

## Workshop #4

Secure Development by **Z** OpenZeppelin

09/16 - 12PM PST / 7PM UTC

# Strategies for Secure Governance with Smart Contracts

Martin Abbatemarco

Security Researcher at OpenZeppelin



# Security solutions for industry leaders

Our mission is to protect  
the open economy

 OpenZeppelin

## Audits

200+ audits completed

## Defender

3,000+ users in the first  
six months of launch,  
including many top DeFi  
projects

## Contracts

\$83B+ TVL in DeFi  
protocols, and thousands of  
NFTs including Beeple's  
\$69M built on Contracts



cøsmos



UNISWAP



Futureswap



Aave



Decentraland



brave



coinbase



Set



Balancer



augur



opyn



linch



Optimism

GNOSIS



δY/δX

Polkadot.

Series of sessions

# Secure Development

The dangers of token integration



Strategies for secure access controls



The dangers of price oracles



**Strategies for secure governance**

Do's and don'ts of smart contract upgrades

...

**so, governance**



who participates ?

how ?

what kind of actions ?

what components of the system can be affected ?

binding ?

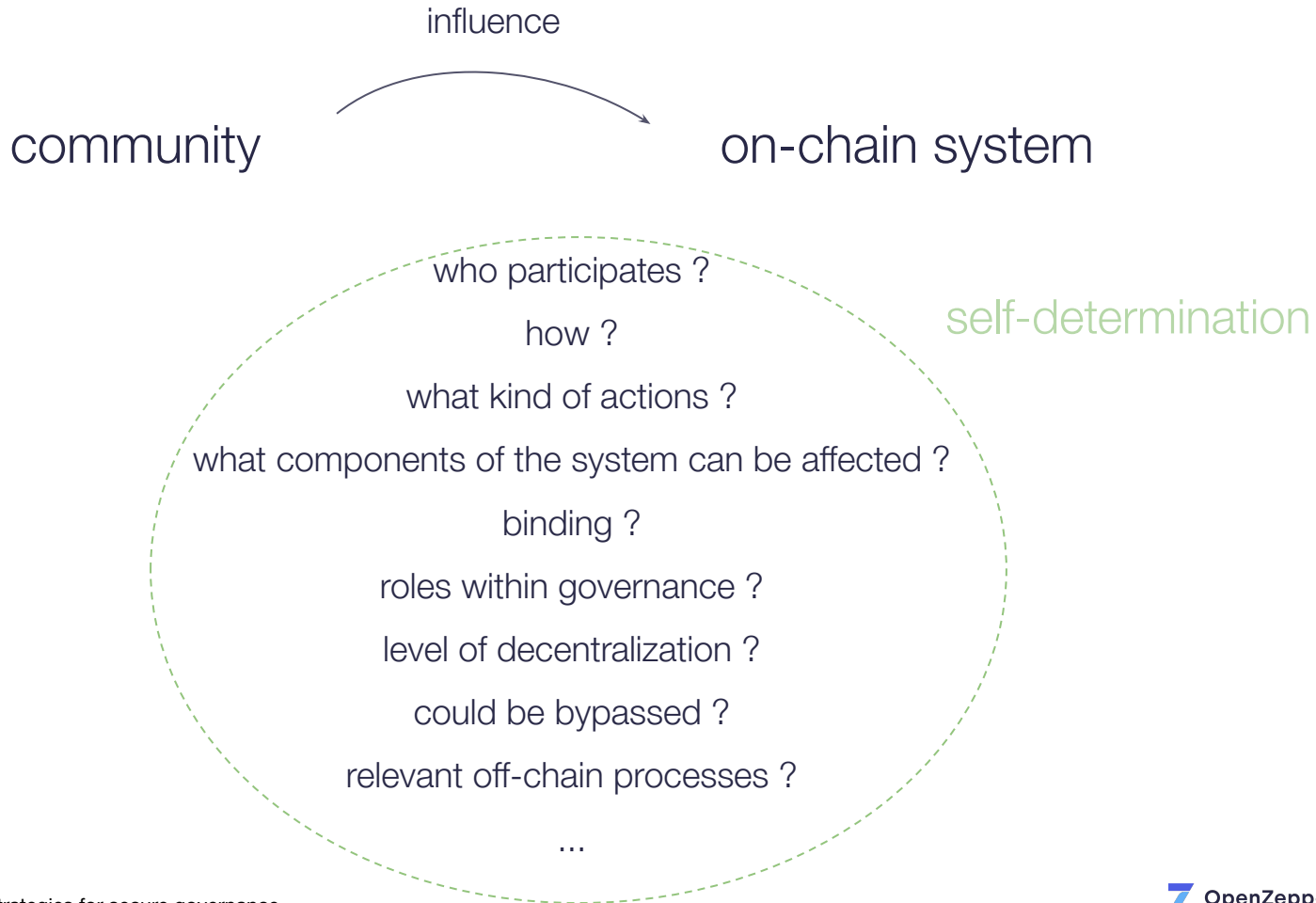
roles within governance ?

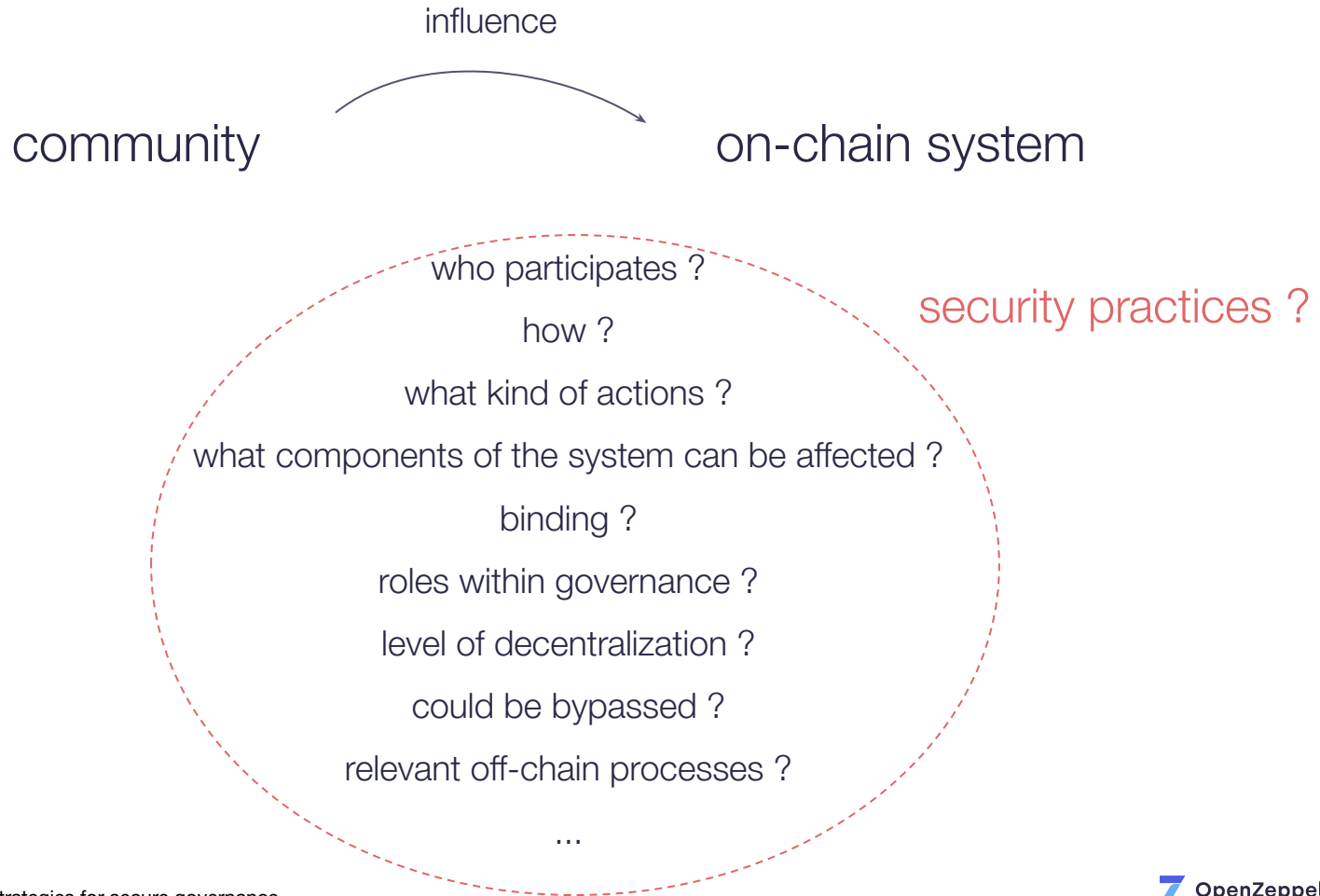
level of decentralization ?

could be bypassed ?

relevant off-chain processes ?

...







**we're figuring it out**





# **figuring it out**



- vote delegation
- minimum quorum
- specialized committees
- continuous voting
- augmented voting power via token lock
- off-chain voting for signaling / polling
- on-chain actions from off-chain voting by way of oracles
- upgradeable smart contracts
- time delays on sensitive actions
- emergency actions via Multisig (e.g. pause, shutdown)
- legal entities (e.g. foundation) providing support for the DAO
- tools for DAO treasury management
- token issuance to cover protocol expenses
- incentivized voting
- incentivized off-chain engagement (e.g. forum participation)

[docs.balancer.fi/core-concepts/governance/multisig](https://docs.balancer.fi/core-concepts/governance/multisig)



vote delegation

minimum quorum

specialized committees

continuous voting

augmented voting power via token lock

off-chain voting for signaling / polling

on-chain actions from off-chain voting by way of oracles

ungradeable

tools, infrastructure, processes, risks

delays on sensitive actions

emergency actions via Multisig (e.g. pause, shutdown)

legal entities (e.g. foundation) providing support for the DAO

tools for DAO treasury management

token issuance to cover protocol expenses

incentivized voting

incentivized off-chain engagement (e.g. forum participation)

[docs.balancer.fi/core-concepts/governance/multisig](https://docs.balancer.fi/core-concepts/governance/multisig)

# learning from others

AAVE

UNI

MKR

SNX

COMP

BAL

CRV

UMA

YFI

*(and many others)*

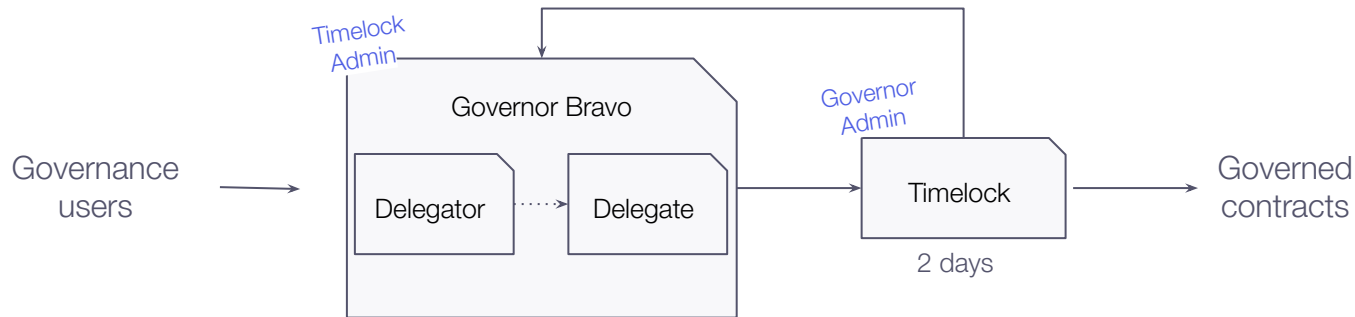


**what can we learn ?**



# COMP

[compound.finance/docs/governance](https://compound.finance/docs/governance)

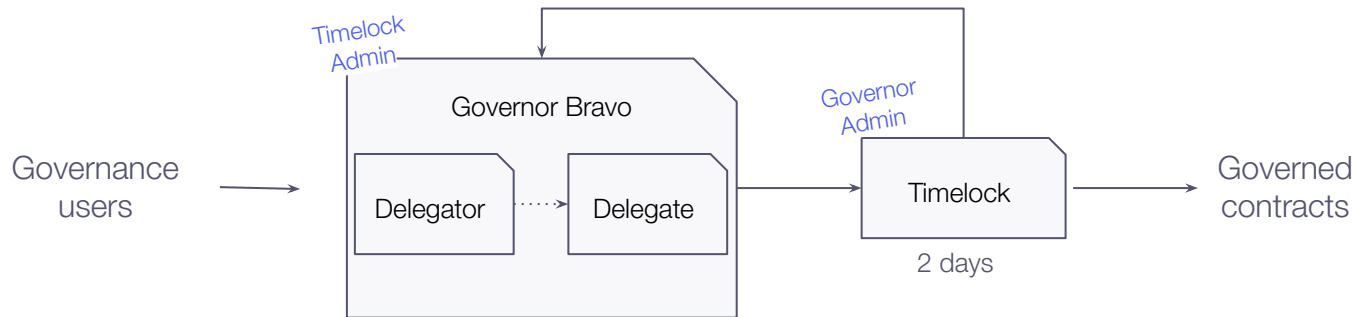


```
function _setImplementation(address implementation_) public {
    require(msg.sender == admin, "GovernorBravoDelegator::_setI
    require(implementation_ != address(0), "GovernorBravoDelega
```

- **Upgrades**
- Transfer admin powers
- Change voting period, threshold, delays
- Flash-loan protections
- Delegation of voting power

## COMP





## COMP

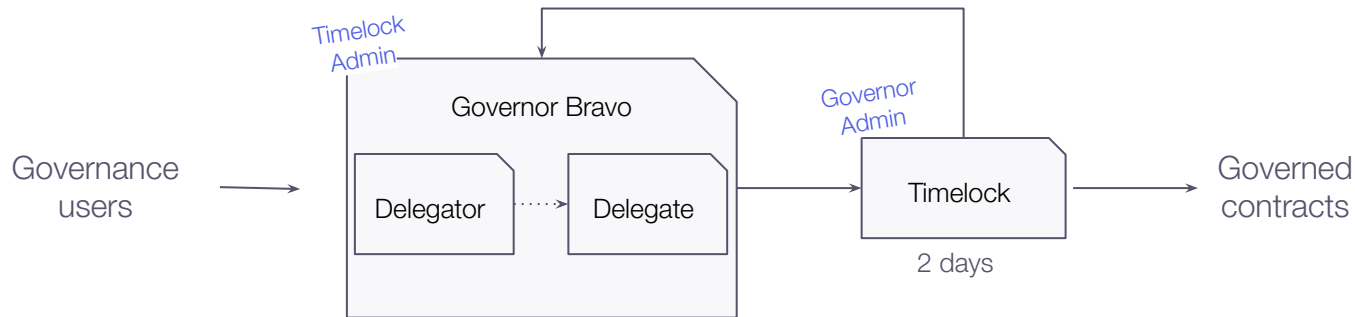
- Upgrades
- Transfer admin powers
- Change voting period, threshold, delays
- Flash-loan protections
- Delegation of voting power

```
function _setVotingDelay(uint newVotingDelay) external {
    require(msg.sender == admin, "GovernorBravo::_setVotingDelay: admin only");
}
```

```
function _setVotingPeriod(uint newVotingPeriod) external {
    require(msg.sender == admin, "GovernorBravo::_setVotingPeriod: admin only");
}
```

```
function _setProposalThreshold(uint newProposalThreshold) external {
    require(msg.sender == admin, "GovernorBravo::_setProposalThreshold: admin only");
}
```

```
function _setPendingAdmin(address newPendingAdmin) external {
    // Check caller = admin
    require(msg.sender == admin, "GovernorBravo::_setPendingAdmin: admin only");
}
```



- Upgrades
- Transfer admin powers
- Change voting period, threshold
- **Flash-loan protections**
- Delegation of voting power

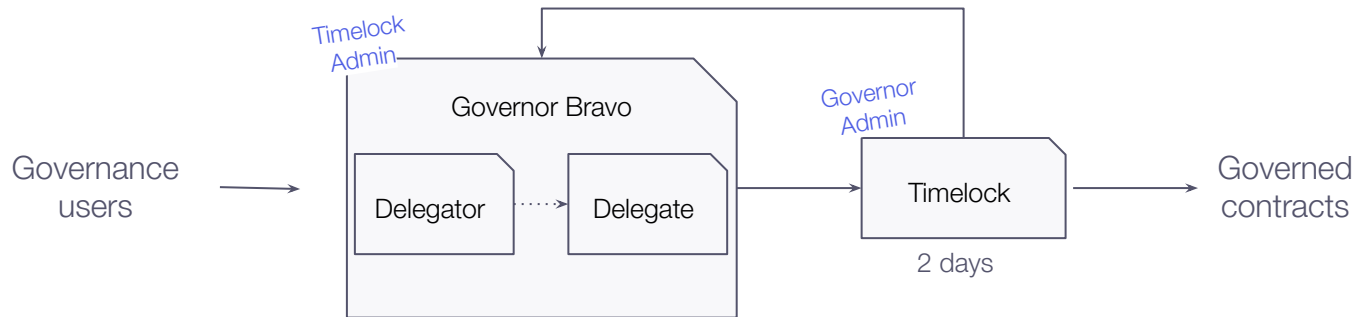
## COMP

```
function propose(address[] memory targets, uint[] memory values, string[] memory signatures) public {
    // Reject proposals before initiating as Governor
    require(initialProposalId != 0, "GovernorBravo::propose: Governor Bravo not active");
    require(comp.getPriorVotes(msg.sender, sub256(block.number, 1)) > proposalThreshold,
    "GovernorBravo::propose: proposer votes below proposal threshold");

    Proposal storage proposal = Proposal({proposalId: proposalId});
    proposal.targets = targets;
    proposal.values = values;
    proposal.signatures = signatures;
    proposal.state = ProposalState.Active;

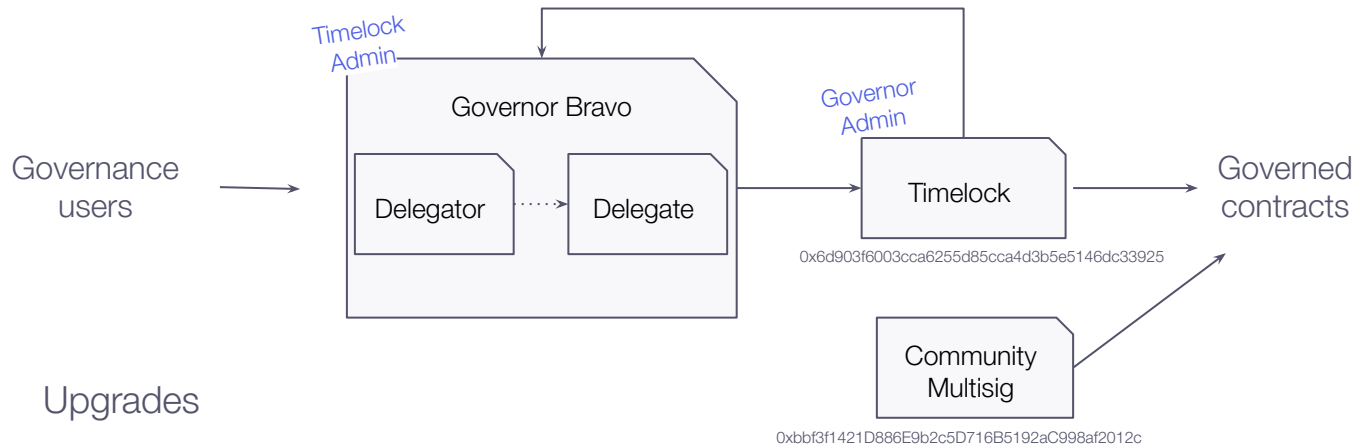
    Receipt storage receipt = Receipt({receiptId: receiptId});
    receipt.hasVoted = true;
    receipt.votes = votes;

    emit ProposalCreated(proposalId, targets, values, signatures);
    emit ProposalStateChange(proposalId, ProposalState.Active);
    emit ReceiptCreated(receiptId, receipt);
    emit ReceiptVote(receiptId, msg.sender, votes);
}
```



- Upgrades
- Transfer admin powers
- Change voting period, threshold, delays
- Flash-loan protections
- **Delegation of voting power**

```
function delegate(address delegatee) public {  
    return _delegate(msg.sender, delegatee);  
}  
  
function delegateBySig(
```

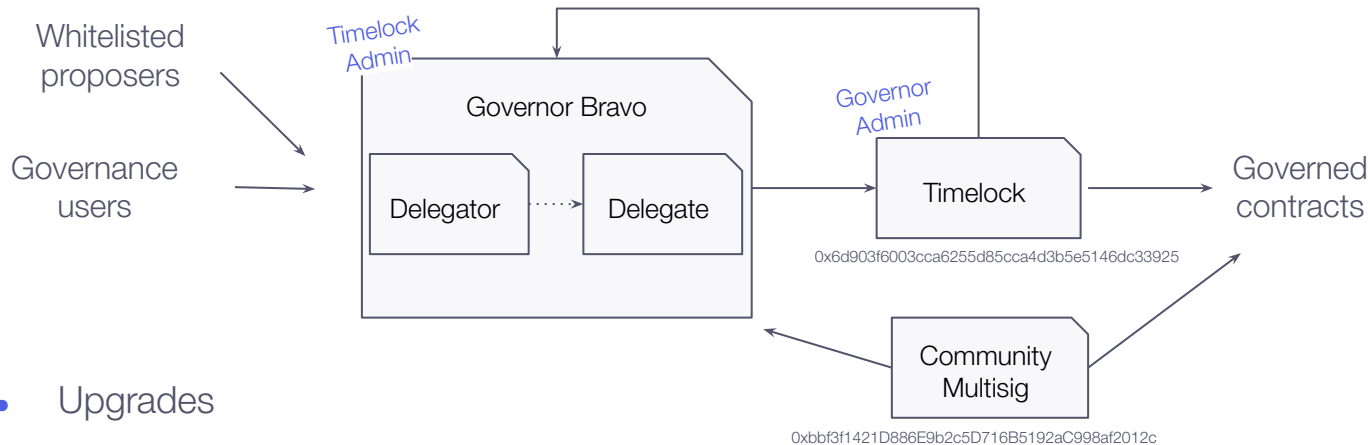


## COMP

- Upgrades
- Transfer admin powers
- Change voting period, threshold, delays
- Flash-loan protections
- Delegation of voting power
- **Governance bypass for sensitive actions**

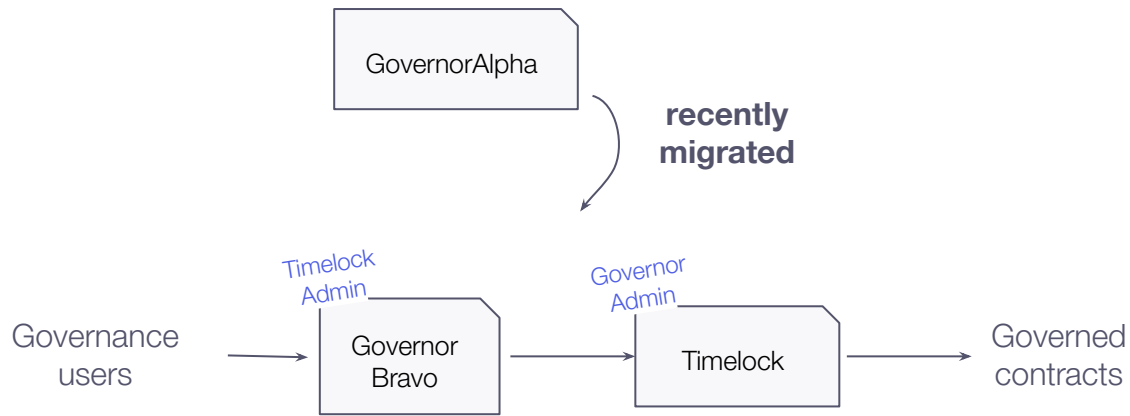
## COMP

- Upgrades
- Transfer admin powers
- Change voting period, threshold, delays
- Flash-loan protections
- Delegation of voting power
- Governance bypass for sensitive actions
- **Whitelisted proposers**



# UNI

[docs.uniswap.org/protocol/concepts/governance/overview](https://docs.uniswap.org/protocol/concepts/governance/overview)



# UNI

hayden.eth 🐼🔵 @haydenzadams · Aug 30  
1/3

Uniswap community members recently proposed upgrading Uniswap governance contracts to Bravo (also used by Compound gov). Today, a Labs team member discovered a minor issue that would have made the next 8 votes ineffective.

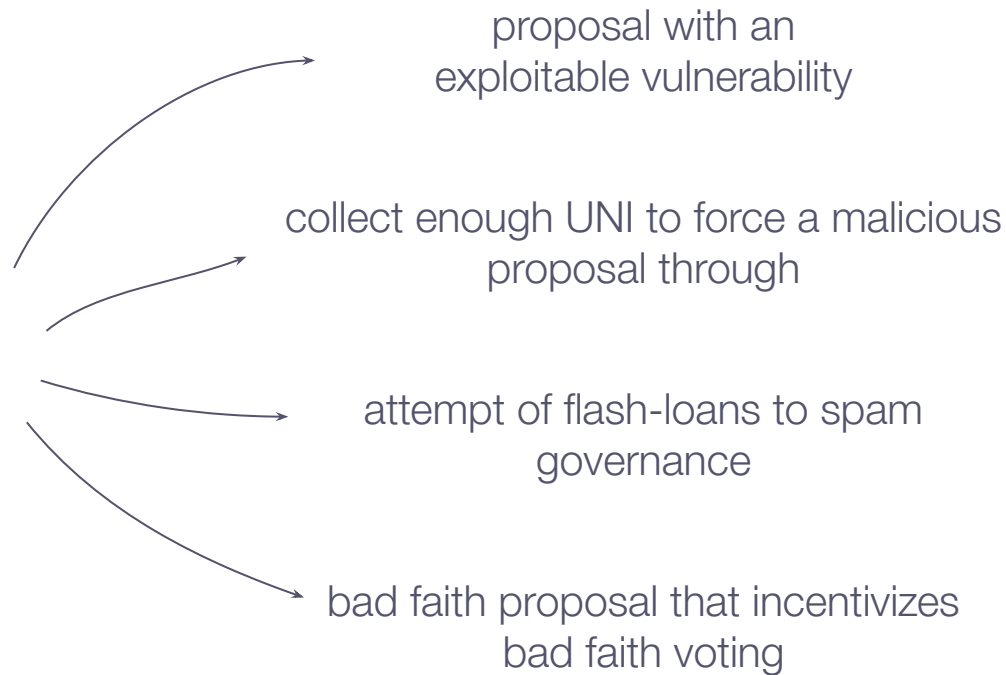
Now fixed, governance working!

<https://twitter.com/haydenzadams/status/1432496728230535175>

Be careful with  
migrations

# UNI

## Adversarial circumstances



<https://docs.uniswap.org/protocol/concepts/governance/adversarial-circumstances>



# SNX

[docs.synthetix.io/governance](https://docs.synthetix.io/governance)

**SNX**

**Governance  
artifacts**



improvement  
proposals



configuration  
changes

**Governance  
bodies**



several DAOs with  
different levels of  
responsibilities

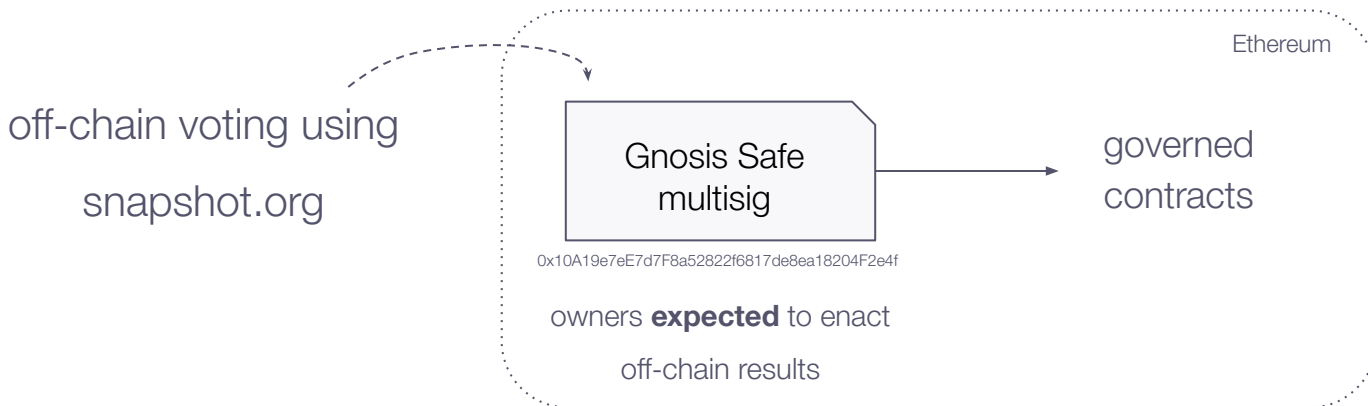
## SNX

- Use of battle-tested multisig wallets
- Documentation for transparency
- Clear off-chain formal processes
- Use of NFTs to show membership to council
- Use of off-chain voting via [snapshot.org](https://snapshot.org)

# BAL

[docs.balancer.fi/core-concepts/governance](https://docs.balancer.fi/core-concepts/governance)

# BAL

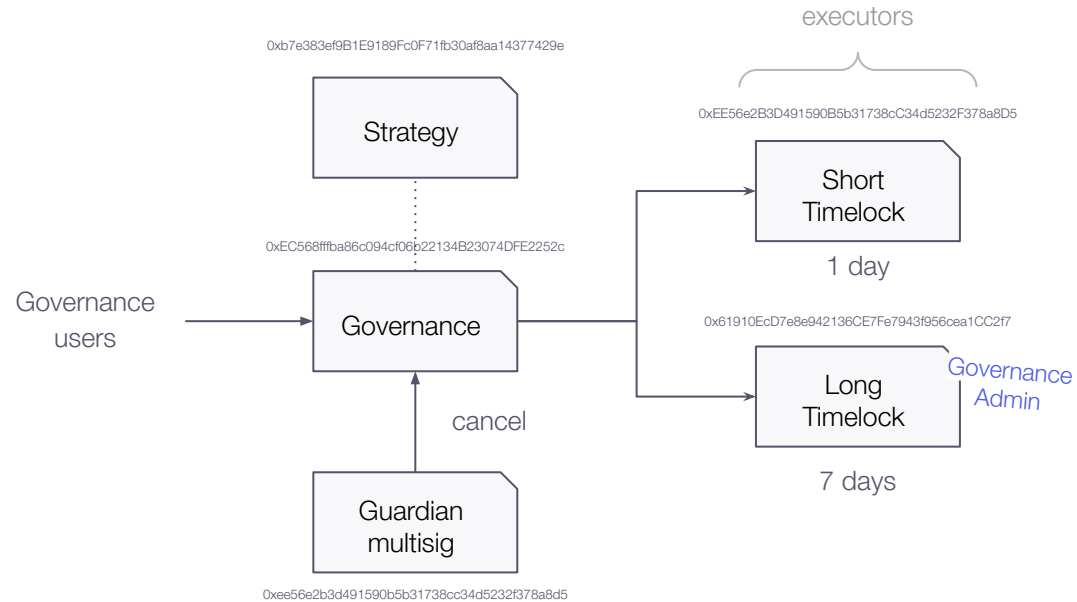


- Highlighting powers, limitations & responsibilities
- Linking to people in the community
- Expected roadmap
- Rationale behind these decisions
- Designing for rogue multisigs

[docs.balancer.fi/core-concepts/governance/multisig](https://docs.balancer.fi/core-concepts/governance/multisig)

# AAVE

[docs.aave.com/developers/protocol-governance/governance](https://docs.aave.com/developers/protocol-governance/governance)



# AAVE

- Use of a guardian to cancel proposals
- Use of a strategy for changes in how power is measured
- Documentation of different types of “policies”
- Different delays and thresholds for different actions



# MKR

[docs.makerdao.com/smart-contract-modules/governance-module](https://docs.makerdao.com/smart-contract-modules/governance-module)

**MKR**

Governance  
users

lock / free / vote

0x0a3f6849f78076aefadf113f5bed87720274ddc0

Chief



0x0a3f6849f78076aefadf113f5bed87720274ddc0

**MKR**

Governance  
users

lift

Chief

hat

Spell

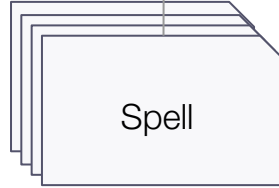
**MKR**

Governance  
users

0x0a3f6849f78076aefadf113f5bed87720274ddc0



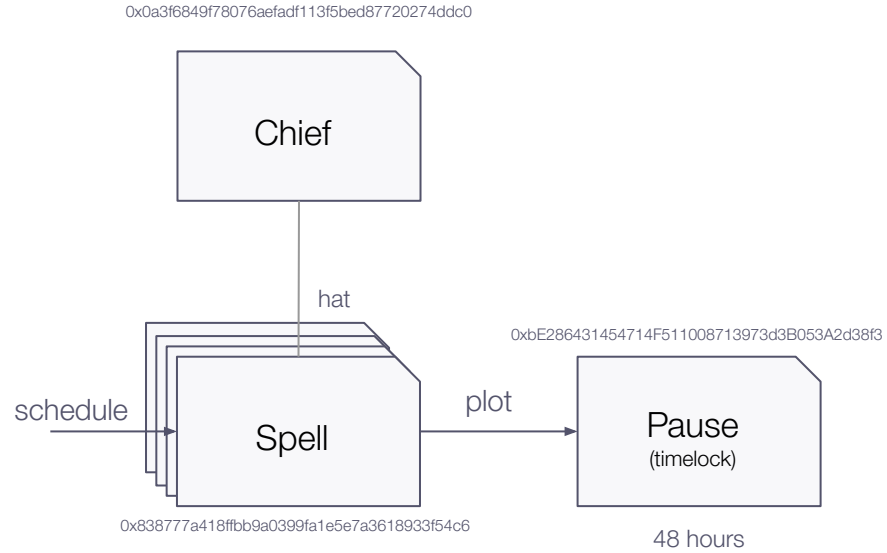
hat



0x838777a418ffbb9a0399fa1e5e7a3618933f54c6

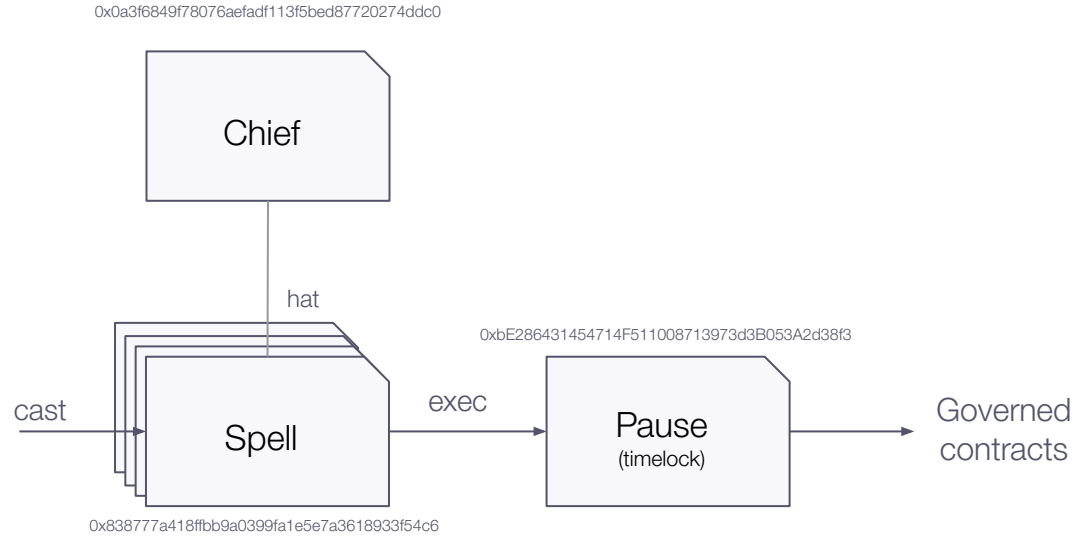
**MKR**

Governance  
users




**MKR**

Governance  
users



## MKR

- Battle-tested mechanism for approval voting
- Use of delays
- Well documented sources of user error + failure modes 

[docs.makerdao.com/smart-contract-modules/governance-module](https://docs.makerdao.com/smart-contract-modules/governance-module)



# issues



## domain-specific issues

<https://blog.openzeppelin.com/makerdao-critical-vulnerability>

## low-level calls to arbitrary contracts with arbitrary data

```
(bool success, bytes memory returnData) = target.call.value(value)(callData);  
(bool success, bytes memory returndata) = targets[i].call{value: values[i]}(calldatas[i]);
```

- check success conditions
- handle returned data if necessary
- EOA vs. account with code
- beware of reentrancy

## Use of ETH

- Ensure there's a way to deposit ETH if actions require ETH :)
- Beware of msg.value in batched governance actions

## flash-loans of governance tokens

- Avoid measuring voting power at current block
- Consider checks for EOAs

## Governance takeover

- Monitor the market for your governance token
- Adjust governance parameters accordingly
- Use of delays
- Use of community-vetted guardians

<https://forum.makerdao.com/t/urgent-flash-loans-and-securing-the-maker-protocol/4901>

<https://medium.com/coinmonks/how-to-turn-20m-into-340m-in-15-seconds-48d161a42311>

## Migrations and upgrades

- Beware of sensitive changes - they can brick governance
- Require security assessments (peer-reviews / 3rd parties)
- Document processes, include checklists and off-chain validations

On governance

## Closing thoughts

1

Use well-known contracts (Governor, Gnosis Safe)

2

Consider upgrades in governance logic (proxies, or strategies)

3

Protect against flash-loans of governance tokens

4

Delegation of voting power

5

Different delays and thresholds for different actions

On governance

## Closing thoughts

6

Guardians via multisigs with reputable members

7

Off-chain voting via snapshot.org

8

Documentation for powers, limitations, responsibilities, processes

9

Audits and peer-reviews for modifications

10

Incentives are challenging



On governance

## Where do I learn more ?

- References and developer documentation
- Public security audits of governance systems
- <https://ethereum.org/en/dao>
- <https://wiki.withtally.com/docs/other-protocols>

Series of sessions

# Secure Development

The dangers of token integration



Strategies for secure access controls



The dangers of price oracles



**Strategies for secure governance**



The dos and don'ts of smart contract upgrades

...

# We're hiring!

## Open Roles

- Blockchain Security Engineer
- Full Stack Ethereum Developer
- Open Source Developer
- Site Reliability Engineer
- and more!

Check out more

**[zpl.in/join](https://zpl.in/join)**

# Thanks!

## Learn more

[openzeppelin.com](https://openzeppelin.com)

**defender**.[openzeppelin.com](https://openzeppelin.com)

**blog**.[openzeppelin.com](https://openzeppelin.com)

**forum**.[openzeppelin.com](https://openzeppelin.com)

## Contact

 [@tinchoabbate](https://twitter.com/tinchoabbate)

[tincho@openzeppelin.com](mailto:tincho@openzeppelin.com)