一、以太坊的升级路线图 M2SVPS

The Merge

在 The Merge 阶段，POW 共识机制将过度为
POS，信标链将合并在一起。为了便于理解，我们将以太坊结构简化为下图：

我们在这里先定义什么是分片：简单理解就是水平分割数据库以分散负载的过程

在转为 POS 后：区块提议者和区块验证者分离，POS
工作流程如下（根据上图理解）：

- 在 Rollup 上提交交易
- 验证者将交易添加到分片区块中
- 信标链选择验证者以提出新的块
- 其余的验证者组成随机的委员会并验证分片上的提议

提出区块和证明提议都需要在一个 slot 内完成，一般是 12s。每 32 个 slot
组成一个 epoch 周期，每个 epoch 将打乱验证者排序并重新选举委员会。

合并后，以太坊将为共识层实现提议者-构建者分离（PBS）。Vitalik 认为，所有区块链的终局都将是拥有中心化的区块生产和去中心化的区块验证。由于分片后的以太坊区块数据非常密集，出于对数据可用性的高要求，区块生产的中心化是必要的。同时，必须有一种能够维护一个去中心化的验证者集的方案，它可以验证区块并执行数据可用性采样。

矿工和区块验证分离。

矿工进行区块的构建，然后将区块提交给验证者。向验证者出价投标选择自己的区块，然后验证者投票来决定区块是否有效。

分片是一种分区方式，可以在 P2P 网络中分散计算任务和存储工作负载，经过这种处理方式，每个节点不用负责处理整个网络的交易负载，只需要维护与其分区（或分片）相关的信息就可以了。每个分片都有自己的验证者网络或者节点网络。

分片的安全性问题：例如整个网络有 10 条分片链，破坏整个网络需要 51% 的算力，那么破坏单个分片只需要 5.1% 的算力。因此后续的改进就包含了一个 SSF 算法，这个算法能够有效防止 51% 的算力攻击。根据 vitalik 总结，转向 SSF 是一个多年的路线图，即使目前做了大量工作，它也将是以太坊较晚推行的重大变化之一，并且远在以太坊的 PoS 证明机制、分片和 Verkle 树完全推出之后。

信标链，负责生成随机数，将节点分配给分片，捕捉单个分片的快照和其他各种功能，负责完成分片间的通信，协调网络的同步。

信标链的执行步骤如下：

- 区块生产者对区块头和出价一起进行承诺。
- 信标链上的区块者（验证者）选择获胜的区块头和投标，无论区块打包者是否最终生成区块体，都将无条件获得中标费。
- Committee（验证者中随机选取）投票确认获得的区块头。
- 区块打包者披露区块体。

The Surge

该路线的主要目标是推动以 Rollup 为中心的扩容。Surge 指的是添加了以太坊分片，这是一种扩容解决方案，
以太坊基金会声称：该解决方案将进一步启用低 gas 费的二层区块链，降低 rollup 或捆绑交易的成本，并使用户更容易操作保护以太坊网络的节点。

该图仍然可以通过以下简图来理解：

以 zkrollup 运行原理为例：在 zkrollup 中分为排序器（sequencer）和聚合器（aggregator），排序器负责将用户交易排序，并且将其打包成批次（batch），发送给聚合器。聚合器执行交易，从前状态根（prev state root），生成后状态根（post state root），然后生成证明（proof），聚合器最后将前状态根、后状态根、交易数据，证明发送到 L1 上的合约，合约负责校验证明是否有效，交易数据存储在 calldata 内。Zkrollup 数据可用性可以让任何人能够根据链上存储的交易数据，还原出账户的全局状态。

但是使用 calldata 的费用非常昂贵，因此整个 EIP-4844 协议（可能随时改变）提出了将交易区块的大小改为 1～2MB，为未来的 rollup 与数据分片打下坚实基础。目前以太坊的区块大小大约是 60KB ~100KB，以 EIP-4844 为例，大概可以提升 10～34x 的区块大小极限。该区块格式被称为 blob（也可以称为数据分片 data shard）。

The Scourge

该阶段 Scourge 是路线图的补充，主要用于解决 MEV 的问题。那什么是 MEV？

MEV 全名 Miner Extractable Value / Maximal Extractable Value，这一概念最先应用在工作量证明的背景下，最初被称为「矿工可提取价值（Miner Extractable Value）」。这是因为在工作量证明中，矿工掌握了交易的包含、排除和顺序等角色能力。然而，在通过合并过渡为权益证明后，验证者将负责这些角色，而挖矿将不再适用（此处介绍的价值提取方法在这次过渡后仍将保留，因此需要更改名称）。为了继续使用相同的首字母缩写词以便确保延续性，同时保持相同基本含义，现在使用「最大可提取价值（Maximal Extractable Value）」作为更具包容性的替代词。

套利空间包括：

通过压缩存储空间，来获得 gas 费用的价差；

裁判员抢跑：广泛的搜索 mempool 上的交易，机器在本地执行计算，看看是否会有利可图，如果有则用自己的地址发起相同交易，并且使用更高的 gas 费；

寻找清算目标：机器人竞相以最快的速度解析区块链数据，从而确定哪些借款人可以被清算，然后成为第一个提交清算交易并自行收取清算费的人。

夹心交易：搜索人会监视内存池内 DEX 的大额交易。例如，有人想要在 Uniswap 上使用 DAI 购买 10,000 UNI。这类大额交易会对 UNI / DAI 对产生重大的影响，可能会显著提高 UNI 相对于 DAI 的价格。搜索人可以计算该大额交易对 UNI / DAI 对的大致价格影响，并在大额交易之前立即执行最优买单，低价买入 UNI，然后在大额交易之后立即执行卖单，以大额订单造成的更高价格卖出。

MEV 的缺陷：

某些形式的 MEV，如夹心交易，会导致用户的体验明显变差。被夹在中间的用户面临更高的滑点和更差的交易执行。在网络层，一般的抢跑者和他们经常参与的矿工费拍卖（当两个或更多的先行者通过逐步提高自己交易的矿工费，从而使他们的交易被打包到下一个区块），导致网络拥堵和试图运行正常交易的其他人的高矿工费。除了区块内发生的，MEV 也可能会在区块间产生有害的影响。如果一个区块中可用的 MEV 大大超过了标准区块的奖励，矿工可能会被激励去重新开采区块并为自己捕获 MEV，进而导致区块链的重新组织和共识的不稳定。

大部分 MEV 是由称为「搜索者」的独立网络参与者提取的。搜索者在区块链数据上运行复杂的算法来检测盈利的 MEV 机会，并且有机器人自动将这些盈利交易提交到网络。以太坊上的 MEV 问题涉及使用机器人来利用网络交易，导致拥塞和高额费用。

The Verge

Verge 将实现「 Verkle 树」（一种数学证明）和「无状态客户端」。这些技术升级将允许用户成为网络验证者，而无需在他们的机器上存储大量数据。这也是围绕 rollup 扩容的步骤之一，前文提到过 zk rollup
的简易工作原理，聚合器提交了证明，layer 1 上的验证合约只需要验证 blob 内的 KZG 承诺和生成的证明即可。这里简单介绍一下 KZG
承诺，就是确保所有的交易都被包含了进来。因为 rollup
可以提交部分交易，生成证明，如果使用
KZG，那么所有的交易都会被确保包含进来生成证明。

The Verge 就是确保验证非常简单，只需要下载 N
个字节数据，执行基本的计算就可以验证 rollup 提交的证明。

值得一提的是，ZK rollup 有许多种方案，stark、snark 或者 bulletproof
等。各个方案对于证明和验证的方式都不尽相同，因此出现了权衡。SNARKs
目前比 STARKs 技术更易上手，技术也更完善，因此许多项目刚开始都是使用 SNARKs，但是随着 STARKs 技术上的迭代后，最终会逐渐转向抗量子攻击的 STARKs。虽然以太坊为了与 rollup 适配，EIP-4844 主要改进之一是交易格式 blob，扩大了区块容量，但目前所有的零知识证明的主要瓶颈仍然在于自身证明算法，一方面通过改进算法来解决证明问题，另一方面通过堆叠硬件来提高证明效率，也因此衍生了 ZK 挖矿赛道。有兴趣的可以转到这篇文章。

The Purge

The Purge 将减少在硬盘驱动器上存储 ETH 所需的空间量，试图简化以太坊协议并且不需要节点存储历史。这样可以极大的提升网络的带宽。

EIP-4444：

客户端必须停止在 P2P 层上提供超过一年的历史标头、正文和 recipient。客户端可以在本地修剪这些历史数据。保存以太坊的历史是根本，相信有各种带外方式来实现这一点。历史数据可以通过 torrent 磁力链接或 IPFS

等网络打包和共享。此外，Portal Network 或 The Graph 等系统可用于获取历史数据。客户端应允许导入和导出历史数据。客户端可以提供获取 / 验证数据并自动导入它们的脚本。

The Splurge

该路线主要是一些零碎的优化修复，如账户抽象、EVM 优化以及随机数方案 VDF 等。

这里提到的账户抽象（Account Abstraction，AA）一直都是 ZK 系 Layer 2 想要首先实现的目标。那什么是账户抽象？在实现账户抽象之后，一个智能合约账户也可以主动发起交易，而无需依赖「元交易」的机制（这在 EIP-4844 里被提出）。

在以太坊内，账户分为合约账户和外部账户。目前的以太坊的事务类型只有一种，必须由外部地址发起，合约地址无法主动发起事务。因此，任何合约自身状态的改变，必须依赖于一个外部地址发起的事务，无论是一个多重签名账户，还是混币器，或是任何智能合约的配置变更，都需要由至少一个外部账户触发。

无论使用以太坊上的什么应用，用户都必须持有以太坊（并承担以太坊价格波动的风险）。其次，用户需要处理复杂的费用逻辑，gas price，gas limit，事务阻塞，这些概念对用户来说过于复杂。许多区块链钱包或应用试图通过产品优化提高用户体验，但效果甚微。

以账户为中心的方案的目标是为用户创建一个基于智能合约管理的账户。实现账户抽象后的好处是：

现在的合约可以持有 ETH，直接提交包含所有签名的事务，用户不一定需要为交易支付 gas 费用，完全取决于项目。

由于实现了自定义密码学，因此未来不会强制要求使用 ESCDA 椭圆曲线来进行签名，未来一台手机的指纹识别、面部识别、生物识别等技术均可以作为签名方式。

从而显著改善了用户与以太坊的交互体验。

二、以太坊的模块化

整个以太坊目前已经出现了模块化的趋势，执行层就是由 Layer 2 负责（如 arbitrum、zksync、starknet、polygon zkevm 等）。他们负责执行 L2 上用户的交易，并且提交证明。Layer 2 一般使用的是 OP 技术 / ZK 技术，ZK 技术在理论上 TPS 是远高于 OP 的，目前大量生态在 OP 系，但是未来，随着 ZK 技术的完善，会有越来越多应用迁移到 ZK 系。该部分是对路线图的详细描述与补充为什么和怎么样。

目前以太坊只是将执行层剥离开来，实际上，其它层级仍然混作一谈。在 celestia 的愿景中，执行层只进行两件事：对单笔交易而言，执行交易并发生状态更改；对同批次的交易而言，计算该批次的状态根。当前以太坊执行层的一部分工作分给了 Rollup，即我们熟知的 StarkNet 、zkSync 、Arbitrum 和 Optimism。

现在无论是 optimism、polygon、starknet、zksync 等都往模块化的道路上探索。

Optimism 提出了 bedrock / op stack，polygon 也在研发 polygon avail 作为数据可用性层，supernets 则用以简化链的创建与共享验证者集。

结算层：可以理解为主链上的 Rollup 合约验证上文提到的前状态根、后状态根、证明的有效性（zkRollup）或欺诈证明（Optimistic Rollup）的过程。

共识层：无论采用 PoW 、PoS 或其他共识算法，总之共识层是为了在分布式系统中对某件事达成一致，即对状态转换的有效性达成共识（前状态根经过计算转换成的后状态根）。在模块化的语境下，结算层和共识层的含义有些相近，故也有一些研究者把结算层和共识层统一起来。

数据可用性层：确保将交易数据完整上传到数据可用性层，验证节点能够通过该层的数据复现所有状态变更。

这里需要辨析的是数据可用性和数据存储的区别：

数据可用性与数据存储是明显不同的，前者关注的是最新区块发布的数据是否可用，而后者则是涉及安全地存储数据并保证在需要时可以访问它。

1、结算层上的各种 Rollup

从结算层看，目前认为 rollup 的焦点在 ZK 系。如果通过 ZK 系的 rollup 来改进 ZK 证明系统的大小、gas
消耗、成本，再结合递归和并行处理的话就能够极大的拓展其
TPS。那么我们就先从 ZK rollup 开始。

随着以太坊扩容之路的发展，零知识证明（Zero Knowledge
Proof，ZKP）技术被 Vitalik 认为是有望成为扩容之战的终局的方案。

ZKP 的本质是让某人证明他们知道或所拥有某些东西。例如，我可以证明我拥有开门的钥匙，而无须将钥匙拿出来。证明知道某个账户的密码，而无需输入密码并冒着被暴露的风险，这项技术对个人隐私、加密、企业甚至核裁军都有影响。通过姚氏百万富翁问题修改版本来加深理解：这个问题讨论了两个百万富翁爱丽丝和鲍勃，他们想在不透露实际财富的情况下知道他们中的哪一个更富有。

假设公寓每月的租金为 1000
美元，若要符合出租人选的标准，则至少要支付一个月租金的 40
倍。那么我们（租客）需要证明我们的年收入要有 4 万美元以上才行。但房主不想我们找到漏洞，因此选择不公布具体的租金，他的目的是测试我们是否符合标准，而答案仅仅是符合或者不符合，而不对具体金额负责。

现在有十个盒子，以 1 万美元为增量，标记为 10～100k
美元。每个都有一个钥匙和一个插槽。房主带着盒子走进房间毁掉 9
把钥匙，拿走标有 40k 美元盒子的钥匙。

租客年薪达到 7.5 万美元，银行代理人监督开具资产证明的文件，不写明具体资金，这个文件的本质是银行的资产声明可验证索赔文件。随后我们将该文件投入
10k～70k 的箱子中。那么房主使用 40k
的钥匙打开箱子，看到里面的可验证索赔文件时，则判定该租客符合标准。

这里面涉及到的点包括，声明人（银行）出具资产达标证明，验证者（房主）通过钥匙验证租客是否具有资格。再次强调，验证结果只有两个选择——具有资格和不具有资格，并不会也不能对租客具体资产数额作出要求。

我们仍然可以用下图作为理解，交易在 layer 2 上执行，在分片提交交易。layer 2 一般采用 rollup 的形式，也就是在 layer 2 上将多笔交易打包成一个批次来处理事务，然后再提交给 layer 1 的 rollup 智能合约。这里包含新旧状态根，layer 1 上的合约会验证两个状态根是否匹配，如果匹配那么主链上的旧状态根就会更换为新状态根。那如何验证批次处理后得到的状态根是正确的呢，这里就衍生出了 optimistic rollup 和 zk rollup。分别使用欺诈证明和 zk 技术进行交易的确认以及状态根的验证。

这里的 layer 2（rollup）就相当于上文例子中的声明人（银行），其打包操作就是这份声明操作，并不会对具体数额作出声明，而是确认是否达到标准。打包后提交给 layer 1 的就是这份可索赔的声明文件。验证新旧状态根就是房主通过钥匙验证自己期望的租客经济实力是否达标。状态根验证问题就是银行提交的声明，该如何进行声明才能使问题可信。

基于 optimistic 也就是欺诈证明的 rollup 来说，主链的 Rollup 合约记录了该 Rollup 内部状态根变更的完整记录，以及每个（触发状态根变更的）批次处理的哈希值。如果有人发现某个批次处理对应的新状态根是错误的，他们可以在主链上发布一个证明，证明该批次处理生成的新状态根是错误的。合约校验该证明，如果校验通过则对该批次处理之后的所有批次处理交易全部回滚。

这里的验证方式相当于声明人（银行）提交了可验证资产声明文件，然后将资产文件全部公开到链上，并且数据也要公开到链上，其他挑战者根据原始数据进行计算看可验证的资产文件是否存在错误或伪造的情况，如果有问题，则提出挑战，挑战成功则向银行索赔。这里最重要的问题就是需要预留时间给挑战者收集数据并且验证该份文件的真实性。

对于使用零知识证明（Zero Knowledge Proof，ZKP）技术的 Rollup 来说，其每个批次处理中包含一个称为 ZK-SNARK 的密码学证明。银行通过密码学证明技术来生成资产声明文件。这样就不需要预留时间给挑战者，从而也就没有挑战者这一角色存在了。

2、现在 ZK 系 Rollup 不及预期的原因

目前 polygon 系的 hermez 已经发布，zksync dev 主网、starknet 主网也已经上线。但是他们的交易速度似乎与我们理论上还相差过大，特别是 starknet 的用户能明显感知到，其主网速度慢的令人惊讶。究其原因还是在于零知识证明技术生成证明难度仍然很大，成本开销仍然很高，还有需要对以太坊的兼容性和 zkevm 性能上的权衡。Polygon 团队也承认：「Polygon zkEVM 的测试网版本也具有有限的吞吐能力，这意味着它远不是作为优化扩展机器的最终形式。」

3、数据可用性层

以太坊的抽象执行步骤如下所示：

在以太坊的去中心化过程中，我们也可以在 The Merge 路线图上看到——去中心化验证者。其中最主要的就是实现客户端的多样性以及降低机器的入门门槛，增加验证者的人数。因此有些机器不达标的验证者想要参与网络，就可以使用轻客户端，轻节点的运行原理是通过临近的全节点索要区块头，轻节点只需要下载和验证区块头即可。如果轻节点不参与进来，那么所有的交易都需要全节点去执行验证，因此全节点需要下载和验证区块中的每笔交易，同时随着交易量的增多，全节点承压也越来越大，因此节点网络逐渐倾向于高性能、中心化。

但是这里的问题是，恶意的全节点可以给予缺失 / 无效的区块头，但是轻节点没办法证伪，对此问题有两种办法，刚开始是使用欺诈证明，需要一个可信的全节点来监控区块的有效性，在发现无效区块后构造一个欺诈证明，在一段时间内未收到欺诈证明则判定为有效区块头。

但是这里明显需要一个可信的全节点，即需要可信设置或者诚实假设。但是区块生产者能够隐藏部分交易，欺诈证明就明显失效，因为诚实的节点，也依赖于区块生产者的数据，若数据本身就被隐藏，那么可信节点就认为提交的数据是全部数据，那么自然也不会生成欺诈证明。

Mustarfa AI-Bassam 和 Vitalik 在合著的论文中提出了新的解决方案——纠删码。采用纠删码来解决数据可用性的问题，比如 celestia，polygon avail

均采用的是 reed-solomon
纠删码。但是如何确保传输的数据是完整的数据呢，结合 KZG 承诺 /
欺诈证明即可。

在 KZG 承诺 / 欺诈证明中，能够确保区块生产者发布完整的数据，不会隐藏交易，然后将数据通过纠删码进行编码，再通过数据可用性采样，这样就可以让轻节点正确地验证数据。

Rollup 内聚合器提交的数据都是以 calldata 形式存储在链上的，这是因为
calldata 数据相对于其它存储区域更便宜。

Calldata cost in gas = Transaction size 16 gas per byte

每笔交易主要的开销在 calldata 成本，因为在链上存储费用极其昂贵，该部分占到 rollup 成本的 80%～95% 之多。

由于这个问题，我们才提出了 EIP-4844 的新交易格式
blob，扩大区块容量，降低提交到链上所需的 gas 费。

4、数据可用性层的链上与链下

那么如何解决链上数据昂贵的问题呢？有以下几种方法：

首先是压缩上传到 L1 的 calldata 数据大小，这方面已经有了许多的优化。

其次是降低在链上存放数据的成本，通过以太坊的 proto-danksharding 和
danksharding 来为 rollup
提供「大区块」，更大的数据可用性空间，采用纠删码和 KZG
承诺来解决轻节点的问题。如 EIP-4844。

第三个是，把数据可用性放在链下，这部分的通用方案包括，celestia / polygon avail 等。

通过数据可用性存放的位置，我们将其分为下图所示：

Validium 的方案：将数据可用性放在链下，那么这些交易数据就由中心化的运营商来维护，用户就需要可信设置，但成本会很低，但同时安全性几乎没有。之后 starkex 和 arbitrum nova 都提出成立 DAC 来负责交易数据的存储。DAC 成员都是知名且在法律管辖区内的个人或组织，信任假设是他们不会串通和作恶。

Zkporter 提出 guardians（zksync token 持有者）来质押维护数据可用性，如果发生了数据可用性故障，那么质押的资金将被罚没。

Volition 则是用户自己选择链上 / 链下数据可用性，根据需求，在安全与成本之间选择。

这时候，celestia 和 polygon avail 就出现了。如果 validium 有链下数据可用性的需求，又害怕去中心化程度低，从而引发类似跨链桥的私钥攻击，那么去中心化的通用 DA 方案则可以解决这个问题。Celestia 和 polygon avail 通过成为一条单独的链，来为 validium 提供链下 DA 的解决方案。但是通过单独的链，虽然提升的安全性，但相应会提高成本。

Rollup 的拓展实际上有两部分，一部分是聚合器的执行速度，另一方面则需要数据可用层的配合，目前聚合器是中心化的服务器来运行，假设交易执行的速度能达到无限大的程度，那么主要拓展困境在于其受到底层数据可用性解决方案的数据吞吐量的影响。如果 rollup 要最大化其交易吞吐量，则如何最大化数据可用性解决方案的数据空间吞吐量是至关重要的。

再回到开头，使用 KZG 承诺或者欺诈证明来确保数据的完整性，通过纠删码来拓展交易数据帮助轻节点进行数据可用性采样，进一步确保轻节点能够正确验证数据。

也许你也想问，到底 KZG 承诺是如何运行来确保其数据的完整性的呢？或许可以稍微解答一下：

KZG 承诺：证明多项式在特定位置的值与指定的数值一致。

KZG 承诺无非就是多项式承诺中的一种，能够在不给定具体消息的情况下验证消息。大概流程如下：

将数据通过纠删码化为多项式，将其拓展。使用 KZG 承诺确保我们的拓展是有效的，且原数据是有效的。然后利用拓展可以 reconstruct 数据，最后进行数据可用性采样。

提交者（commiter）生成承诺（commitment），将其与消息绑定。

将绑定后的消息传送给验证者，这里的 communication 方案就关系到证明规模（proof size）的大小。

验证者（verifier），带入有限域的多个值验证是否仍然等于 a（这就是可用性采样的过程），基本原理就是验证次数越多那么正确的概率就越高。

Celestia 要求验证者下载整个区块，现在的 danksharding 则利用数据可用性采样技术。

由于区块存在部分可用的情况，因此任何时候我们都需要在重构区块的时候保证同步。在区块确实部分可用时，节点之间通信，将区块拼凑出来。

KZG 承诺和数据欺诈证明的比较：

可以看到 KZG 承诺能确保拓展和数据是正确的，而欺诈证明引入第三方进行观察。最明显的区别是，欺诈证明需要一个时间间隔来给观察者进行反应，然后再报告欺诈，这时候需要满足节点直接的同步，从而整个网络能够及时收到欺诈证明。KZG 则明显的比欺诈证明更快，其使用数学方法来确保数据的正确，而不需要一个等待时间。

它能够证明数据以及其拓展是正确的。但是由于一维 KZG 承诺需要耗费更大的资源，因此以太坊选择二维 KZG 承诺。

比如 100 行 100 列，那就是 100,00 个份额（shares）。但每采样一次，都不是万分之一的保证。那么扩展四倍意味着在整个份额中至少要有 1/4 的份额不可用，你才可能抽到一个不可用的份额，才表示真正不可用，因为恢复不出来。只有在

1/4 不可用的情况下才恢复不出来，才是真正有效的发现错误，所以抽一次的概率大概是 1/4。抽十多次，十五次，可以达到 99% 的可靠性保证。现在在 15–20 次的范围之内做选择。

5、EIP-4844（Proto-Danksharding）

在 proto-danksharding 实现中，所有验证者和用户仍然必须直接验证完整数据的可用性。

Proto-danksharding 引入的主要特征是新的交易类型，我们称之为携带 blob 的交易。携带 blob 的事务类似于常规事务，不同之处在于它还携带一个称为 blob 的额外数据。Blob 非常大（~125 kB），并且比类似数量的调用数据便宜得多。但是，这些 blob 无法从 EVM 访问（只有对 blob 的承诺）。并且 blob 由共识层（信标链）而不是执行层存储。这里其实就是数据分片概念逐渐成型的开始。

因为验证者和客户端仍然需要下载完整的 blob 内容，所以 proto-danksharding 中的数据带宽目标为每个插槽 1 MB，而不是完整的 16 MB。然而，由于这些数据没有与现有以太坊交易的 gas 使用量竞争，因此仍然有很大的可扩展性收益。

尽管实现全分片（使用数据可用性采样等）是一项复杂的任务，并且在 proto-danksharding 之后仍然是一项复杂的任务，但这种复杂性包含在共识层中。一旦 proto-danksharding 推出，执行层客户端团队、rollup 开发人员和用户不需要做进一步的工作来完成向全分片的过渡。Proto-danksharding 还将 blob 数据与 calldata 分离，使客户端更容易在更短的时间内存储 blob 数据。

值得注意的是，所有工作都是由共识层更改，不需要执行客户端团队、用户或 Rollup 开发人员的任何额外工作。

EIP-4488 和 proto-danksharding 都导致每个插槽（12 秒）的长期最大使用量约为 1 MB。这相当于每年大约 2.5 TB，远高于以太坊今天所需的增长率。

在 EIP-4488 的情况下，解决此问题需要历史记录到期提案 EIP-4444

（路线图部分有提及），其中不再要求客户端存储超过某个时间段的历史记录。

6、数据分片

在这里，将尽可能多的以小白的视角讲清楚以太坊扩容过程中大家都在讨论的问题。所以我们回到分片，再次强调一下对于分片的片面概念：简单理解就是水平分割数据库以分散负载的过程。

在这里，我们的数据分片有一个很重要的问题就是，在 PBS 中（提议者与区块构建者分离，路线图 The Merge 处有提及），在分片中，每个节点群只处理该分片内的交易，交易在分片间会相对独立，那么 AB 两用户处于不同分片上，相互转账该如何处理呢？那这里就需要很好的跨片通信的能力。

过去的方式是数据可用性层分片，每个分片都有独立的提议者（proposers）和委员会（committee）。在验证者集中，每个验证者轮流验证分片的数据，他们将数据全部下载下来进行验证。

缺点是：

需要严密的同步技术来保证验证者之间能够在一个 slot 内同步。

验证者需要收集所有的 committee 的投票，这里也会出现延迟。

而且验证者完全下载数据对其压力也很大。

第二种方法是放弃完全的数据验证，而是采用数据可用性采样的方法（该方法在 The Surge 后期实现）。这里又分为两种随机采样方式，1）区块随机采样，对部分数据分片采样，如果验证通过后，验证者进行签名。但是这里的问题是，可能会出现遗漏交易的情况。2）通过纠删码将数据重新解释为多项式，再利用特定条件下多项式能够恢复数据的特点，来确保数据的完整可用性。

「分片" 关键就是验证者不负责下载所有数据，而这就是为什么 Proto-danksharding 不被认为是 "分片的" 的原因（尽管它的名字里有 "分片 sharding"）。Proto-danksharding 要求每个验证者完整地下载所有分片 blob 来验证它们的可用性；Danksharding

则随后将引入采样，单个验证者只需下载分片 blob 的片段。

三、以太坊的未来之 Layer 3

被视为以太坊拓展未来的 ZK 系 Layer 2 如 zksync、starknet 都纷纷提出了 Layer 3 的概念。简单理解就是 Layer 2 的 Layer 2。

以太坊上高昂的交易成本正在推动它（L3）成为 L2
的结算层。相信在不久的将来，由于交易成本显着降低、对 DeFi
工具的支持不断增加以及 L2 提供的流动性增加，最终用户将在 L2
上进行大部分活动，而以太坊逐渐成为结算层。

L2 通过降低每笔交易的 gas 成本和提高交易率来提高可扩展性。同时，L2s 保留了去中心化、通用逻辑和可组合性的好处。但是，某些应用程序需要特定的定制，这可能更好地由一个新的独立层提供服务：L3！

L3 与 L2 相关，就像 L2 与 L1 相关一样。只要 L2
能够支持验证者（Verifier）智能合约，L3 就可以使用有效性证明来实现。当 L2
也使用提交给 L1 的有效性证明时，就像 StarkNet
所做的那样，这将成为一个非常优雅的递归结构，其中 L2 证明的压缩优势乘以 L3 证明的压缩优势。理论上说，如果每一层都实现了例如 1000 倍的成本降低，那么 L3 可以比 L1 降低 1,000,000 倍——同时仍然保持 L1 的安全性。这也是 starknet 引以为豪的递归证明的真实用例。

这里需要用到《数据可用性层的链上与链下》部分知识。整个 Layer 3 包括了：

Rollup（链上数据可用性），validium（链下数据可用性）。两种分别对应不同的应用需求。对价格、数据敏感的 web2 企业可以使用
validium，将数据放在链下，这样极大的降低了链上 gas 费用，并且可以不公开用户数据实现隐私性，让企业完成自己对数据的掌控力，使用自定义的数据格式，以前企业的数据商业模式仍然能够跑通。

L2 用于扩展，L3 用于定制功能，例如隐私。

在这个愿景中，没有尝试提供「二次方级可扩展性」；相反，这个堆栈中有一层可

以帮助应用程序扩展，然后根据不同用例的定制功能需求分离各层。

L2 用于通用扩展，L3 用于自定义扩展。

自定义扩展可能有不同的形式：使用除 EVM 之外的其他东西进行计算的专用应用程序，其数据压缩针对特定应用程序的数据格式进行优化的
rollup（包括将「数据」与「证明」分开，并用每个区块的单个 SNARK
完全替换证明）等。

L2 用于无信任扩展（rollup），L3 用于弱信任扩展（validium）。

Validium 是使用 SNARK 来验证计算的系统，但将数据可用性留给受信任的第三方或委员会。在我看来，Validium
被严重低估了：特别是，许多「企业区块链」应用程序实际上可能最好由运行
validium 证明者并定期将哈希提交到链的中心化服务器来提供最佳服务。

Validium 的安全等级低于 rollup，但可以便宜得多。

对于 dApp 的开发者来说，在基础设施上可以有以下几种选择：

自己开发一个 Rollup（ZK Rollups 或者 Optimistic Rollups）

优势是你可以继承以太坊的生态（用户），还有它的安全性，但是对于一个 dApp 团队来说，Rollup 的开发费用显然过高。

选择 Cosmos、Polkadot 或者是 Avalanche

开发的费用会更低（例如 dydx 就选择了
Cosmos），但是你将失去以太坊的生态（用户），以及安全性。

自己开发一个 Layer 1 区块链

带来的开发费用和难度很高，但是却能拥有最高的控制权。

我们对比一下三种情况：

难度/费用：Alt-layer 1 > Rollup > Cosmos

安全性：Rollup > Cosmos > Alt-layer 1

生态/用户：Rollup > Cosmos > Alt-layer 1

控制权：Alt-layer 1 > Cosmos > Rollup

作为一个 dApp 的开发者，如果想继承以太坊上的安全性和流量，那就不能重新开发一条链，那只能选择 rollup。但是自己开发一个 layer 2 rollup 又非常贵，那么合适的解决方案就变成了利用 layer 3 SDK 开发一个自己的应用专用的 Rollup（application-specific rollup），即 Layer 3。

四、Layer 2 的未来发展

由于以太坊是基于账户模型设计的，所有的用户均处在一整个状态树内，因此无法进行并行，因此以太坊本身的桎梏就让其需要剥离执行操作，将 rollup 的多笔交易合成为一笔交易，作为结算层的存在。现在所有的问题就集中在 layer 2 的吞吐量的提升上。不仅仅是用 Layer 3 可以提高交易的吞吐量，还有在 Layer 2 上实行并行处理，也可以极大提高整个网络的吞吐量。

并行化问题 starknet 也在积极探索，虽然目前证明算法仍然是桎梏，但是预计未来将不会成为阻力。潜在的瓶颈包括：

排序器 tx 处理：一些排序器的工作似乎天生就是串行的。

带宽：多个排序器之间的互连将受到限制。

L2 状态大小

在 starknet 社区中，成员也提出了 aptos 的并行处理方式非常不错。就 Starknet 而言，目前也在推进排序器内部 tx 并行排序的能力。

五、总结

以太坊正在将执行层剥离，所有的行为都朝着其「全球」结算层愿景的方向前进。目前整个以太坊虽然进度缓慢，也就是由于其整体过于庞大，每次更新都牵扯了许多利益与权衡。但不可否认的是，以太坊正在经历重大变革，以太坊大量的链上活动、经济机制改进以及以太坊 2.0 可扩展性，其引领的创新 IC0、Defi、NFT 等很多东西值得以太坊社区兴奋与期待。相信伴随着越来越多国家部署以太坊的节点，

比如阿根廷首都政府计划在 2023
年部署以太坊验证节点，在不久的将来，以太坊真的能够实现其宏伟愿景。