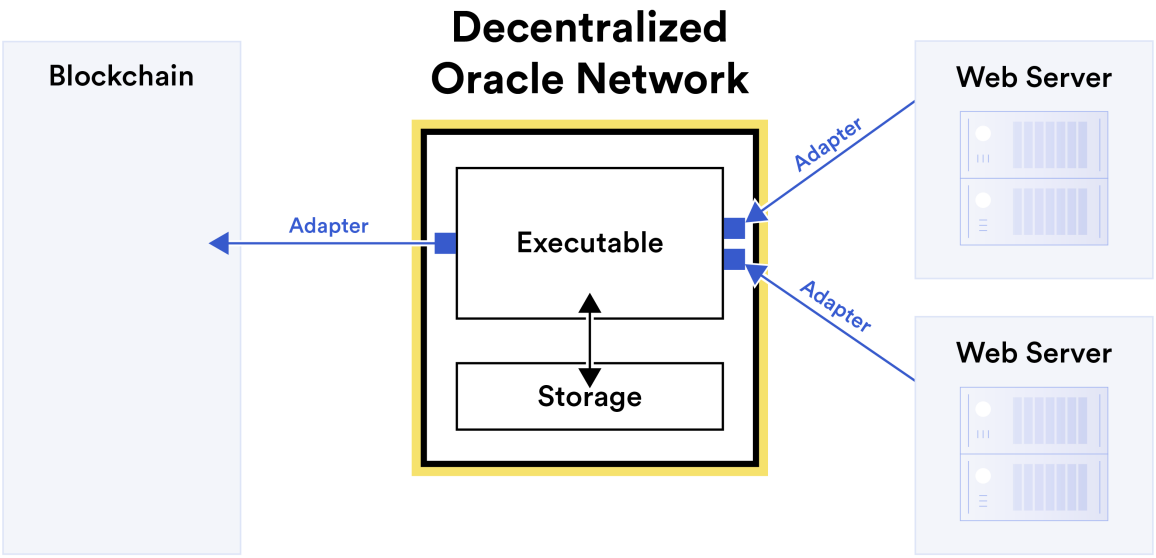
# Chainlink——白皮书简析(whitepaper v2)

以目前区块链公链比较成熟的生态以太坊为例，为了保证账本的准确性和智能合约执行的确定性，以太坊节点虚拟机会被运行在一个隔离的环境中，因此在虚拟机中运行的智能合约代码无法跟传统编程语言一般直接从链下或者互联网获取数据，所有链下的数据都需要通过链下主动往链上发起交易并经过节点共识上链后才能被智能合约读取和使用。

而在实际的业务场景中，为了满足各种各样的需求，不可避免地智能合约需要去获取链下的数据作为依据或数据源，这也就催生了当前区块链中非常重要的一类项目——预言机（Oracle），预言机项目的作用简单来说就是聚合链下的数据然后将数据上链以供其它合约使用，或者换句话说，预言机就是一个“上链的链下数据源”。（注：区块链中的预言机和传统计算机技术中的预言机不是一个概念）

## 去中心化预言机网络框架(Decentralized Oracle Networks)

在实际的以太坊业务中，由于数据上链后不可更改，且错误的数据源所带来的破坏无法估量，因此为了防止预言机无意甚至是恶意地将错误数据同步到链上，现在许多的预言机项目或者框架都被设计成去中心化的。



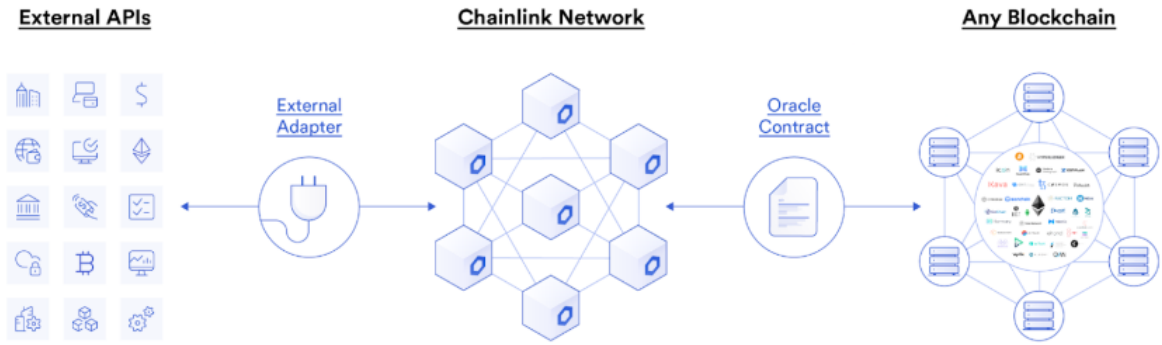

*去中心化预言机网络框架图（来源：Chainlink Whitepaper V2）*

一般情况下，去中心化预言机网络中会有多个聚合数据的节点，每个节点各自通过适配器或代码中间件(图中的Adapter)从不同的数据源(Web Server)中获取数据，然后将数据拉进节点的虚拟机环境(Executable)进行处理，在这过程中节点节点虚拟机内的代码可执行文件会从数据库(Storage)中获取数据辅助处理从数据源获得的信息(主要做数据异常判断)，并最终得到一个数据结果，一些相应的信息也会被写回数据库。

预言机网络中的每个节点都会根据其连接的数据源得到一个最终结果，在大多数情况下，当预言机网络的大多数节点经过共识最终得到一个一致的结果时，预言机网络会推选出一个节点将这个结果发布到预言机的链上合约中，最终完成链下数据的上链过程。

## Chainlink框架

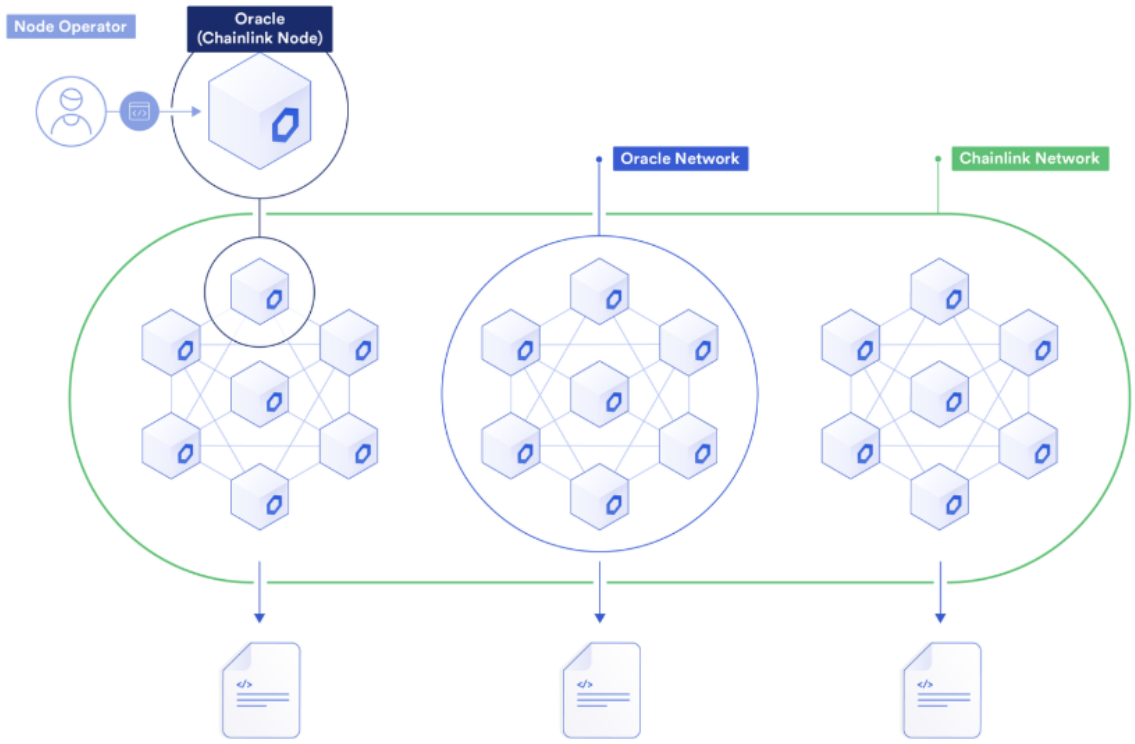
Chainlink是目前区块链生态中比较成熟的去中心化预言机网络项目。




*Chainlink宏观架构图（来源：blog.chain.link）*

Chainlink网络的每个节点通过其自定义的外部适配器从链下的数据源获取数据，然后节点之间经过特定的共识达成一个结果并将结果通过交易传至链上的预言机合约以供链上的项目使用(实际上是将所有可信节点的数据上链然后在链上取中位数)。

这里需要特别强调的是，Chainlink网络并不是单单指某一个预言机网络，它实际上指的是一个预言机框架。

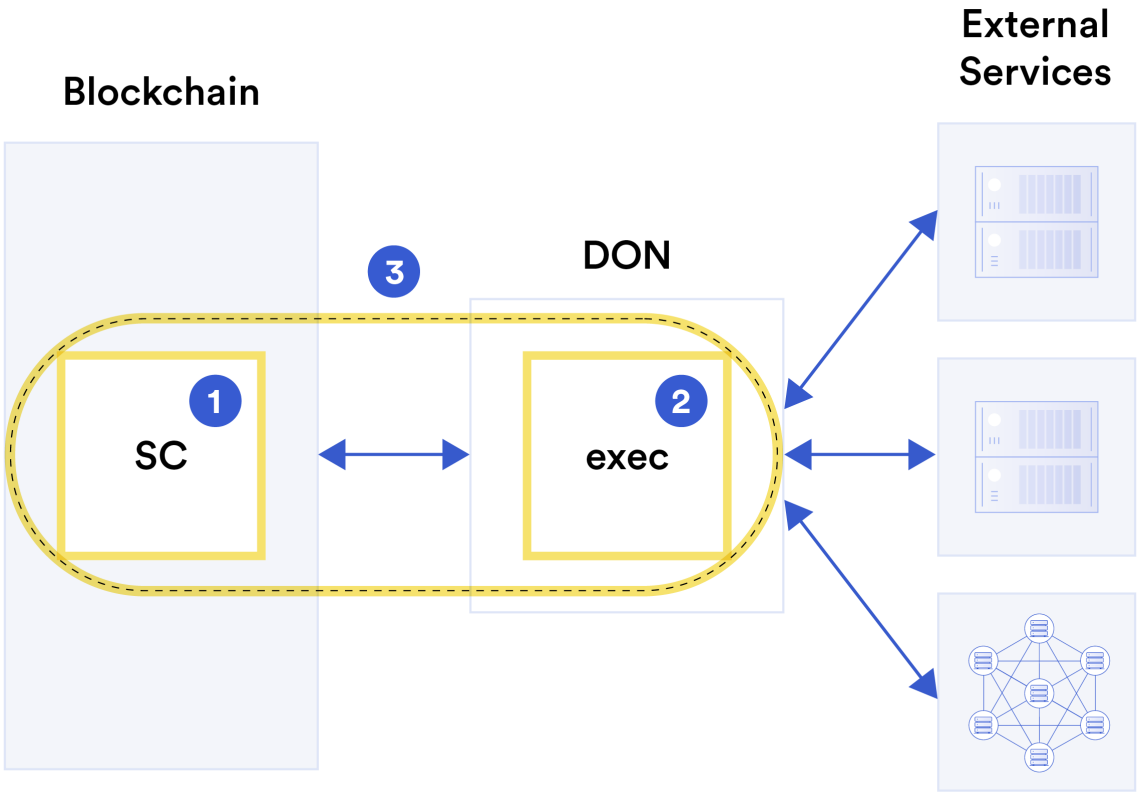



*Chainlink网络架构（来源：blog.chain.link）*

在一个Chainlink框架中可以包括多个预言机网络，而预言机节点之间可以是异构的，他们之间通过高度封装的接口进行交互。

## 混合型智能合约(Hybrid Smart Contract)

Chainlink在其白皮书v2版本中提出了一个“混合型智能合约”的概念。相较于传统代码，在链上执行智能合约代码速度更慢，代价更高并且无法从现实世界获取数据，因此Chainlink认为若要实现智能合约的全部潜力，就需要将智能合约和链下的组件安全地结合起来，这种将合约和链下代码组件的组合就是混合型智能合约。

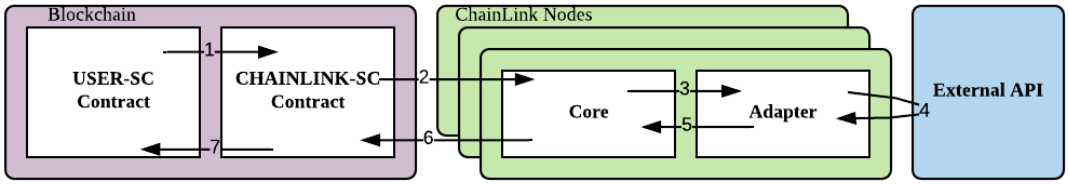



*混合型智能合约概念图（来源：Chainlink Whitepaper V2）*

混合型智能合约(图中的③)主要由两部分组成：第一部分是运行在区块链上的链上合约SC(①)，第二部分是在Chainlink预言机网络中运行的可执行文件exec(②)，而Chainlink网络作为这两个组件之间的桥梁，则负责将混合型智能合约与链外资源，如网络服务、其他区块链网络、分布式存储资源等连接起来。

注：DON表示的是去中心化预言机网络(Decentralized Oracle Network)，这里指的是Chainlink网络。

为了更深入地了解混合型智能合约，这里简单描述一下工作流程。




*Chainlink工作流图（来源：Chainlink Whitepaper V1）*

1.有获取链下数据需求的用户合约向Chainlink部署在链上的预言机合约发起获取数据请求交易，交易上链后，该请求事件会被广播；

2.Chainlink节点的辅助中间件或者适配器在监听到该请求事件后，会将事件发送至节点中的可执行文件(图中的Core，也就是exec)；

3.Chainlink的Core获取事件后会根据事件请求内容给适配器分配相应的数据聚合任务；

4.适配器从链下数据源获取数据并返回，一个节点会从多个数据源获取多个数据；

5.适配器将获取的数据返回到Core；

6.Core对数据进行处理，打包成链下报告(Off-Chain Reporting)，并主动调用Chainlink链上预言机合约的发送报告函数发起交易，将链下数据上链；

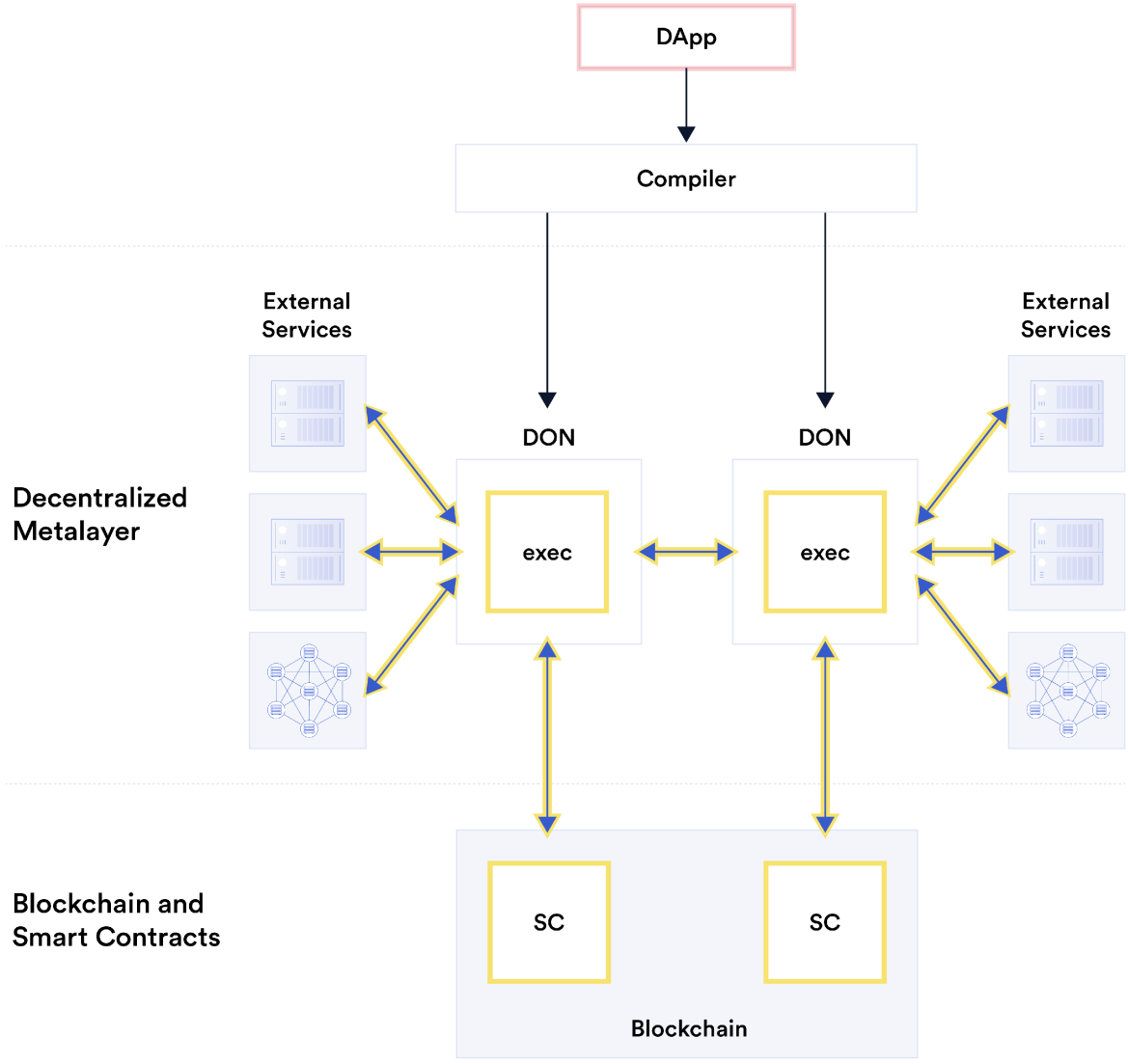
7.用户合约从链上预言机合约获得链下数据。

而在实际的业务流程中，根据不通的业务需求上述的流程会有些不同的调整，比如获取资产价格数据时，用户合约直接调用链上预言机合约的"lastestRoundData()"函数就能直接获得链下的资产价格，而不需要在发起请求后等待预言机将数据上链才能获取。原因是Chainlink的资产价格维护策略主要是两个：①当链上存储的价格和链下实际的价格偏差超过一定的阈值后触发价格更新；②心跳，一般是一小时主动更新一次。

## 抽象化接口——元层（Metalayer）

基于降低Dapp(去中心化应用)的开发门槛和让现有传统非区块链企业系统以最小的代价接入区块链生态的目的，Chainlink提出了一个叫"元层"的框架。简单来说，就是对区块链生态和预言机项目进行高度封装并对外暴露接口，这样Dapp开发者或者传统非区块链企业系统开发者就可以简单地通过与接口交互而将自己项目或系统接入区块链而不需要做出太大的修改。

并且由于元层对外暴露的接口是与去中心化预言机网络交互的，这也就意味着对于元层的使用者来说，他们可以无缝地去使用链上或链下地数据进行交互，链上链下数据交互产生的各种问题对他们来说是无感的，全都由预言机网络去解决。



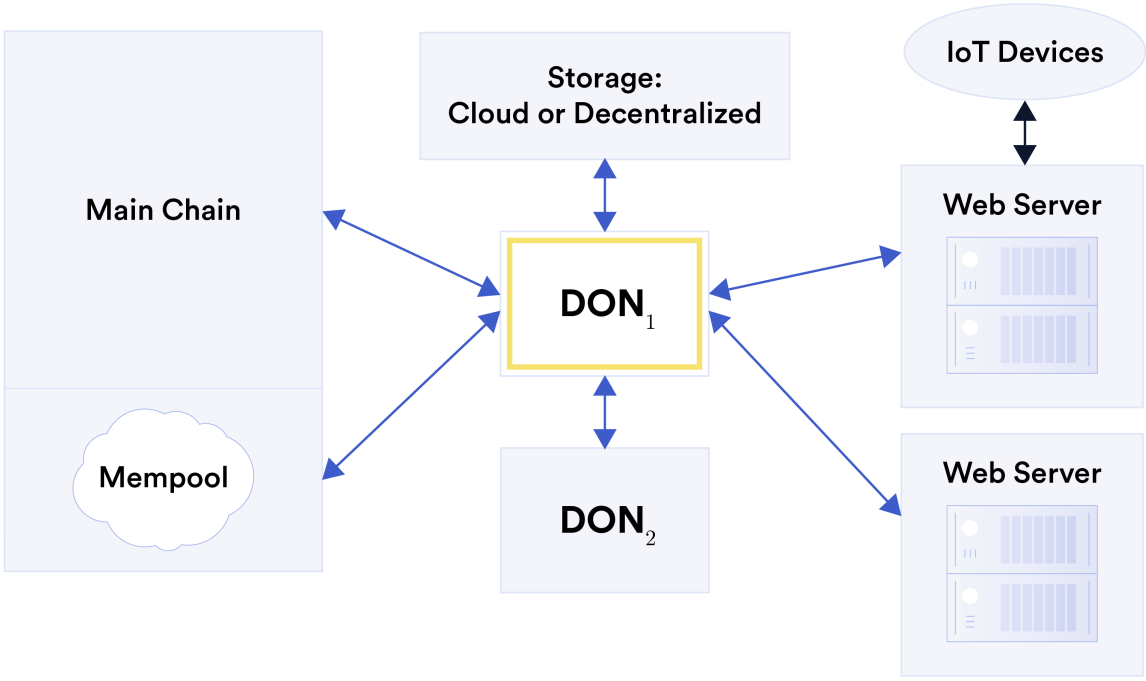

*去中心化元层概念图（来源：Chainlink Whitepaper V2）*

在元层架构中，Dapp会被特定的编译器编译成去中心化预言机网络节点可识别的对象并直接参与到预言机网络活动中。

从这里其实可以看出Chainlink项目的目标就是成为整个区块链生态面向外部世界的代言人。

## 适配器（Adapters）

适配器是运行在Chainlink节点上的可执行文件的接口，可以发送或接受链下的数据。适配器本质上就是一个代码中间件，为Chainlink节点服务，可以用于链接Chainlink网络和其他数据源或网络。




*适配器的应用场景（来源：Chainlink Whitepaper V2）*

上图中的蓝色双向箭头表示的就是适配器，从上图中可以看出，适配器除了可以用于连接Chainlink网络和Web服务器、区块链和云服务存储外，还可以用来连接其他的去中心化预言机网络，换句话说，Chainlink可以从其他的预言机项目获取数据。

## 可执行文件和启动器（An executable and Initiators）

可执行文件是Chainlink节点上代码的基本单元，可以用exec=(logic,init)来表示，其中logic表示一个确定性的程序，有许多入口点(logic1，logic2…）；init是一组相应的启动器(init1，init2…)。为了确保Chainlink的确定性和可审核性，可执行文件的logic从底层账本L(白皮书中提到去中心化预言机网络可通过无许可共识的方式高效地维护一个共同的主账本)读取输入数据，并将对应的输出信息存回账本中，换句话说，任何输入可执行文件的数据都必须首先存储在账本L上。

启动器主要用于触发logic内事务的执行，Chainlink节点的启动器与可执行文件相关联，并依赖于外部的状态决定是否启动其对应的logic代码，因此启动器与logic不同，具有非确定性。

下面举一个ETH-USD资产价格更新的例子说明启动器和可执行文件的关系：

**Example 1** (Deviation-triggered price feed). *A smart contract* SC *may require fresh price-feed data (see Section 3.6.3) whenever there is a substantial change, e.g., 1%, in the exchange rate between a pair of assets, e.g., ETH-USD. Volatility-sensitive price feeds are supported in Chainlink today, but it is instructive to see how they can be realized on a DON by means of an executable* $\mathsf{exec}_{\mathsf{feed}}$.

*The executable* $\mathsf{exec}_{\mathsf{feed}}$ *maintains the most recent ETH-USD price* $r$ *on* $\mathcal{L}$*, in the form of a sequence of* $\langle \texttt{NewPrice} : j, r \rangle$ *entries, where* $j$ *is an index incremented with each price update.*

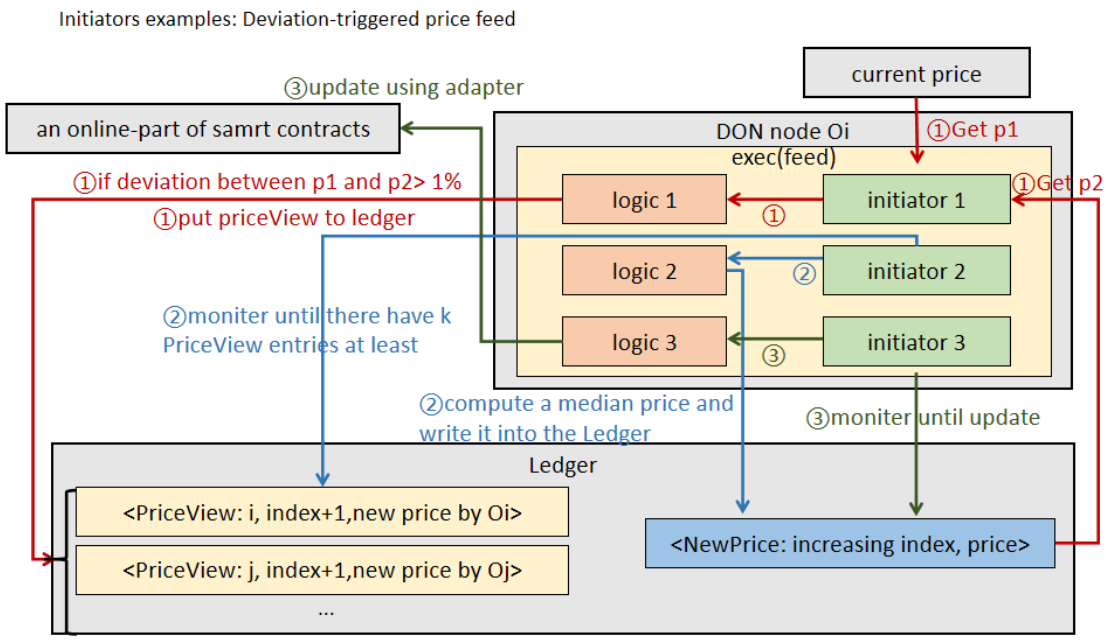
*An initiator* $\mathsf{init}_1$ *causes each node* $\mathcal{O}_i$ *to monitor the current ETH-USD price for deviations of at least* 1% *from the most recently stored price* $r$ *with index* $j$*. Upon detection of such a deviation,* $\mathcal{O}_i$ *writes its current view* $r_i$ *of the new price to* $\mathcal{L}$ *using an entry of the form* $\langle \texttt{PriceView} : i, j+1, r_i \rangle$.

*A second initiator* $\mathsf{init}_2$ *fires when at least* $k$ *such* `PriceView`*-entries with new price values for index* $j+1$ *created by distinct nodes have accumulated on* $\mathcal{L}$*. Then,* $\mathsf{init}_2$ *invokes an entry point* $\mathsf{logic}_2$ *to compute the median* $\rho$ *of the first* $k$ *fresh, valid price-view values and writes a fresh value* $\langle \texttt{NewPrice} : j+1, \rho \rangle$ *to* $\mathcal{L}$ *. (Operationally, nodes may take turns as designated writers.)*

*A third initiator* $\mathsf{init}_3$ *watches for* `NewPrice` *entries on* $\mathcal{L}$*. Whenever a new report* $\langle \texttt{NewPrice} : j, r \rangle$ *appears there, it invokes an entry point* $\mathsf{logic}_3$ *that pushes* $(j, r)$ *to* SC *using an adapter.*

*白皮书对应的例子原文图片（来源：Chainlink Whitepaper V2）*

为了方便理解，我将上述的例子画成简易的流程图：




*资产价格更新流程图*

第一阶段(图中标红部分)：Oi节点的可执行文件中的启动器init1监控ETH-USD币对的最新价格和账本p1以及L中存储的该币对的价格p2，当p1和p2偏差超过1%时，init1调用logic1，logic1会通过适配器从链下聚合价格并且将聚合的新价格以PriceView记录的形式存入L中，并标明该记录由Oi存入账本；

第二阶段(图中标蓝部分)：启动器init2监控账本L中的PriceView记录，当发现有超过k个预言机节点存入的价格处在某个合理的区间内时(k可以为某个设定好的阈值)，调用logic2读取前k条记录的价格并取其中位数，然后生成最新的价格记录NewPrice并写入账本L中；

第三阶段(图中标绿部分)：启动器init3时刻监控账本L中的NewPrice记录，当发现该条记录更新时，调用logic3将最新的价格通过适配器更新至链上的预言机合约中。

# Chainlink节点的经济激励/惩罚措施

Chainlink团队发行了Link代币作为Chainlink经济生态中关键的一环，当用户向Chainlink请求链下数据，比如随机数时，需要向Chainlink支付一定量的Link代币；而当Chainlink网络中的节点成功向Chainlink的链上合约发送正确的链下数据时，也能获得一定量的Link代币作为奖励，反之，当节点发送错误的数据时，则会扣除一定量的Link代币，这就是最基础的Link经济生态。

为了更好地了解经济激励或惩罚措施对节点所带来地驱动或限制影响，下面引用一段白皮书v2中描述的节点进行数据聚合时可能出现的情况的原文：

*Complete agreement:* In the best case, nodes are in complete agreement: all nodes are available and have provided a timely report of the same value $r$ (either `true` or `false`). In this case, the network need only forward $r$ to relying contracts and reward each node with a fixed per-round payment $\$p$, which is much smaller than $\$d$.

*Partial agreement:* It is possible that some nodes are offline or there is disagreement about which value is correct, but most nodes report `true` and only a minority reports `false`. This case is also straightforward. The majority value (`true`) is computed, resulting in a correct report $r$. All nodes that reported $r$ are rewarded with $\$p$ while the oracles that reported incorrectly have their deposits slashed modestly, e.g., by $\$10p$.

*Alert:* In the event that a watchdog believes the output of the network is incorrect, it publicly triggers an alert, escalating the mechanism to the second-tier network. There are then two possible results:

- *Correct alert:* If the second-tier network confirms that the output of the first-tier network was incorrect, the alerting watchdog node receives a reward consisting of all slashed deposits, and thus more than $\$dn/2$.
- *Faulty alert:* If the second-tier and first-tier oracles agree, the escalation is deemed faulty and the alerting node loses its $\$d_w$ deposit.

*白皮书中节点进行数据聚合时可能的情况原文图片（来源：Chainlink Whitepaper V2）*

`$p`表示每一轮的数据聚合中`Chainlink`链上合约发放给正确提供数据的节点的奖励；

`$d`表示预言机节点为了加入`Chainlink`网络作为数据提供节点赚取收益所抵押的押金；

`$dw`表示预言机节点想要作为一个监督节点时所要抵押的监督押金，以防止恶意监督。

这些抵押或奖励都以`Link`代币结算。

在每一轮的数据聚合和报告中，所有预言机节点都会将自己所聚合到的数据提交至预言机网络账本中(代码中表现为数据库，而Chainlink使用的是PostgreSQL开源数据库)，然后由所有的节点相互监督其他节点提交数据的正确性。在所有节点提交自己的数据报告时，会有以下几种情况：

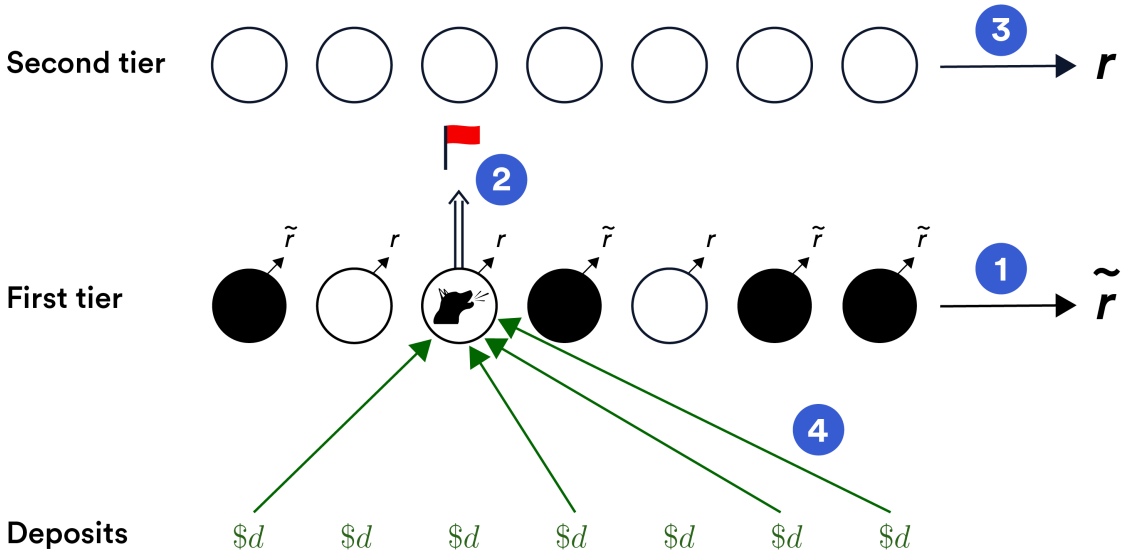
完全一致：在最好的情况下，所有节点提交的数据完全一致，或者说都处在某个可以聚合的区间内，所有节点的数据都可用，并且及时都提供了相同的聚合值r的报告。在这种情况下，Chainlink网络只需要将r转发给链上的预言机合约，并奖励每个节点一个固定的每轮支付的奖励$p即可，$p比节点的押金$d小得多；

部分一致：可能由于一些节点离线，或者对哪个值是正确的存在分歧等原因，可能出现某一轮的数据聚合中有少数节点的报告是错误的，但是大多数节点的数据报告是正确的情况，在这种情况下Chainlink网络会根据这大多数的报告计算出一个聚合值r，然后所有这部分报告r的节点都能得到奖励$p，而发送其它报告值的节点则扣除一定量的押金，例如10倍的$p。

警报：由于Chainlink网络本身无法得知哪些是正确的数据，因此如果有攻击者收买大多数节点用来汇报一个错误的数据(类似于区块链的51%攻击)，那么Chainlink网络完全有可能把这个错误的数据发送至链上从而造成难以预估的损失。因此为了防止这种情况，Chainlink设计了一种监督机制，所有的节点都可以成为一个监督节点(watchdog),如果一个监督节点认为Chainlink网络当前这一轮的数据报告有错误，那么该节点就可以发起警报。若Chainlink网络在某一轮的数据聚合中有节点发起警报，那么这一轮的数据聚合就会有一个第二层的网络接入，由这个第二层网络判断这一轮的数据报告是否有误，因此在有监督节点发起警报的情况下，有两个可能的结果：

①警报正确：第二层网络确认这一轮的数据报告确实有错误，则所有发出错误报告的节点的押金$d会被扣除(原文中用的是slash这个词，结合前后文，押金$d会被扣除绝大部分甚至全部)。 而这部分扣除的押金会全部奖励给发起警报的节点。由于Chainlink网络获得一个错误的聚合值需 要超过一半的节点发出错误的数据报告，因此监督节点获得的奖励将超过$dn/2，也就是获得整个Chainlink网络所有节点押金总和的一半以上。

②警报错误：第二层网络确认这一轮的数据报告正确无误，则扣除其监督押金$dw。




*监督节点发送警报示意图（来源：Chainlink Whitepaper V2）*

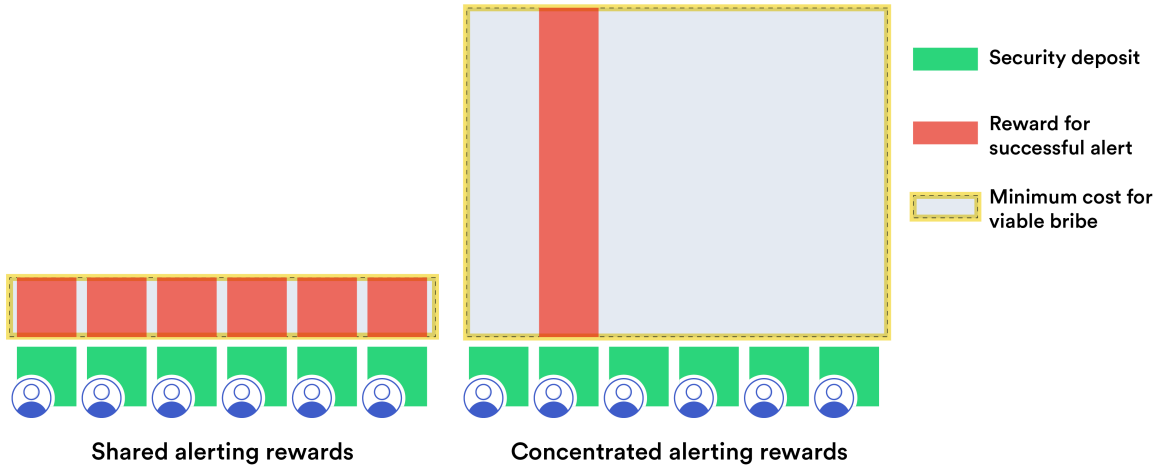
监督节点发送警报的大致流程如上图所示大致有四个步骤：

1.大多数节点(图中黑色部分)发送一个错误的数据报告，并且Chainlink网络通过这些错误的报告聚合出一个错误的值R(图中带~的r)；

2.监督节点向二层网络发送警报；

3.二层网络再次进行数据聚合得到一个正确的值r，并且发现与一层网络(ChainLink网络)所聚合的值R不一致，则调用链上相关合约对作恶节点进行处罚；

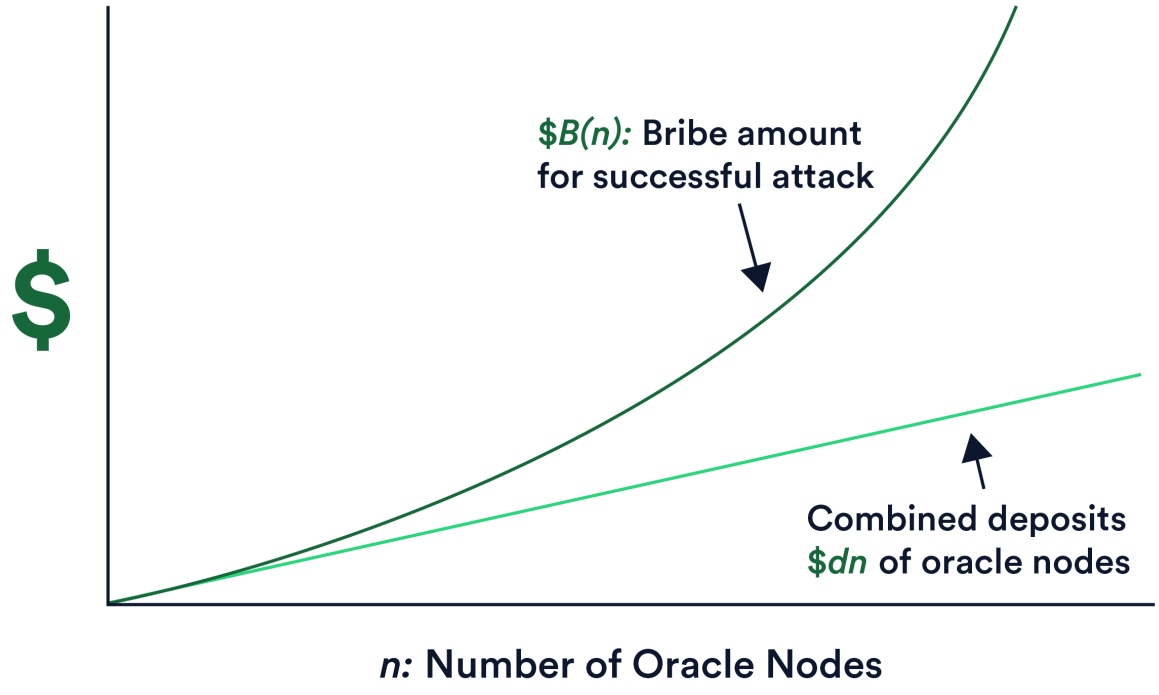
4.链上相关的监督合约没收所有作恶节点的押金，并将这些押金奖励给监督节点。




*通过集中监督奖励提高攻击者行贿成本（来源：Chainlink Whitepaper V2）*

从上面的描述中可以看出当监督节点发出正确的警报时会获得非常大量的奖励，这也就意味当有错误报告生成时所有的节点都想当监督节点。在一般的做法中，监督所得奖励理应平分给所有的监督节点以作鼓励，但是Chainlink认为把所有的监督奖励进行平均分配会降低攻击者的行贿成本，因为攻击者只需拿出大于$dn/2的资金就可以让每个监督节点收到比它进行警报后得到的奖励还要更多的资金，这也就意味着攻击者只需要网络所有节点押金总和一半以上的资金就可以操作一轮预言机网络的数据投喂，以这样的行贿成本来防止数据操纵攻击相对于Defi(去中心化金融)的攻击收益来说是不太保险的。

因此为了提高攻击者的行贿成本，Chainlink选择将监督奖励集中到一个节点上。在每轮进行数据聚合时，Chainlink网络会随机给所有节点分配一个优先级，当有多个节点发出警报并且警报正确时，则由发出警报的所有节点中优先级最高的节点获得所有奖励，这就意味着即使攻击者买通本轮优先级最高的节点让他不发送警报，但是优先级第二的节点为了获得大量的监督奖励也会向二层网络发出警报，因此在这种机制下如果攻击者想要操作预言机的数据，就需要用高于$dn/2的金额买通所有的节点。




*操作预言机数据成本与Chainlink网络规模的关系（来源：Chainlink Whitepaper V2）*

由于贿赂金额$dn/2会随着Chainlink网络节点规模增大而增大，并且这笔贿赂金需要支付的节点数量也会随着n的增大的变多，这也就意味着攻击者成功操控预言机数据上链的成本和Chainlink网络节点规模之间是指数增长的关系，如上图所示，因此随着Chainlink的网络规模增大，攻击者操控数据的成本就越高，上链的数据就越可信。

# Chainlink二层网络(Second Tier)

上一部分提到当Chainlink一层网络最终聚合出一个错误的数据时，会由二层网络来进行裁定，因此这个二层网络应该是个比一层网络更可靠，并且获取数据成本更高的网络。

Because of the rareness of adjudication and opportunity for extended-time execution noted above, in contrast to the first tier, nodes in the second tier can:

1. Be highly compensated for conducting adjudication.
2. Draw on additional data sources, beyond even the diverse set used by the first-tier.
3. Rely on manual and/or expert inspection and intervention, e.g., to identify and reconcile errors in source data and distinguish between an honest node relaying faulty data and a misbehaving node.

*二层网络的实现部分原文（来源：Chainlink Whitepaper V2）*

白皮书中提到二层网络可以由在一层网络中服务时间最长并且最可靠的节点组成，但是二层网络的节点并不一定需要是一层网络的节点。而与一层网络相比，二层网络的节点可以有下面三个特点：

1.在进行裁决时应该得到高额的奖励；

2.需要使用与一层网络所使用的数据源不同的额外的数据源；

3.依靠人工或专家的检查和干预，来识别和协调源数据中的错误，并区分出诚实节点和恶意节点。

# 补充——Price Feeds数据多层聚合

Chainlink为了获得可信的资产价格数据和避免单点故障，对最终要上链的资产价格数据进行了多层聚合。


*三次数据聚合（来源：blog.chain.link）*

Chainlink对最终要上链的资产价格数据总共进行了三层聚合：

1.在数据源层面聚合。首先数据会在数据源先聚合一次，交易所如Coinbase、Uniswap等会基于交易活动聚合原始市场数据，专业的数据聚合公司(CoinMarketCap、CoinFecko等)会从各个交易平台收集原始数据，并会通过不同平台的交易量生成交易量加权平均价，在这过程中一些无效交易和异常值会被剔除。

2.在节点层面聚合。从这一层开始数据才由Chainlink网络节点接手，每个Chainlink节点都会接入多个优质付费数据聚合商的API接口获取数据并返回这些数据的中位数。在这次聚合中Chainlink节点同样会自动剔除异常值，提升服务可靠性。

3.在预言机网络层面聚合。所有Chainlink节点各自聚合的数据最后会生成一份预言机报告，报告中包含每个节点上传的数据以及签名(异常数据会被剔除，对应节点会被扣除一定量押金)。然后这份报告会被保存在Chainlink的链上预言机合约，当用户想要获取价格数据时，Chainlink的链上预言机合约会取报告中数据集的中位数返回给用户。

一个Chainlink网络中至少需要2/3的节点上传结果和签名该预言机报告才会被链上合约接受，在这过程中还使用了“阈值签名”技术，具体技术细节可以在Chainlink Whitepaper V1中查看。

# 最后

Chainlink白皮书的内容非常多，v2版本的白皮书有136页，这里我只挑出了一些我个人认为比较核心的，且有助于理解整个Chainlink框架的部分进行描述。

其实白皮书中还有一些重要内容，比如公平交易序列(Fair Transaction Sequencing)和交易执行框架(Transaction-Execution Framework)，这两个部分的内容简单来说就是利用预言机网络本身来提前处理交易，将一些复杂的操作从执行效率低的公链(如以太坊)转移到执行效率高且获取数据成本更低的预言机网络执行，最后将交易结果按照提交到预言机网络的时间顺序打包并上链，从而实现交易上链顺序的公平性进而了避免“gas war”，并且提高了公链处理业务的效率。个人认为这两个方案更偏向于“layer-2”技术，所以就不在这里过多说明，而其他一些诸如节点声誉(Node Reputation)和信任最小化(Trust Minimization)等内容也不再做展开。

本文的内容是我阅读完白皮书后写下的一些记录，其中一些白皮书中没有过多涉及的内容则是我自行搜索资料阅读源码并进行逻辑自洽后自行补充的，所以难免会有错误的地方，欢迎各位大佬指正，我会及时修改。