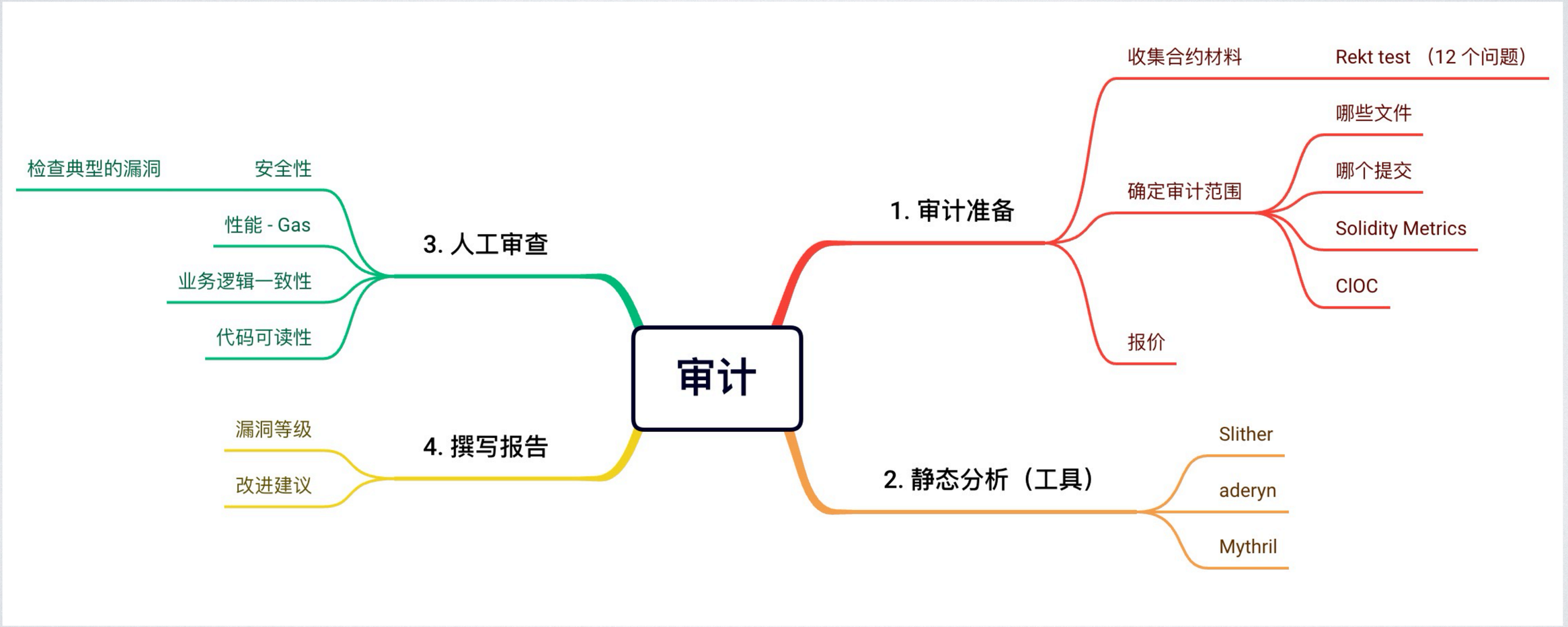
Solidity 安全

安全是一个过程，而非一个产品。它是一种思维方式，必须贯穿软件开发过程的方方面面

安全全过程

- 部署前：文档、编码、测试、审计
- 运行中：监控
- 事故分析
  - 分析资金流，尝试定位
  - 迁移

合约审计


审计
1. 审计准备
收集合约材料
Rekt test （12 个问题）
确定审计范围
哪些文件
哪个提交
Solidity Metrics
CIOC
报价
2. 静态分析（工具）
Slither
aderyn
Mythril
3. 人工审查
检查典型的漏洞
安全性
性能 - Gas
业务逻辑一致性
代码可读性
4. 撰写报告
漏洞等级
改进建议

审计准备

- 完整的工程、详细的文档、完备的测试
  - Rekt Test: https://blog.trailofbits.com/2023/08/14/can-you-pass-the-rekt-test/
- 工具:
  - CLOC： https://github.com/AlDanial/cloc
  - Solidity Metrics (VS Code extensions)
  - Solidity Visual developer (VS Code extensions)

静态分析

- Slither : https://github.com/crytic/slither
- Aderyn
- https://mythx.io/

人工审查

- 一个优秀的合约审计工程师：
  - 熟悉EVM特性、常见协议、常见安全问题
    - https://github.com/ZhangZhuoSJTU/Web3Bugs
  - 跟踪安全事件、了解新攻击点
    - https://substack.com/@blockthreat
    - https://solodit.xyz/
    - https://rekt.news/

审计报告

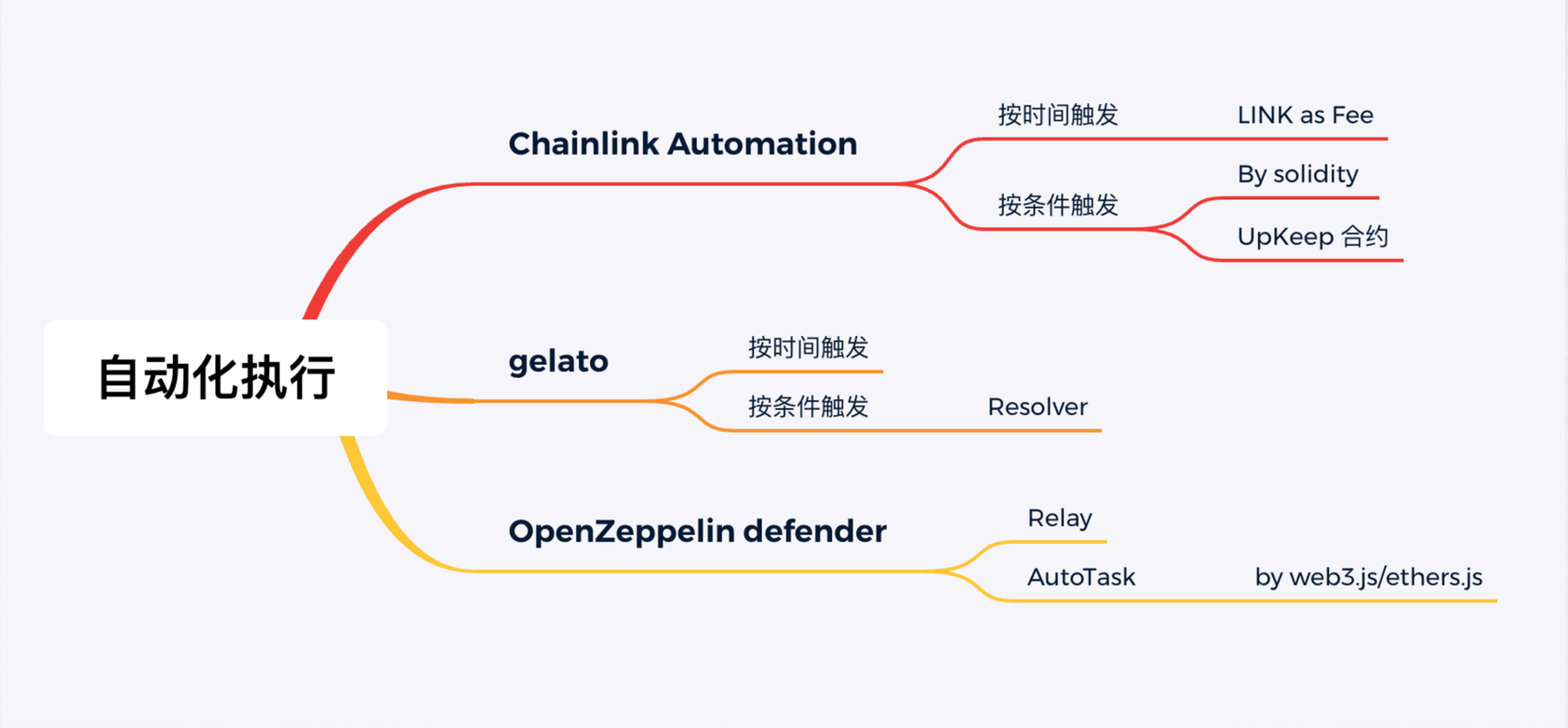
- 模板
- https://github.com/Cyfrin/audit-report-templating

运行中：监控

- OpenZepplin defender Monitor

合约自动化执行

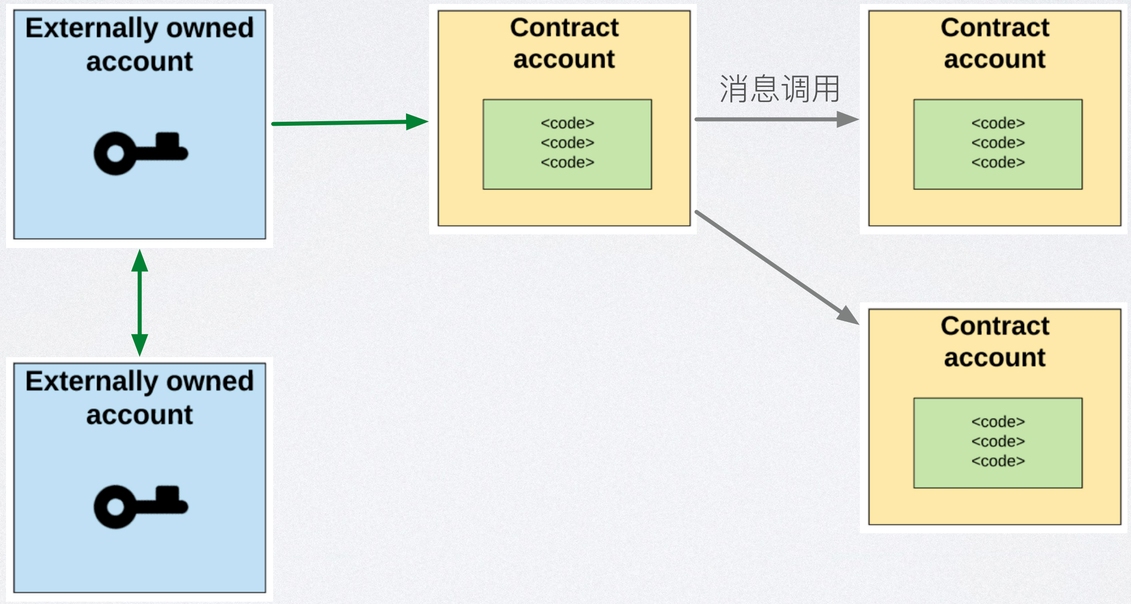
运行中：合约自动化执行


自动化执行
Chainlink Automation
按时间触发
LINK as Fee
按条件触发
By solidity
UpKeep 合约
gelato
按时间触发
按条件触发
Resolver
OpenZeppelin defender
Relay
AutoTask
by web3.js/ethers.js

合约只能被动响应

1. 由 EOA 支付Gas
2. 一次发起一笔交易

合约账户只能被动响应、没有自动运行



合约自动化执行

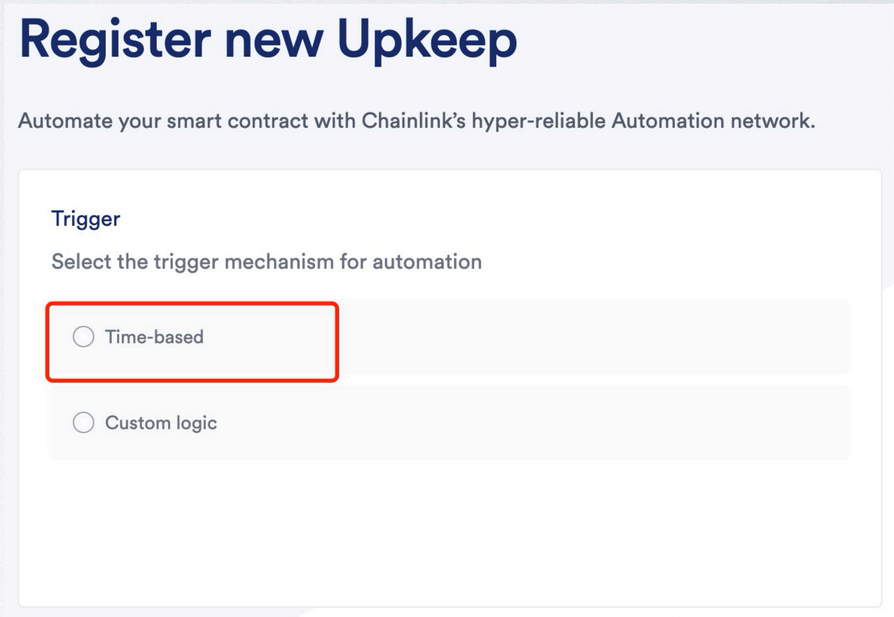
- 如何实现周期任务/定时任务/条件任务?
- 编写后端程序，常驻后端执行
- 主要问题：单点故障、热钱包泄漏

Chainlink Automation

- 超可靠和去中心化的自动化平台
- 根据时间或条件自动执行合约函数
- 若按条件，需编写 Upkeep 合约
  - checkUpKeep()
  - performUpKeep()

ChainLink Automation- 按时间执行

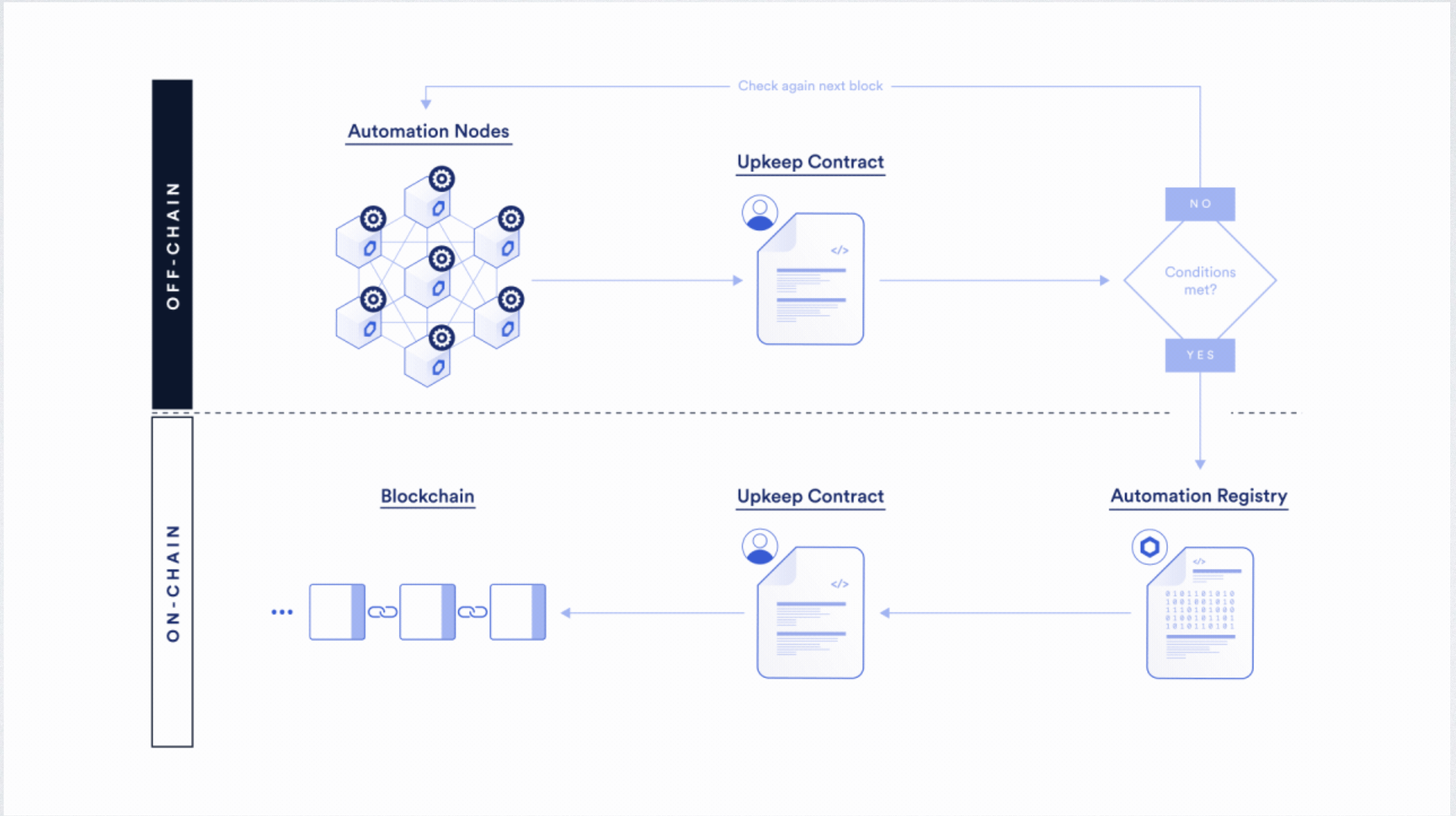
- 开发步骤：
- 在 https://automation.chain.link/ 注册
- 选择“Time-based”
- 填入要执行的合约地址
- 填入时间周期



ChainLink Automation - 按条件执行

- 编写 UpKeep 合约处理进行逻辑判断及调用
  - checkUpkeep (判断条件)
  - performUpkeep(执行)

CHAINLINK AUTOMATION



ChainLink Automation - 按条件执行

- 案例：当 Bank 存款大于 2 个 Token 时，自动转出

代码：w5-auto/contracts/AutoCollectUpKeep.sol

任务地址：https://automation.chain.link/sepolia/27157894517625986686394990837133401010366092253963320486987630466239869405442

Gelato Functions

- 按时间执行, 无需代码 （automated-transaction）
- 按链上条件执行 （Solidity Functions）， checker 合约
- 按链上条件执行 （Typescript Functions）

OpenZepplin Defender Action

- 通过web3.js / ethers.js 来定制执行
- Relay: 生成独立的账户
- AutoTask

https://www.openzeppelin.com/defender

事故分析

- https://phalcon.blocksec.com/explorer
  - https://www.youtube.com/watch?v=eXeirKUy1XA
  - https://www.youtube.com/watch?v=uiqCrhIU0To
- Foundry Transaction Replay Trace/Debugger
- Tenderly Debugger

练习题

- 修改 Bank 合约：
  - 用户可以通过 deposit 进行存款（先Approve） — 之前已经实现
  - (自动化)当存款超过 x 时，转移一半的存款的到指定的地址，如Owner.

可使用 ChainLink Automation 、Gelato、OpenZepplin Defender Action

https://decert.me/quests/072fccb4-a976-4cf9-933c-c4ef14e0f6eb