VRF（可验证随机数）

1.随机数生成器（RNG）- 预言机方案

- 用户合约给预言机发送随机数请求
- 预言机获取种子，生成随机数以及相关的Proof
- VRF合约验证随机数是否由预言机按照约定算法生成
- 用户合约接受已验证的随机数

2.可验证随机输（VRF）定义

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

3.VRF是由3个函数组成

1.密钥生成函数（Key Gen）

G(r) =>(PK,SK)

PK:public key，公钥

SK:secret key，密钥

2.随机数生成函数（Evaluate）

E(SK,seed) => (Randomness,Proof)

seed:RNG的种子

Randomness：随机数

Proof：证明

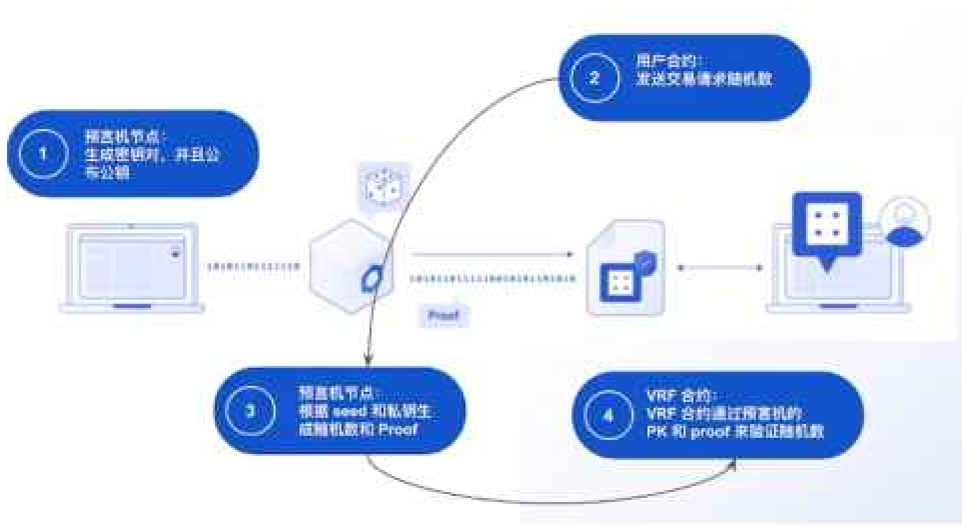
3.验证函数（Verify）

V(PK,seed,Randomness,Proof) =>(true or false)

true：验证成功

False：验证失败

4.Chainlink VRF 业务流程


1 预言机节点：生成密钥对，并且公布公钥
2 用户合约：发送交易请求随机数
Proof
3 预言机节点：根据 seed 和私钥生成随机数和 Proof
4 VRF 合约：VRF 合约通过预言机的 PK 和 proof 来验证随机数

1.预言机节点：生成密钥对，并且公布公钥

2.用户合约：发送交易请求随机数

3.预言机节点：根据seed和私钥生成随机数和Proof

4.VRF合约：VRF合约通过预言机的PK和proof来验证随机数

5.Chainlink VRF 技术架构

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

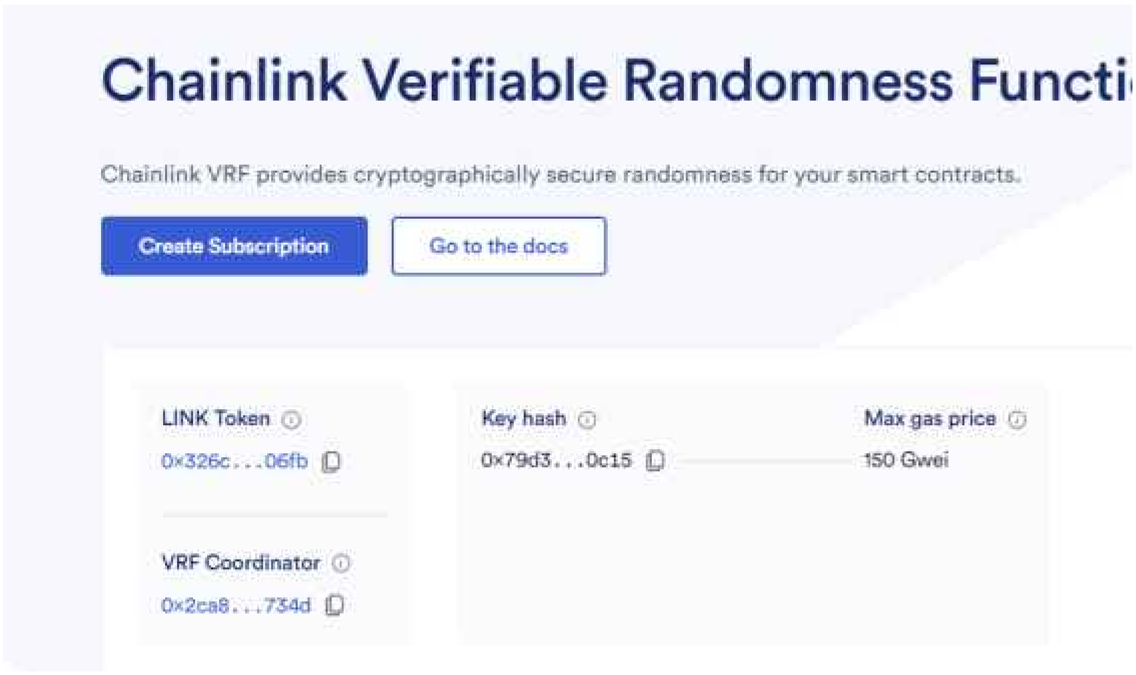
6.VRF演示

Remix

1.注册VRF

地址:

https://vrf.chain.link/


Chainlink Verifiable Randomness Functi

Chainlink VRF provides cryptographically secure randomness for your smart contracts.

LINK Token 0x326c...06fb

VRF Coordinator 0x2ca8...734d

Key hash 0x79d3...0c15

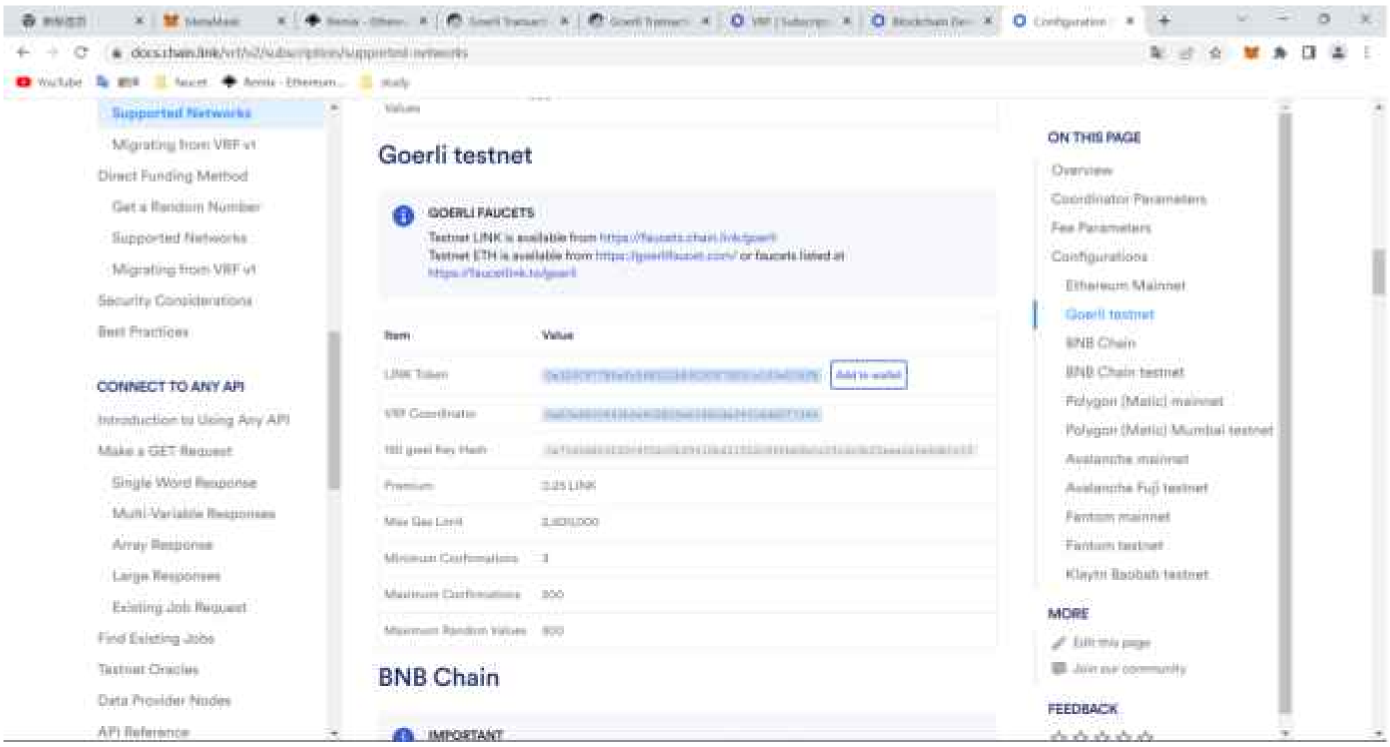
Max gas price 150 Gwei

2.合约代码

```
// SPDX-License-Identifier: GPL-3.0

pragma solidity ^0.8.7;

import "@chainlink/contracts/src/v0.8/interfaces/VRFCoordinatorV2Interface.sol";
import "@chainlink/contracts/src/v0.8/VRFConsumerBaseV2.sol";

contract ChainlinkVRFDemo is VRFConsumerBaseV2 {
  VRFCoordinatorV2Interface COORDINATOR;
  address vrfcoordinatorAddr =0x2Ca8E0C643bDe4C2E08ab1fA0da3401AdAD7734D;

  bytes32 keyHash =
0x79d3d8832d904592c0bf9818b621522c988bb8b0c05cdc3b15aea1b6e8db0c15;
  uint64 s_subId;
  uint16 minimumRequestConfirmations = 3;
  uint32 callbackGasLimit = 200000;
  uint32 numWords = 5;
  address owner;

  uint256[] public s_randomWords;
  uint256 public requestId;

  constructor(uint64 subId) VRFConsumerBaseV2(vrfcoordinatorAddr){
    COORDINATOR = VRFCoordinatorV2Interface(vrfcoordinatorAddr);
    s_subId= subId;
    owner = msg.sender;
  }

  function requestRandomWords() external {
    require(msg.sender == owner);
    COORDINATOR.requestRandomWords(
      keyHash,
      s_subId,
      minimumRequestConfirmations,
      callbackGasLimit,
      numWords
    );

  }

  function fulfillRandomWords(uint256 requestId, uint256[] memory randomWords)
internal override{
        s_randomWords = randomWords;
    }

}
```

vrfcoordinatorAddr地址：0x2Ca8E0C643bDe4C2E08ab1fA0da3401AdAD7734D

Keyhash地址：0x79d3d8832d904592c0bf9818b621522c988bb8b0c05cdc3b15aea1b6e8db0c15


docs.chain.link/vrf/v2/subscription/supported-networks

Supported Networks
Migrating from VRF v1
Direct Funding Method
Get a Random Number
Supported Networks
Migrating from VRF v1
Security Considerations
Best Practices

CONNECT TO ANY API
Introduction to Using Any API
Make a GET Request
Single Word Response
Multi-Variable Responses
Array Response
Large Responses
Existing Job Request
Find Existing Jobs
Testnet Oracles
Data Provider Nodes
API Reference

Goerli testnet

GOERLI FAUCETS
Testnet LINK is available from https://faucets.chain.link/goerli
Testnet ETH is available from https://goerlifaucet.com/ or faucets listed at https://faucetlink.to/goerli

Item	Value
LINK Token	[illegible]
VRF Coordinator	[illegible]
150 gwei Key Hash	[illegible]
Premium	0.25 LINK
Max Gas Limit	2,500,000
Minimum Confirmations	3
Maximum Confirmations	200
Maximum Random Values	500

BNB Chain

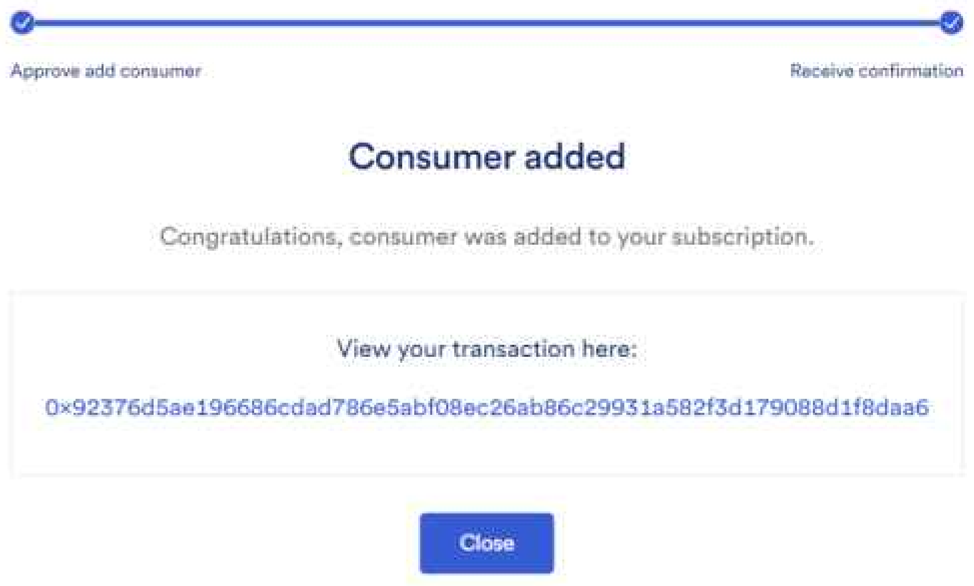
IMPORTANT

ON THIS PAGE
Overview
Coordinator Parameters
Fee Parameters
Configurations
Ethereum Mainnet
Goerli testnet
BNB Chain
BNB Chain testnet
Polygon (Matic) mainnet
Polygon (Matic) Mumbai testnet
Avalanche mainnet
Avalanche Fuji testnet
Fantom mainnet
Fantom testnet
Klaytn Baobab testnet

MORE
Edit this page
Join our community

FEEDBACK

3.将用户合约加入到订阅

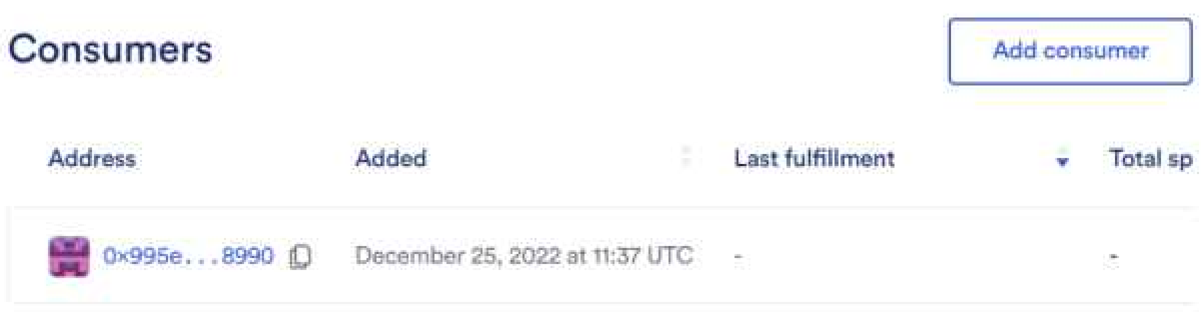


Consumer added

Congratulations, consumer was added to your subscription.

View your transaction here:

0x92376d5ae196686cdad786e5abf08ec26ab86c29931a582f3d179088d1f8daa6

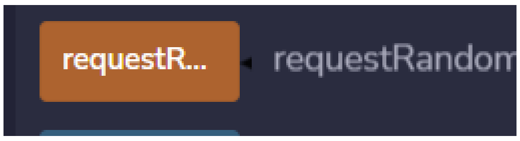


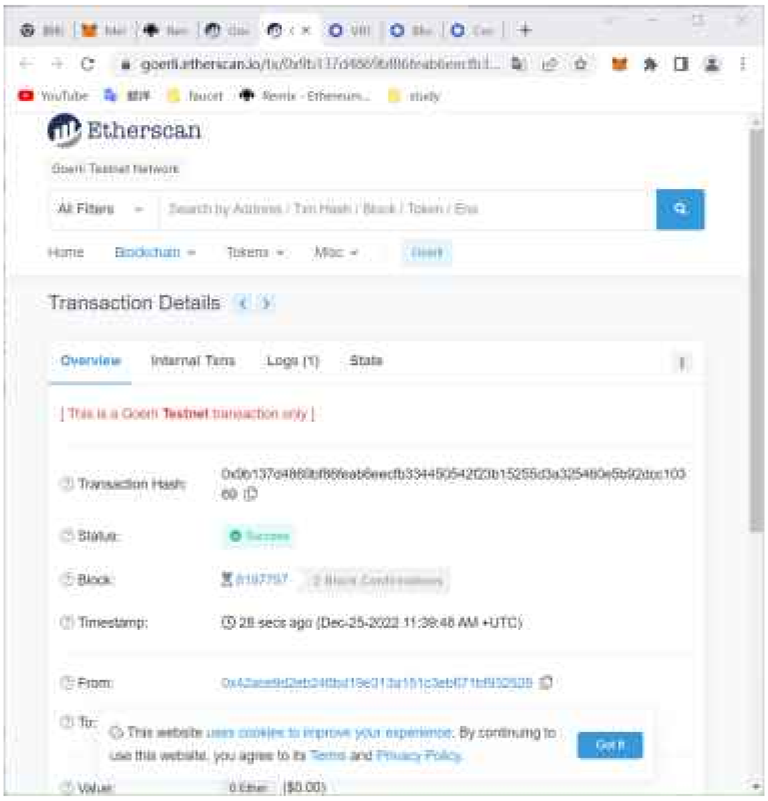
Consumers

Add consumer

Address	Added	Last fulfillment	Total sp
0x995e...8990	December 25, 2022 at 11:37 UTC	-	-

4.通过用户合约调用VRF合约请求随机数





Transaction Details

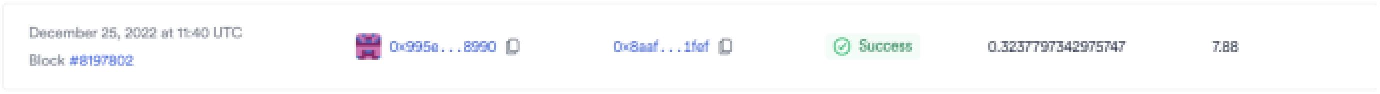
Overview | Internal Txns | Logs (1) | State

[This is a Goerli **Testnet** transaction only]

Transaction Hash:	0x0b137d4869bf86feab6eecfb33445054223b15255d3a325480e5b92dcc103 60
Status:	Success
Block:	8197797 2 Block Confirmations
Timestamp:	28 secs ago (Dec-25-2022 11:39:48 AM +UTC)
From:	0x42aca9d2eb246bd19e313a151c3eb671bf932529
To:	
Value:	0 Ether ($0.00)

This website uses cookies to improve your experience. By continuing to use this website, you agree to its Terms and Privacy Policy. Got it

5.接收随机数



December 25, 2022 at 11:40 UTC Block #8197802	0x995e...8990	0x8aaf...1fef	Success	0.3237797342975747	7.88