Approaching Infinity:
Governance, and the case for experimentation

By Brett Sun
Approaching Infinity:
Governance, and the case for experimentation

By Brett Sun
Or perhaps better put,

Let’s throw things on the wall... and see what sticks

(And why we want that)
So what am I going to talk about?

“So Micro” (intra-network) governance, e.g.

Communities

NOT blockchains
The most successful projects will be communities, not startups
A brief history of governance

- Global scale (interdependence)
- Individual representation (decentralization)
A brief history of governance

- Hunter-gatherers
- Global scale (interdependence)
- Individual representation (decentralization)
A brief history of governance

- Hunter-gatherers
- Agricultural tribes
- Global scale (interdependence)
- Individual representation (decentralization)
A brief history of governance

Global scale (interdependence)

Individual representation (decentralization)

Hunter-gatherers

Agricultural tribes

Nobility & Monarchies
A brief history of governance

- Hunter-gatherers
- Agricultural tribes
- Nobility & Monarchies
- Mercantile Empires

- Global scale (interdependence)
- Individual representation (decentralization)
A brief history of governance

- Hunter-gatherers
- Agricultural tribes
- Nobility & Monarchies
- Mercantile Empires
- Liberal Democracies

Global scale (interdependence) vs. Individual representation (decentralization)
Well, what's the point?
“Centralized system[s] will necessarily need people who are not directly exposed to the cost of errors”

Nassim Taleb,
Skin in the Game
“Computers and networks are cheap... computer science secures money [and scales society] far better than accountants, police, [politicians,] and lawyers”

Nick Szabo,
Money, blockchains, and social scalability
Do you really want to trust those without skin in the game to maximize your community’s ability to socially scale?
The future of governance

- Individual representation (decentralization)
- Global scale (interdependence)
The future of governance

- Global scale (interdependence)
- Individual representation (decentralization)

Valley of coordination
The future of governance

Individual representation (decentralization)

Global scale (interdependence)
The future of governance

- Individual representation (decentralization)
- Global scale (interdependence)

???
The future of governance

Global scale (interdependence)

Individual representation (decentralization)

Peak of self-representation

???
Governance can be experimented with at the speed of software

(let’s throw things on the wall and see what sticks)
A framework for programming governance

github.com/aragon/aragonOS
Core design tenets

1. Primitives-focused
2. Default to opt-in
3. Interchangeable, malleable components
Let’s make some governance happen!
Access Control List (ACL)

```solidity
interface IACL {

    function initialize(address permissionsCreator) external;
    function hasPermission(
        address who,
        address where,
        bytes32 what
    ) external view returns (bool);

    /* Non-interface suggestions for implementations:

    function createPermission(address who, address where, bytes32 what) external;
    function grantPermission(address who, address where, bytes32 what) external;
    function revokePermission(address who, address where, bytes32 what) external;
    */
}
```

Interface:  https://github.com/aragon/aragonOS/blob/dev/contracts/acl/IACL.sol
Implementation:  https://github.com/aragon/aragonOS/blob/dev/contracts/acl/ACL.sol
Make me a DAO

```solidity
contract DAOFactory {
    function newDAO(address root) public returns (Kernel) {
        Kernel dao = new Kernel();
        dao.initialize(root); // Assume this also creates an ACL for the DAO
        return dao;
    }
}
```

Factory: https://github.com/aragon/aragonOS/blob/dev/contracts/factory/DAOFactoy.sol
And sprinkle on some (light) governance...
contract DAOFactory {
    function newDAO(address root) public returns (Kernel) {
        Kernel dao = new Kernel();
        dao.initialize(root); // Assume this also creates an ACL for the DAO

        ACL acl = ACL(dao.acl());

        Vault vault = Vault(dao.newAppInstance("vault"));
        Voting voting = Voting(dao.newAppInstance("voting"));
        Finance finance = Finance(dao.newAppInstance("finance"));
        TokenManager tokenManager = TokenManager(dao.newAppInstance("token-manager"));

        acl.createPermission(finance, vault, vault.TRANSFER_ROLE());
        acl.createPermission(voting, finance, finance.CREATE_PAYMENTS_ROLE());
        acl.createPermission(voting, finance, finance.EXECUTE_PAYMENTS_ROLE());
        acl.createPermission(voting, tokenManager, tokenManager.MINT_ROLE());
        acl.createPermission(voting, tokenManager, tokenManager.ASSIGN_ROLE());

        return dao;
    }
}
Goodies

• Program upgradability
• Package management

(which are inherently software governance questions)
Join us

Aragon Labs, Aragon Nest
Token Curated Registries, Liquid Voting

(We're hiring)
Come find us

Brett
@sohkai
Come find us

Tatu
@smokyish
Come find us

John
@lightcoin
Thank you
(and let’s go experiment!)