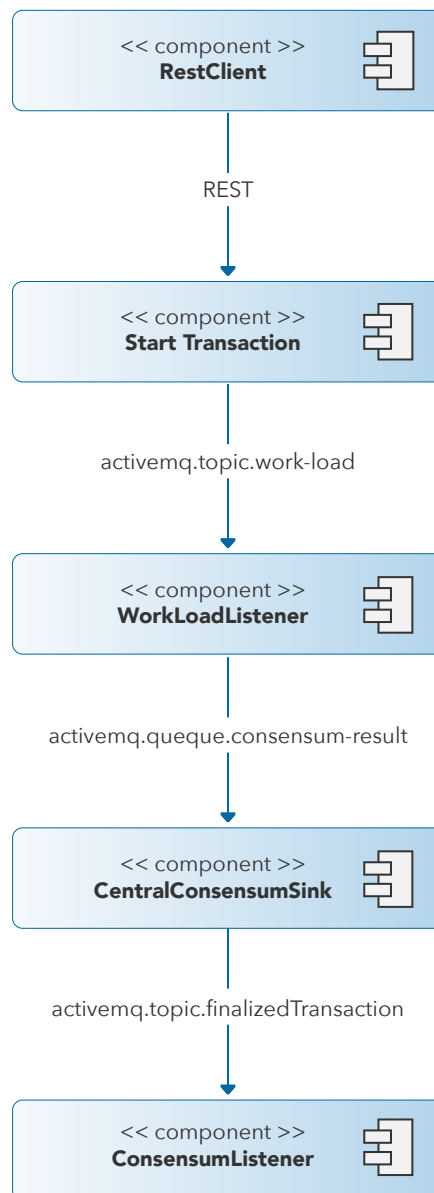
的语言为了简化基于我们的Crypto-Relational Database技术的应用程序的开发，并且相对于现有技术软化学习曲线，Multiversum将使用基于SQL的18语法来使用标准的持久存储函数（CRUD）。

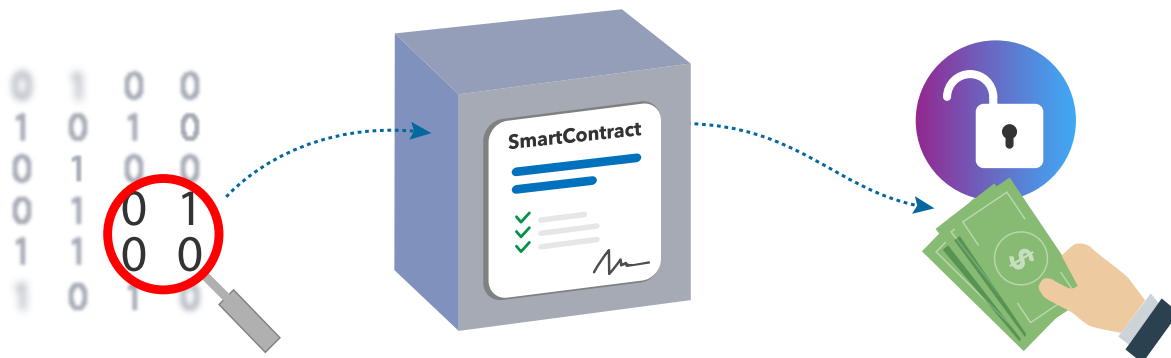
### 19. 全路径数据流量

交易的接受，控制，确认和持续过程通过下面的模式化和简化的过程进行：交易被发送到REST客户端，其必要的的数据用私钥签名；REST客户端将事务发送给协调集群的领导者节点：它将使用专有协调协议将节点拆分到节点之间；他们将对数据完整性，签名，资金可用性，已经使用的哈希，实际的钱包状态，被阻止的钱包或用户进行初步检查；来自发件人ID的任何附加操作现在都锁定在易失性存储器中，而特定的数据字段已完成（如先前要链接到的事务，时间戳和先前的散列）；交易通过必须定义的协议（AMQP用于试验，MQTT和其他定义）发送到主题消息队列并与工作节点并行分发；工作者节点验证他们对处理请求的兴趣（他们可能缺少必要的数据，已经忙碌且需要评估其他条件），并继续创建新的钱包状态，恢复先前链接事务的相关哈希并将其添加到事务记录中。完整性证明结果现在被添加；几艘以哈希计算；最佳案例完整路线场景结束。



Logic data flux  
Processtroom in detail





## 智能合约

Multiversum相信提出改进智能合约对公众的重要性，但在撰写本文时，除非研究范围有所调整，还没有决定探索这种可能性。因此，我们希望在Multiversum技术中纳入最适合我们需求的开源解决方案，并根据其许可模式作为参考实施。

## 基础设施

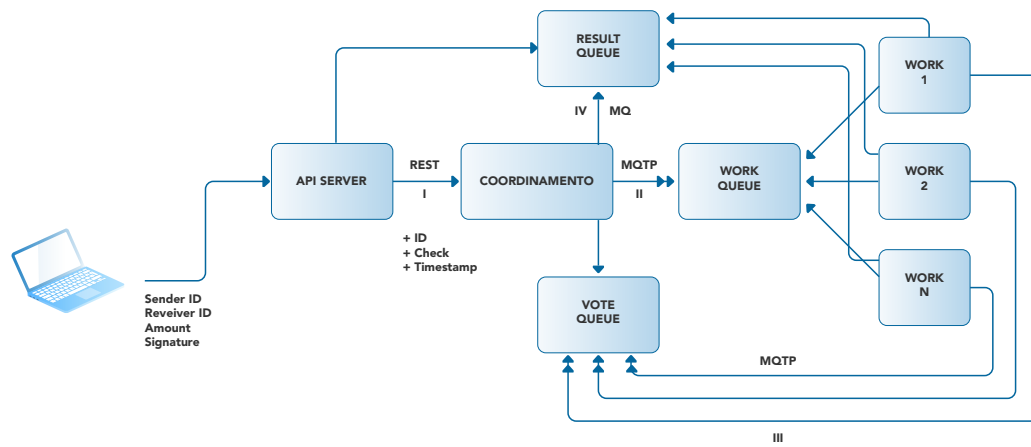
Multiversum基础设施旨在确保弹性和可达性。根据每个节点的技术规范，开发了能够自行选择其成员为特定角色的节点集群，其中包括：

- 计算能力
- 内存容量
- 互惠延迟
- 连锁数据完整性
- 机器可靠性
- 对完整性证明的怀疑

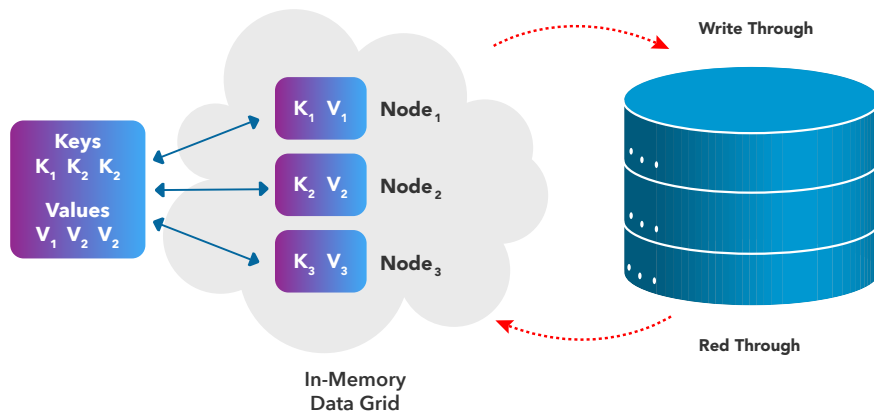
节点将有一个或多个角色：

- 客户端节点
- 协调节点
- 消息传递节点
- 工作节点
- 持久性节点
- 备份节点

每个可以提供有效证书的节点都能够注册到集群并获得角色。在一个或多个节点崩溃的情况下，集群将能够自动重新分配任务，优化角色。



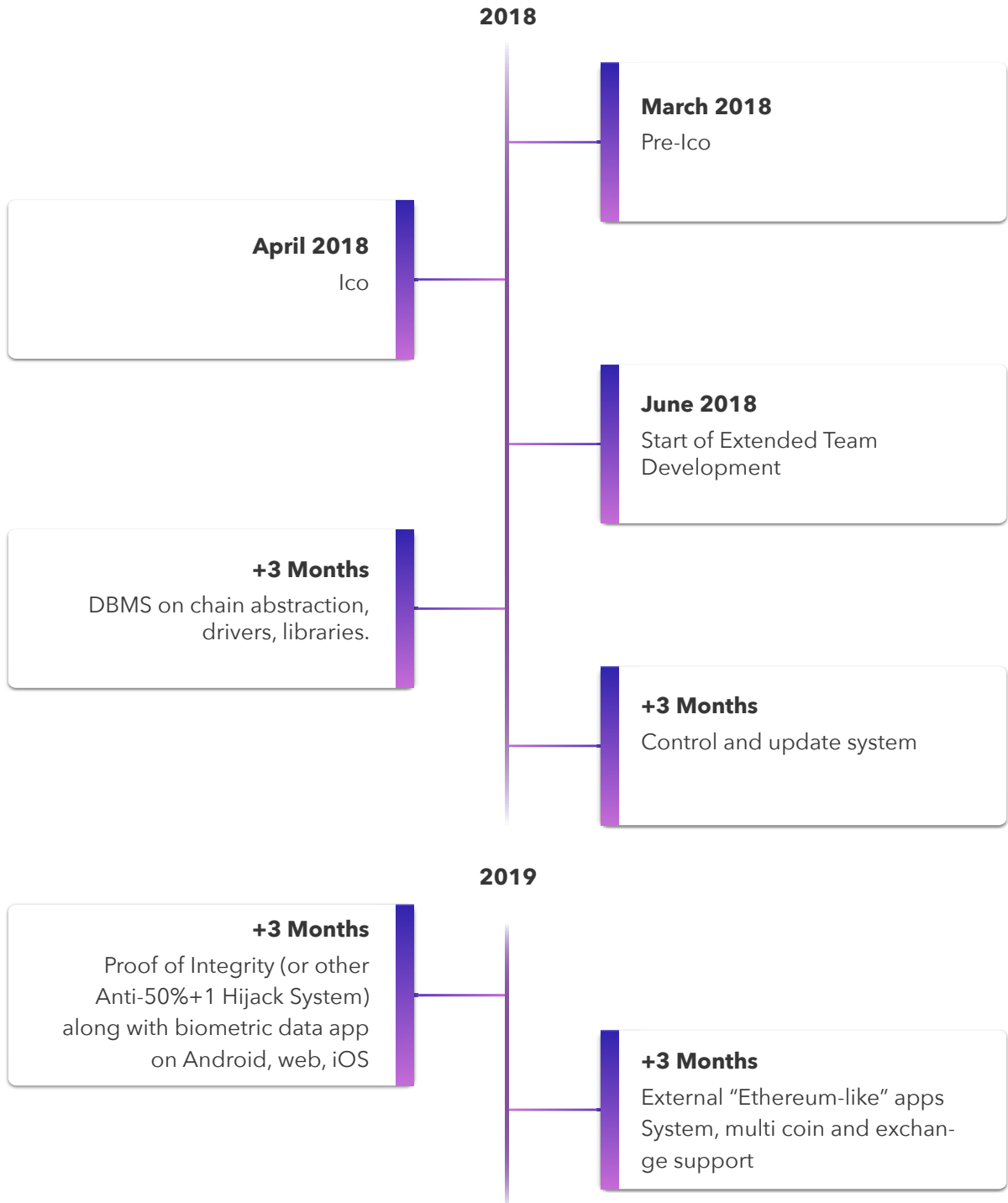
共享高速缓存内部JVM的组件将作为内存数据库存在，从而允许：在查看物理内存之前，读取直接在易失性存储器中执行数据读取查询。通过写入，在执行大容量插入之前将数据加载到易失性存储器中以保存数据，以便优化性能。



关于安全性的注意事项在开发过程中，“骇客悬赏”将提供给暴露漏洞的开发人员，并能够提供有效的修复。



# Technical Road Map



## References

- 1 [https://en.wikipedia.org/wiki/Scalability#Horizontal\\_and\\_vertical\\_scaling](https://en.wikipedia.org/wiki/Scalability#Horizontal_and_vertical_scaling)
- 2 [https://en.wikipedia.org/wiki/Proof-of-work\\_system](https://en.wikipedia.org/wiki/Proof-of-work_system)
- 3 <https://en.wikipedia.org/wiki/Proof-of-stake>
- 4 [https://en.wikipedia.org/wiki/Agile\\_software\\_development](https://en.wikipedia.org/wiki/Agile_software_development)
- 5 [https://en.wikipedia.org/wiki/Scope\\_\(project\\_management\)](https://en.wikipedia.org/wiki/Scope_(project_management))
- 6 [https://en.wikipedia.org/wiki/Shard\\_\(database\\_architecture\)](https://en.wikipedia.org/wiki/Shard_(database_architecture))
- 7 [https://en.wikipedia.org/wiki/High-availability\\_cluster](https://en.wikipedia.org/wiki/High-availability_cluster)
- 8 [https://en.wikipedia.org/wiki/Single\\_point\\_of\\_failure](https://en.wikipedia.org/wiki/Single_point_of_failure)
- 9 <https://en.wikipedia.org/wiki/Microservices>
- 10 [https://en.wikipedia.org/wiki/Serverless\\_computing](https://en.wikipedia.org/wiki/Serverless_computing)
- 11 <http://goo.gl/CVBzJd> Biometric Digital Signature Key Generation and Cryptography Communication Based on Fingerprint”
- 12 <https://en.wikipedia.org/wiki/ERC20>
- 13 [https://en.wikipedia.org/wiki/Byzantine\\_fault\\_tolerance](https://en.wikipedia.org/wiki/Byzantine_fault_tolerance)
- 14 [https://en.wikipedia.org/wiki/Security-Enhanced\\_Linux](https://en.wikipedia.org/wiki/Security-Enhanced_Linux)
- 15 [https://en.wikipedia.org/wiki/Spring\\_Framework](https://en.wikipedia.org/wiki/Spring_Framework)
- 16 <https://en.wikipedia.org/wiki/ACID>
- 17 [https://en.wikipedia.org/wiki/Models\\_of\\_communication#Transactional\\_Model](https://en.wikipedia.org/wiki/Models_of_communication#Transactional_Model)
- 18 <https://en.wikipedia.org/wiki/SQL>
- 19 [https://en.wikipedia.org/wiki/Message\\_queue#Standards\\_and\\_protocols](https://en.wikipedia.org/wiki/Message_queue#Standards_and_protocols)
- 20 [https://en.wikipedia.org/wiki/Smart\\_contract](https://en.wikipedia.org/wiki/Smart_contract)
- 21 <https://en.wikipedia.org/wiki/Reachability>
- 22 [https://en.wikipedia.org/wiki/Java\\_virtual\\_machine](https://en.wikipedia.org/wiki/Java_virtual_machine)

# 营销策略

在不断变化的IT市场运营，将相应地更新我们的策略，沟通技巧和公司使命，重点为利益相关者创造价值，并确保短期和长期管理逻辑之间的适当平衡。

我们计划的关键点是：

- 公司使命
- 业务目标
- 业务战略
- 业务活动组合



主要工具之一将是社交媒体营销：在社交网络上开展的活动，以提高品牌知名度，识别潜在消费者，建立联系并与客户建立有意义的关系。 我们的社交媒体策略师将开展多项行动，这些行动属于单一战略计划的一部分，首先是使用专用工具和社区开发的渠道管理和监控，重点是基于获得的结果的内容和互动以及策略效率评估。



MULTIVERSUM

HERE TO STAY