

BoardRoom presents

# ARESPROTOCOL

Governance smart-contracts for decentralized investment funds

## Presentation Agenda

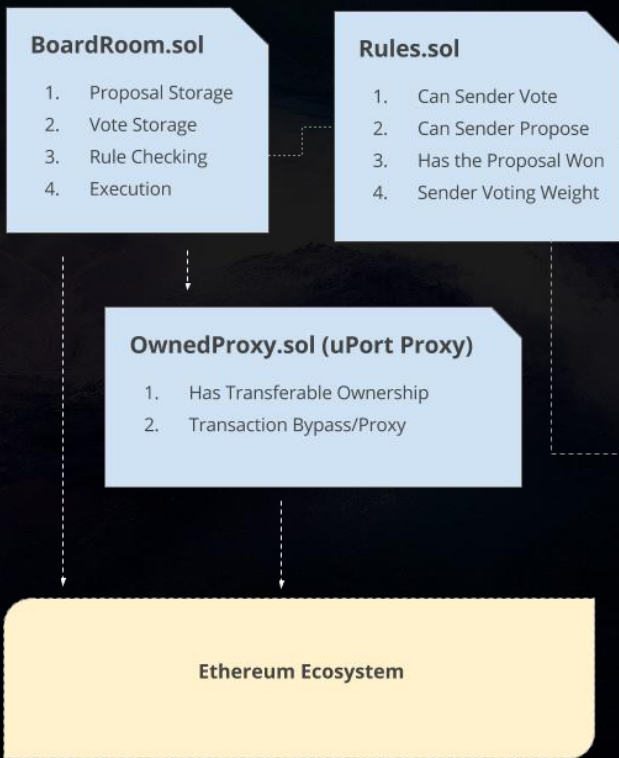
- 1) **BoardRoom** - project and design pattern (Nick Dodson)
- 2) **ARES Protocol** - design and implications (Dino Mark)
- 3) Planning, roadmap, contact information (Nick)

## **BoardRoom** Mission

To make smart-contract governance **generic** and **accessible to everyone**.

## BoardRoom Design Pattern

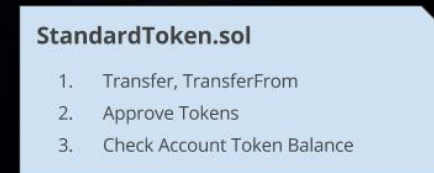
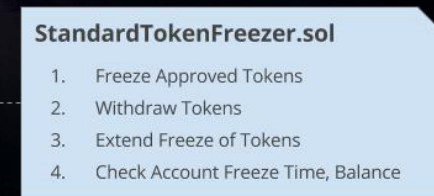
### BoardRoom Contract Design



### Registry Defined Membership & Voting Weight



### Token Defined Membership & Voting Weight



## BoardRoom Features

- Composability
- Simplicity: 1 contract, 2 interfaces
- **Modularity, evolutionary**: entire rulesets and structures can be swapped (in

1 TX.)

- Parsimony: complex rules, simplified execution
- Clarity/transparency: readable, predictable, comparable rule interfaces
- Identity persistence (via proxy)
- Robustness: generic design
- Ease of assembly
- Common interfaces: saves time on UI/contract design
- Extensible/diverse: hundreds of rulesets can be generated

## Interface Design

[github.com/boardroom-project/boardroom-contracts](https://github.com/boardroom-project/boardroom-contracts)

```
contract Board {  
  
    function newProposal(string _name, address _proxy, uint  
    _debatePeriod, address _destination, uint _value, bytes _calldata)  
        public returns (uint proposalID) {}  
  
    function vote(uint _proposalID, uint _position)  
        public returns (uint voteWeight) {}  
  
    function execute(uint _proposalID, bytes _calldata) public {}  
  
    function changeRules(address _rules) public {}  
  
}
```

```
contract Rules {  
  
    function canExecute(address _sender, uint _proposalID) public  
    constant returns (bool);  
  
    function canVote(address _sender, uint _proposalID) public constant  
    returns (bool);  
  
    function canPropose(address _sender) public constant returns (bool);  
  
    function votingWeightOf(address _sender, uint _proposalID) public  
    constant returns (uint);  
  
}
```

## Example: OpenRules.sol

[github.com/boardroom-project/boardroom-contracts](https://github.com/boardroom-project/boardroom-contracts)

```
import "BoardRoom.sol";
import "Rules.sol";

contract OpenRules is Rules {
    function canExecute(uint _proposalID) constant returns (bool) {
        BoardRoom board = BoardRoom(msg.sender);
        uint nay = board.positionWeightOf(_proposalID, 0);
        uint yea = board.positionWeightOf(_proposalID, 1);

        if(yea > nay) {
            return true;
        }
    }

    function canVote(address _sender, uint _proposalID) constant returns (bool) {
        return true;
    }

    function canPropose(address _sender) constant returns (bool) {
        return true;
    }

    function votingWeightOf(address _sender, uint _proposalID) constant returns (uint) {
        return 1;
    }
}
```

## Example Rulesets

[github.com/boardroom-project/boardroom-contracts](https://github.com/boardroom-project/boardroom-contracts)

- BondRules: required proposal bonds
- DelegatedVotingRules: delegated voting
- LiquidDemocracyRules: complete liquid democracy rules
- SingleAccountRules: single account controls all
- MultiSigRules: a wallet, with signatories
- MetaRules: proposal types, each with their own Rules contracts
- OpenRules: anyone can vote, propose, execute
- WeiFundRules: weifund campaign contributors > members
- CuratorRules: curated proposals
- TokenFreezerRules: required frozen tokens
- **ARES Rules...**



Introducing:

# **ARES** Protocol

Smart-contract governance for investment funds and collaborative decision making (without curators).

[ares.sh/white-paper](https://ares.sh/white-paper)

**DAO 2.0?**



## ARES Protocol

Improves on the old: through **simplification**, better game theory, and the use of **BoardRoom** design patterns.

# **ARES** Protocol | Basic Structure

## **No Curators**

Bonded Proposals

Instant Withdrawals

Grace Periods

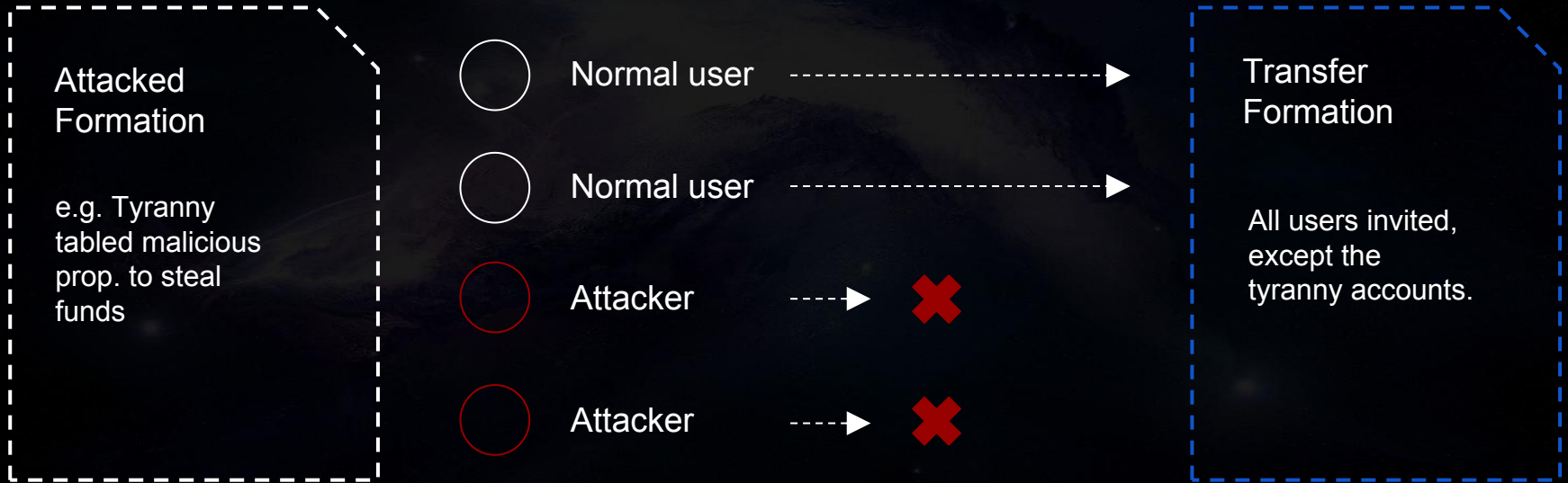
Rule Change Votes

Dynamic Quorum

Anti Re-Entrancy Safe-Sends

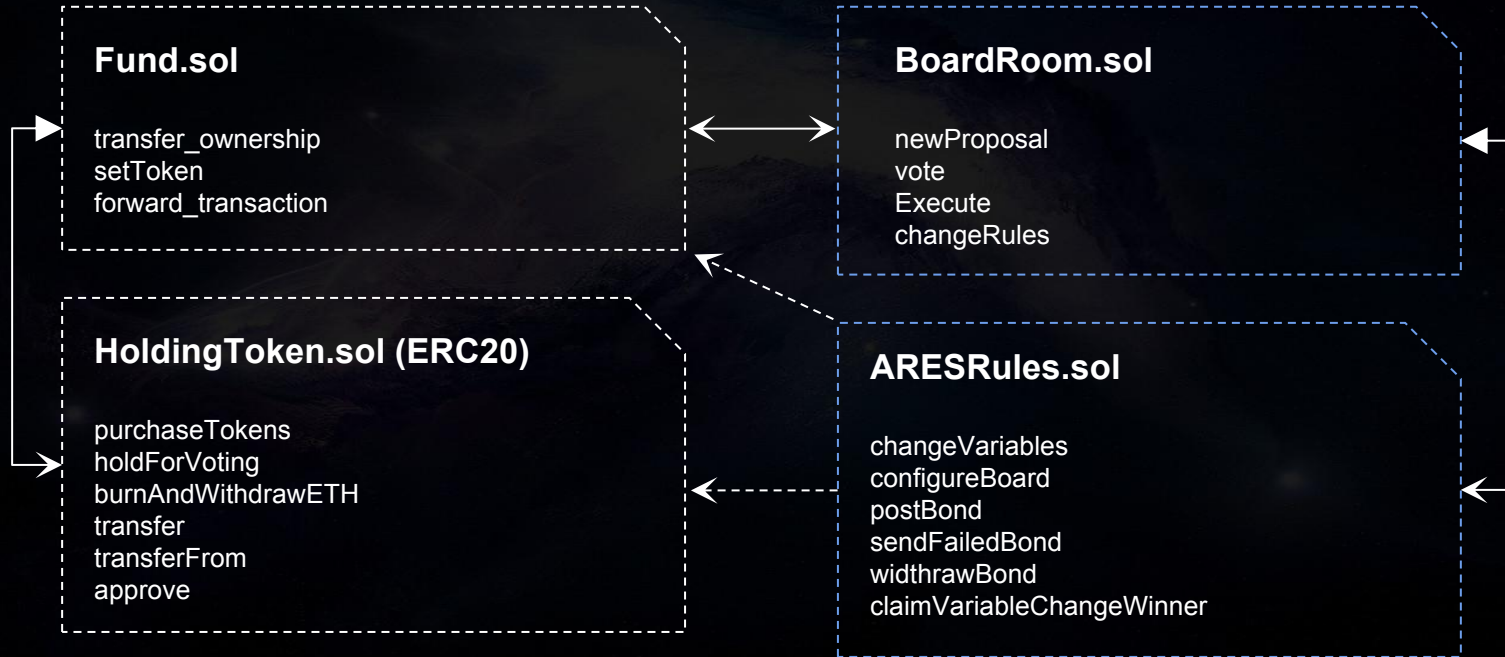
Modular Design allows future in-flight upgrades

## ARES Protocol | Thwarting Tyranny



Users can burn and buy in, the attacker is not invited.

## Contract Design



## ARES Additional Features

**Early warning detection** systems (email, sms, community policy), proposal and social policy, social and transactional data collection/research, prototype before polish (testing at small scales).

## RoadMap 2017

1. March - Multi third-party audit
2. Late **March** - Bug bounty
3. Late **April** - UI/Warning Systems/Social Policy
4. Early **May** - Ethjs-dao (small scale community DAO)
5. **July** - Formal verification begins
6. **August** - Data research release



## **WARNING!**

Do not attempt, in any way, to launch an ARES “DAO 2.0”.

We must conduct more research before we can verify its safety.

This protocol and design is highly experimental.

## BoardRoom.to

[github.com/boardroom-project](https://github.com/boardroom-project)

follow @GoBoardRoom

## Nick Dodson

[github.com/SilentCicero](https://github.com/SilentCicero)

follow @IAmNickDodson

## Ares.sh

[github.com/ares-protocol](https://github.com/ares-protocol)

[ares.sh/white-paper](https://ares.sh/white-paper)

## Dino Mark

[github.com/ares-protocol](https://github.com/ares-protocol)