

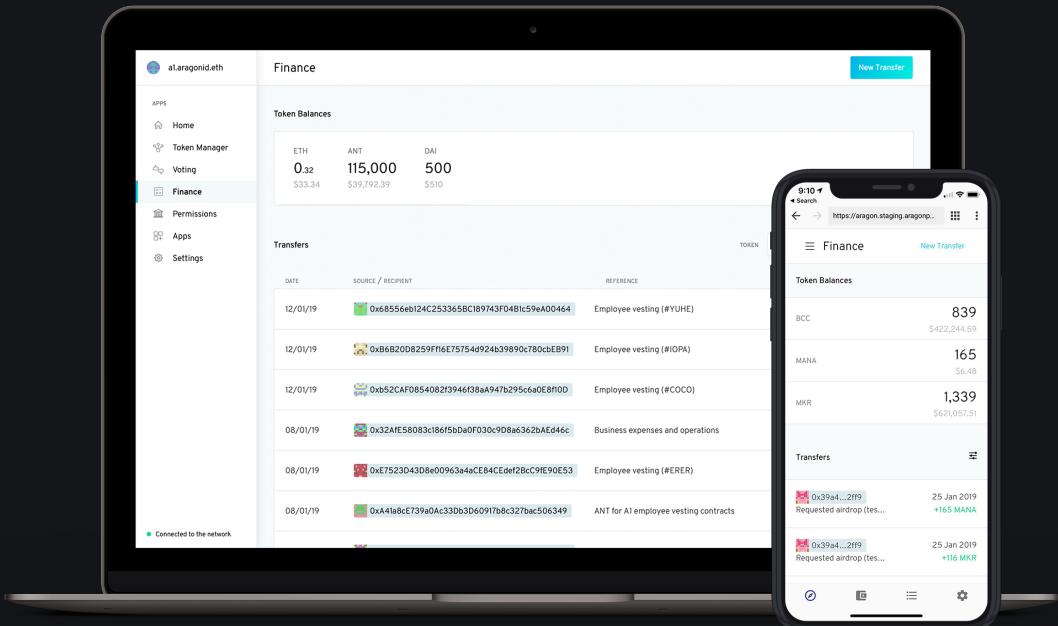


Bring your governance model to Aragon

Jorge Izquierdo - Aragon One, CTO
M-1 - Feb 8th, 2019

Aragon

- DApp to create and manage decentralized organizations
- Built on the Ethereum VM
- Available on the web, desktop and mobile
- 295 DAOs running on the Ethereum mainnet



app.aragon.org

Aragon OG

- Delaware C-Corps on the blockchain
- Focused on one specific usecase that we knew very well
- Bombarded with 'how can I do X on Aragon?' questions

The screenshot shows the Aragon web interface with the URL <https://alpha.aragon.one/#/ownership/stock/new>. The main sidebar includes links for Home, Ownership (highlighted), Voting, Roles, Fundraising, Accounting, Bylaws, and Settings. The main content area displays a table of current stock classes:

Shareholder	Stock class	Shares
Company reserves	CVS	0
Me	CVS	9900000
Purple Frog	CVS	100
Young Wave	CVS	100000

A modal window titled "Add stock class" is open on the right side. It contains fields for "Stock name" (Unicorn Stock), "Stock symbol" (ARN-), "Voting power" (10), "Economic rights" (10), and a checkbox for "Use existing ERC20 token" which is checked. Below the modal, a note states: "A voting will be needed to Add new stock Requires 50% support in a 7 day voting". At the bottom right of the modal is a purple button labeled "Create 'Add new stock' voting".

aragonOS

- The most powerful and flexible smart contract framework
- Upgradeability and access control for free
- Audited by the White Hat Group and Consensys Diligence
- +\$1m AUM
- +3,000 live aragonOS contracts



aragonOS

github.com/aragon/aragonOS

UNIX design philosophy

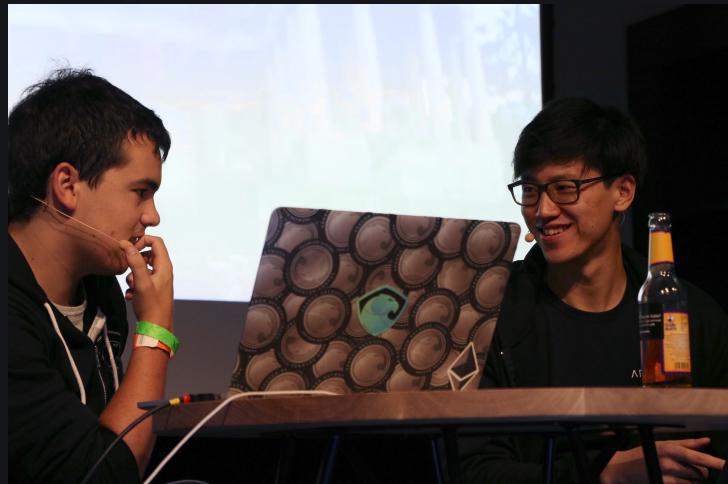
1. Make each *program* **do one thing well and one thing only**. To do a new job, build afresh rather than complicate old programs by adding new "features".
2. Expect the **output of every program to become the input to another**, as yet unknown, program. Don't clutter output with extraneous information. Don't insist on interactive input.



Ken Thompson and Dennis Ritchie

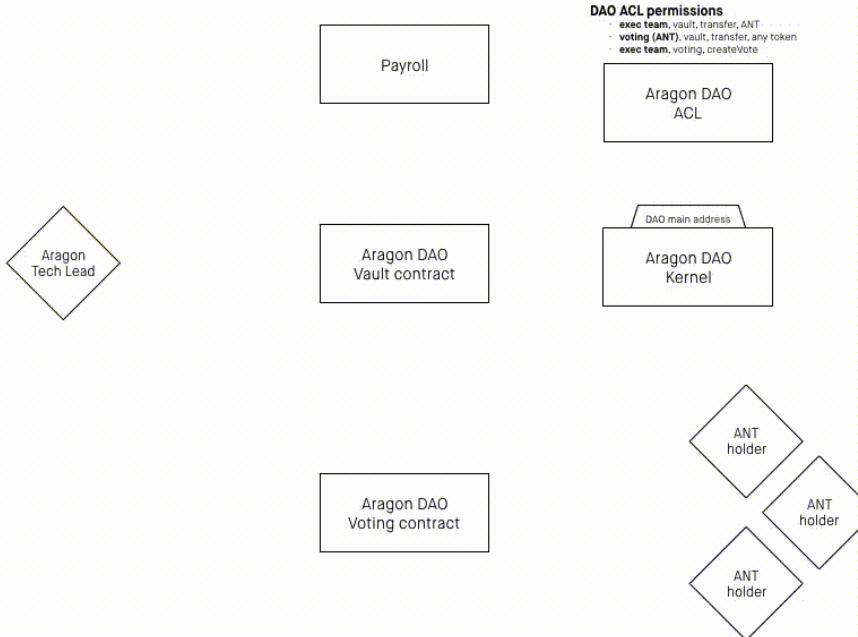
aragonOS design philosophy

1. Make each *Aragon app* **do one thing well and one thing only**. To do a new job, build afresh rather than complicate old programs by adding new "features".
2. Expect the **output of every contract to become the input to another**, as yet unknown, contract. Don't clutter output with extraneous complexity. Don't insist on interactive input.

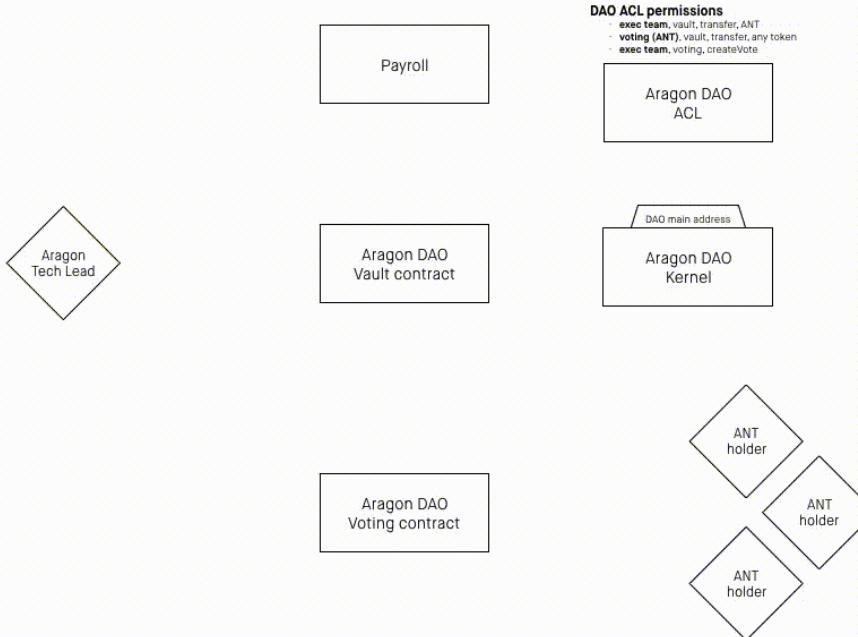


Brett and Jorge

aragonOS flow



aragonOS flow



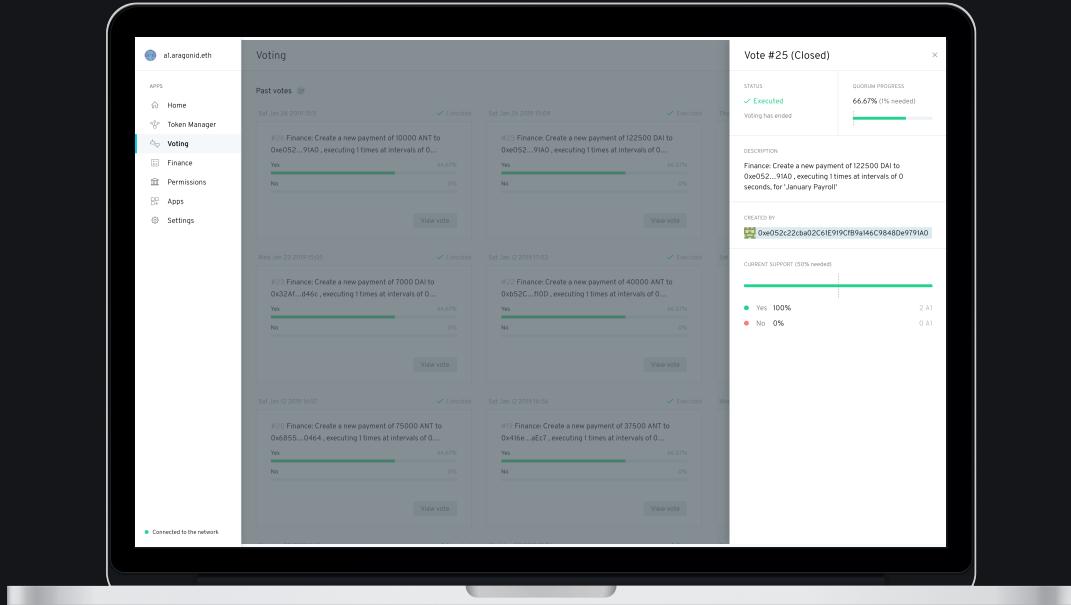
Governance as access control

```
interface IACL {  
    function hasPermission(address who, address where, bytes32 what, bytes how)  
        external view  
        returns (bool);  
}
```

aragonOS: acl/IACL.sol

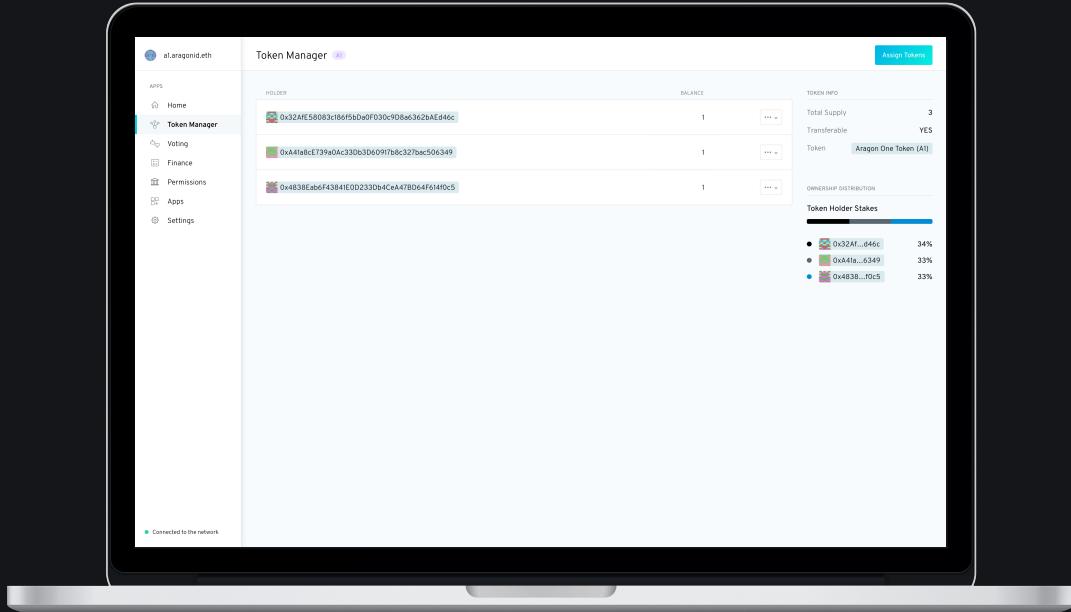
Aragon apps

- Voting



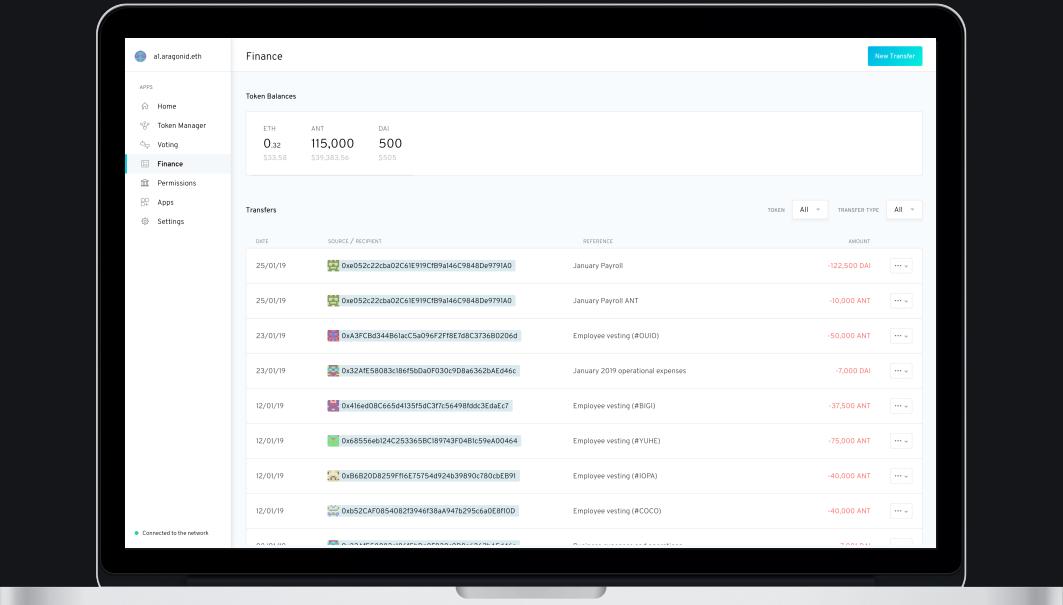
Aragon apps

- Voting
- Token Manager



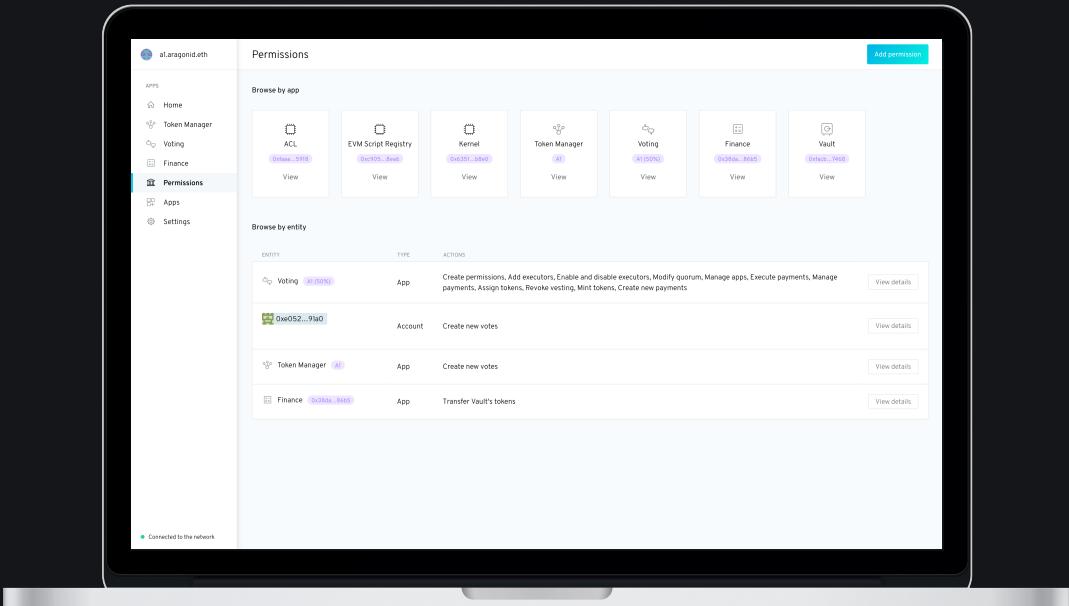
Aragon apps

- Voting
- Token Manager
- Finance



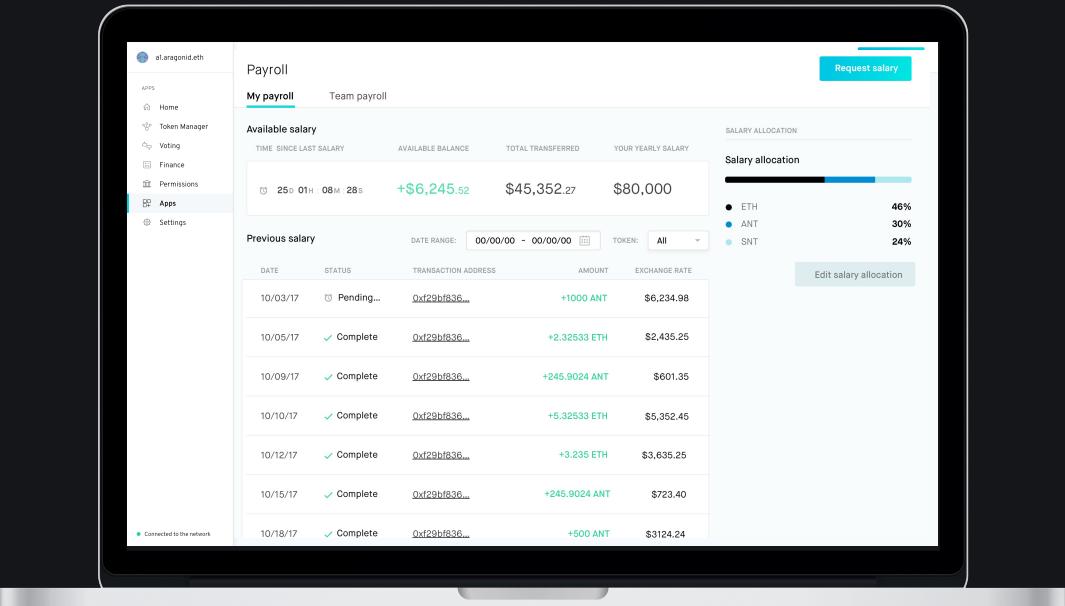
Aragon apps

- Voting
- Token Manager
- Finance
- Permissions (native)



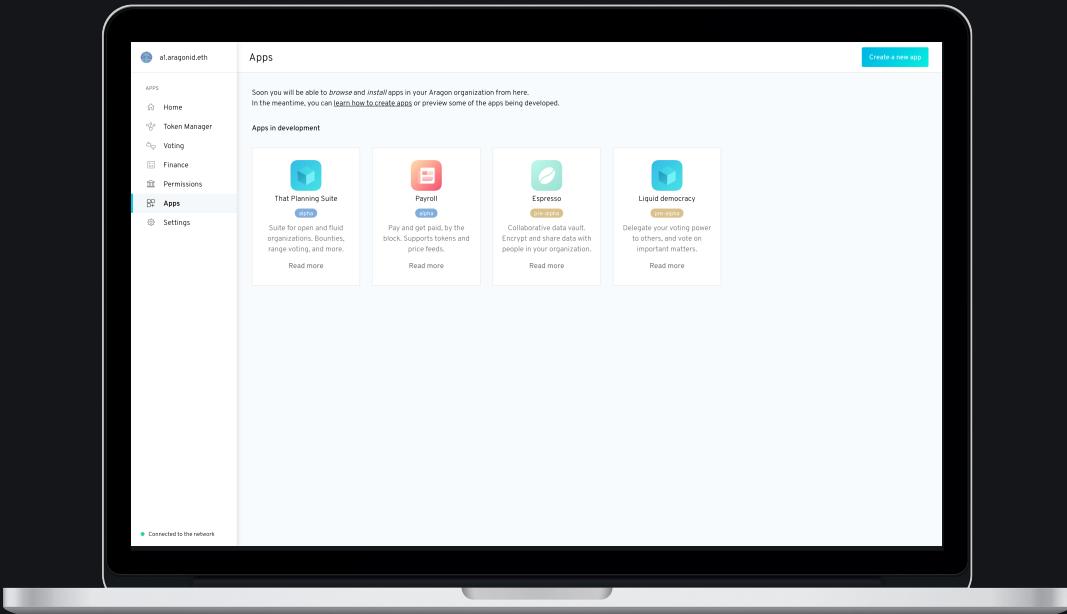
Aragon apps

- Voting
- Token Manager
- Finance
- Permissions (native)
- Payroll (soon!)



Aragon apps

- Voting
- Token Manager
- Finance
- Permissions (native)
- Payroll (soon!)
- An app platform



aragonSDK

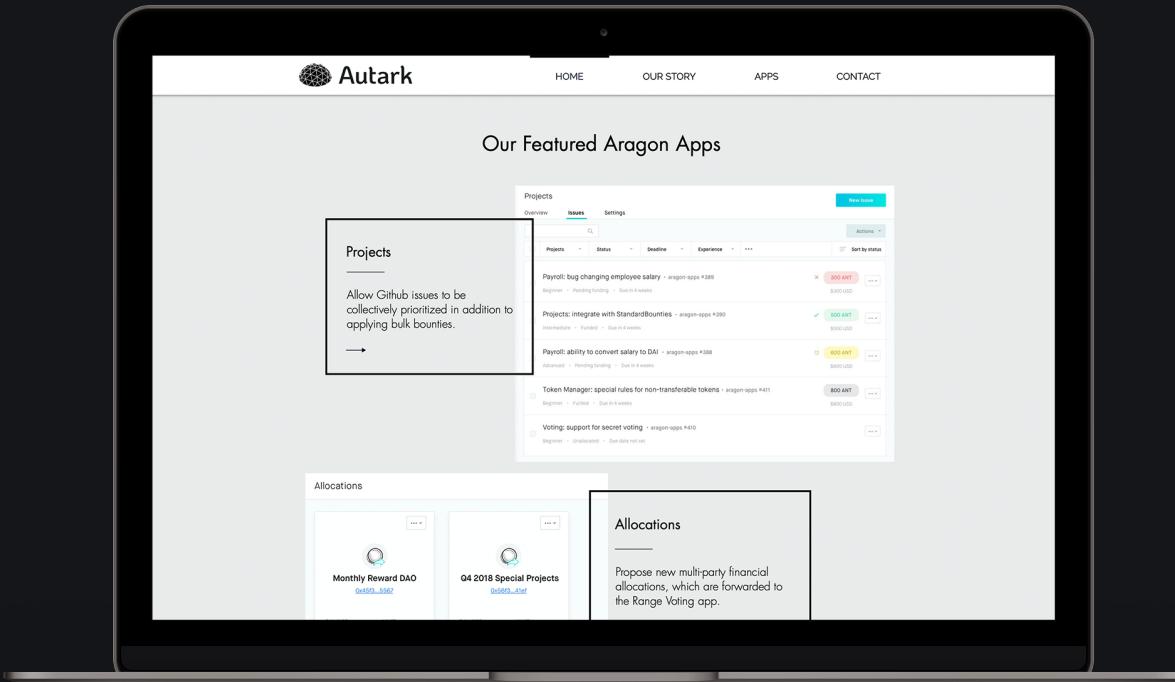
- The easiest way to build true decentralized apps
- +10 apps in development
- DAO apps and beyond



aragon**SDK**

hack.aragon.org

Autark: Planning Suite



autark.xyz/aragon-apps

Althea: Subnet DAOs

The screenshot shows a web-based application interface for managing nodes. On the left, a sidebar titled "APPS" contains links: "Home", "Node List" (which is selected and highlighted in blue), "Permissions (SOON)", "Apps (SOON)", and "Settings". The main content area has a header "QUERY FOR AN EXISTING NODE *". Below it is a search input field and a "Submit" button. Further down, there are two input fields: "ETHEREUM ADDRESS" containing the value "0xb4124ceb3451635dacedd11767f004d8a28c6ee7" and "IP ADDRESS" containing "0xc0a8010ac0a8010a". Each of these fields has a "Submit" button to its right. At the bottom center is another "Submit" button.

altheamesh.com/governance-paper

Kits: DAO setup scripts

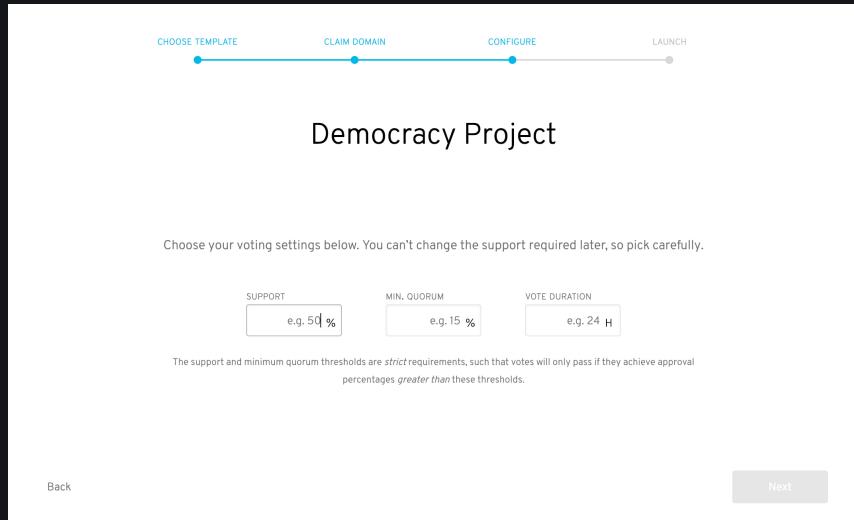
- A Kit atomically create and configure a specific type of DAO
- Public kits: Democracy and Multisig
- Configuring with code removes human mistakes when setting up complex DAOs

```
contract DemocracyKit is BetaKitBase {
    function newInstance(
        string name,
        address[] holders,
        uint256[] tokens,
        uint64 supportNeeded,
        uint64 minAcceptanceQuorum,
        uint64 voteDuration
    ) public {
        MiniMeToken token = popTokenCache(msg.sender);
        Kernel dao;
        ACL acl;
        Voting voting;
        (dao, acl, , , voting) = createDAO(
            name,
            token,
            holders,
            tokens,
            uint256(-1)
        );
    }
}
```

dao-kits: kits/democracy

Kits: DAO setup scripts

- A Kit atomically create and configure a specific type of DAO
- Public kits: Democracy and Multisig
- Configuring with code removes human mistakes when setting up complex DAOs



dao-kits: kits/democracy

AGP voting kit

```
contract AGP1Kit is KitBase, APMNamehash, IsContract {
    uint64 constant public MAIN_VOTING_SUPPORT = 50 * 10**16; // > 50%
    uint64 constant public MAIN_VOTING_QUORUM = 0; // Just 1 vote is enough
    uint64 constant public MAIN_VOTING_VOTE_TIME = 48 hours;

    uint64 constant public META_TRACK_VOTING_SUPPORT = 666666666666666666; // > two thirds
    uint64 constant public META_TRACK_VOTING_QUORUM = 0; // Just 1 vote is enough
    uint64 constant public META_TRACK_VOTING_VOTE_TIME = 48 hours;

    uint64 constant public FINANCE_PERIOD_DURATION = 7889400; // 365.25 days / 4

    bytes32 constant public financeAppId = apmNamehash("finance");
    bytes32 constant public vaultAppId = apmNamehash("vault");
    bytes32 constant public votingAppId = apmNamehash("voting");

    function newInstance(MiniMeToken _ant, address _multisig) external returns (Kernel) {
        Kernel dao = fac.newDAO(this);
        ACL acl = ACL(dao.acl());

        acl.createPermission(this, dao, dao.APP_MANAGER_ROLE(), this);
```

dao-kits: kits/agp1

AGP voting kit

The screenshot shows the governance application interface with the following sections:

- Left Sidebar:** APPS (Home, Voting, Finance), Permissions (selected), Apps, Settings.
- Permissions Section:** A grid of icons and details for various entities:
 - ACL (0x169...7e8) - View
 - EVM Script Registry (0x7fd...6d8) - View
 - Kernel (0x2d8...128) - View
 - Voting (ANT (50%)) - View
 - Voting (ANT (67%)) - View
 - Finance (0xd970...10f) - View
 - Vault (0xaacc...9e69) - View
- Browse by entity:** A table listing entities and their actions:

ENTITY	TYPE	ACTIONS
Voting (ANT (67%))	App	Create permissions, Manage apps, Modify quorum, Modify support, Modify quorum, Modify support
0xcafe...5f2c	Account	Create new votes, Create new votes
Finance (0xd970...10f)	App	Transfer Vault's tokens
Voting (ANT (50%))	App	Create new payments, Execute payments, Manage payments
- Bottom Status:** Connected to the network.

dao-kits: kits/agp1

Aragon Concierge



Add Discussion

Concierge

+ :: Concierge refers to the projects we actively work with to make sure Aragon fits their needs and helping them realize their governance vision using Aragon.

Projects

Active	2	Backlog	0	Completed	1	Hidden Columns
Melon Council	Jorge Izquierdo	+ New				
Althea	Luke Duncan	1				
DAppNode token v1	Jorge Izquierdo	+ New				
No Status	2					

github.com/aragon/concierge

Concierge: Melon Council

Governance structure

The Melon Council is composed of 2 sub-groups:

- Melon Technical Council (MTC)
- Melon Business/Users representatives (MEB)

There can only be up to 2 MEB members for every 5 MTC members.

Decisions

The entire council (General membership or membership), formed by members of both sub-groups (1 person 1 vote):

- Inflation allocation (approving requests for MIP funding and project funding)
- Setting priority MIPs
- Setting the protocol fee (AMGU)
- Inclusion of a new member in MTC (supermajority)
- **Election of Chair and Vice-chair of the Melon Council at the beginning of each year** (purely signaling, no on-chain powers)



The MTC sub-group:

- Approval of new contracts
- Resolution of the ENS name

The MEB sub-group (this will happen off-chain and the membership is bound to executing it via a real world legal contract):

- Remove MTC members

Governance model

Permissions

The relevant permissions that will be set are:

Permissions table

App	Permission	Grantee	Manager	+ New
Kernel	Set app	Voting (Membership)	Voting (Membership)	
ACL	Create permissions	Voting (Membership)	Voting (Membership)	
Voting (Membership)	Create votes	TM (Membership)	Voting (Membership)	
Voting (MTC)	Create votes	TM (MTC)	Voting (MTC)	
Voting (Membership 2/3)	Create votes	TM (Membership)	Voting (Membership 2/3)	
Vault	Transfer tokens	Finance	Voting (Membership)	
Finance	Create payments	Voting (Membership)	Voting (Membership)	
Finance	Execute payments	Voting (Membership)	Voting (Membership)	
Finance	Disable payments	Voting (Membership)	Voting (Membership)	
TM (Membership)	Mint tokens	Voting (Membership)	Voting (Membership)	
TM (MTC)	Mint tokens	Voting (Membership 2/3)	Voting (Membership 2/3)	
Agent (protocol)	Execute action	Voting (Membership)	Voting (Membership)	
Agent (technical)	Execute action	Voting (MTC)	Voting (MTC)	

COUNT 13

The Voting (Membership) will be the root authority in the DAO, being the manager of most permissions and in charge of DAO contract upgrades, app installation and creating new permissions. Any permission not explicitly set here will either be granted to the root authority or not created (the root authority can create them at any time in the future).

Kit specification

Concierge: Melon Council

Permissions
The relevant permissions that will be set are:

Permissions table

App	Permission	Grantee	Manager
Kernel	Set app	Voting (Membership)	Voting (Membership)
ACL	Create permissions	Voting (Membership)	Voting (Membership)
Voting (Membership)	Create votes	TM (Membership)	Voting (Membership)
Voting (MTC)	Create votes	TM (MTC)	Voting (MTC)
Voting (Membership 2/3)	Create votes	TM (Membership)	Voting (Membership 2/3)
Vault	Transfer tokens	Finance	Voting (Membership)
Finance	Create payments	Voting (Membership)	Voting (Membership)
Finance	Execute payments	Voting (Membership)	Voting (Membership)
Finance	Disable payments	Voting (Membership)	Voting (Membership)
TM (Membership)	Mint tokens	Voting (Membership)	Voting (Membership)
TM (MTC)	Mint tokens	Voting (Membership 2/3)	Voting (Membership 2/3)
Agent (protocol)	Execute action	Voting (Membership)	Voting (Membership)
Agent (technical)	Execute action	Voting (MTC)	Voting (MTC)

+ New
COUNT 13

The **Voting (Membership)** will be the root authority in the DAO, being the manager of most permissions and in charge of DAO contract upgrades, app installation and creating new permissions. Any permission not explicitly set here will either be granted to the root authority or not created (the root authority can create them at any time in the future).



```
contract MelonKit is KitBase, APMNamehash, IsContract {
    string constant private MAIN_TOKEN_SYMBOL = "MGM";
    string constant private MAIN_TOKEN_NAME = "Melon General Membership";
    string constant private MTC_TOKEN_SYMBOL = "MTC";
    string constant private MTC_TOKEN_NAME = "Melon Technical Council";

    uint64 constant public MAIN_VOTING_SUPPORT = 50 * 10**16; // > 50%
    uint64 constant public MAIN_VOTING_QUORUM = 0; // Just 1 vote is enough
    uint64 constant public MAIN_VOTING_VOTE_TIME = 48 hours;

    uint64 constant public SUPERMAJORITY_VOTING_SUPPORT = 6666666666666666; // > two thirds
    uint64 constant public SUPERMAJORITY_VOTING_QUORUM = 0; // Just 1 vote is enough
    uint64 constant public SUPERMAJORITY_VOTING_VOTE_TIME = 48 hours;

    uint64 constant public MTC_VOTING_SUPPORT = 50 * 10**16; // > 50%
    uint64 constant public MTC_VOTING_QUORUM = 0; // Just 1 vote is enough
    uint64 constant public MTC_VOTING_VOTE_TIME = 48 hours;

    uint64 constant public FINANCE_PERIOD_DURATION = 7889400; // 365.25 days / 4

    function newInstance1(address[] mbeMembers, address[] mtcMembers) external {
        Kernel dao = fac.newDAO(this);
        ACL acl = ACL(dao.acl());

        acl.createPermission(this, dao, dao.APP_MANAGER_ROLE(), this);
    }
}
```

Kit specification

Kit implementation
concierge: kits/melon

Concierge: Melon Council

```
contract MelonKit is KitBase, APMNamehash, IsContract {  
    string constant private MAIN_TOKEN_SYMBOL = "MGM";  
    string constant private MAIN_TOKEN_NAME = "Melon General Membership";  
    string constant private MTC_TOKEN_SYMBOL = "MTC";  
    string constant private MTC_TOKEN_NAME = "Melon Technical Council";  
  
    uint64 constant public MAIN_VOTING_SUPPORT = 50 * 10**16; // > 50%  
    uint64 constant public MAIN_VOTING_QUORUM = 0; // Just 1 vote is enough  
    uint64 constant public MAIN_VOTING_VOTE_TIME = 48 hours;  
  
    uint64 constant public SUPERMAJORITY_VOTING_SUPPORT = 6666666666666666; // > two thirds  
    uint64 constant public SUPERMAJORITY_VOTING_QUORUM = 0; // Just 1 vote is enough  
    uint64 constant public SUPERMAJORITY_VOTING_VOTE_TIME = 48 hours;  
  
    uint64 constant public MTC_VOTING_SUPPORT = 50 * 10**16; // > 50%  
    uint64 constant public MTC_VOTING_QUORUM = 0; // Just 1 vote is enough  
    uint64 constant public MTC_VOTING_VOTE_TIME = 48 hours;  
  
    uint64 constant public FINANCE_PERIOD_DURATION = 7889400; // 365.25 days / 4  
  
    function newInstance1(address[] mebMembers, address[] mtcMembers) external {  
        Kernel dao = fac.newDAO(this);  
        ACL acl = ACL(dao.acl());  
  
        acl.createPermission(this, dao, dao.APP_MANAGER_ROLE(), this);  
    }  
}
```



Kit implementation
concierge: kits/melon

Live DAO

Melon Council on Aragon

Aragon

https://aragon-staging.aragonpm.com/#/0xf8085ca7bed7747d4455ce7861d83b760

Buscar

Add permission

Permissions

APPs

Browse by app

ACL	EVM Script Registry	Kernel	Token Manager	Token Manager	Voting
View	View	View	View	View	View
0xdbb9...9138	0xc874...ba75	0x1808...77h2	MGM	MTC	MGM (50%)

Browse by entity

ENTITY	TYPE	ACTIONS
Voting	App	Create permissions, Manage apps, Create new payments, Execute payments, Manage payments, Issue tokens, Assign tokens, Burn tokens, Mint tokens, Modify quorum, Modify support, Execute actions
Finance	App	Transfer Vault's tokens
Token Manager	App	Create new votes, Create new votes

Connected to the network

9:14 🔍

Search

https://aragon-staging.aragonpm...

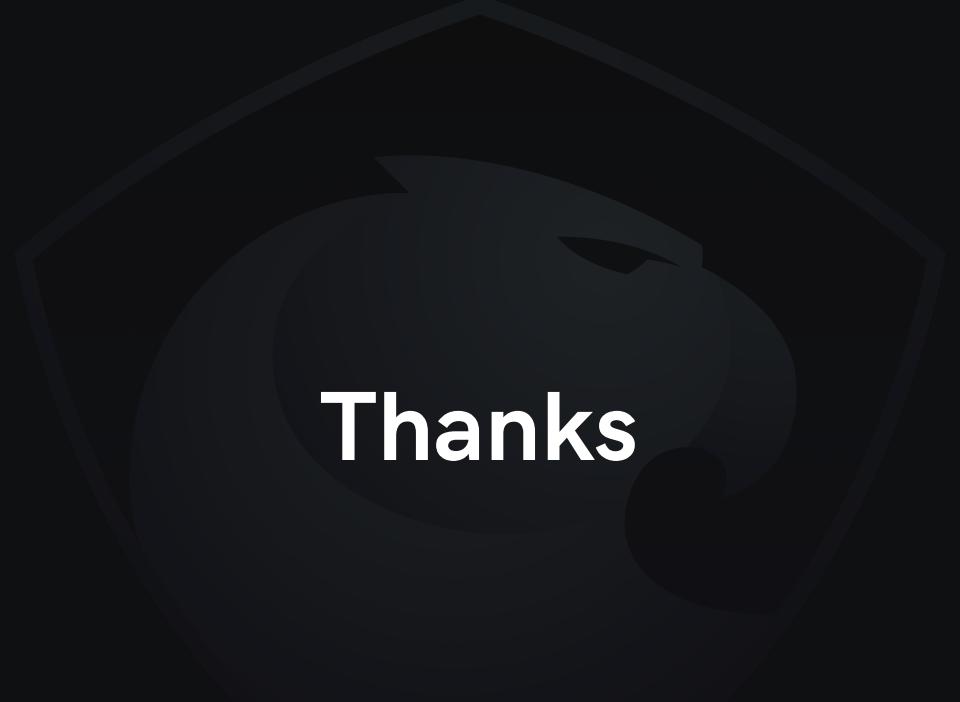
Permissions

Token Manager	Voting
MTC	MGM (50%)
View	View
Voting	Voting
MGM (67%)	MTC (50%)
View	View
Finance	Vault
0xab91...a14b	0x70c9...ab51
View	View

≡

☰

⚙️



Thanks

aragon.org

@AragonProject @AragonOneTeam @izqui9