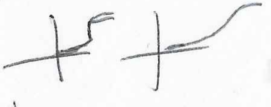
This is what I was thinking when it comes to calculating clustering co-efficients within social network topologies. but you'd likely run the result through a function (Sigmoid):



$\frac{1}{1+e^{-x\delta}}$ shifted →.

(not sure how feasible).

3.2. Colony

Colony is a DAO framework based on a reputation system (i.e., decision power is weighted by the user reputation). It aims to help organizations create their own DAOs, named 'colonies', providing financial management, ownership, structure and authority.

The Colony network is composed of a suite of smart contracts which are deployed on the Ethereum blockchain. However, although it is planned to be included in the near future, at this moment, organizations cannot customize colonies with smart contract modules in order to implement their specific governance model.

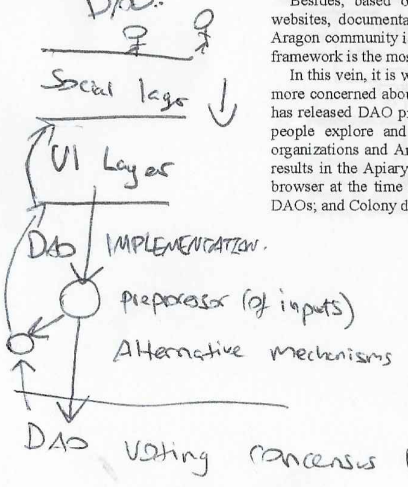
According to Colony's whitepaper [18], the structure of a colony is based on *domains* and the *permissions* that accounts may have in each domain.

3.3. DAOstack

DAOstack supports the process of DAO development by providing a library of governance protocols and user interfaces that facilitates their creation and management.

From the point of view of DAOstack, a DAO is seen as a network of stakeholders who make non-hierarchical decisions about shared resources [19]. In DAOstack, decisions are initiated by *proposals*. The framework of DAOstack is composed of a set of several modules or layers [11].

This is how I originally Modelled a DAO:



This Stack based model was before I'd done ANY reading on DAOs.

3.4. Aragon, Colony and DAOstack comparison

Implementing DAOs with rich functionality requires the next mechanisms: (i) Financial management (i.e., funding); (i) A voting system; (ii) Tokens for membership management and voting power; (iii) Ability to create new governance models relying on smart contracts; (iv) Templates of organization models; and (v) Permissions, which are of major importance when different roles are accessing to a DAO. The comparison among the three frameworks (Aragon, Colony and DAOstack) in relation to these main mechanisms associated with the process of building DAOs are summarized in Table 1.

How do you control monetary Policy – through Voting I assume?

As can be seen in Table 1, among these three frameworks, only Aragon offer prototypes of DAOs (organization templates), which can be configured, and provide mechanisms to add smart contracts that enable the definition of new governance models. On the other hand, DAOstack does not support the definition of permissions and roles. Therefore, Aragon is more flexible as it satisfies all the above requirements for DAO development.

Besides, based on our findings searching the Web (blockchain-based project websites, documentation, social media and forums, and overall dissemination), the Aragon community is the most active in the field of DAOs [20, 21]. Furthermore, this framework is the most developed and most widely adopted among developers.

In this vein, it is worth highlighting the case of 1Hive [22], one of the communities more concerned about DAOs and their adoption using the Aragon framework. 1Hive has released DAO profiles in Apiary [23], their DAO explorer. Apiary aims to help people explore and understand the Aragon ecosystem indexing all the Aragon organizations and Aragon apps on the Ethereum mainnet network and listing all the results in the Apiary Browser [24]. 1544 Aragon organizations were indexed in this browser at the time of this writing. Note in comparison, DAOstack has ~60 public DAOs; and Colony does not provide a number at this time. [25, 26].

Considerations.

Table 1. Comparison of Aragon, Colony and DAOstack.

Mechanism	Aragon	Colony	DAOstack
Token	✓	✓	✓
Reputation	✓	✓	✓
Funding	✓	✓	✓
Permissions	✓	✓	X
Voting system	✓	✓	✓
Organization templates	✓	X	X
New governance models	✓	X	X

4. DAO development: Task management case study

Although we have implemented several DAOs focused on testing and validating the Aragon framework, the case study in this section aims to be just complex enough to illustrate DAO development using the Aragon framework, as it shows how to incorporate new functionality. Thus, in this section the development process of an example Aragon DAO will be described.

In order to facilitate replicability, and to exemplify the current state of the framework, we will make explicit the points in the development process that were more challenging due to the lack of resources, documentation or relevant issues in the framework. In addition, this detailed description may serve well to computer scientists and software developers intending to use Aragon for practical purposes.

Requi for a framew

The objective of the case study is to implement a DAO that supports collaborative activities of researchers, located in different countries, who participate in common research projects. The DAO has to include a Dapp that manages those tasks necessary to obtain the deliverables associated with a specific project.

Although a formalized overview of the Aragon architecture is missing in its available documentation, as mentioned earlier, we have experimented with some implementations, and we have modelled an approach in order to formalize its understanding, that is shown in Fig. 1 as a UML class diagram.