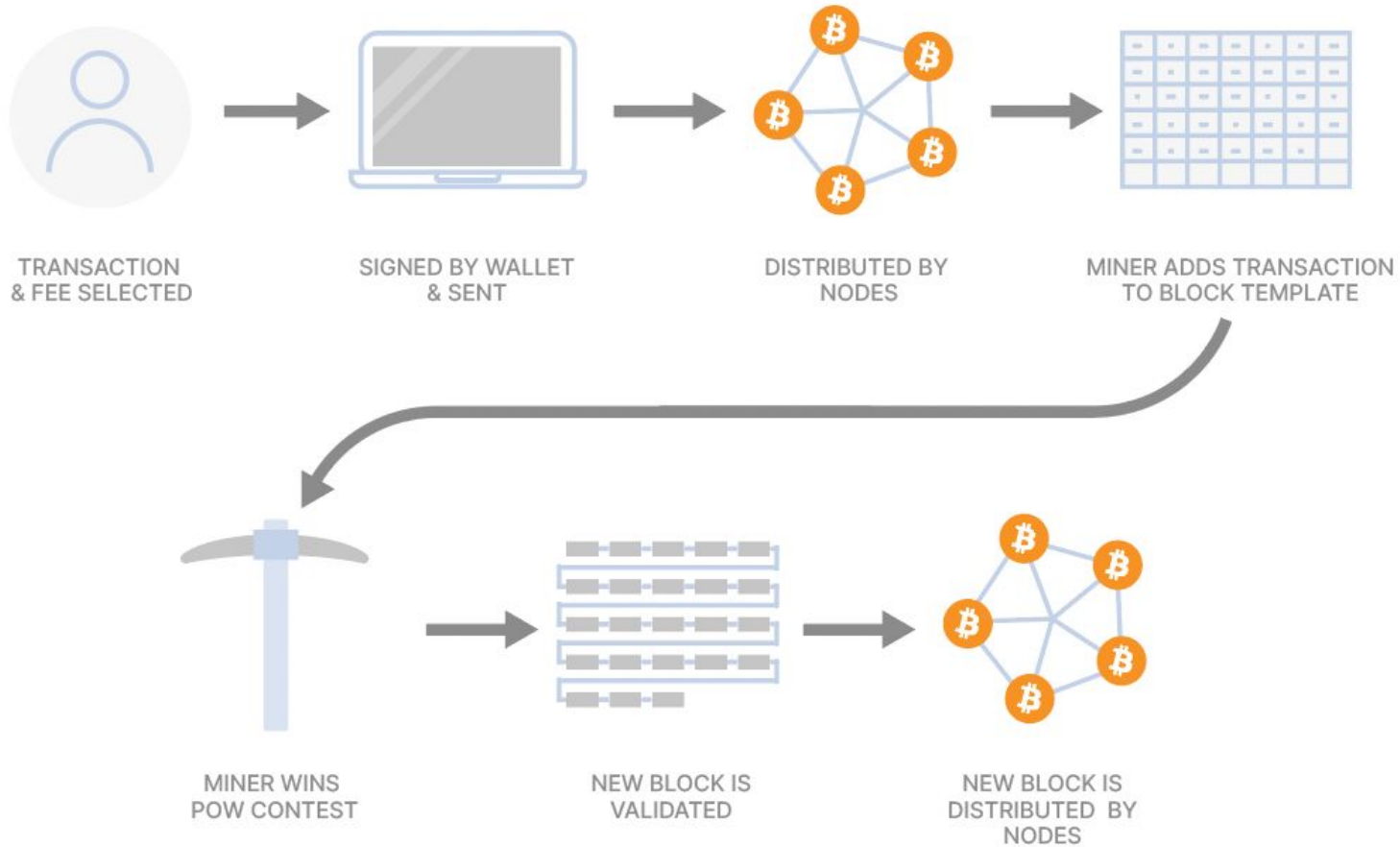


The software supply chain of crypto and decentralization

Martin Monperrus, César Soto-Valero,
Javier Ron, Benoit Baudry and friends





TL;DR: “The crypto ecosystem is as good
as its software supply chain”

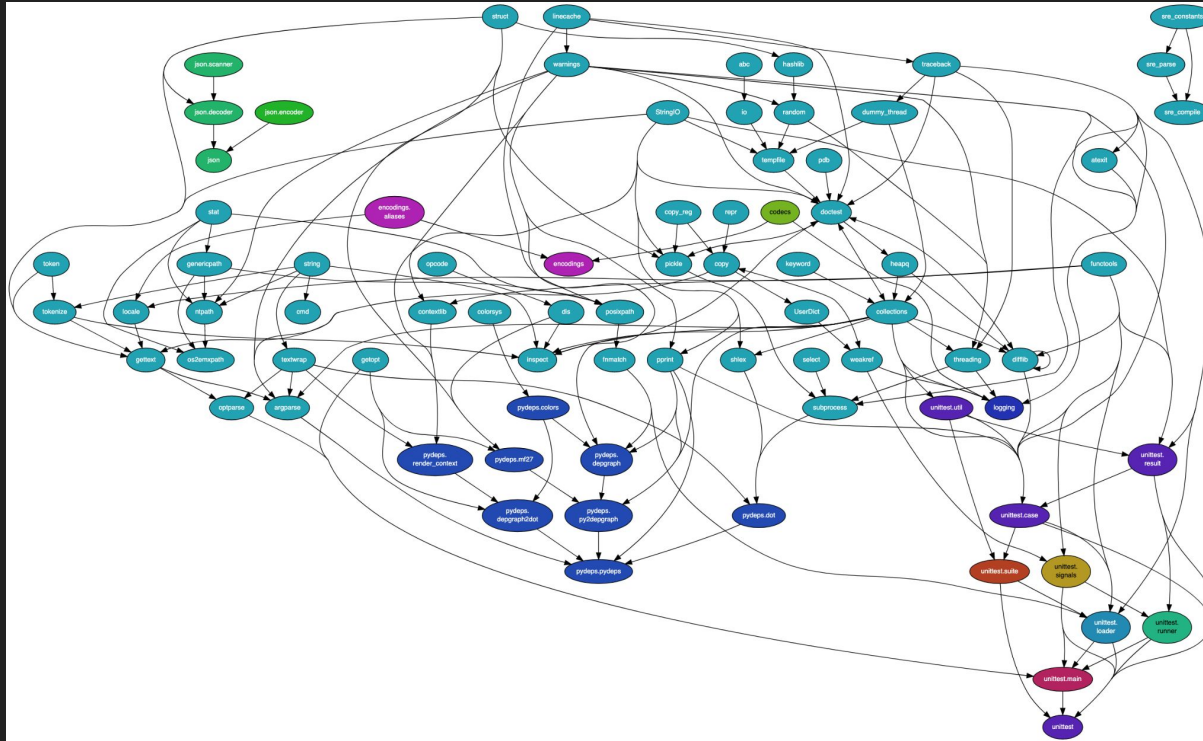
Software supply chain

“The set of all dependencies and tools used to build, deploy and run a software system”

Starting with

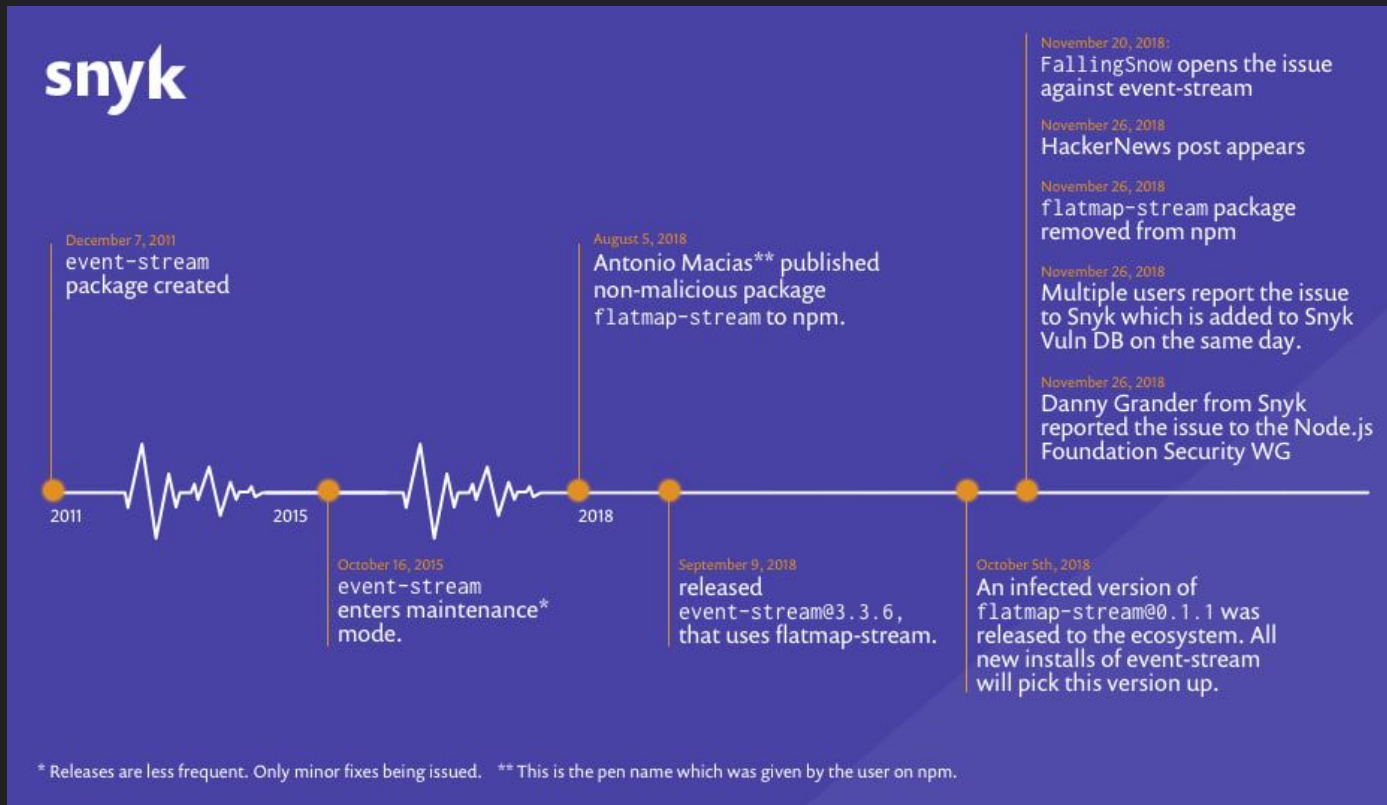
- The dependencies (Maven, NPM, etc)
- The core (compiler, linker, minifier)
- The pipelines (CI/CD)

Dependencies



Bugs? Breaking changes? Malicious code injection?

Example attack on dependency (event-stream, target copay wallet)



<https://snyk.io/blog/a-post-mortem-of-the-malicious-event-stream-backdoor/>

Other examples

- DyDx
<https://www.mend.io/resources/blog/popular-cryptocurrency-exchange-dydx-has-had-its-npm-account-hacked/>
- Sushiswap
<https://blog.sonatype.com/3-million-cryptocurrency-heist-malicious-github-commit?hsLang=en-us>
- Cryptomining in docker
<https://www.darkreading.com/attacks-breaches/container-supply-chain-attacks-cashing-in-on-cryptojacking>
- And counting.... <https://chains.proj.kth.se/software-supply-chain-attacks-crypto.html>

Countermeasures (KUTE)

- Know your dependencies (Software bill of materials)
 - <https://github.com/CycloneDX>
- Update your dependencies (Regularly and Automatically)
 - [Dependabot](#), [Renovate](#), [DepFu](#)
- Track changes when updating (changelog/authors/keys)
 - <https://github.com/lightbend-labs/jardiff>
- Ensure at runtime (Software integrity)
 - <https://chains-project.github.io>

All of this is platform/language dependent

Know your dependencies (Ethereum Java Nodes)

Challenges of Producing Software Bill Of Materials for Java

Musard Balliu, Benoit Baudry, Sofia Bobadilla, Mathias Ekstedt, Martin Monperrus, Javier Ron, Aman Sharma, Gabriel Skoglund, César Soto-Valero, Martin Wittlinger

Abstract—Software bills of materials (SBOM) promise to become the backbone of software supply chain hardening. We deep-dive into 6 tools and the accuracy of the SBOMs they produce for complex open-source Java projects. Our novel insights reveal some hard challenges for the accurate production and usage of SBOMs.

The Multibillion Dollar Software Supply Chain of Ethereum

César Soto-Valero, Martin Monperrus, and Benoit Baudry, KTH Royal Institute of Technology

Ethereum is the single largest programmable blockchain platform today. Ethereum nodes operate the blockchain, relying on a vast supply chain of third-party software dependencies.

Know your dependencies (Ethereum Java Nodes)

January 2022

- Besu (Eth1)
- Teku (Eth2)

	Besu (Eth1)	Teku (Eth2)
Lines of Java code	268,356	209,860
Commits	3,125	3,142
Contributors	115	65
Unique internal dependencies	41	57
Unique third-party dependencies	355	293
Unique suppliers	165	146
Unique third-party dependencies introduced since January 2021	127	79
Unique third-party suppliers introduced since January 2021	49	22
Unique third-party dependency versions modified since January 2021	171	150

\$ build-info-go

\$ gradle dependencies --scan

Mitigate Build Attacks (reproducible)

“The ability to fully control the produced binaries in a deterministic way”



- Gitian (invented by bitcoin, 2010): <https://gitian.org/>
- Guix <https://guix.gnu.org/> (bitcoin today)
- NIX <https://nixos.org/>
- Build attestations <https://github.com/bitcoin-core/guix.sigs>
- Geth builds are not reproducible

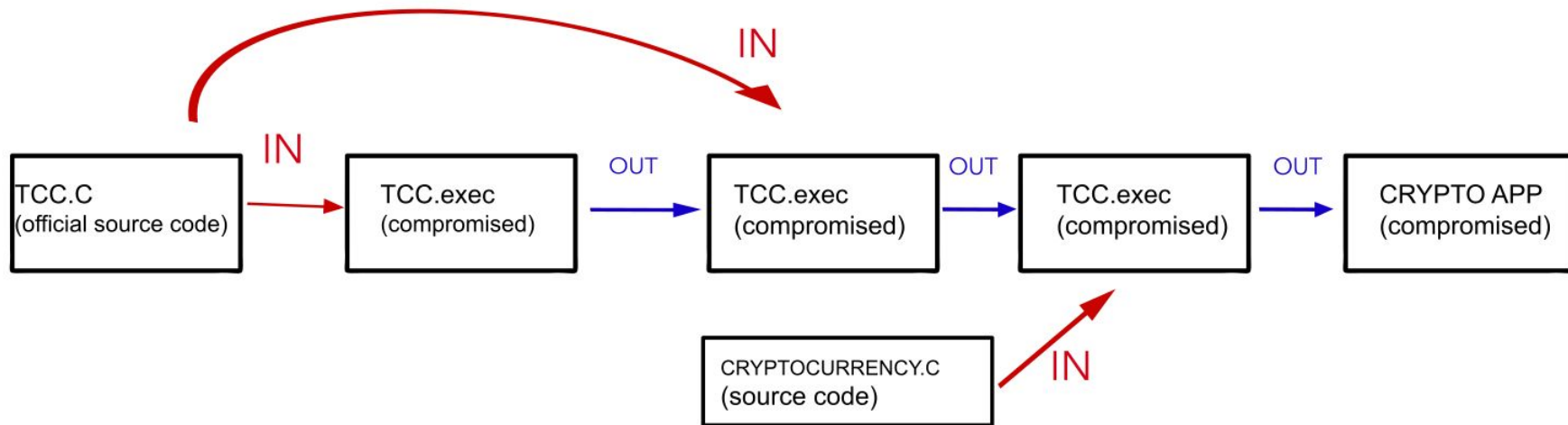
SHA256:

5a6b35d1a348a402f2d2d6ab5aed6
53a1a1f13bc63aaaf51605e3501b0
733b7a

<https://github.com/chains-project/btc-supply-chain>

Mitigate Trusting Trust Attacks

“To what extent should one trust a statement that a program is free of Trojan horses.”

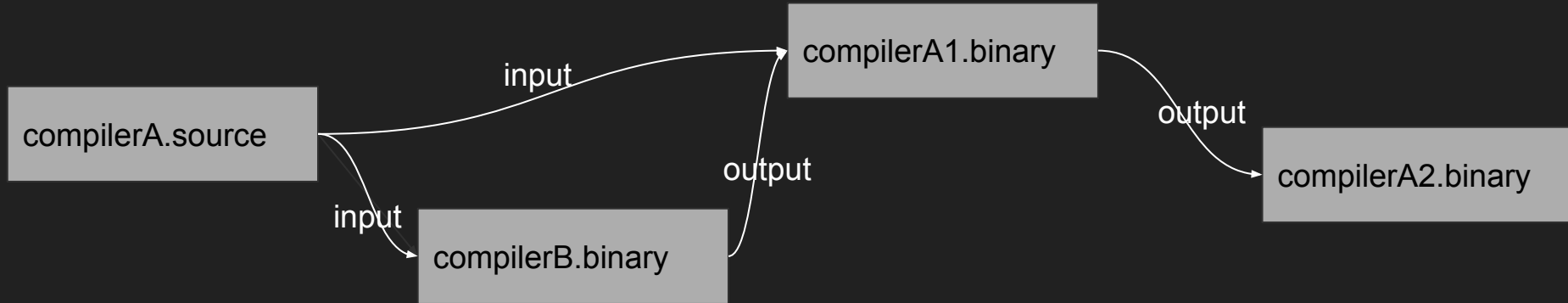


Niklas Rosencrantz's Master's thesis, KTH, 2022

[Diverse Double-Compiling to Harden Cryptocurrency Software](#)

Diverse Double Compilation

“The usage of two different compilers for the same language to counter trusting trust attacks”



Challenges:

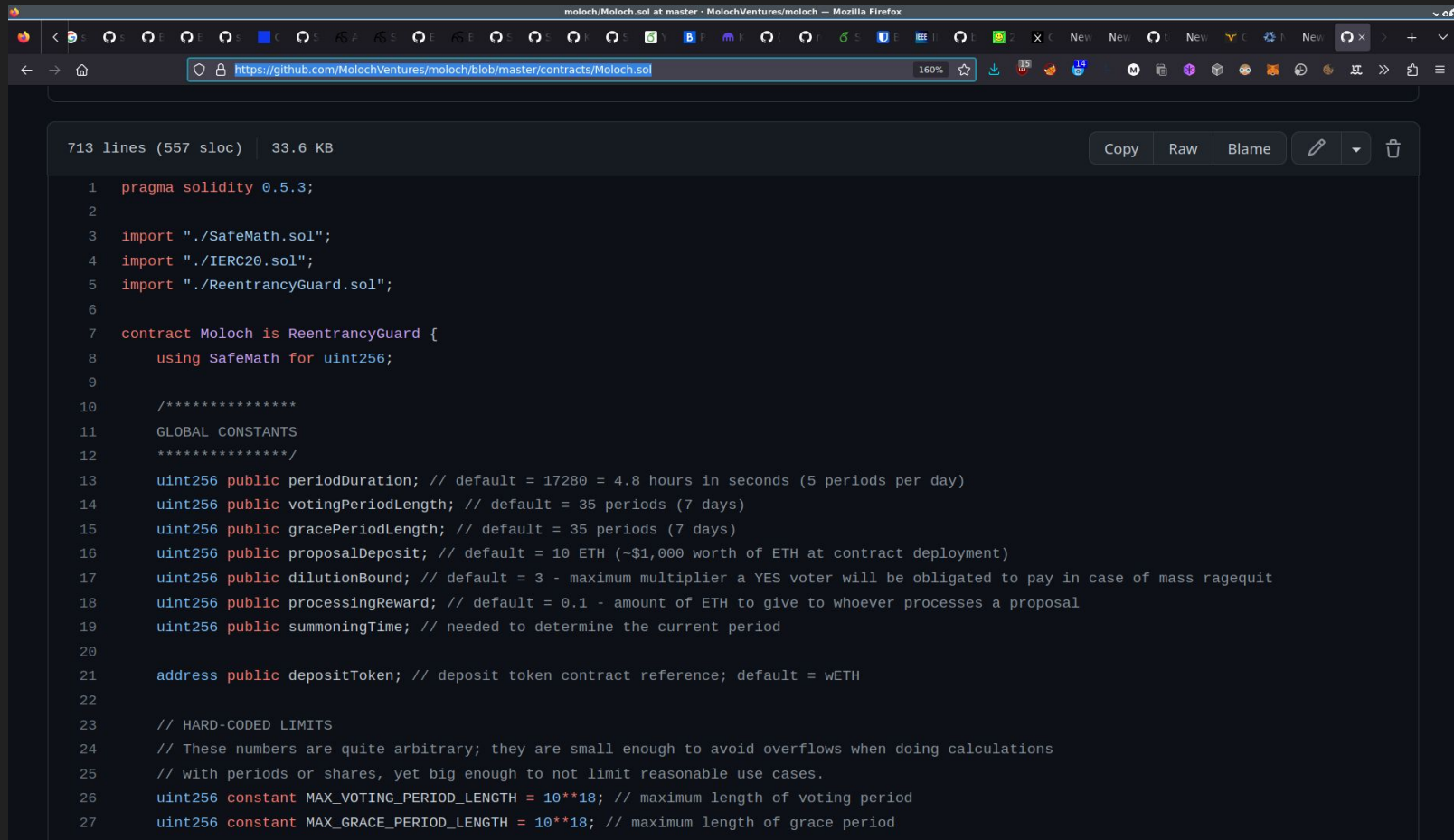
- Have two different compatible compilers
- Integrate them in mainstream continuous delivery of compilers

Software supply chain of decentralization

- “Governance code (“Code is law”)
 - Example:
 - Voting code
 - Tax code
 - Grant allowance code
- Core properties
 - Transparency = Democracy
 - Immutability = Political Stability
- State-of-the-art: “Decentralized Autonomous Organization” (DAO)

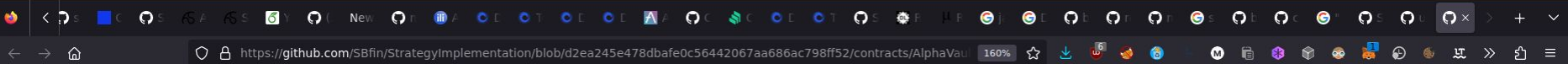
Smart contracts for governance

<https://github.com/MolochVentures/moloch/blob/master/contracts/Moloch.sol>



```
713 lines (557 sloc) | 33.6 KB
Copy Raw Blame

1 pragma solidity 0.5.3;
2
3 import "./SafeMath.sol";
4 import "./IERC20.sol";
5 import "./ReentrancyGuard.sol";
6
7 contract Moloch is ReentrancyGuard {
8     using SafeMath for uint256;
9
10    /*****
11    GLOBAL CONSTANTS
12    *****/
13    uint256 public periodDuration; // default = 17280 = 4.8 hours in seconds (5 periods per day)
14    uint256 public votingPeriodLength; // default = 35 periods (7 days)
15    uint256 public gracePeriodLength; // default = 35 periods (7 days)
16    uint256 public proposalDeposit; // default = 10 ETH (~$1,000 worth of ETH at contract deployment)
17    uint256 public dilutionBound; // default = 3 - maximum multiplier a YES voter will be obligated to pay in case of mass ragequit
18    uint256 public processingReward; // default = 0.1 - amount of ETH to give to whoever processes a proposal
19    uint256 public summoningTime; // needed to determine the current period
20
21    address public depositToken; // deposit token contract reference; default = wETH
22
23    // HARD-CODED LIMITS
24    // These numbers are quite arbitrary; they are small enough to avoid overflows when doing calculations
25    // with periods or shares, yet big enough to not limit reasonable use cases.
26    uint256 constant MAX_VOTING_PERIOD_LENGTH = 10**18; // maximum length of voting period
27    uint256 constant MAX_GRACE_PERIOD_LENGTH = 10**18; // maximum length of grace period
```



154 lines (120 sloc) | 5.07 KB

Copy

Raw

Blame



```
1 pragma solidity 0.7.6;
2
3 import "../interfaces/external/IWETH9.sol";
4 import "@openzeppelin/contracts/math/Math.sol";
5 import "@openzeppelin/contracts/math/SafeMath.sol";
6 import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
7 import "@openzeppelin/contracts/token/ERC20/IERC20.sol";
8 import "@openzeppelin/contracts/token/ERC20/SafeERC20.sol";
9 import "@openzeppelin/contracts/utils/ReentrancyGuard.sol";
10 import "@uniswap/v3-core/contracts/interfaces/callback/IUniswapV3MintCallback.sol";
11 import "@uniswap/v3-core/contracts/interfaces/callback/IUniswapV3SwapCallback.sol";
12 import "@uniswap/v3-core/contracts/interfaces/IUniswapV3Pool.sol";
13 import "@uniswap/v3-core/contracts/libraries/TickMath.sol";
14 import "@uniswap/v3-periphery/contracts/libraries/LiquidityAmounts.sol";
15 import "@uniswap/v3-periphery/contracts/libraries/PositionKey.sol";
16 import "../interfaces/IVault.sol";
17 import "../interfaces/IOrbitVault.sol";
18 import "../AlphaVault.sol";
19
20 contract AlphaVaultUtility is
21     ...ReentrancyGuard
22     ...{
23     ...using SafeERC20 for IERC20;
24     ...using SafeMath for uint256;
25
26     ...address public immutable weth;
27     ...AlphaVault public immutable alphaVault;
```


The DAO update regression

- Problem: how to secure a governance update? (= a code update)
- Prerequisite: Votes on patches
- Solution: Automated deployment with high integrity



<https://compound.finance/governance/proposals/139>


Compound | Proposal Detail #139 — Mozilla Firefox

← OVERVIEW

OpenZeppelin Security Partnership - 2023 Q1 Compensation

Passed

139 • Executed December 17th, 2022

 OpenZeppelin
0xec40...ec1e

For

1,182,845

79 Addresses

Votes

Polychain Capital

306,103.155

0x9AA8...cCF1

256,015.7022

Gauntlet

126,106.1087

VIEW ALL

Against

0

0 Addresses

Votes

—

—

—

—

—

—

VIEW ALL

Details

1

Transfer 26178.56 COMP to 0x57C970568668087c05352456a3F59B58B0330066

Background

Starting on Dec 21st, 2021, OpenZeppelin was **selected** to offer the Compound DAO security services including continuous audit, security advisory, and monitoring. At the start of every quarter OpenZeppelin will create a proposal to renew the partnership and perform the next service fee payment.

Compensation Structure

Proposal History

Created

December 10th, 2022 – 4:46pm

Active

December 12th, 2022 – 12:48pm

Succeeded

December 15th, 2022 – 6:55am

Queued

December 15th, 2022 – 6:55am

Executed

December 17th, 2022 – 6:56am

18

“Governance in programmable societies vitally requires software integrity”



KTH / [programmable-society](#) Public



Code



Issues

16



Pull requests



Discussions



Actions



Settings

New Course DD2485, KTH, Nov-Dec 2023

References

[The Multibillion Dollar Software Supply Chain of Ethereum](#) (César Soto-Valero, Martin Monperrus and Benoit Baudry), In IEEE Computer, 2022.

[Challenges of Producing Software Bill Of Materials for Java](#) (Musard Balliu, Benoit Baudry, Sofia Bobadilla, Mathias Ekstedt, Martin Monperrus, Javier Ron, Aman Sharma, Gabriel Skoglund, César Soto-Valero and Martin Wittlinger), Technical report 2303.11102, arXiv, 2023.

[Highly Available Blockchain Nodes With N-Version Design](#) (Javier Ron, César Soto-Valero, Long Zhang, Benoit Baudry and Martin Monperrus), Technical report 2303.14438, arXiv, 2023.