官网地址：https://rollkit.dev/
github地址：https://github.com/rollkit
文档地址：https://rollkit.dev/docs/intro/
博客地址：https://rollkit.dev/blog/
Rollkit-Cosmos-sdk：https://github.com/rollkit/cosmos-sdk
Rollkit-Cometbft：https://github.com/rollkit/cometbft
Rollkit：https://github.com/rollkit/rollkit
Rollkit-celestia-da：https://github.com/rollkit/celestia-da
Rollkit-go-da：https://github.com/rollkit/go-da；go-da 为模块化区块链定义了通用数据可用性接口。
目前已实现的此通用数据可用性接口的da实例：

The following implementations are available:

- celestia-da implements Celestia as DA.
- avail-da implements Polygon Avail as DA.

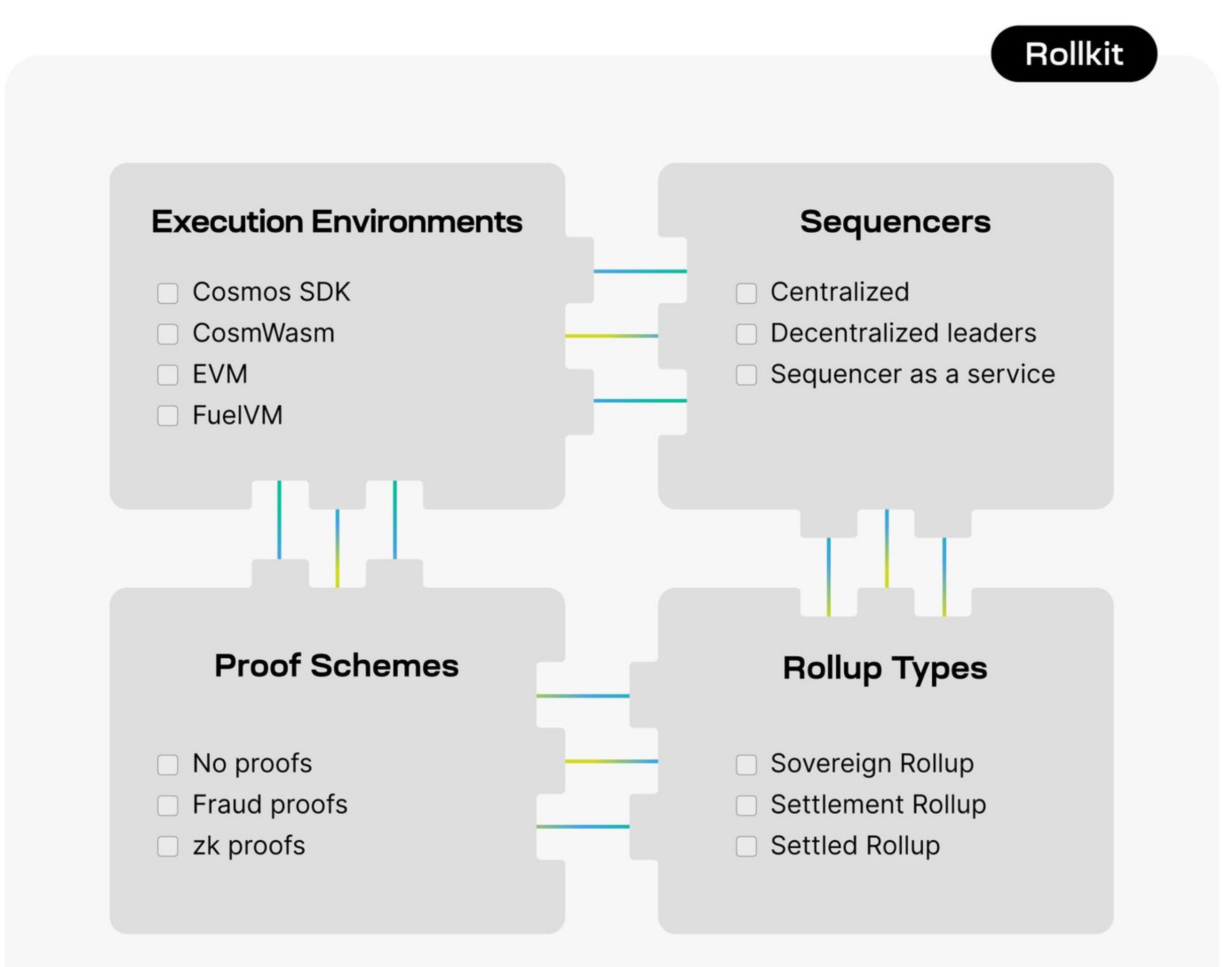
Rollkit 堆栈

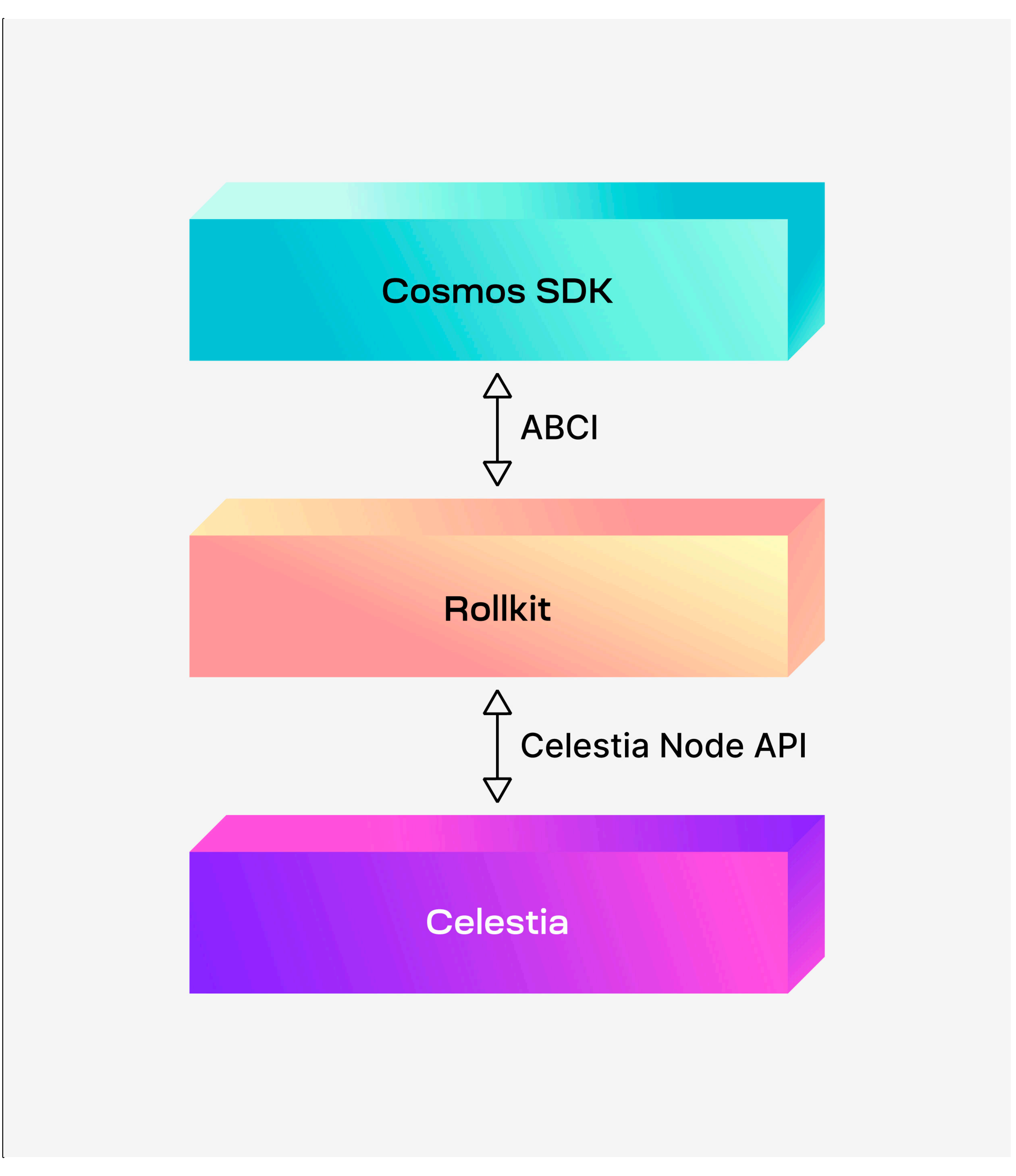
本页将介绍 Rollkit 的主要组件。

Rollup排序器节点收集用户的交易，将它们聚合成块，并将块发布到数据可用性（DA）层（例如Celestia）以进行排序和最终确定。

全节点执行并验证汇总区块，并且在乐观汇总的情况下，在需要时传播欺诈证明。

轻客户端将接收标头，验证证据（欺诈、zk 等），并对有关状态的信任最小化查询进行身份验证。





ABCI接口

Rollkit 是一款功能齐全的应用程序区块链接口（ABCI）客户端软件。
它可以用作任何 ABCI 应用程序的 CometBFT 替代品。
由于这种兼容性，您可以使用**abci-cli**等工具来测试和调试您的汇总。

Cosmos SDK

您想将 Cosmos SDK 应用程序更改为 Rollkit 汇总吗？没问题！

您需要将 Cosmos SDK Go 依赖项替换为支持 Rollkit 的版本，可以在**rollkit/cosmos-sdk**存储库中找到该版本。

请注意，**rollkit/cosmos-sdk**存储库遵循上游 Cosmos SDK 的发布分支，但好处是使用 Rollkit 而不是 CometBFT 作为 ABCI 客户端。

并且不要忘记将另一个依赖项 替换CometBFT为，**rollkit/cometbft**它具有增强的 ABCI 接口，其中包括状态欺诈证明所需的方法。

Data availability-数据可用性：将数据提交到DA层网络

可以使用通用**接口来访问数据可用性（DA）**。此设计允许与任何 DA 层无缝集成。新的实现可以通过编程方式插入，无需分叉 Rollkit。

该DataAvailabilityLayerClient接口包括基本的生命周期方法（Init、Start、Stop）以及数据可用性方法（SubmitBlocks、RetrieveBlocks）。

该BlockRetriever接口用于实现数据可用性层的全节点同步。请务必记住，DA 层块高度和汇总高度之间没有直接关联。

每个 DA 层块可以包含任意数量的汇总块。

Celestia-塞拉斯蒂娅

Celestia 是为 Rollkit 实现的数据可用性集成的一个示例。它通过包使用**Celestia Node APIrollkit/celestia-da**。

要在 Celestia 上部署 Rollkit rollup，您还必须**运行 Celestia 轻节点**。

Rollkit 还支持使用其他数据可用性 (DA) 层，并且比特币、Mock、gRPC 的研究集成正在进行中。

可以使用**go-da**接口添加新的 DA 层集成。

Node components

1、Mempool

内存池的灵感来自 CometBFT 内存池。默认情况下，事务以先到先服务 (FCFS) 方式处理。交易的排序可以在应用程序级别实现；

目前，这可以通过在 CheckTx 上返回优先级来实现，一旦我们支持 ABCI++，也可以通过PrepareProposal 和application mempool.实现这一点。

2、Block manager

块管理器包含通过 Go 通道进行通信的例程 AggregationLoop、RetrieveLoop 和 SyncLoop。

这些 Go 例程在 **Rollkit** 节点启动 (**OnStart**) 时运行。

只有定序器节点运行 AggregationLoop，它根据 BlockManager 中的 BlockTime 使用计时器控制汇总的块生成频率。

所有节点都会运行SyncLoop并查找以下操作：

- **接收区块头**：通过通道接收区块头HeaderInCh，Rollkit 节点尝试使用相应的区块数据来验证该区块。
- **接收区块数据**：通过通道接收区块体blockInCh，Rollkit 节点尝试验证该区块。
- RetrieveLoop根据BlockManager中的DABlockTimein定时器发出信号。

所有节点还运行 RetrieveLoop，负责与数据可用性层交互。

它检查最后更新的 DAHeight 以检索具有由 SyncLoop 发出信号的计时器 DABlockTime 的块。

请注意，汇总的 DA 层的起始高度 DAStartHeight 可在 BlockManager 中配置。

RPC

Rollkit的RPC完全实现了CometBFT RPC接口和用于查询的API：
有关汇总节点的信息：节点的运行状况、状态和网络信息等信息。
Rollup 区块链：获取有关 Rollup 区块链的信息，例如区块和区块头。
汇总交易：获取交易信息并广播原始交易，具有搜索功能。
ABCI：汇总应用程序信息。
目前支持以下 RPC 协议：
  HTTP 上的 URI
  基于 HTTP 的 JSON-RPC
  基于 WebSocket 的 JSON-RPC

P2P层
Rollkit 的 P2P 层支持 rollup 节点之间的直接通信。 它用于八卦交易、新创建区块的标头以及状态欺诈证明。

Rollkit 使用基于 DHT 的主动对等点发现。 启动节点会连接到预先配置的引导对等点，并在 DHT 中通告其命名空间 ID。

该解决方案很灵活，因为多个汇总网络可以重用相同的 DHT/引导节点，但特定的汇总网络也可能决定使用专用节点。

Rollkit 节点类型
Rollkit节点是在node包中实现的。

1、全节点
全节点验证所有区块，并为乐观汇总生成欺诈证明。 由于他们完全验证所有汇总块，因此他们不依赖欺诈或有效性证明来确保安全。

2、轻节点（正在进行中）
轻节点是验证区块头的轻量级汇总节点，可以通过欺诈证明或有效性证明来保护。
建议使用低资源设备的普通用户使用它们。 运行轻节点的用户可以对汇总状态进行信任最小化查询。
目前，Rollkit轻节点仍在开发中。

3、定序器节点

汇总可以利用定序器节点。 排序器是汇总的区块生产者，负责将交易聚合到区块中，并且通常执行交易以生成状态根，供汇总的轻客户端使用。

Rollkit 计划支持多种不同的可插拔音序器方案：

		比 L1 更快的软确认	控制 Rollup 的事务顺序	与其他汇总的原子可组合性	抵制审
集中测序仪	需要启动定序器	是的✅	是的✅	没有❌	最
分散式测序仪	需要旋转一套音序器	是的✅	是的✅	没有❌	实
共享分散式测序仪	是的✅	是的✅	没有❌	是的✅	实
纯分叉选择规则	是的✅	没有❌	也许🟡	也许🟡	最

状态有效性模式
1、悲观（仅限全节点）
悲观汇总是仅支持全节点的汇总，该全节点会重放汇总中的所有事务以检查其有效性。 Rollkit 默认支持悲观汇总。
悲观汇总类似于 Tether 通过 OmniLayer 使用比特币作为数据可用性层的方式。

2、乐观（欺诈证明）（正在进行中）
Rollkit 当前的设计由将块发布到 DA 层的单个定序器和多个（可选）完整节点组成。
定序器将块头八卦到全节点，全节点从 DA 层获取发布的块。
然后，全节点执行这些区块中的交易以更新其状态，并通过 P2P 网络将八卦区块头传送到 Rollkit 轻节点。
一旦启用状态欺诈证明，当一个区块包含欺诈性状态转换时，Rollkit 全节点可以通过比较交易之间的中间状态根（ISR）来检测它，
并生成可以通过 P2P 网络传播到 Rollkit light 的状态欺诈证明 节点。
这些 Rollkit 轻节点可以使用这个状态欺诈证明来验证自己是否发生了欺诈性的状态转换。
总体而言，只要系统中至少有一个诚实的全节点生成状态欺诈证明，状态欺诈证明就可以实现全节点和轻节点之间的信任最小化。
请注意，Rollkit 状态欺诈证明仍在进行中，并且需要 ABCI 之上的新方法，具体来说，GenerateFraudProof、VerifyFraudProof 和 GetAppHash。
您可以在此架构决策记录 (ADR) 中找到当前的详细设计以及推动状态欺诈证明完成所需的剩余工作。

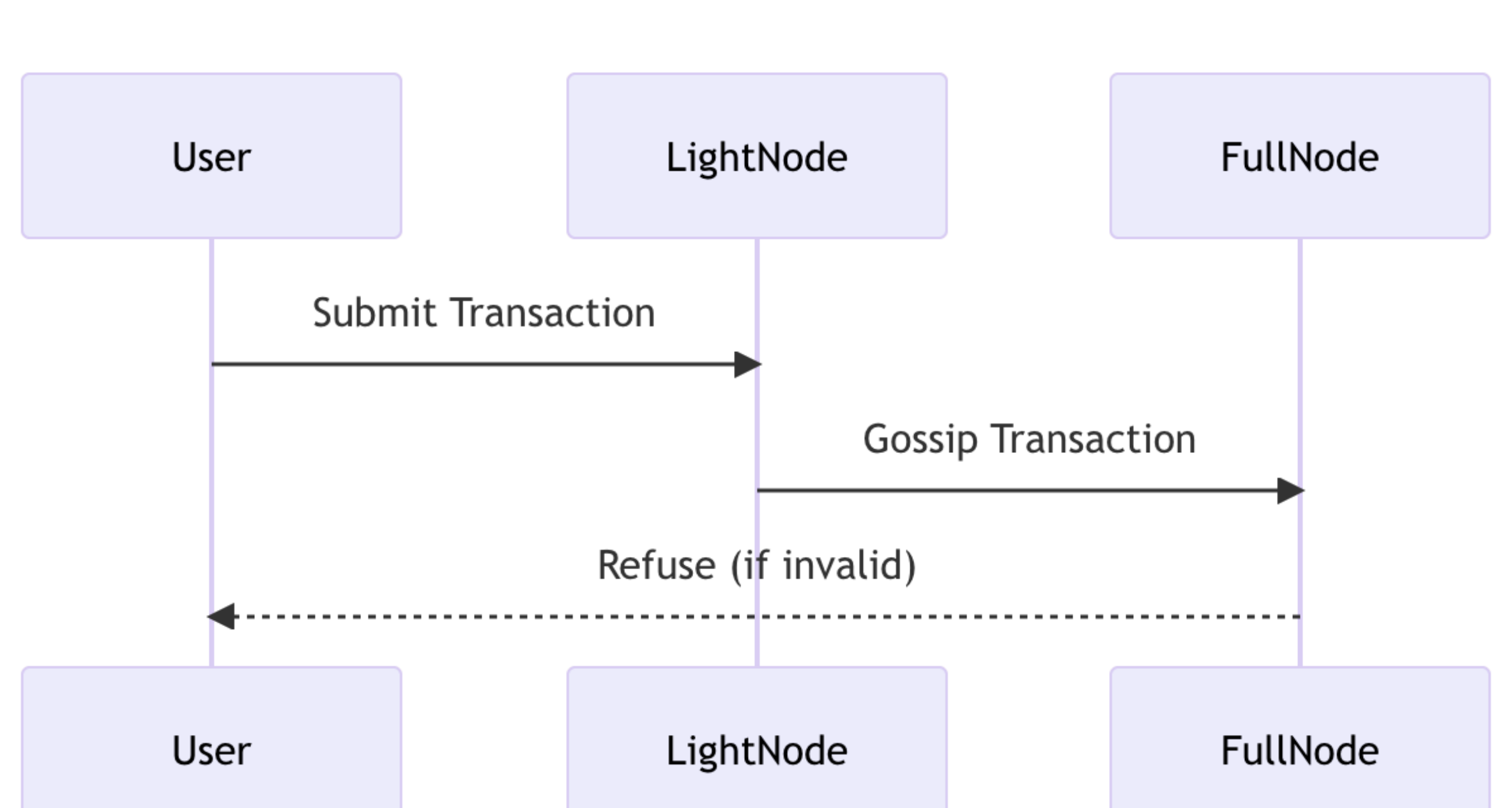
3、有效性（ZK 证明）

有效性 (ZK) 汇总已计划，但 Rollkit 目前不支持。

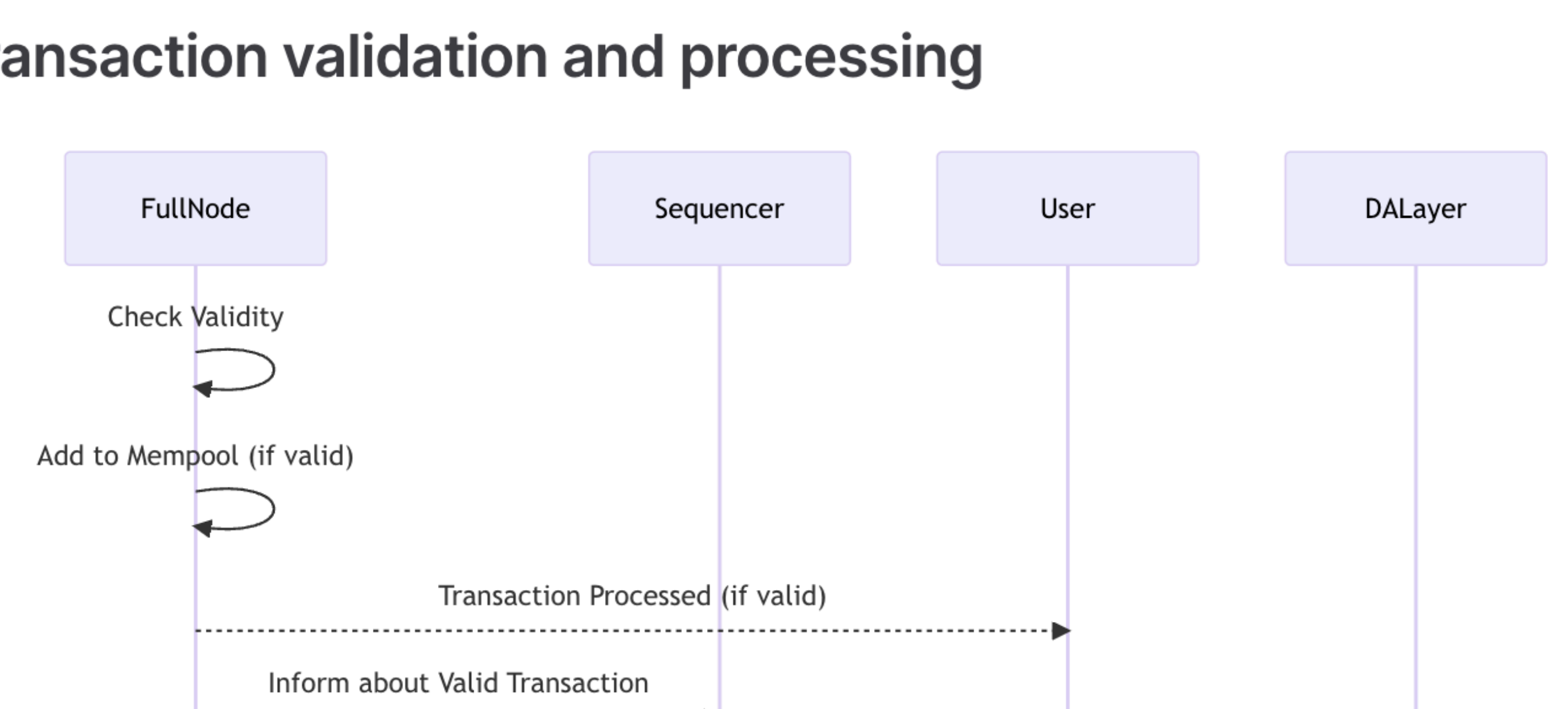
Transaction flow

Rollup 用户使用轻节点与 rollup P2P 网络进行通信主要有两个原因：
  1、提交交易
  2、八卦标题和欺诈证据

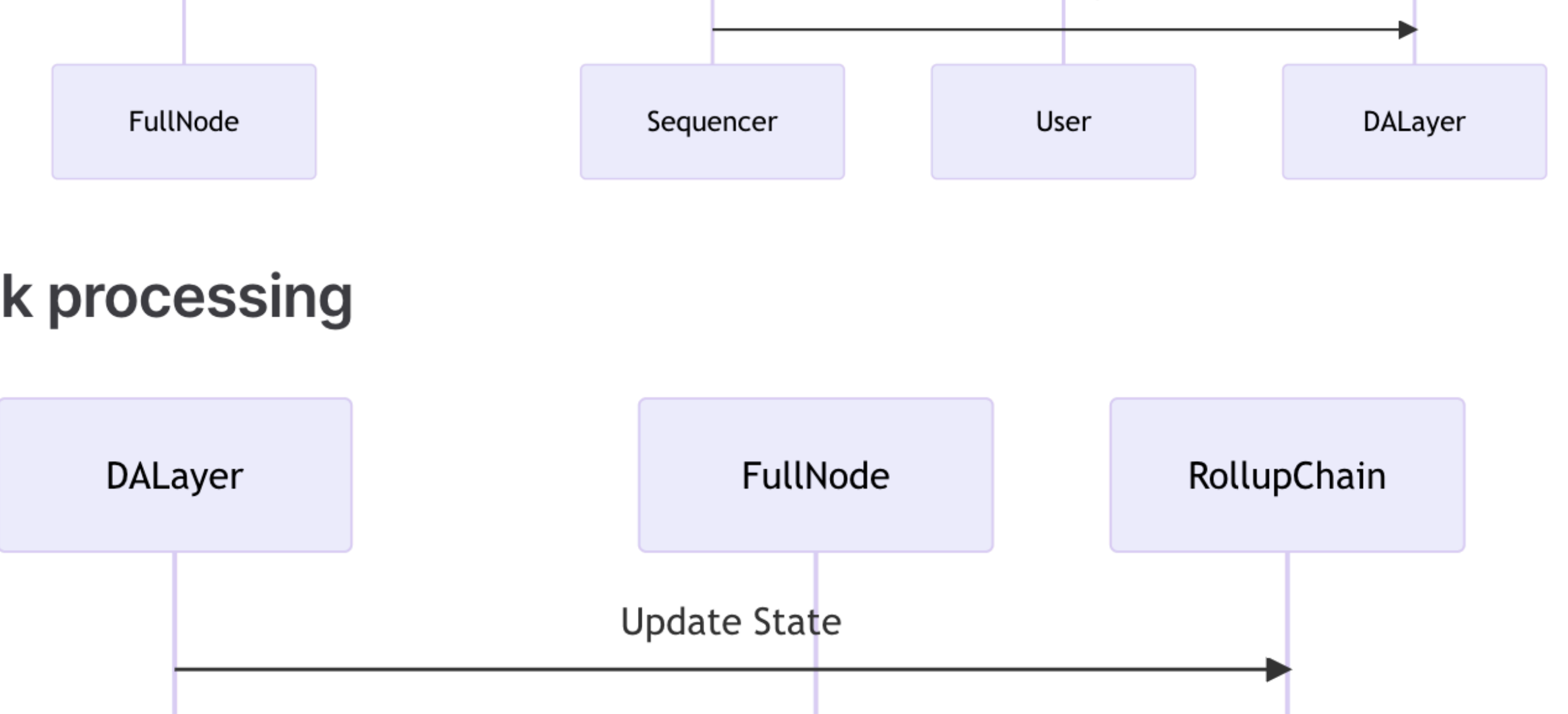
Transaction submission



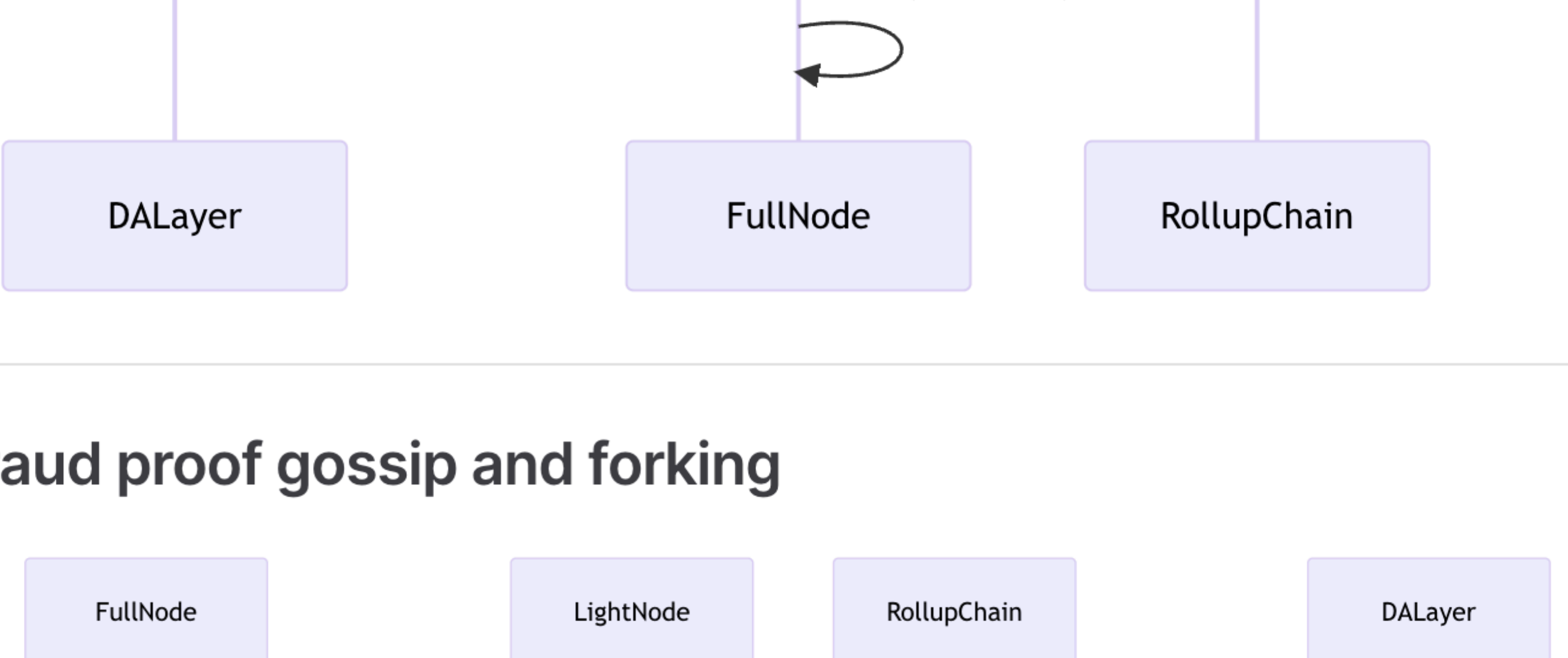
Transaction validation and processing



Block processing



Fraud proof gossip and forking



为了进行交易，用户将交易提交到他们的轻节点，轻节点将交易传送到全节点。

在将交易添加到内存池之前，完整节点会检查其有效性。

有效的交易会被纳入mempool，而无效的交易会被拒绝，用户的交易不会被处理。

如果交易有效并且已包含在内存池中，排序器可以将其添加到汇总块中，然后将其提交到数据可用性（DA）层。

这会为用户带来成功的交易流，并且汇总的状态也会相应地更新。

区块提交到DA层后，全节点下载并验证该区块。然而，排序器有可能恶意向 DA 层提交一个包含无效交易或状态的区块。

在这种情况下，rollup链的完整节点将认为该块无效。

在乐观汇总的情况下，如果他们发现块无效，他们会生成欺诈证明并在 P2P 网络中与其他全节点和轻节点进行八卦。

结果，rollup链将停止，网络将通过社会共识决定分叉链。

将来，当去中心化排序器方案到位时，将提供其他选项，例如削减排序器或选择另一个完整节点作为排序器。

然而，无论如何，都必须创建一个新的区块并将其提交给 DA 层。您可以**在此处**阅读有关定序器节点的更多信息。