# 概述

市场上市值较大的公链起步较早，在新的发展需求下已经略显疲态，通用的做法是在已有的生态下试图用不同的技术弥补现有的不足。

受限于当年的思想和技术实现，修修补补，虽有局部的一些创新，但很难解决已有的问题，同时兼容已有的技术和社区用户等，创新解决方案越来越复杂。

新的技术创新的不断涌现，提供了新的机遇，具体分析现有的问题和用户痛点，结合新的解决问题思路，我们提出了自己的解决方案。

## 问题

公链作为金融基础设施，提供基本的数字货币价值支撑，同时承载大量的 dapp，主要面列以下问题。

**问题一**：TPS 问题

去中心化是底层公链的价值基础，网络传输延迟的限制，制约了区块大小和出块频率，分布式系统状态复制的固有时间复杂度等，造成底层公链 TPS 不高。

**问题二**：手续费问题

底层公链作为价值所在层，大量的 DApp 交易竞争有限的交易池大小，竞争有限的区块 gaslimit，共享竞价的结果是交易手续费水涨船高。

**问题三**：互操作性问题

每个区块链网络都有自己的安全和信任机制，有自己的价值。不同网络之间价值和信息的流动，受限于信任假设等，造成安全性问题，进一步加剧了互操作性问题。

## 发展格局

### 高性能公链生态

#### 1. DAG公链

以比特币为代表的区块链中, 除创世区块外, 每个区块有且只有一个前驱父区块和后继子区块, 区块之间构成单链.如果两个区块被同时提议, 将导致区块链出现分叉, 根据最长链原则, 最终只会保留一个区块在主链上, 而另一个被丢弃。这种结构使得针对区块链的账本操作只能够以串行化的方式进行。

但在基于DAG的分布式账本中, 每个账本的基本单元可以引用一至多个前驱单元, 且可以被一个或多个后继单元同时引用。这种基于DAG的账本操作支持并发进行, 多个节点可以同时向账本中新增交易或区块单元, 从而极大地提高了系统吞吐量。

代表性公链有Conflux，filecoin，avalanche

#### 2. PE公链

像以太坊这这样的智能合约公链，顺序性一次执行一个交易，所有其他的交易置于暂停状态，直到前面交易执行完成，再顺序执行后面交易。

PE并行处理，可以让交易在不同的处理核心分别处理，充分发挥硬件的并行特性，提升网路吞吐量，有效改进了交易确认的延迟。

目前 PE 实现都需要更改智能合约。比特币类的 UTXO 模型天然支持并行，但是需要扩充才能支持智能合约，比如 Cardano 的 extended UTXO model。

账户模型下，需要识别链的状态改变，避免避免不同交易之间的状态竞争。如 Solana 和 Sui 的要求交易明确声明它们修改的区块链状态部分。

**3. 共识创新公链**

活性和安全性前提下，在高性能的驱动下共识创新层出不穷。主流算法一般是基于 pos 和 bft 的结合，进一步流水线和并行处理提高共识效率。

PBFT normal case 需要 3 阶段达成共识：

| **normal case 的消息类型** | **格式** | **补充说明** |
|---|---|---|
| pre-prepare | <pre-prepare, view, txn, proof> | 主节点（Primary）广播预准备消息（Preprepare）到各副本节点（Replica） |
| prepare | <prepare, view, hash(txn)> | 该阶段是各个节点告诉其他节点我已经知道了这个消息，一旦某个节点收到了包含n-f 个prepare消息（我们将使用QC也就是Quorum Certificate来指代，下同）则进入prepared状态 |
| commit | <commit, view, hash(txn)> | 该阶段是各个节点以及知道其他节点知道了这个消息，一旦某个节点收到了n-f个commit消息（QC）则进入committed状态 |

View Change 也需要 3 各阶段：

| **normal case 的消息类型** | **格式** | **补充说明** |
|---|---|---|
| view-change | <view_change, v+1, txn, view-change-proof> | 从节点认为主节点有问题时，会向其它节点发送view-change消息，当前存活的节点编号最小的节点将成为新的主节点 |
| view-change-ack | <view-change-ack, v+1, (validTxn, validRound), prepare-proof> | 其他节点收到 veiw-change 消息，如果检查 view-change 消息有效，停止低于 v+1 的协议，并回复 ack 给leader |
| new-view | <pre-prepare, (txn, v+1), ack-proof> | Leader 等待 2f+1 个 ack 消息，并将2f+1 个 ack 消息打包记作 ack-proof，向其他节点广播并开启新的 veiw |

tendermint 共识算法，PBFT view change 的通信复杂度是 O(n3), tendermint 提出 lockedTxn 和 timeout 来解决 view-change 复杂度高的问题。但是一个区块的 commit 还是经过 pbft 的 3 个阶段，且只有上一次区块

commit 之后，下一个出下一个区块，进入新的 3 阶段 round。目前 cosmos 区块链采用 tendermint 共识算法。

hotstuff 共识算法，视图以单调递增的方式不断切换。每个视图内都有一个唯一的主节点负责提案、收集和转发消息并生成QC，同时一个视图只能出一个区块。hotstuff 中如果我们对某一区块达成了两轮共识，在更换主节点时便能确定，主节点只需要基于最新的两轮共识节点产生新节点就是安全的，这个想法降低了在视图切换时候的通信复杂度。同时 hotstuff 中各个阶段流程都高度的相似，进而提出各个阶段以流水线方式进行处理交易，进一步简化消息类型，降低消息复杂度，提高共识效率。目前 sui 采用了基于 hotstuff 的演变算法 Tusk 共识算法，Aptos 采用了基于HotStuff共识算法演进算法 Aptos BFT共识算法。

## 分层模块化生态

高性能的底层公链，也不可能面面俱到，解决所有应用问题。从以太坊为代表区块链技术的发展和迭代看，链的分层和角色分工越来越清晰，Layer1做见证和结算层，Layer2的定位是计算和执行层。基于这种思路演化出了不同技术实现。

**1. 状态通道**

两个用户之间开辟一个区块链之外的通道，只有第一步开辟通道和最后一步关闭通道需要在区块链上发送交易。其他的交易在通道内完成，但是这些交易是有效的区块链交易，一方作恶，另一方随时可以将最新的通道交易发到区块链上赎回金额并惩罚作恶方。状态通道不能与不属于通道的人交易，另一个缺点是它们要求通道双方锁定流动性以实例化通道以防止恶意交易对手不会真正支付承诺资金的情况，同时需要定期监管网络的人，监督是否作恶。状态通道主要应用比特币的闪电网络，基于以太坊的Raiden，connext等。

**2. sidechain**

sidechain 侧链是与主区块链挂钩的独立区块链。单向挂钩，将主区块链的代币发送到不可用的地址来烧掉代币，然后在侧链上铸造等价的代币。双向挂钩，将主链上代币锁定到固定的地址，在侧链上铸造出代币或者在侧链上销毁代币，在主链上解锁一定的主链代币。侧链拥有主链转移过来的价值的同时拜托主链的共识等约束，但是没有主链的庞大网络和安全性。侧链的主要应用有比特币的侧链 Liquid Network，以太坊的侧链 polygon pos。

**3. Plasma**

Plasma 本质上是一系列在主区块链之外运行的智能合约。以太坊是主干，每条 Plasma 链都是一个分支，每个分支都被视为具有自己的区块链历史和计算的区块链。「 根区块链 」（即以太坊区块链）使用称为「 欺诈证明 」 的东西来强制 Plasma 链中状态的有效性。Plasma 链的运营商提交 Plasma 链上发生的转移的 Merkle 根。虽然每个 Plasma 链都可以使用自己的机制来验证交易，但它仍然使用以太坊区块链作为最终的真相仲裁者。因为区块链中只保存了hash值，一个可能的攻击是 operator 不将证明数据发给用户。这样子用户既无法控告 operator，智能合约又无法区分是谁在作恶，也就是数据可用性问题。

**4. rollup**

Rollups的思路跟Plasma相似，除了交易数据的hash值，我们还将交易的必要数据保存到main-chain中。这样给人的感觉就是一种“压缩”，这样 operator 一次可以向 main-chain 提交几百甚至上千的压缩后的“交易数据”。这些state相当于保存在main-chain的log数据。转账人的签名等数据放到链下operator那里。如何保证数据的一致性呢？为了这个一致性问题，出现了两种不同的技术路线，分别是Optimistic rollup和ZK rollup。前者以乐观的方式使用“challenge-response”的方式让智能合约来做裁判来解决争端。它的一个缺点是，用户要等待一周时间才能将钱从rollup中取出，因为需要留时间给可能存在的争端。后者需要operator在本地生成零知识证明

（zkSNARK），智能合约通过验证该证明来判断数据是否一致，它的一个优点是，用户可以很快就能将钱从rollup中取出，但缺点是需要operator有很强的算力来生成zk证明。

### 多链互操作生态

跨链操作是通过连接相对独立的区块链系统来实现不同账本之间的可信互操作。

随着跨链操作向着纵深发展，实现链上与链下协同、信息网络与价值网络融合成为跨链发展的新需求，区块链预言机（oracle）作为将数据从区块链外传输到区块链上的工具开始引起研究者的重视并发挥其作用。

**1. 公证人机制**

通过选择产生一个或多个作为共同依赖的公证人，负责监听来自不同链中的请求，在对请求进行验证后，执行该请求事务，然后在目标链执行约定的操作，实现对请求事务的响应。公证人是跨链操作的关键，可由跨链各方共同指定，也可通过密码学技术确定。公证人通过预设的共识算法决定对接收到的请求是否执行，如果执行将达成共识。目前较常用的是多重签名公证人机制和分布式多重签名公证人机制。

多重签名公证人机制利用多重签名技术，由多位公证人同时对交易进行签名，当验证通过后跨链交易才能被确认。降低了单签名过程对公证人可靠性的依赖。

分布式多重签名公证人机制改进了签名方式：首先，利用密码学技术生成系统中唯一的私钥 k ，并将 k 拆分成 key1( part1) ， key2( part2) ， …， keyn ( partn) 共 n 份；然后，从网络中随机选取 n 个互不信任的节点作为公证人，将拆分后的私钥段分发给每个公证人；最后，由这 n个公证人共同签名完成交易的验证和确认。

**2. 中继**

当某一平行链中的节点需要进行跨链操作时，该节点首先发起跨链操作请求，该请求报文将发送到中继链中的验证节点，经验证通过后再由中继链发送到目标平行链。

中继方案基于轻客户端验证技术来实现，验证节点不需要跨链下载完整的账本，只需要执行能够实现轻客户端功能的智能合约，就能够跨链验证某笔交易是否存在。

**3. 预言机**

来自源链的智能合约调用预言机的消息路由器，该路由器将利用预言机的去中心化预言机网络安全地将消息发送到目标链，在目标链中，另一个消息路由器验证接收到的消息，通过验证后将其发送到目标智能合约，使智能合约能够通过跨链通信向其他区块链读写数据。

## 整体架构

### 子链钱包

1. 传统的插件钱包（metamask，coinbase），在玩游戏过程中，每次和合约交互都会调起插件钱包进行签名确认，大量的确认操作，严重验证影响游戏的体验。
2. 资产在layer1主链上面，在layer2链上的账户需要将原生代币兑换到layer2上才能支付发送合约调用交易的手续费。操作步骤较多，造成体验较差。

**解决方案**

1. 鉴于目前链的架构。layer2链上的游戏具有一定的生命周期。在短暂的生命周期内构造一个临时私钥代替工作私钥签名。由于临时私钥在生命周期结束之后会销毁，安全性可以弱化。在游戏合约运行期间。弱

化为不需要每次签名合约交易都弹出对话框进行确认。整个游戏运行期间在后台完成签名，实现用户的无感签名，带来游戏的丝滑体验。

2. 签名交易的手续费统一由游戏发行方提供，在代签名交易执行过程中将gas费用的扣款方更改为游戏的发行方。

**关键名词**

游戏发行方：游戏合约的部署者，需要在layer2链上由足够的费用代付游戏合约调用的手续费。

临时私钥：代替工作私钥签名的临时性私钥。具有一定的游戏阶段性。阶段开始时候生成，阶段结束销毁。只在阶段中具有有效性。

工作私钥：游戏资产需要从layer1转移到layer2上面。在layer2上面具有有效游戏资产的账户私钥，即游戏过程中实际动账的账户私钥。

**问题和平衡**

1. 游戏gas费代付，游戏玩家可能进行频繁重复操作，耗尽游戏发行方费用。

游戏发行方，对游戏玩家也就是 workAddress 有一定授信额度，此额度可以设置为无限也就是持续代付，也可以设置为有限值，也就是有限额度代付。

2. 避免游戏临时私钥权限过大，造成游戏资产不必要的损失。

提出抽象的阶段概念，游戏合约更具自己业务需要是否要在合约方法内进行阶段的判断。

**游戏合约基类**

```
contract GameBase {
    // 获取合约发行方
    function issuer() external view returns(address);

    // 获取授信额度
    function lineOfCredit() external view returns (uint256);
}
```