OpenZeppelin

How PoolTogether Automates Operations

using OpenZeppelin Defender

Aodhgan Gleeson
aodhgan@pooltogether.com
@gawnieg

Santiago Palladino
santiago@openzeppelin.com
@smpalladino

OpenZeppelin

Our mission is to protect the open economy

OpenZeppelin is a software company that provides **security audits** and **products** for decentralized systems.

Projects from any size -from new startups to established organizations- trust OpenZeppelin to build, inspect and connect to the open economy.

Security, Reliability and Risk Management

OpenZeppelin provides a complete suite of **security and reliability products** to build, manage, and inspect all aspects of software development and operations for Ethereum projects.

Contracts

4+ million downloads

Build

Security and Reliability

Inspect

Manage

Audits

200+ audits

Defender

Automated Monitoring of Balances using Defender Sentinels

Santiago Palladino
santiago@openzeppelin.com
@smpalladino

Aodhgan Gleeson
aodhgan@pooltogether.com
@gawnieg

OpenZeppelin

Our mission is to protect the open economy

OpenZeppelin is a software company that provides **security audits** and **products** for decentralized systems.

Projects from any size -from new startups to established organizations- trust OpenZeppelin to build, inspect and connect to the open economy.

Security, Reliability and Risk Management

OpenZeppelin provides a complete suite of **security and reliability products** to build, manage, and inspect all aspects of software development and operations for Ethereum projects.

Contracts

4+ million downloads

Build

Security and Reliability

Inspect

Manage

Audits

200+ audits

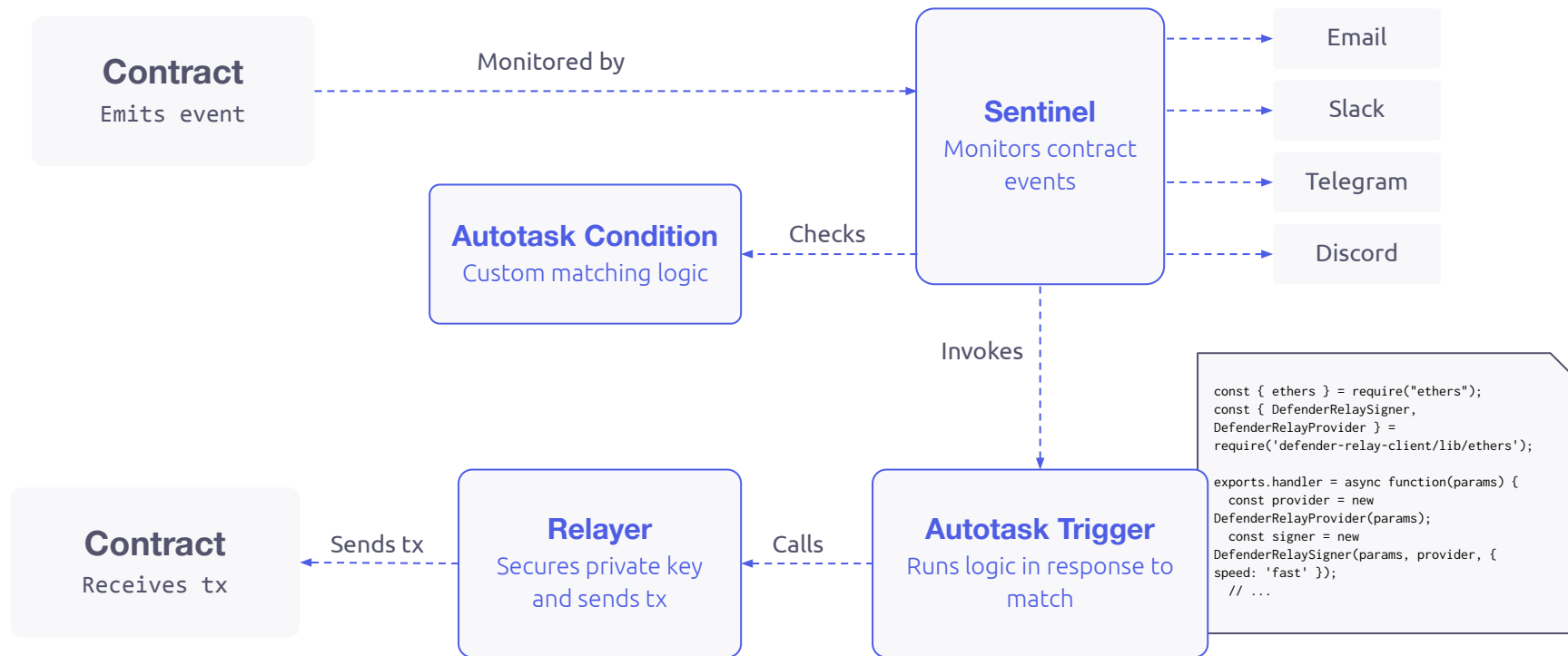**Defender**

Automating operations on Smart Contracts

Monitoring, sending txs, automation, administration, etc

Defender Components for this workshop

- **Sentinels** for monitoring transactions

- **Autotasks** for complex matching and triggers

- **Relayers** for sending transactions

OpenZeppelin

Sentinel → Autotask → Relayer

Contract
`Emits event`

Monitored by

Sentinel
Monitors contract events

Email

Slack

Telegram

Discord

Autotask Condition
Custom matching logic

Checks

Invokes

```
const { ethers } = require("ethers");
const { DefenderRelaySigner,
DefenderRelayProvider } =
require('defender-relay-client/lib/ethers');

exports.handler = async function(params) {
  const provider = new
DefenderRelayProvider(params);
  const signer = new
DefenderRelaySigner(params, provider, {
speed: 'fast' });
  // ...
```

Contract
`Receives tx`

Sends tx

Relayer
Secures private key and sends tx

Calls

Autotask Trigger
Runs logic in response to match

OpenZeppelin

Thank you!

Learn more

openzeppelin.com/**defender**
forum.openzeppelin.com
docs.openzeppelin.com

Contact

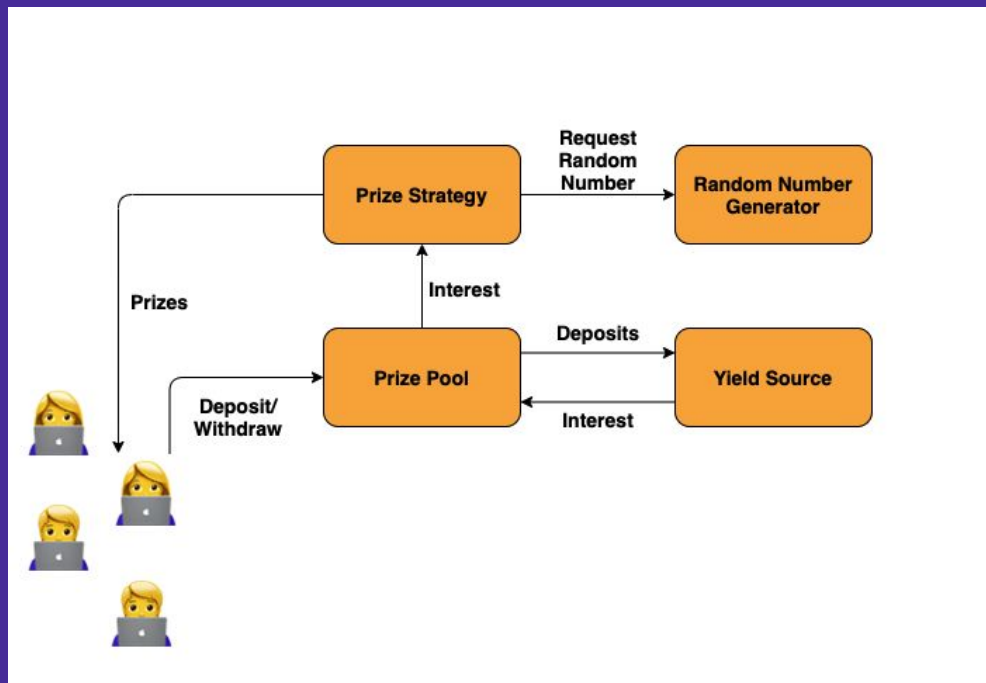🐦 @smpalladino
santiago@openzeppelin.com

Agenda

- **What is PoolTogether?**
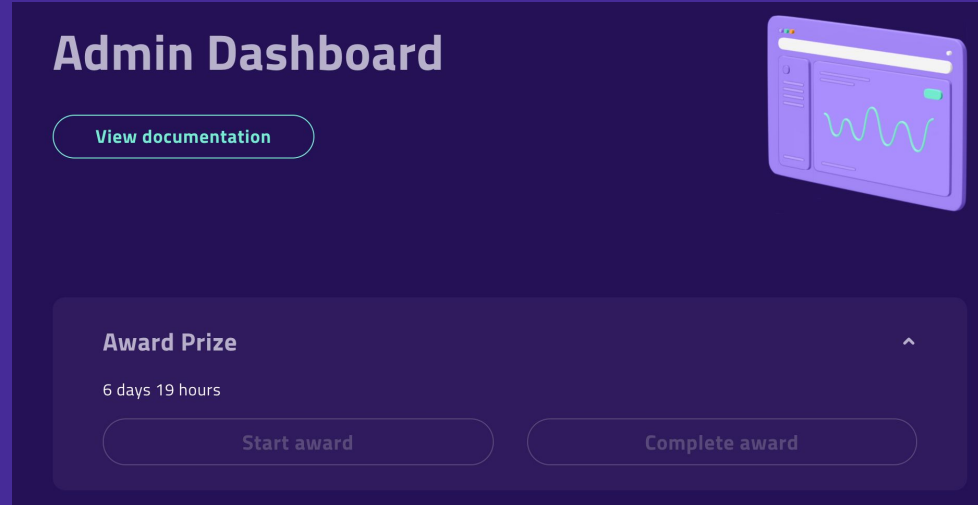
- **The Problems**

- **The Solutions**

What is PoolTogether?

- **Prize Linked Savings protocol**
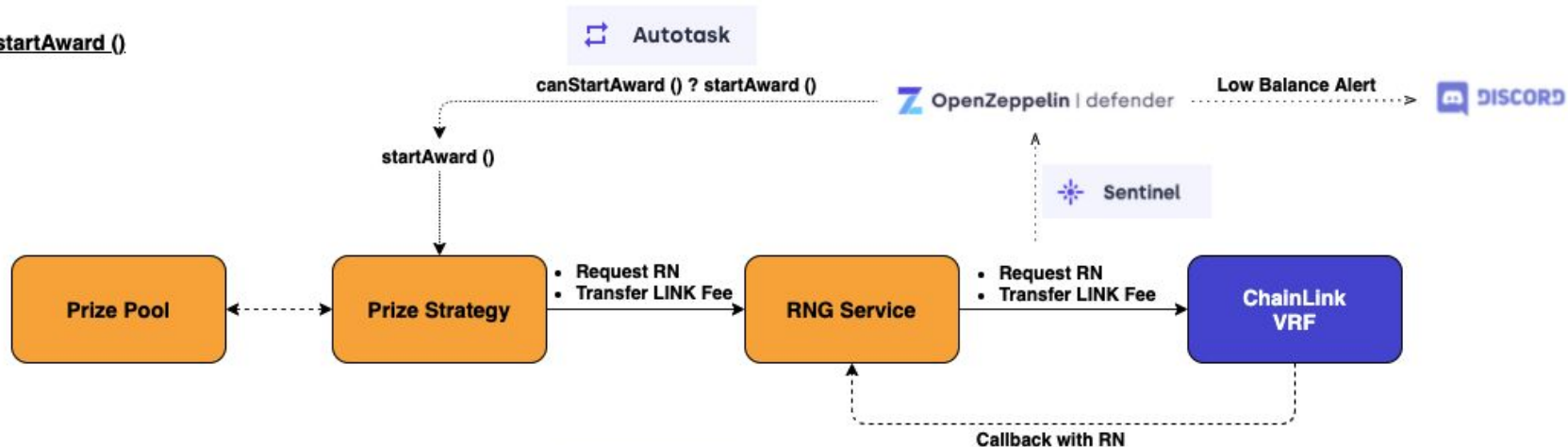- **Periodic randomly draw awards winners**
- **Prize Pool Builder**

Problem

- **Automate Award process**

- **Maintain LINK balance**
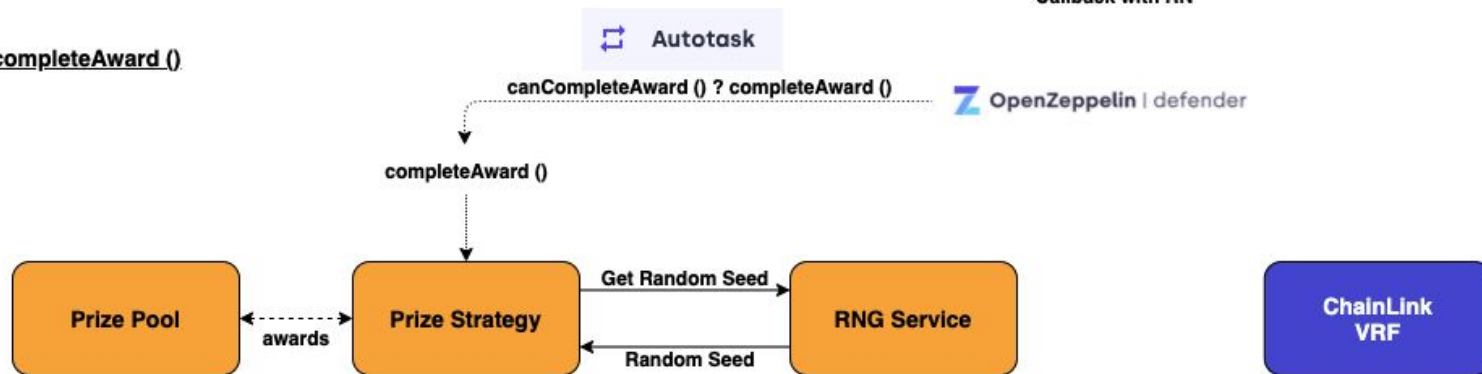
- **Across multiple networks**

- **Using Discord**

Admin Dashboard

View documentation

Award Prize

6 days 19 hours

Start award   Complete award

Award Process - Smart Contract Architecture

LINK Balance Solution: Sentinel + Cond. Autotask

Network

Connect to mainnet for production or pick testnet for development.

PRODUCTION

Mainnet   xDai   Fuse   Binance BSC   Matic

Avalanche C-Chain

TESTNET

Rinkeby   Ropsten   Kovan   Goerli   Sokol

BSC Testnet   Moonbase Alpha   Mumbai   FUJI C-Chain

Address

ChainLink: LINK To...   ✕

ABI

Paste your contract's ABI code here to watch Events and Functions.

```
269      //
270      {
271          "indexed": false,
272          "name": "data",
273          "type": "bytes"
274      }
275      ],
276      "name": "Transfer",
277      "type": "event"
278    },
279    {
280      "anonymous": false,
281      "inputs": [
```

Confirmation Blocks

Confirmed (1 blocks)

Sentinel conditions

Transaction properties

These parameters act as filters, If a transaction match these conditions it will trigger the sentinel.

Variables: value, gasPrice, gasUsed, to, from, nonce, status ('success' or 'failed'), input, or transactionIndex.

Contract conditions

Specify which events and function calls should be monitored. You can monitor specific conditions within events and functions.

EVENTS

☑ `Transfer(address,address,uint256,bytes)`

```
from == "RNG-SERVICE-ADDRESS"
```

Variables: from to value data

☐ `Approval(address,address,uint256)`

Test Sentinel conditions

Backtest this Sentinel by searching for transactions that match the conditions.

Block Range

Specific transaction

Transaction Hash (0x)

Find matches

Autotask condition

If other conditions match, the sentinel will invoke this autotask to further evaluate. The Autotask is given a list of events and should returns all hashes that it decides to match. Autotasks may optionally add metadata per transaction which is included in notifications. Errors are treated as a non-match. Read more here.

Check Link Balance   ✕

Previous   **Next**

Implementation Notes

- **Conditional Autotask objective**

- **Use a historical triggering tx**

  - **Testing does not fire notification**

- **If transaction != match**

  - **Look at the Autotask run**

- **Autotask Code network agnostic**

```javascript
19
20    for(const evt of events) {
21        const prizeStrategyAddress = evt.matchReasons[0].params.from //get prize strategy addres
22        const feeAmount = ethers.BigNumber.from(evt.matchReasons[0].params.value) // get fee amo
23        const chainlinkTokenAddress = evt.sentinel.address // get Chainlink Token address for th
24
25        console.log(`calling balanceOf(${prizeStrategyAddress}) @ ${chainlinkTokenAddress}`)
26
27        const linkContract = new ethers.Contract(chainlinkTokenAddress, chainLinkTokenABI, signe
28        const balance = await linkContract.balanceOf(prizeStrategyAddress)
29        const decimalBalance = ethers.utils.formatUnits(balance, await linkContract.decimals())
30
31        console.log(`LINK balance: ${decimalBalance}`)
32
33        if(balance.lt(feeAmount.mul(ethers.BigNumber.from(NOTICE_PERIOD)))){
34            console.log(`low balance of ${decimalBalance} detected!`)
35
36            matches.push({
37                hash: evt.hash,
38                metadata: {
39                    "balance" : decimalBalance,
40                    "prize strategy" : prizeStrategyAddress
41                }
42            });
43        }
```

Summary

- **PoolTogether award process**

- **Automate award cycles**

- **Monitor LINK balance**

- **Autotasks + Sentinel**

Thank you!

Learn more

openzeppelin.com/**defender**
forum.openzeppelin.com
docs.openzeppelin.com