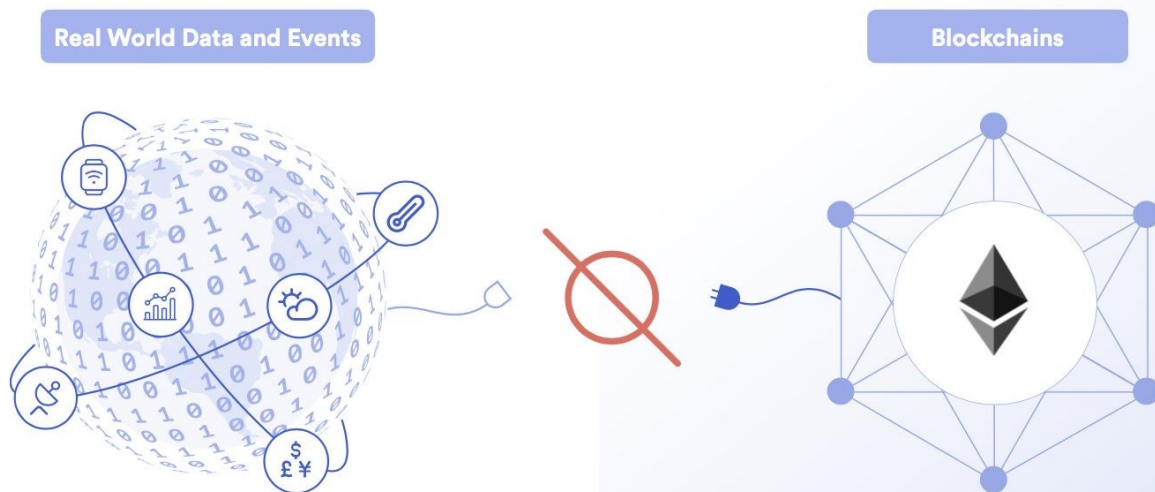




Chainlink 原理

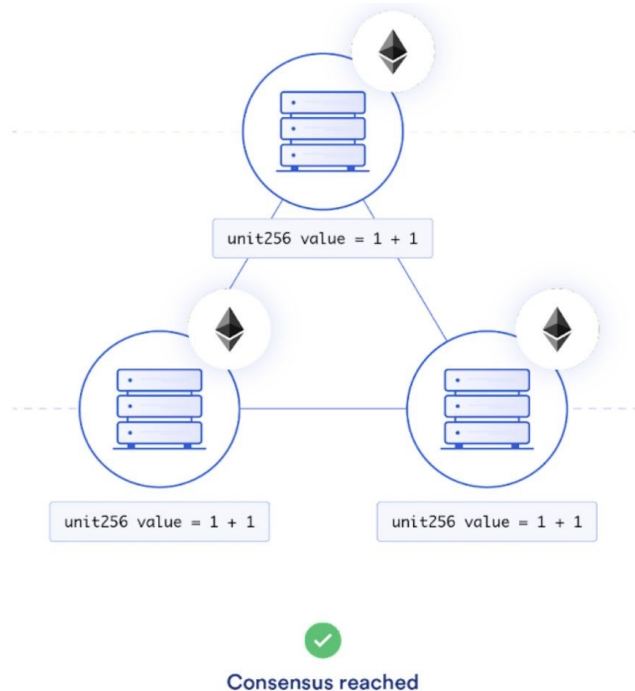
预言机问题

智能合约无法获取区块链以外的数据，外部 API 提供的数据和任何其他链下资源都无法获取。

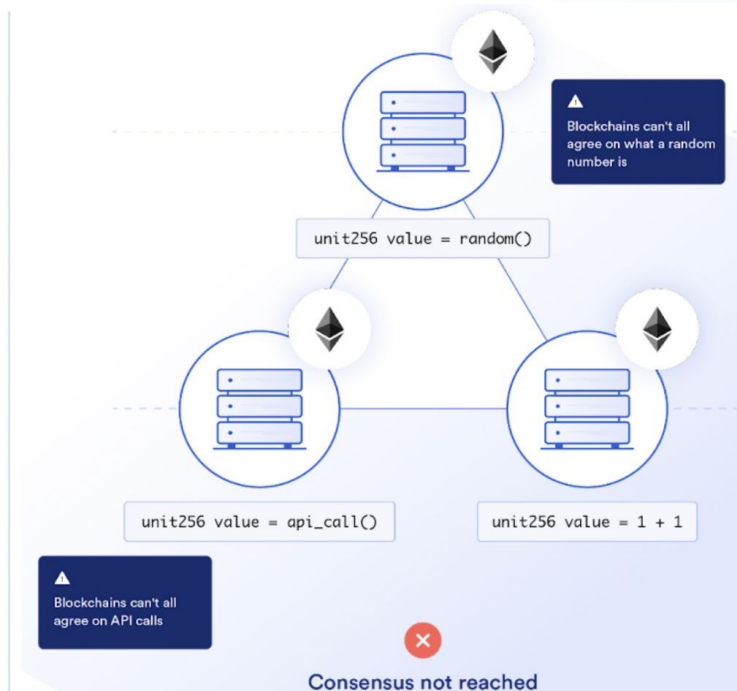


非确定性操作

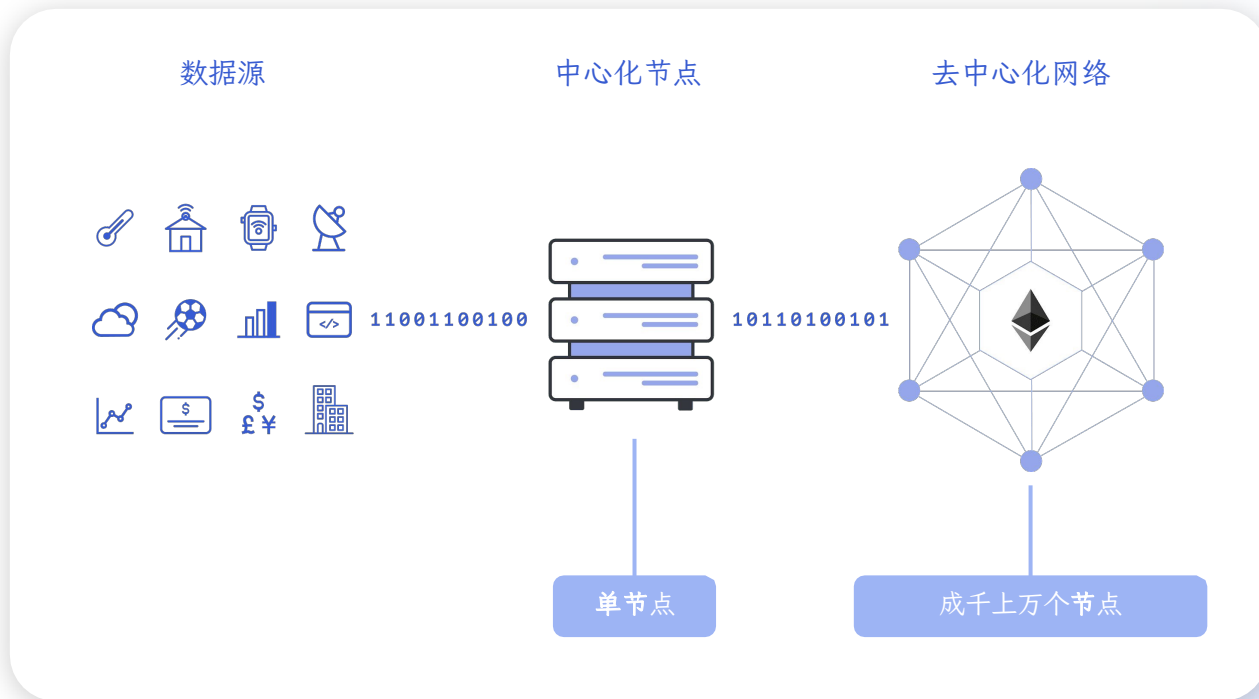
确定性操作



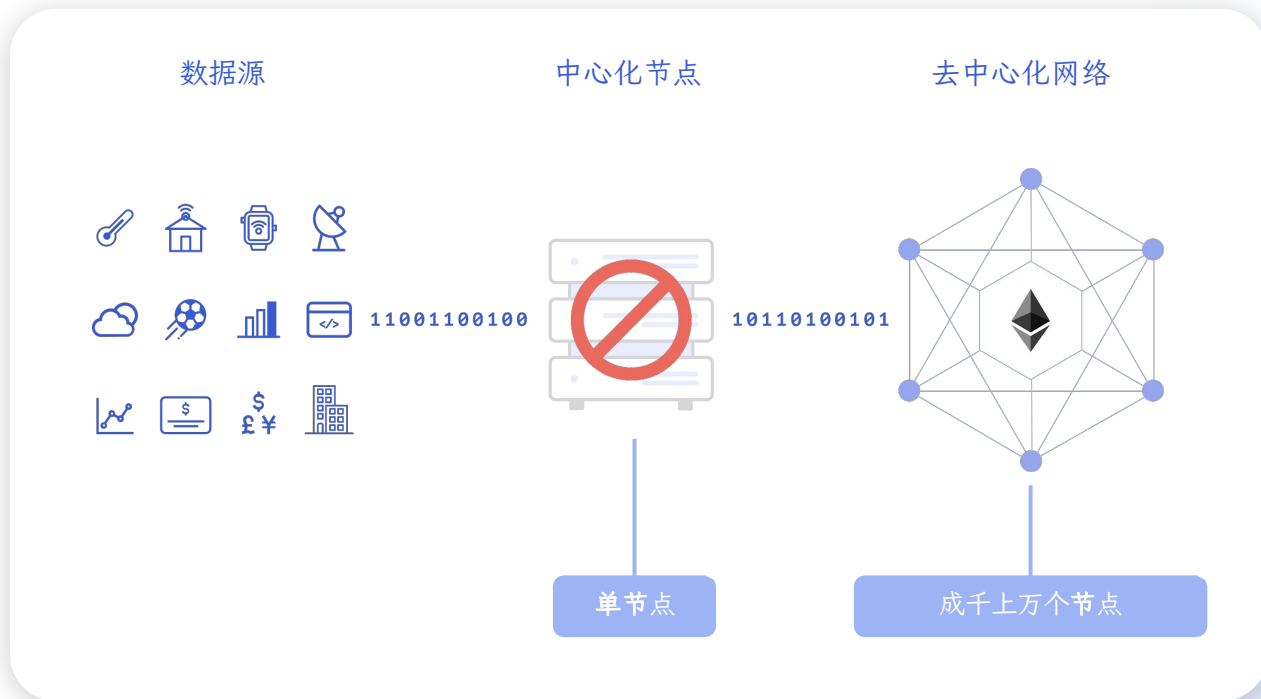
非确定性操作



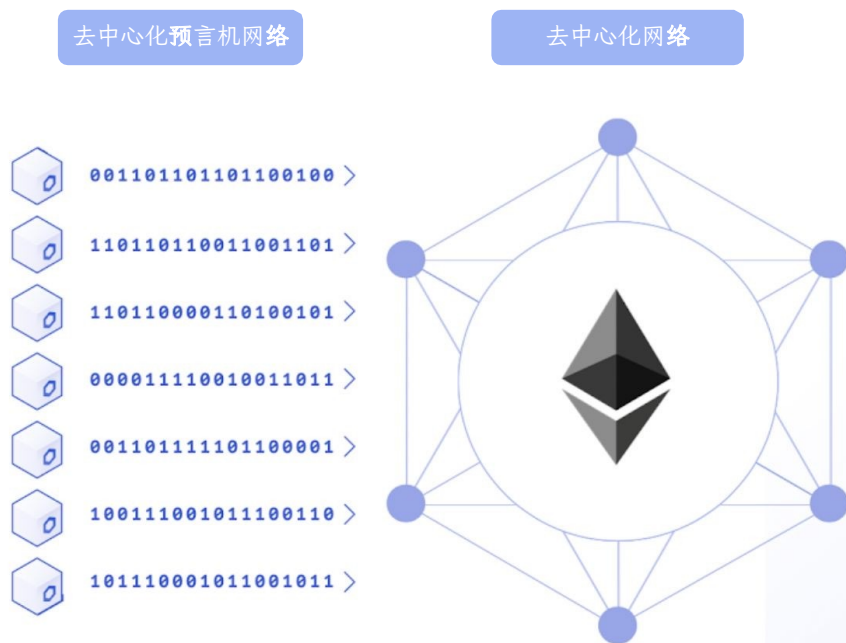
中心化预言机工作流程



中心化预言机单点失败风险



去中心化预言机网络



去中心化预言机网络

多个数据节点形成去中心预言机网络，每个节点都会收集数据，达成共识后输入到区块链上的智能合约

1. 技术上，避免了单点失败风险
2. 数据上，通过网络对多个数据源进行验证



Chainlink Data Feed

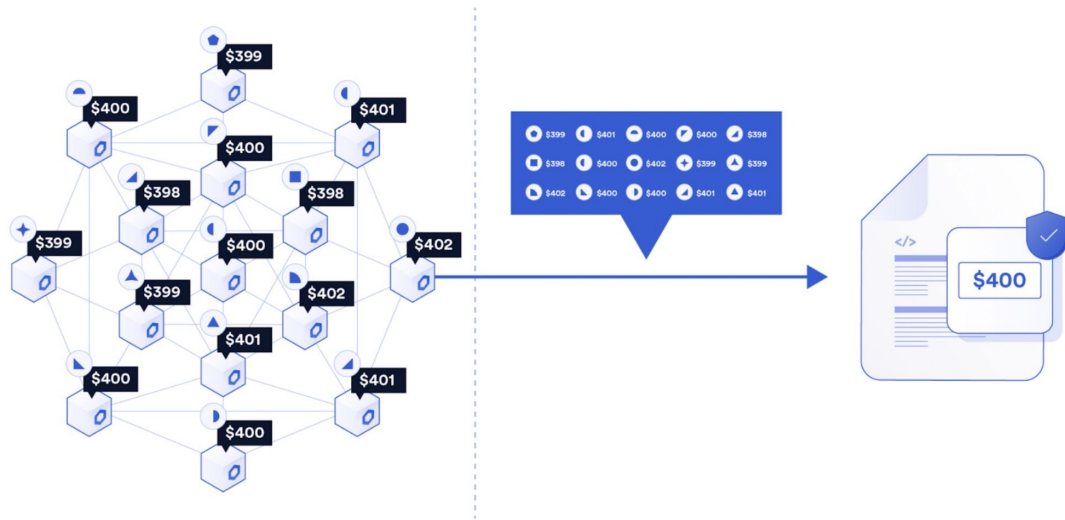
Chainlink Data Feed

去中心化预言机网络

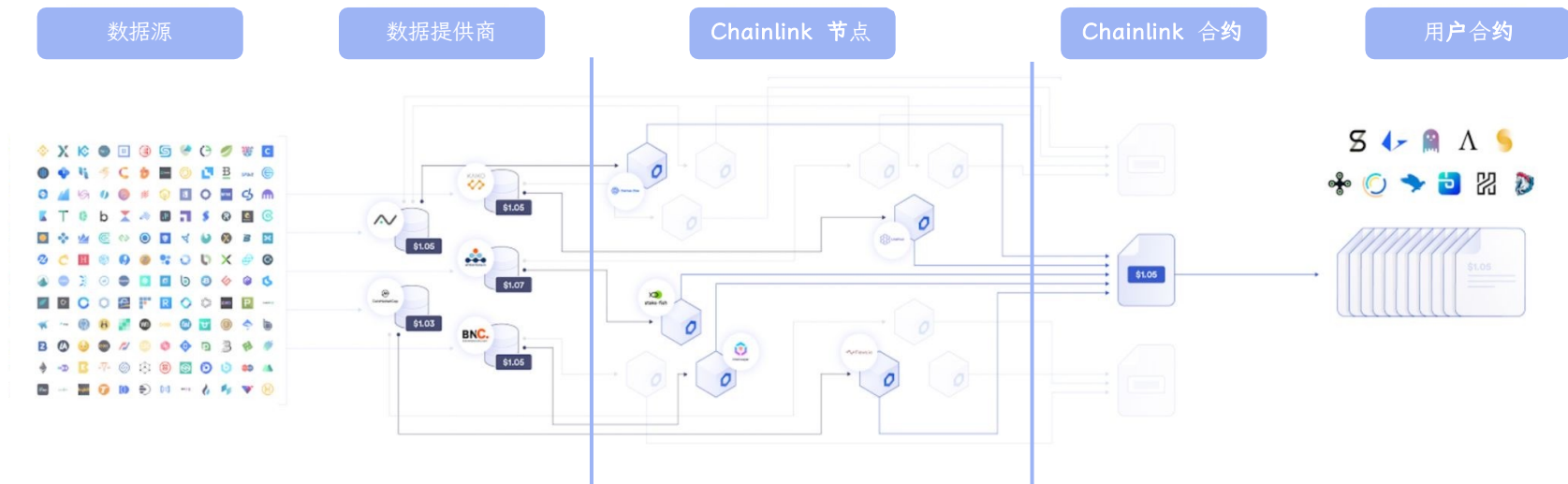
Chainlink 预言机节点

区块链网络

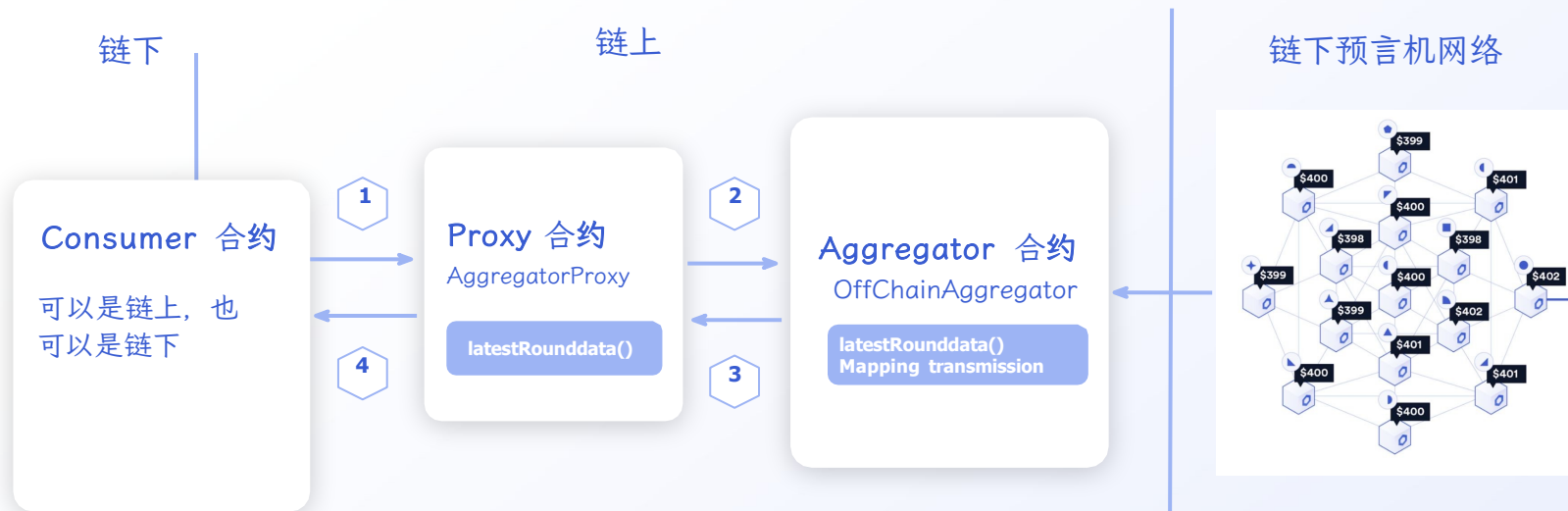
智能合约



Chainlink Data Feed 业务流程



Chainlink Data Feed 技术架构



1. Consumer 调用 Proxy 合约函数 `latestRoundData()`
2. Proxy 调用 Aggregator 合约函数 `latestRoundData()`
3. Aggregator 返回 Transmission 结果给 Proxy
4. Proxy 返回结果给 Consumer

Chainlink 预言机网络更新 Aggregator 合约中的价格信息

1. 每 30 分钟会更新一次
2. 通证价格波动超过 0.5%

Chainlink Data Feed 用户案例



Lending and borrowing

Issue and settle loans, liquidate undercollateralized positions, trigger collateral swaps, and help protect against insolvency.






Mirrored assets

Generate mirrored versions of real-world and on-chain assets using on-chain collateral and Price Feeds as the reference point for minting and redemption.





Stablecoins

Use financial market data to determine the collateralization of stablecoins, automate mint/burn operations, and trigger rebasing functions.






Asset management

Enable the automated management of capital pools and the marking of funds to market by referring to Price Feeds for rebalances.






Options and futures

Power advanced financial instruments and ensure platform solvency by dynamically setting the funding rate and settling agreements.

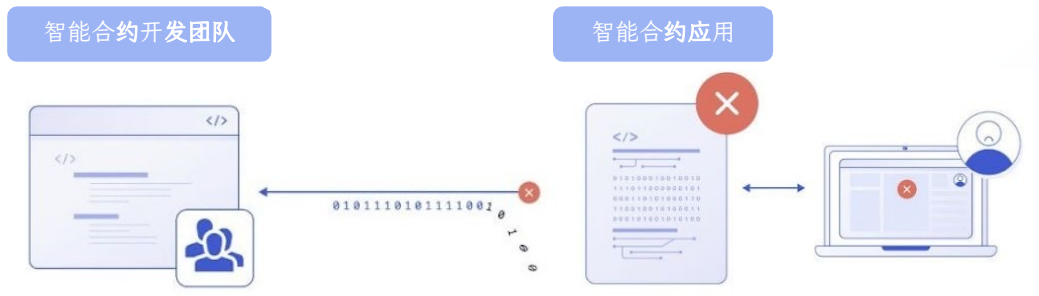






Chainlink Keepers

合约自动化执行



1

手动 DevOp & 中心化服务器

开发者人员通过一个中心化服务器去执行 Solidity 的 Cron job, 监控合约状态, 并且发送交易给链上合约。

- 单点失败风险
- 占用团队时间和资源

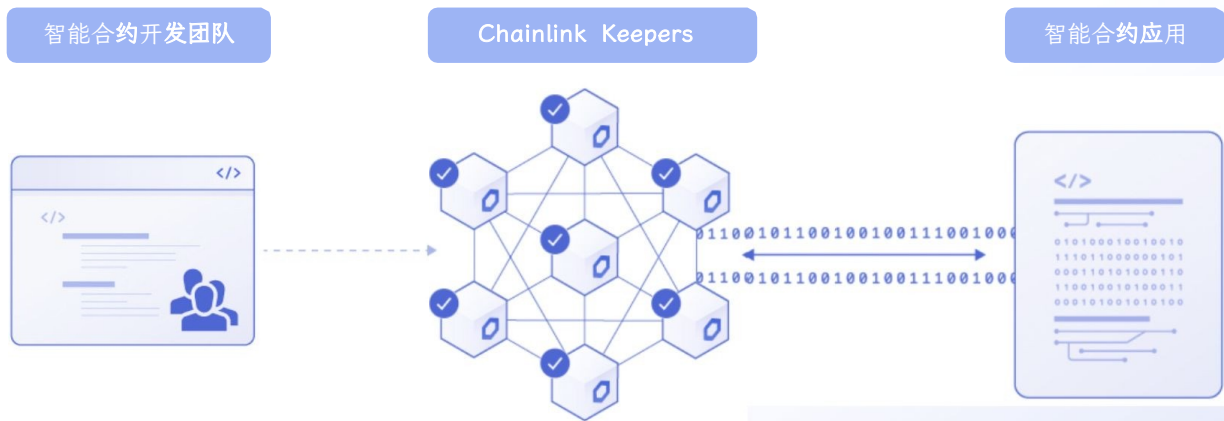
2

Bounty 模式(ETH Alarm clock)

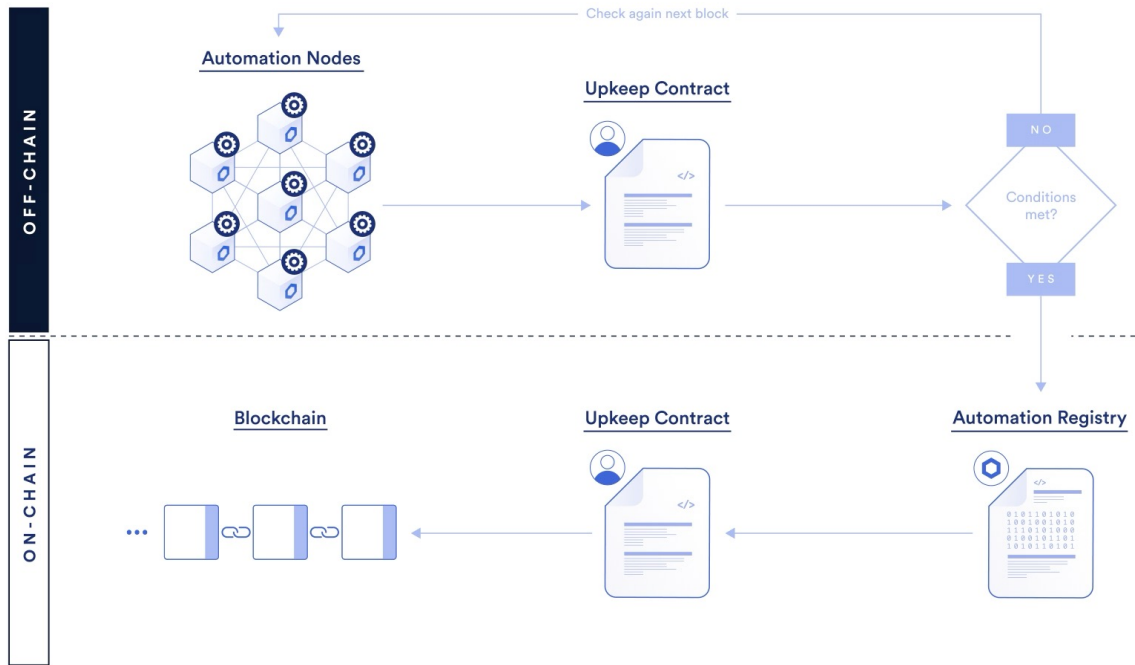
给交易触发的个人账户提供赏金, 交易执行成功即可获得经济激励。

- winner-takes-all reward
- 增加链的拥挤程度
- 没有 direct commitment

Chainlink Keepers (合约自动化执行工具)

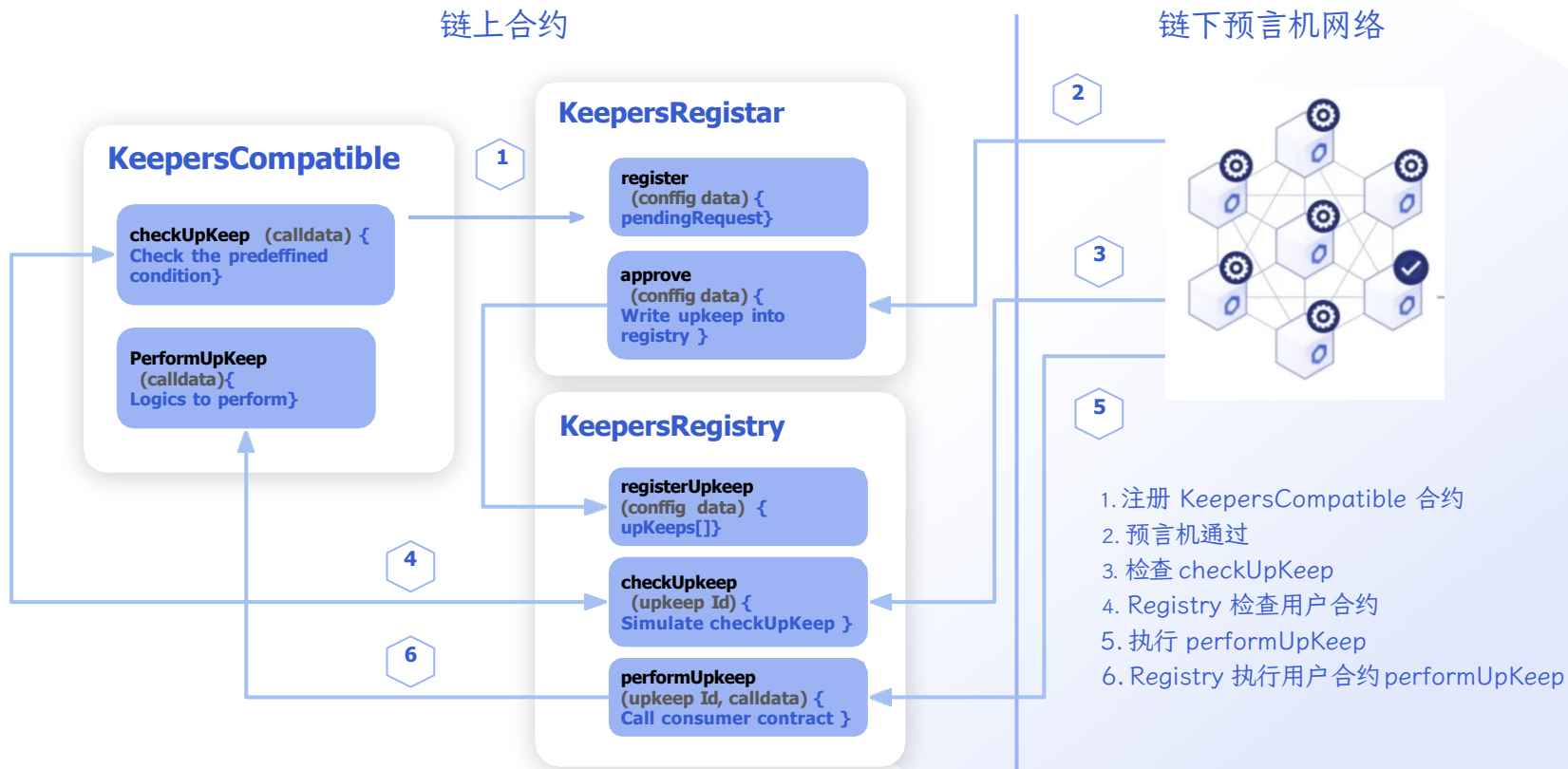


Chainlink Keepers 业务流程



1. `eth_call`, 判断是否满足条件
2. 下个区块继续检查
3. 调用 `keepers` 注册合约
4. `Keepers` 注册合约调用用户合约

Chainlink Keepers 技术架构



Chainlink Keepers 使用场景

自动复利 & yield (Yield Harvesting and Compounding)



流动性管理 (Liquidity management)



借贷平台清算 (Liquidation)



跨链 NFT 铸造 (cross-chain minting)



DEX 限价单 (DEX limit orders)

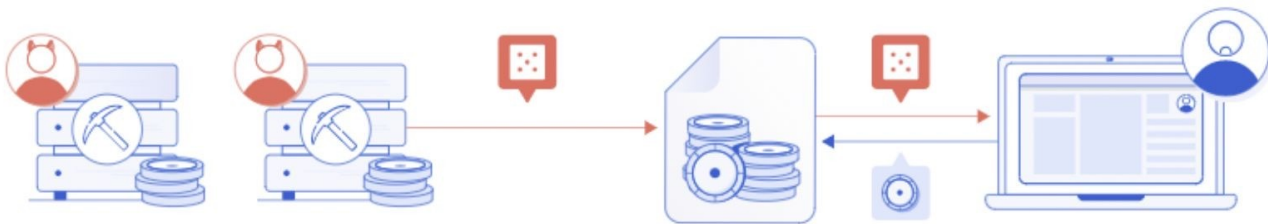


动态 NFT (Dynamic NFT)



Chainlink VRF

随机数生成器 (RNG) - 链上方案



1

恶意矿工选择性打包
交易

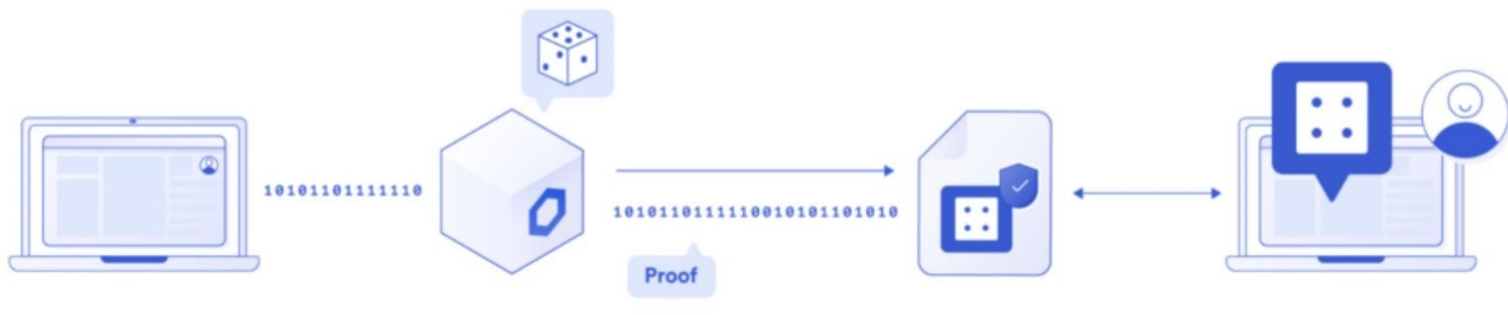
随机数函数的方案，所以依赖的随机数，比如说区块哈希有可能被矿工操纵

2

合约中的随机数生成
被所操纵区块影响

被操纵的随机数会影响到智能合约输入，进而又可能导致用户资产的损失

随机数生成器 (RNG) - 预言机方案



1

用户合约给预言机发送随机数请求

2

预言机获取种子, 生成随机数以及相关的Proof

3

VRF 合约验证随机数是否由预言机按照约定算法生成

4

用户合约接受已验证的随机数

可验证随机数函数 (VRF)

可验证随机数(VRF)定义：

In [cryptography](#), a verifiable random function (VRF) is a public-key [pseudorandom function](#) that provides proofs that its outputs were calculated correctly.

1. 可证明性(**Provability**)
2. 独特性(**Uniqueness**)
3. 伪随机性(**Pseudorandomness**)

VRF 是由 3 个函数组成：

1. 密钥生成函数(**Key Gen**)

$G(r) \Rightarrow (PK, SK)$

PK: public key, 公钥

SK: secret key, 密钥

2. 随机数生成函数(**Evaluate**)

$E(SK, seed) \Rightarrow (Randomness, Proof)$

seed: RNG的种子

Randomness: 随机数

Proof: 证明

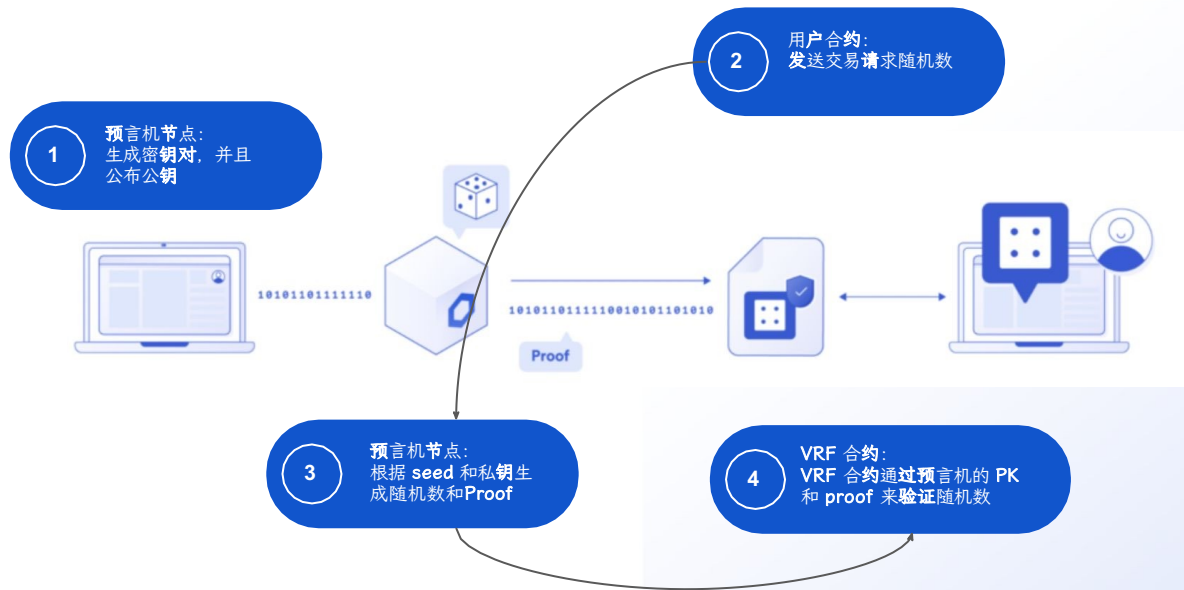
3. 验证函数(**Verify**)

$V(PK, seed, Randomness, Proof) \Rightarrow (true \text{ or } false)$

true: 验证成功

false: 验证失败

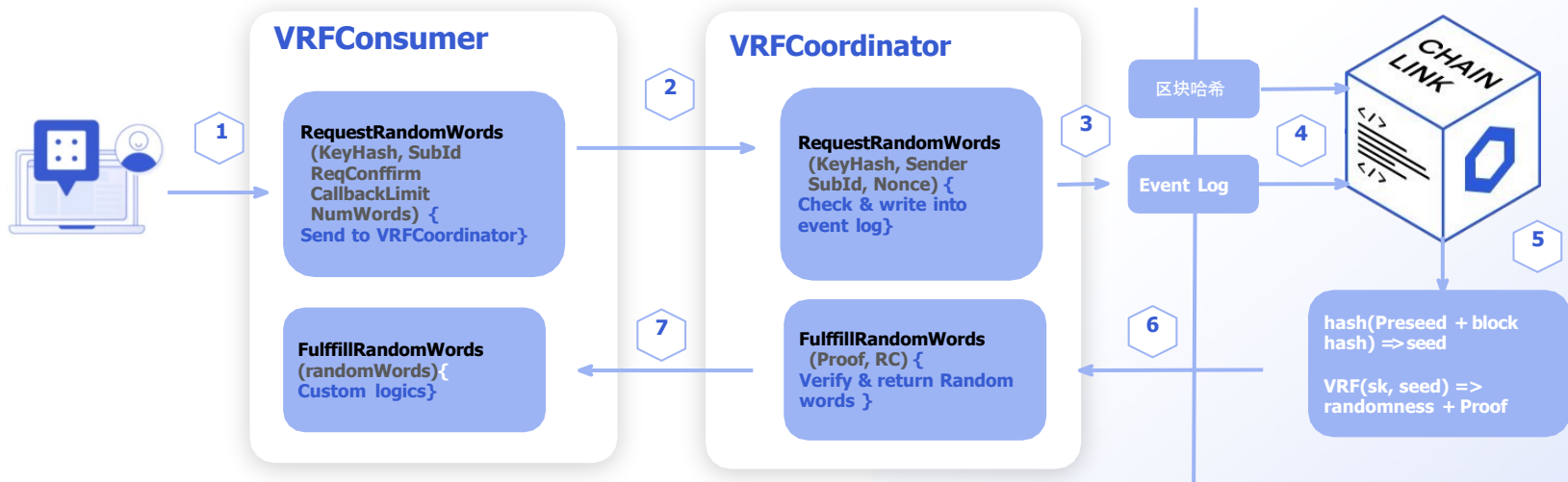
Chainlink VRF 业务流程



Chainlink VRF 技术架构

链上

链下预言机节点



1. 调用 Consumer 合约的函数请求随机数
2. 用户合约调用 Coordinator 合约的函数请求随机数
3. 将 PreSeed 写入 Event log
4. 预言机读取 Event log 中的 PreSeed 和 blockhash

5. 预言机通过 VRF 生成随机数和 Proof
6. 预言机将 rc 和 proof 写入 Coordinator
7. Coordinator 进行验证 & 将随机数写入 Consumer 合约

Chainlink VRF 使用场景

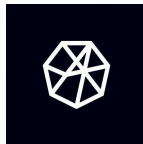
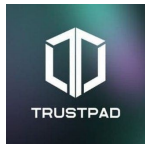
NFT 创建和分发(NFT Creation & Distribution)

- 通过 VRF 给要生成的NFT 分配随机属性
- 给 NFT collection 参与者随机分配稀有NFT



公平抽奖(Fair selecting)

- 给孵化项目的参与者发放白名单



查看 35 个应用场景

<https://blog.chain.link/blockchain-rng-use-cases-enabled-by-chainlink-vrf/>



Chainlink 学习资料

学习资料

NEW

Registration for SmartCon 2022 is now open. [Secure your spot.](#)

Chainlink Architecture

Basic Request Model

Decentralized Data Model

Off-Chain Reporting

DATA FEEDS

Introduction to Data Feeds

Using Data Feeds

Historical Price Data

Feed Registry

API Reference

Using ENS with Data Feeds

Contract Addresses

Ethereum Data Feeds

BNB Chain Data Feeds

Polygon (Matic) Data Feeds

Gnosis Chain (xDai) Data Feeds

HECO Chain Data Feeds

Avalanche Data Feeds

Fantom Data Feeds

Arbitrum Data Feeds

Harmony Data Feeds

Optimism Data Feeds

```
AggregatorV3Interface internal priceFeed;

/**
 * Network: Rinkeby
 * Aggregator: ETH/USD
 * Address: 0x8A753747A1Fa494EC906cE90E9F37563A8AF630e
 */
constructor() {
    priceFeed = AggregatorV3Interface(0x8A753747A1Fa494EC906cE90E9F37563A8AF630e);
}

/**
 * Returns the latest price
 */
function getLatestPrice() public view returns (int) {
    (
        /*uint80 roundID*/,
        int price,
        /*uint startedAt*/,
        /*uint timeStamp*/,
        /*uint80 answeredInRound*/
    ) = priceFeed.latestRoundData();
    return price;
}
```

NEW

Registration for SmartCon 2022 is now open. [Secure your spot.](#)

Chainlink's Request & Receive Data cycle and receive a single response.

Table of Contents

- Example
- Response Types
- Setting the LINK token address, Oracle, and JobID

Example

This example shows how to:

- Fetch a single word response in a single call.

The `Cryptocompare GET /data/pricemultifull` API returns the current trading info (price, vol, open, high, low) of any list of cryptocurrencies in any other currency that you need. To check the response, you can directly paste the following URL in your browser: `https://min-api.cryptocompare.com/data/pricemultifull?fsyms=ETH&tsyms=USD` or run this command in your terminal:

```
curl -X 'GET' \
'https://min-api.cryptocompare.com/data/pricemultifull?fsyms=ETH&tsyms=USD' \
-H 'accept: application/json'
```

The response should be similar to the following example:

```
{
  "RAW": {
    "ETH": {
      "USD": {
```

<https://docs.chain.link/>

学习资料

登链社区

首页 文章 问答 讲堂 专栏 招聘 文档 集市

搜一搜，发现更多精彩内容

我的主页

我的回答

我的提问

我的文章

我的课程

我的专栏

我的学分

我的贡献

我的粉丝

我的关注

我的收藏

最近动态

1天前 发表了文章

十大DeFi安全最佳实践

1天前 发表了文章

学习 Solidity, 全栈 Web3, Javascript 和区块链开发的教程

2022-07-06 11:58 发表了文章

一文读懂元宇宙

2022-06-21 18:17 发表了文章

Chainlink预言机在智能合约中的77种应用方式 (三)

2022-06-16 16:41 发表了文章

如何创建加密货币

2022-06-08 08:56 发表了文章

如何创建NFT

2022-06-06 11:44 发表了文章

一文读懂去中心化交易平台 (DEX)

2022-05-31 09:01 发表了文章

Chainlink预言机在智能合约中的77种应用方式 (二)

2022-05-25 09:08 发表了文章

Chainlink预言机在智能合约中的77种应用方式 (一)

问题 chainlink 节点运行错误

* 部署文档 https://docs.chainlink.com/docs/running-a-chainlink-node[https://docs.chainlink.com/docs/running-a-chainlink-node] * 错误信息 `` 2021-02-02T01:20:07Z [FATAL] Unable to initialize ORM: dial tcp 127.0.0.1:5432: connect: connection refused unable to open postgres://postgres:12...

问题 如何使用chainlink外部适配器获取股票数据？

**如何使用chainlink外部适配器获取股票数据？ ** 我想通过chainlink 外部适配器去抓取 https://finance.sina.com.cn 中的股票数据同步到eth链上，我应该怎么做？

问题 Chainlink预言机问题

有两个疑惑。第一个，chainlink的返回的这5个字段分别代表什么意思。在文档中没找到解释： `` (uint80 roundID, int price, uint startedAt, uint timeStamp, uint80 answeredInRound) = pri...

问题 Chainlink节点部署中，遇到ETH ChainID与Config.ChainID不吻合的问题

I have an issue while launching my chainlink node with the ethereum client service (infura.io). I have this warning : `` Failed to connect to ethereum node wss://kovan.infura.io/ws/v3/ services/head_tracker.go:288 err=verifyEthereumChainID failed: ethereum ChainID doesn't match chainlink c...

问题 chainlink 节点job是怎么把数据同步上链的？

chainlink 节点job是怎么把数据同步上链的？

问题 chainlink 生成的随机数如何是验证？

chainlink 生成的随机数如何是验证？

登链社区

<https://learnblockchain.cn/people/398>