



# DAOstack

## Le système d'exploitation pour les Organisations Autonomes Décentralisées (DAO)

**White Paper V1.0**

29 Octobre 2017

Travail en cours, les commentaires sont les bienvenus

**[fmatan@gmail.com](mailto:fmatan@gmail.com)**

### Résumé

DAOstack est un système d'exploitation pour une nouvelle forme d'organisation: le DAO. Les Organisations Autonomes Décentralisées ont capturé l'imagination des meilleurs esprits dans l'espace de la blockchain, mais malgré leurs promesses, elles sont restées une idée abstraite. L'un des principaux points d'échec est le manque de cadre solide pour la gouvernance décentralisée de la blockchain. A la base de DAOstack , Arc<sup>1</sup> est un cadre ouvert et universel de contrats intelligents pour la gouvernance décentralisée et la gestion de la valeur collective sur la blockchain. Tout comme HTTP permet la création et l'interopérabilité de sites Web et d'applications Web, DAOstack permet la création et l'interopérabilité de sociétés Web, d'applications de collaboration et de DAO, ainsi que l'alignement de leurs intérêts. Le résultat est un nouveau réseau de collaboration ouverte, dans lequel les collectifs peuvent s'auto-organiser autour d'objectifs et de valeurs partagées, au-delà de la croissance économique pure. Nous pensons que ce sera une étape cruciale dans l'évolution de la société vers un avenir plus coopératif et durable.

---

<sup>1</sup> Dérivé du mot grec "Arche", qui signifie «méthode de gouvernement».



## Table des matières

Résumé	1
<b>1. Introduction</b>	<b>4</b>
<b>2. DAO: L'avenir des Organisations</b>	<b>5</b>
Organisations patrimoniales	5
Les agences Blockchain	6
DAO	6
Organisations ouvertes	7
	11
<b>3. Gouvernance de la Blockchain</b>	<b>12</b>
Exemple	13
Structure de gouvernance de la Blockchain	15
Évolutivité	19
Système de vote DAOstack	21
<b>4. Le stack de DAO</b>	<b>23</b>
Arc	24
ArcJS	29
The ArcHives	29
Alchemy	31
<b>5. L'économie du stack</b>	<b>31</b>
Economie du token circulaire	31
Économie des DAO	32
Le modèle de stack	34









@Yg`V]``Yg`gc`]XYg`fYdf`fgYbhYbh`XYg`U[Ybhg`XUbg`  
 dci]c]f`Xfi]bZ`iYbWY`ci``Yif`f`fdihUh]cb`fld`ig`]`g`  
 dggYgg]cb`XY`hc`Yb`bUh]Z`fld`ig``U`VU``Y`Ygh`[fUb  
 dfcdcY`XfiU``ciYf`)`9H<`{``fiU[Ybh`5`dcif`gU`df`fW]  
 `fiYbhfYdf]gY`jchYbhž``Yif`jchY`fhUbh`dcbXfff`dUf  
 f`fdihUh]cb`Ygh`XfiUWWcfX`UjYW``U`dfcdc]h]cbž``Y`W

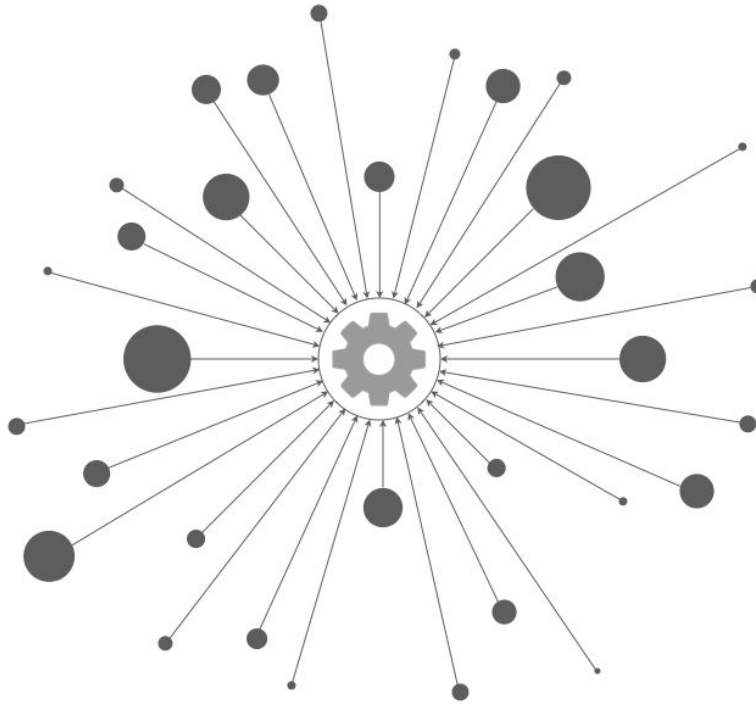
## @YDAOs

@Yg`U[YbWYg`ZcbWh]cbbYbh`UjYW`XYg`WcbhfUhg`]bhY```  
 bY`dYijYbh`dUg`...hfY`Vf]gfYg`Yh`bY`gcbh`acX]Z]UV`Y  
 bcb`...hfY`Ui`hcbcaYg`Yb`ZcbWh]cb`Xi`gmgh`aY`XY`[ci]j  
 cdh]cb`XY`jYhc`g`if`gcb`dfcdfY`dfcWYggi`g`XY`df]gY`

Un DAO est un réseau maillé sans centre d'agences, qui est également une agence en soi. Il n'y a pas de point de contrôle unique ou d'échec dans l'organisation. Au lieu de la gestion centrale, il existe une coordination indirecte entre les agents, également connue en biologie comme la stigmergie, déclenchée par des incitations et du code. Un DAO est une entité auto-organisatrice et, en général, ressemble davantage à un organisme qu'à une organisation.

### Topologie du DAO

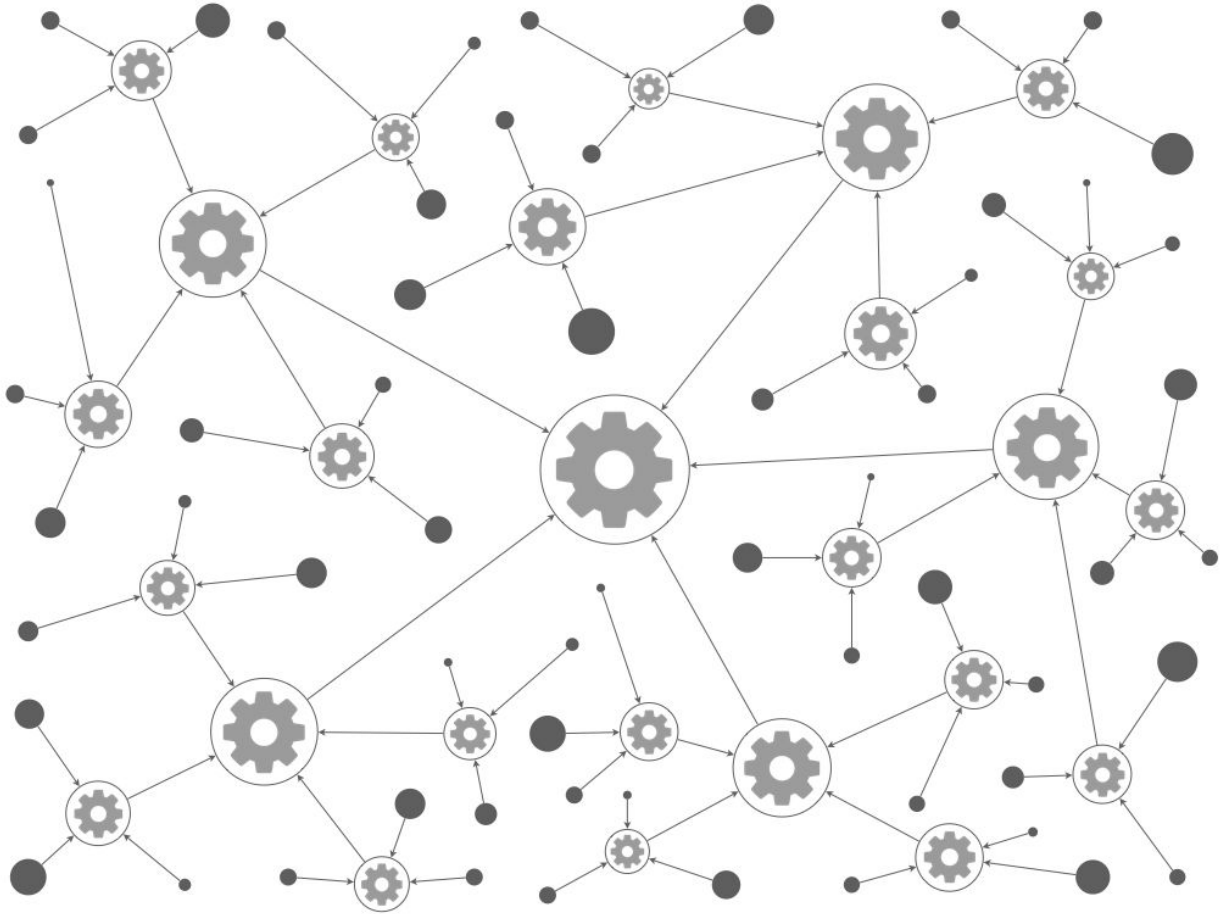
Il existe différents modes de décentralisation et, par conséquent, différents modes de réflexion sur les DAO vis-à-vis d'une agence. La façon communément connue de penser aux DAO est le grand mode d'assemblage:



*Le mode d'assemblage d'un DAO.*

Dans un mode *d'assemblage* d'un DAO, un grand nombre d'agents interagissent dans la prise de décision au sein d'une même agence via son contrat intelligent, en supposant que la réputation, et donc le pouvoir de décision, est équitablement répartie. En dépit d'être le plus simple, ce mode est intrinsèquement en tension avec l'évolutivité et a une limite sur sa capacité de traitement, tout en maintenant la résilience. Nous introduirons aussi des astuces pour étendre la capacité du processus de ce mode DAO, mais en soi, cela ne peut pas être la réponse complète.

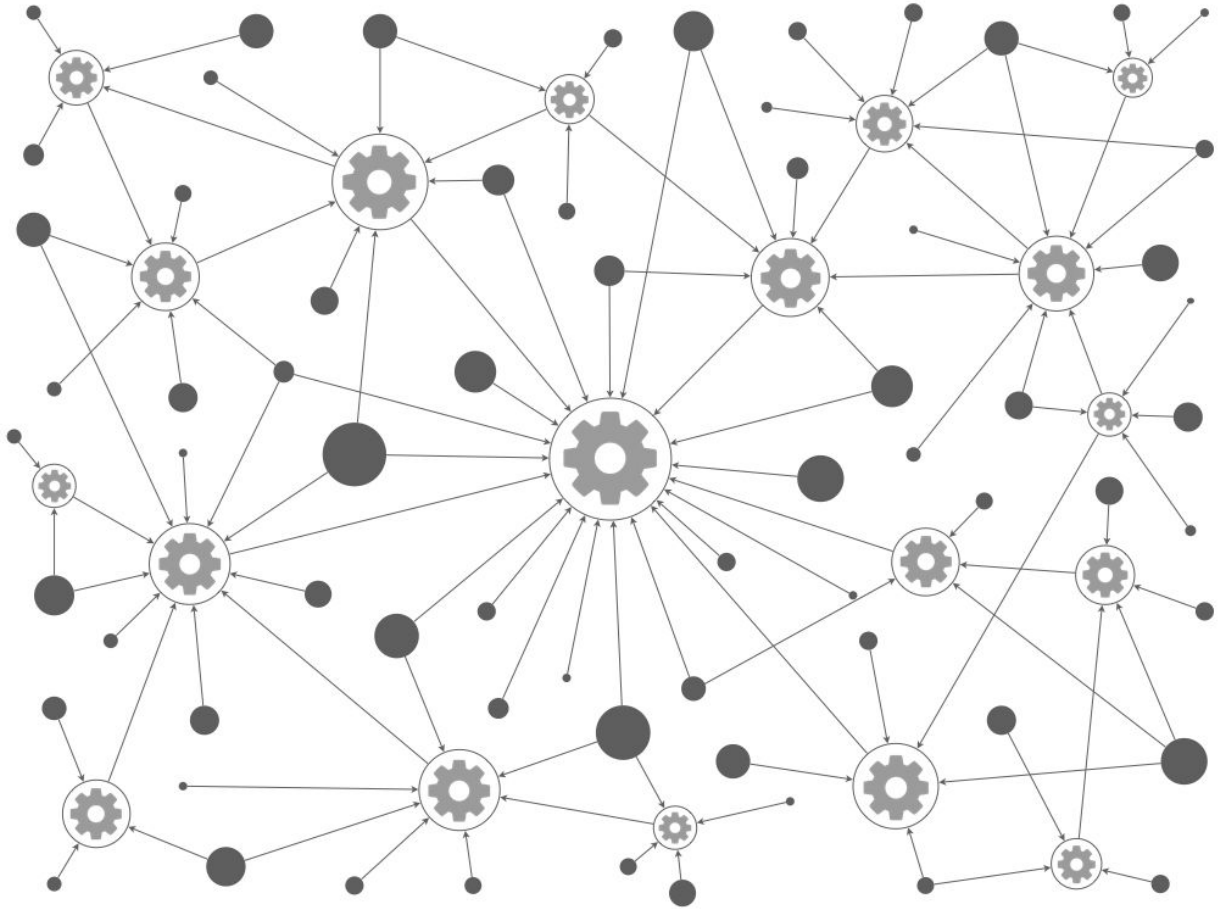
Le deuxième mode de décentralisation est *la gouvernance fédérale fractale*:



*Le mode de gouvernance fédéral fractal d'un DAO.*

Dans le mode de gouvernement fédéral fractal extrême d'un DAO, le DAO est une agence avec quelques agents, chacun étant lui-même une agence avec quelques agents, chacun étant lui-même une agence, et ainsi de suite. En réalité, le DAO se trouverait probablement quelque part entre ces deux modes, et bien au-delà, en tant que réseau maillé imbriqué d'agences, entrelacé avec d'autres DAO par le partage d'Agences:





Le mode réseau maillé complexe d'un DAO.

Le réseau maillé complexe peut provenir d'agences simples avec l'infrastructure fournie par le stack DAO. Le pouvoir peut être bien réparti de manière *méritocratique* à travers le DAO afin d'utiliser efficacement sa sagesse collective et bénéficier à l'organisation dans son ensemble.

### Structures décentralisées connues

Les structures décentralisées sont abondantes dans la nature. Le corps humain est une structure décentralisée faite d'organes, de sous-organes et de sous-sous-organes, jusqu'aux cellules atomiques, qui ont elles-mêmes leur structure interne. La fonctionnalité du corps est assez décentralisée et aucune cellule n'indique aux autres cellules quoi faire. Au contraire, chaque cellule fonctionne de manière autonome en fonction des entrées qu'elle reçoit de son environnement. Le sens d'un organisme - un être humain autonome et sentiel, est un phénomène émergent apparaissant seulement au niveau collectif.



An ant colony is a decentralized structure. It functions without central management or control (the queen doesn't decide about the colony, it lays eggs), and individual ant behaves in reaction to the conditions of its nearest environment. The sensible colony is an emergent phenomena at the collective level, derived from an indirect coordination of ants which need not even communicate directly with one another.

The Internet is a human-based decentralized structure. It is an acentric system which has gracefully scaled to more than 2bn users over fifty years of existence. Its dynamic self-governance helped it evolve and upgrade itself over time just like a living system. The Internet does not support internal value distribution, though, so it lacks an inherent economic incentive model for engagement. Its functionality is thus limited to the distribution of information.

In fact, the blockchain itself is the closest existing thing to a DAO (more accurately, it is DApp). It is a centerless, living organism operated by a wide crowd of engaged members (AKA *miners*). With a new form of internal economic incentive model, it opens the door for growth and adoption levels never seen before. At the time of writing, the Bitcoin blockchain network has grown from zero to almost \$100bn in value without any central management or coordination. The Ethereum blockchain has similarly grown to \$30bn in three years! (Which means the lucky investors of the Ethereum Crowd Sale have seen their investment going up about X1200 times in that time window.) But the functionality of these value-based DApps is limited, and an additional element is needed to enable general-purpose DAOs that could organize around general purposes. The DAO stack is that missing element.

## Open Organizations

Our current economic system is based on an approximately zero-sum, or win-lose game. The force of competition triggers evolution toward peak performance, but which is maximized with respect to local rather than a global win. (I.e. a company optimizes with respect to its own survival, and not with respect to the benefit of a larger whole.<sup>4</sup>) This is the problem of non-cooperative Nash equilibrium. The meaning of Nash equilibrium is that, while another cooperative equilibrium may exist and is better for everyone, no single agent has the incentive to change her strategy alone. It is a problem of large-scale coordination that prevents the transition from competition to cooperation.

This non-cooperative Nash equilibrium prevails almost everywhere, but in the domain of production and development it shows up clearly in the form of closure or obscurity (if not deceit) of knowledge and information, and more generally anti-rival resources. Anti-rival resources are unlimited in usage, and rather

---

<sup>4</sup> The same holds true for any sovereign entity, down to the individual level.



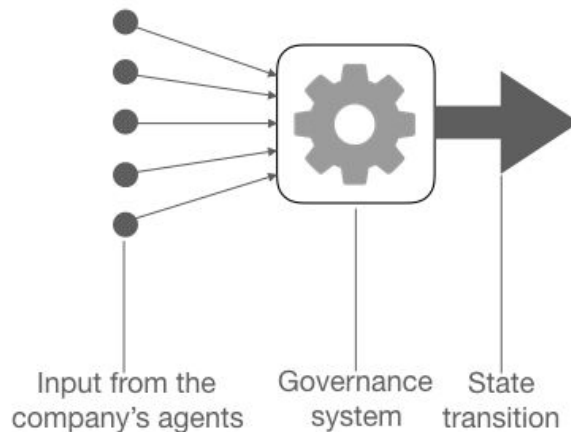
become more valuable the more they are being in use. A common example is code; it is never consumed, and rather the more eyes are on it the better (and safer) the code is. At the same time, companies have no incentive to open-source their code, as otherwise they would asymmetrically give advantage to their competitors. On the other hand, clearly, if ten competing companies produce a similar product, they could have all benefitted from co-producing the shared elements of their products rather than all independently producing them on their own. This conundrum is probably most notable in the pharmaceutical industry.

Intellectual Property (IP) is the traditional means to turn anti-rival resources into scarce elements, thus make them sellable, but it has become less and less sensible, as well as practical, in today's accelerated pace of invention.

Opening up shared resources is inconsistent with the current economy. But on the other hand, it is the very basis for large-scale, open collaboration, or DAOs. To be effective, the DAO needs to incentivize and reward the sharing of reusable components. In result, more existing shareable components will support the growth and effectivity of the DAO. Open organizations is the means to shift from the current non-cooperative Nash equilibrium to a future cooperative state.

### 3. Blockchain Governance

*Agency* is the basic unit of governance over the blockchain. A blockchain governance system is a state-transition function, that collects inputs from blockchain addresses and under certain rules spells out a global state transition.



*Blockchain governance system.*



Instead of unwrapping this statement in formal language, let us explain it via an example.

## Example

The simplest possible agency manages an ETH fund, and internally only has its native FND token. We will call it Fundis. There are only three rules in the Fundis governance system:

1. The only collective decision that Fundis makes is sending ETH from its main wallet to a certain blockchain address (i.e. make an investment). This action is initiated by a proposal that is put up by some agent: say, agent *X* proposes to send 1000 ETH to address *A*. Then agents vote yes or no on that proposal, weighted by the number of FND that the voter has in her address:

Address	FND
0x00eFBC5D0A96e...	6
0x67248B65b6E0f7...	7.5
0x5F42801ac21677...	0.21
0x70f67cD70A50DA...	120
0xEA3C6A6b31dAD...	1.4
⋮	⋮

*FND balances of blockchain addresses.*

Once a majority of FND holders approve a proposal, it is automatically executed.

2. The second rule of Fundis is that upon its establishment, FND tokens are issued and sent to buyers, one FND per ETH, for a period of one week.
3. The third and final rule is that at any time, any FND holder can send his FND to the main Fundis wallet, and get back his share of the ETH existing in that wallet at that moment, pro rata. His FND is destroyed.

There are a few immediate issues with this governance system, such as:



1. It might be very hard to recruit a majority of FND holders to vote on one proposal. This issue is related to governance scalability, which we will discuss in length below.
2. It can be corrupted by a 51% attack. Say, there is \$100m worth of ETH in the Fundis wallet. And, say, that it costs \$60m to buy half of all FND tokens in the market. Then, an agent with enough liquid capital can purchase just above half of the FNDs in the market, and immediately thereafter make a proposal to send all ETH to his own address. That's an easy \$40m hit. This problem is related to resilience, which is the second topic we will cover below.
3. It is not clear (and probably not true) that those who hold more tokens would make the best investment decisions. It is not a deal-breaker, but might be leading to a pretty ineffective investment fund. This is in tension with meritocracy, which is another important criteria.
4. For further illustration, an additional attack is made by making the proposal to distribute all of Fundis ETH into addresses that will vote yes on this proposal, pro rata. Perhaps in the beginning decent agents will refrain from approving this cunning proposal; but once the first approvals will come, there will be a growing pressure for approving it (and not losing all of your money), until an avalanche effect will take place and the cunning proposal will be approved. This to demonstrate that resilience is a tricky issue.
5. Finally, even after the Fundis creators have realized that its governance system is heavily flawed, they cannot do much about it since Fundis governance system is not upgradable, which is another criteria we would like to impose.

This simple governance system highlights almost all of the issues with blockchain governance, which we will discuss below. It also demonstrates the initial statement above: "A blockchain governance system is a state-transition function, that collects inputs from blockchain addresses and under certain rules spells out a global-state transition."

Indeed, there are four types of inputs that agents can inject into the company:

1. Submitting a proposal;
2. Voting yes or no on an existing proposal;
3. Sending ETH to Fundis in order to buy FND in the first week of its establishment;
4. And sending FND to Fundis in order to claim back one's ETH share.

There are three possible actions (resulting with a global-state transitions) Fundis can make, and three rules of its governance system to trigger them correspondingly:

1. Issuing and sending FND to senders of ETH in the first week (1 FND per ETH);



2. Sending ETH from Fundis wallet to senders of FND, pro rata (and burning the FND);
3. Sending ETH to an address following a successful proposal (as described above).

## Reputation Systems

Instead of weighing votes of agents with their native token stake, a company can have different balance sheets to denote the influence of agents inside votes. Generally, reputation scores are non-transferable assets —an agent cannot transfer its reputation to another agent. The simplest case is when a company has a single reputation system —meaning, the same influence score for all types of decisions. More generally, a company can maintain multiple reputation scores that are used for different cases. Reputation is used here interchangeably with influence power, and can form the basis for a meritocratic governance system, where those who are most appreciated —due to their past contributions— have the most influence. One way to allocate reputation to agents would be via proposals. For example, one can propose to allocate 100 FNDr (Fundis reputation score units) to agent A for her recent valuable contribution C. Reputation can also be algorithmically tied with economic remuneration of the company to a contributor of value; as well as with one's votes and evaluations in relative to the collective. This topic is called reputation flow, and will be covered somewhere else.

Reputation systems alone solves most of the problems of the previous token-based governance system, but it still remains unscalable.

## Blockchain Governance structure

The governance of an agency can be divided into two types of actions, the *do's* and the *don't's*:

1. The *do's* are the logical and operational rules under which ***agency functionalities*** can be triggered. For example: if a majority of reputation holders in a DAO approve the issuance of new tokens, token issuance will be automatically triggered by the DAO's smart contract. We call these operational logics: ***schemes***.
2. The *don't's* are the restrictions and limitations that must absolutely be respected by an agency, and that cannot be violated even by an approved scheme. For example, if a DAO approves an upper cap of one million tokens, token-issuance schemes will operate only as long as the total number of tokens issued is less than a million. We call these limitations: ***global constraints***. Constraints can be absolute, or they can be designed to be upgradable under specific conditions.



We collectively denote schemes and global constraints by *elements*. Given the elements of an agency, its entire governance protocol (including the protocol to change the protocol) is unambiguously defined.

## Agency functionalities

An agency deployed on the Ethereum blockchain via the DAO stack can, in principle, do anything that can be done on the blockchain. In particular:

- **Token distribution.** Each agency or DAO can issue and distribute its own native tokens to contributors of value, as valued by the organization. The issuance of native tokens enables the organization the creation of its own separate economy. The utility or benefit of the tokens can be anything that the agency decides about, such as entitling access to the DAO's product (we will call these *utility tokens*), or entitling a share of the agency's revenue (we will call those *share tokens*).
- **Funds allocation.** An organization can earn, or collect via its own-token sale, external tokens such as ETH, STK or other DAO's tokens. It can keep them in reserve, and distribute to third parties in exchange of a particular effort or contribution. This is somewhat analogous to an agency using its funds to compensate contributors, employees or other service providers.
- **Reputation assignment.** Each agency can assign *reputation scores* to its members. Reputation is a representation of one's professional credibility, and thus influence, within the organization. As opposed to traditional blockchain-based tokens, reputation is not transferrable. It is awarded to or earned by specific members, according to their merits and contributions made to the organization. Since reputation is tied with decision-making power in the organization, more reputation should be allocated to those who the organization believes make the best decisions. However, in order not to lock up decision-making power over time, the organization might decide that reputation dissipates over time.
- **Collective data curation.** An organization can manage its own collective databases of objects, and maintain their curation. It can be the curation of articles, website, organizations or anything else. The power of a shared database lies in its network effect; if everyone are looking at the same spot (because it's well-curated), then that spot is valuable (and monetizable too). We will see below a few examples with the collective DAOstack registries, the ArcHives.



- **External activity.** An agency can act within another agency as a single entity. For example, an agency can submit a proposal inside another agency (or DAO), and vote on others' proposals.<sup>5</sup>
- **Governance upgrade.** Each organization can configure and update its own governance system. By approving or removing certain elements, the agency defines how it functions, what it can or cannot do, and what are the mechanisms for changing these governance schemes.<sup>6</sup>

## Schemes

*Schemes* are logical functions made of a series of instructions that take a particular set of inputs, and process them to generate a particular set of outputs. Schemes can be designed to do virtually anything, although most of them will trigger one of the basic functionalities of an agency, such as token and fund allocation, reputation assignment and protocol upgrade. Common schemes are based on *proposals* —the proposition to trigger some of the basic actions of an agency in a particular way. These will then be voted on with *yes* or *no* and be possibly approved, resulting in the automatic execution of the proposal.

### A simple rewarding scheme

For example, a simple rewarding scheme in a DAO can be as follows:

- An agent submits a proposal to the DAO to reward 150 tokens and 200 reputation bits to another agent for contribution made to the DAO.
- Anyone can vote *yes* or *no* on this proposal, to be weighted by their DAO reputation score.
- Once a majority of the DAO reputation holders vote *yes*, the DAO's operating system executes the proposal and allocates 150 tokens and 200 reps to the contributing agent. Note that this agent can also be yet another agency.

Similarly, when looking back at the Fundis example above, it had three rules, which could be written as three distinct schemes. Thus, the entire governance system of Fundis is describable via those three elements. We will use this modularized formalism in the next chapter, when describing the DAO stack and in particular the *Arc* governance framework.

---

<sup>5</sup> More generally, a DAOstack agency can act anywhere within the blockchain domain, thus preserving full interoperability with other systems. E.g. an agency in DAOstack can open a user in a company in Aragon :)

<sup>6</sup> The agency can also upgrade the technology stack itself, either to an upgraded version of DAOstack, or a whole new architecture.





## A more complicated example

For the sake of illustration, a fairly more complicated reward scheme could be described as follows:

- An agent submits a proposition requesting to be rewarded for a particular contribution.
- The scheme triggers an alternative voting system, enabling anyone holding reputation in the DAO to propose a specific number of tokens to award the agent with.
- Votes are weighted by the reputation score of each voter, and the total amount of tokens to be awarded equals the weighted *median* of all voters.<sup>7</sup>
- As soon as 20% or more of reputation holders have expressed a vote, and not more than 2% of reputation was engaged in the vote during the last day, the DAO will allocate the amount of tokens and a proportionate amount of reputation to the contributing agent, following the median at that moment.

More generally, the spectrum of possible scheme's design is nearly endless. Schemes can activate different actions (including the functionalities mentioned above), incorporate a variety of different logics, and rely on a variety of different voting systems which might associate voting power with reputation, tokens, or a combination of both. Another common typology of schemes is the one allowing DAOs to offer their native tokens for sale under some conditions (the so-called token sales).

## Global Constraints

*Global constraints* are specific conditions that can be attached to a particular agency or DAO and that will limit its functionalities. As a general rule, for any operation to go through within a particular agency, it must comply with the list of constraints configured by the organization. Regardless of the list of approved schemes in the agency, none of these schemes will be able to trigger a particular functionality in the organization if doing so would not conform with even a single one of the global constraints.

For instance, global constraints that might be adopted by DAOs and agencies include:

- An upper cap imposed on the total amount of tokens that can be issued by the organization.
- A rate of token inflation (i.e. dynamic cap) over a time period (e.g. 2% per month).
- A burn rate on the usage of the an agency's funds.
- A maximum amount of reputation that can be issued within a particular time period.
- A list of agents that can request the issuance of more tokens or reputation (i.e. whitelisting).
- The registries from which the DAO can approve new schemes.

---

<sup>7</sup> This is the largest number that 50% or more of voting reputation has assigned a larger or equal number of it.



An organization can define certain global constraints to be permanent and others to be modifiable. In the latter case, the organization will also define under what conditions these constraints can be updated (e.g. by a 75% decision of token holders, or only via this or that specific scheme).

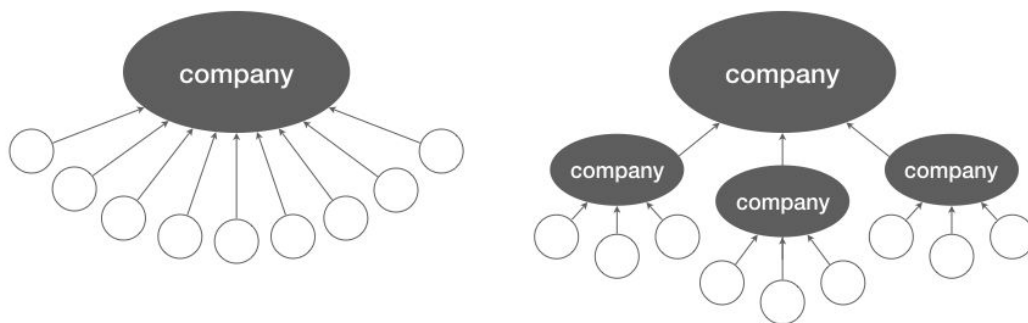
## Scalability

The scalability of a decentralized governance system is in inherent tension with its resilience. Decentralization requires the system to be open for random participants under certain rules (i.e. it uses an *open protocol*). By resilience we mean that we need enough participants to review every decision. But this is clearly in tension with the scarce resource of participants' attention, whether it is computing power —in the case of blockchain, or human attention —in case of DAO governance.

We argue that there are generally three ways to resolve (to some degree) the tension of scalability and resilience in a decentralized governance system, which are: *compositionality*, *attention monetization*, and *relative majority*. The blockchain itself is a form of decentralized governance system, in which these mechanisms are analogous to: *sharding*, *gas*, and *off-chain* computations.

## Compositionality

Consider the following two cases. In the first, nine agents of a company with equal votes. In the second, three agents of a company with equal votes, each of which is by itself a company, composed of three agents with equal votes. A sub-company casts its votes in the mother company as soon as it develops its decision via an internal majority.



The assembly vs. federated governance.



It is easy to see that in the first case, it requires the attention and consensus of five human agents in order to cast a decision in the mother company; whereas it is sufficient to have the consensus of four human agents in the second scenario to form a decision in the mother company.

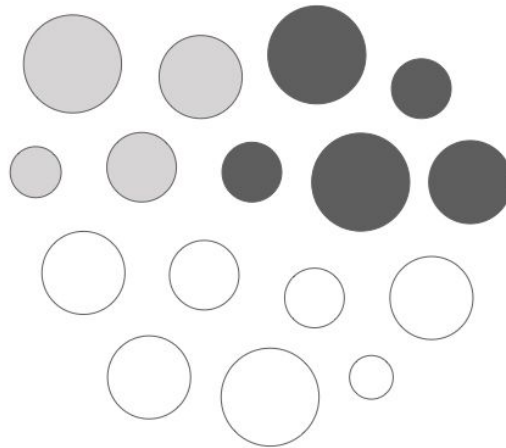
This very simple example illustrates the fact that *compositionality*, or *fractalization*, of governance systems makes them more scalable. One can argue that by this we have compromised resilience (as less human agents can take over the system); but we can also observe that not any configuration of four human agents in the second scenario can form a decision in the mother company. With a deeper scrutiny it can be shown that under different assumptions the resilience in the second case can be weaker or stronger.

### **Monetization of Attention**

Forming a decision requires the attention of voters. The more decisions are there to evaluate, the more attention is needed. However, attention is a scarce resource. In one way or another, to form a resilient decentralized decision-making system, attention should be monetized to reflect its inherent scarcity. In the Bitcoin blockchain it is formed in terms of *transaction fees*, and in the Ethereum blockchain as *gas*. For DAO governance, attention needs to be monetized with the token that reflects the community of agents whose attention is being called for. For internal DAO decisions that would likely be the DAO token, whereas for inter-DAO activities we would use Stacks (STK), the DAOstack ecosystem token. Note that monetization of attention does not buy decisions, but only buys the attention over proposals; we will elaborate on this point further below.

### **Relative Majority**

Naively, a straight-forward way to form scalable decision-making process is by demanding only a relative majority to approve a decision. By relative majority we mean that the majority of approvals is only with respect to those who cast their vote and not all potential voters in the system.



*The size of the balls reflect the voting power of agents. The empty balls are those who have not cast their vote on a specific proposal; the darker balls voted yes, and the lighter balls voted no. Clearly, there is no absolute majority of approvals, but there is a relative majority of approvals out of all voting agents.*

Note that relative majority votes requires a finite time window for the proposal to be considered.

To solve this problem, previous systems of governance introduced a *quorum* —the minimal amount of voting power that needs to be engaged into a decision to make it legitimate for approval. The problem with a quorum is that it is impossible to fix the right amount of it, and will either damage resilience —if set too low, or scalability —if set too high, and possibly both. If at all, quorum needs to be dynamic.

In the proposal below, we provide a resilient governance system, without a quorum, that is based on relative majority and attention monetization.

## DAOstack voting system

In this section we propose the simplest decision-making protocol that is potentially decentralized, resilient and, to some degree, scalable. It is also one of the voting systems implemented in the DAO stack described in the next chapter, as well as the one the the DAOstack organization itself is being launched with in parallel to the release of this paper. We describe the protocol with the following steps:

1. **Proposals.** Decisions are initiated by proposals; proposals are submitted by agents and are then voted on with a *yes* or *no*.



2. **Based on reputation.** Agents' votes are weighted with their reputation. We confine to a single reputation system per company for now. Sub-companies and sub-sub-companies in a DAO will focus on specific branches of production or decision making. In this fractal framework, the notion of a single reputation score per company is not inconceivable.
3. **Finite time.** Once a proposal is *opened* —as we will define below— it has a finite time to be closed (say, 2 weeks). Which means that by the end of this time interval a decision (yes or no) is taken according to the relative majority of voting reputation (i.e. only out of all voters).
4. **Quiet ending.** To avoid finalization attacks, the effective decision (majority of yes or no) cannot be changed in the ending interval (of, say, 1 day). Meaning, if on the last day of opening the majority changed from yes to no (or vice versa), the opening interval is extended by another day. The vote is closed only once there is no change of decision during the last day of voting.
5. **Opening stack.** At each point in time there can be only a finite number of open proposals (say, 10). All other proposals stand in a queue, and are ordered by a ranking system. Every time a proposal in the opening stack is finalized, the proposal with the highest rank in the queue enters the opening stack and is opened.
6. **Boosting.** Anyone can propagate a proposal up in the queue by *boosting* it. Boosting is done by putting tokens on stake.<sup>8</sup> If the proposal is successful the booster gets back her tokens, and otherwise they are destroyed (and in a variation of this protocol are distributed to some or all of the voters).
7. **Ranking system.** The ranking system of proposals in the queue can be pretty general, but a simple sensible choice would be:  $R_+^2 \cdot B$ , where  $R_+$  is the amount of reputation already voting yes on this proposal, and  $B$  is the total amount of tokens boosting that proposal. Note that anyone can boost an existing proposal, not only the proposer. Another option for this formula that we will explore below is:  $R_+^2 \cdot B \cdot S$ , where  $S$  is the amount of tokens (of the same kind as in  $B$ ) that exist and are locked to a minimum period of, say, 6 months in the proposer address.

## 4. The DAO stack

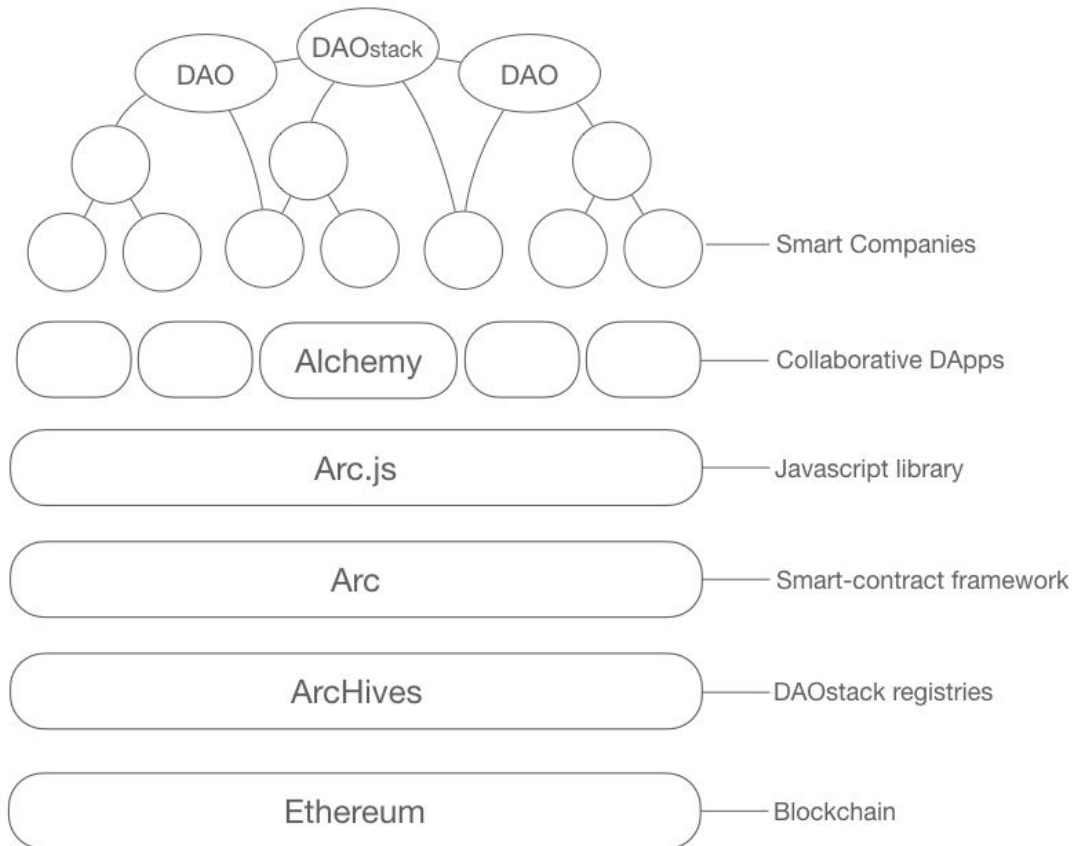
DAOstack provides the foundational tools for the creation, operation and governance of DAOs, internally and externally within a broader ecosystem. In a nutshell, it can be regarded as an analogue of *Wordpress* for DAOs

---

<sup>8</sup> Which tokens will be made clearer in the next two chapters.



—it does for blockchains what *WordPress* has done for the web. This vision is made possible with the following stack of components in place:



A sketch of the DAO stack.

The DAOstack ecosystem is made of a multitude of distinct but interoperable DAOs, interacting with one another in order to maximize the potential benefit of open and distributed collaboration. At the technical level, all DAOs are made of a series of smart contracts, deployed through *Arc*: a Solidity framework of governance allowing to create, configure, deploy and operate DAOs onto the Ethereum blockchain,<sup>9</sup> possibly relying on IPFS as an overlay network for data storage and retrieval.

People can interact with these DAOs either directly, through the execution of blockchain transactions, or indirectly, by relying on a particular front-end to the underlying blockchain ecosystem. *Alchemy* is the collaborative DApp developed internally by DAOstack, enabling anyone to create a new agency or DAO and

---

<sup>9</sup> While Arc is currently based on Ethereum, the framework aims to be blockchain agnostic once other technologies mature and become interoperable with one another.



start collaborating with others in the DAOstack ecosystem. It relies on *Arc.js*, a JavaScript library that operates the Arc Solidity framework via Web3.js. It is designed to make it easy for front-end JavaScript developers to create collaborative applications on top of Arc without the need to interact directly with the Solidity code or the Ethereum blockchain. The *ArcHives* is a public set of registries, curated by the DAOstack community and serving its global ecosystem. It is where the ecosystem and network effect builds up.

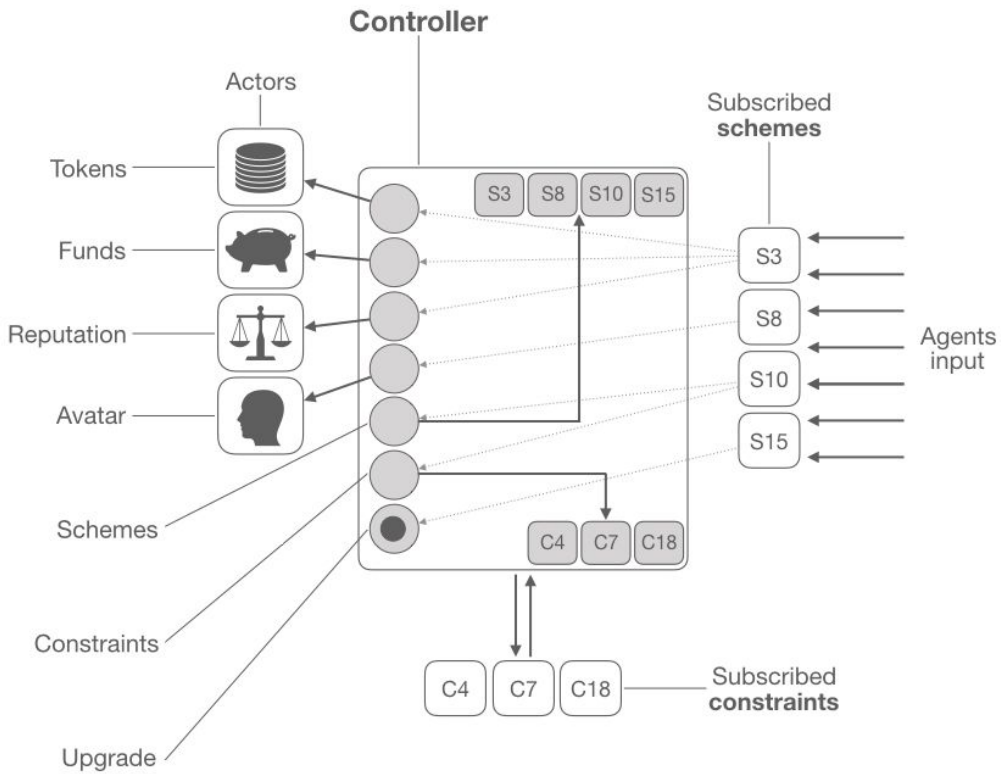
## Arc

Arc is a general governance framework for an interacting internet of blockchain agencies, the basic operating system for DAOs. It is an open-source, modular and general-purpose framework by design, and it comes with an open library of template governance modules, or *elements*, that will evolve by the needs of its users. It also allows an easy upgrade and modification of a governance system to better fit the organization's needs over time.

Arc is not limited to a specific set of governance systems, and it makes it easy for third parties to create their own *elements* per their needs. By combining available *elements* each agency can implement its own governance system that specifies the rules for the issuance, management and assignment of scarce resources, including transferable assets (e.g. tokens) and non-transferable assets (e.g. reputation).

## Architecture

Arc elegantly implements in smart contracts the basic decomposition of governance systems discussed in the previous chapter, with: *actions*, *schemes* and *global constraints* that every agency can be built of. Below is a visual representation of the logic and smart contract architecture of the Arc framework:



A sketch of the Arc contract bundle per agency.

## The Controller

The controller contract is the main engine of the agency. It is “owned” by and gets commands from the *subscribed schemes* alone, which operate its functions. Via its functions it sends commands to the *actors* contracts: the *token* and *reputation* printers, the funds *wallet* and the *avatar*, as well as the upgrading functions of the governance system and the technical architecture itself.

## Subscribed elements

Subscribed schemes and constraints are elements that have been registered by a previous scheme which is allowed to register schemes. An agency usually begins its life with one such scheme, the *Genesis Scheme*, but can add new ones that can register or unregister certain elements under certain conditions.





### Agent's input

The only interface for agents to interact with the agency is via its subscribed schemes. Each scheme comes with its specific “knobs” (functions) which can be operated by external agents (blockchain addresses) via transactions that call and operate these functions.

### Token printer

The token printer issues and allocates the native tokens of the agency to agents. It is “owned” by and gets commands from the controller alone.

### Reputation printer

The reputation printer assigns the agency reputation to agents. It is “owned” by and gets commands from the controller alone.

### Wallet

The *wallet* will be implemented in a next version (at the moment the *avatar* is also the wallet). It holds external funds owned by the agency and distributes them to agents. It is “owned” by and gets commands from the controller alone.

### Avatar

The *avatar* is the “face” of the agency (and its address), which acts externally and is capable of doing —via the help of particular schemes— anything that can be done on the blockchain. In particular, it can participate as an agent in other agencies —e.g. submit and vote on proposals— on behalf of its agency. It is also the identity to which reputation in other agencies is assigned.

### Protocol upgrade

The controller can —when receiving commands from the *subscribed schemes*— register new *schemes* and *constraints*, or unregister old ones. Since the governance protocol of an agency is fully specified via its *elements*, every agency can easily upgrade and modify its governance system (if this ability is rooted in its existing protocol). The conditions to perform a protocol upgrade are defined in subscribed schemes that can operate the elements-(un)registration. The agency can have different conditions to register or unregister different elements. (E.g. it might require a 60% majority of reputation holders to modify the schemes and most of the constraints, but a 75% majority of token holders to increase the token supply.)



## Technical upgrade

The controller has one special function that transfers its ownership over all of the *actor* contracts to a new address. Whatever would be this new address, it will have a complete control over the assets (transferable and non-transferable, such as reputation in other agencies) of the agency. It can be an upgrade to an improved architecture of Arc, or to a whole new architecture, if desired so by the agency.

## Example

We present below an illustration of the interaction between the above elements:

- An agent wants to activate a certain “reward scheme” of a particular DAO.
- The agent will submit a transaction to the smart contract of that scheme (possibly via one of the DApps integrated with Arc), along with the relevant input required for that scheme (e.g. agency, number and type of tokens, recipient address, etc.).
- Other agents may approve this request with their vote.
- Once enough votes approve the request —as defined by that specific reward scheme— the reward scheme will command the controller to execute the proposal (with the specific parameters), and the controller will command the relevant acting contract to execute the reward allocation.
- All functions governed by the controller are subject to the global constraints registered by the DAO. Global constraints act as both an ex-ante and ex-post modifier. Before executing any function, the controller is required to run all global constraints stored in the array to ensure that they all return false at this particular point in time. Given that the state of the DAO might change after a function has been executed, once execution is completed, the controller is required to go —once again— through all global constraints, and will revert everything to the previous state if any of them returns false.

## Universal elements

Elements —schemes and constraints— are universal by design. Meaning, all DAOs can rely on the same universal element contracts rather than deploying their own contract each time. The only contracts that are deployed independently for each agency are the controller and the actors. Such a design contributes to the scalability, functionality and security of the framework.

## Design Principles

Arc was designed with the following design principles in mind:



- **Generality.** Arc is a general framework that supports an indefinite number of governance elements. Arc's library of schemes and global constraints will grow with the addition of new templates and modules, possibly developed by third parties and the open-source DAOstack development community. When combined, it will allow for people to experiment with an increasingly large number of governance protocols, and consequently for the successful ones to float up by evolution and natural selection.
- **Modularity:** The Arc framework is highly modular by design. Every DAO's governance structure is made of small building blocks (the governance modules, or *elements*) that can easily be added, combined, edited or removed. This modularity becomes a point of efficiency in that these building blocks do not need to be redeployed onto the blockchain, rather only referred to, saving both storage and operation cost. Moreover, it makes development of sophisticated governance protocols easier and easier the more building blocks exist. Last but not least, it makes contract security management much more sane, whereas writing entire protocols each time from scratch is inconceivable security wise.
- **Simplicity:** Arc has been designed with a strong emphasis on simplicity —in terms of both technical design and usability. The modularity of the framework makes it possible to focus on individual building blocks, whose complexity can be kept to the minimum. As the building blocks are modular and reused, each will receive a lot of individual scrutiny, affording a higher level of vetting and the corresponding greater level of security. To facilitate ease of use, in Arc, bundles of smart contracts can be deployed with a single transaction.
- **Upgradability:** The governance structure of each DAO can be easily upgraded to use new schemes and constraints or different parameters of existing ones. More granularly, each DAO created through the Arc framework comes with a particular set of rules, which by default include rules to change the rules. In addition, the DAO also comes with the ability to upgrade its technical architecture and evolve to a better and better one over time.
- **Interoperability:** Arc is designed to facilitate interactions and promote interoperability between different agencies and DAOs. In practice, it means that agencies can interface with other agencies in a seamless manner, exchange tokens, form agreements, act as individual agents within other agencies and acquire their tokens and reputation. Arc makes it possible for a mesh network of interconnected agencies to emerge and spontaneously create the DAO ecosystem.
- **Openness:** Open frameworks such as Wordpress, Google add-ons and Android invite independent developers to create their own apps, templates and integrations — thereby instantiating and supporting a vibrant development community and which in turn benefits the framework with a wide set of



applications. Similarly, we expect DAOstack to attract developers from the Ethereum community to develop their own governance modules or frontend DApps —thus creating a thriving plethora of templates, modules, and applications for this emergent ecosystem. The DAOstack code is fully open source, backend and frontend. Another aspect of Arc as an open protocol is that all integrated collaborative DApps are potentially interoperable with one another. Just as users of two web applications can interact with each other since both apps ride on the same open protocol (HTTP), then users of two collaborative DApps riding on Arc will potentially be able to interact with one another (if enabled by the collaborative DApps).

## Arc.JS

Arc is a pretty extended Solidity framework, the integration with which requires a sufficient acquaintance with the Solidity smart-contract programming language and the Ethereum blockchain. To make the integration of third-party apps simple for enthusiastic front-end (JavaScript) developers, we have designed Arc.js as an easy JavaScript gate.

Arc.js is a JavaScript library built on top of Web3.js (Ethereum's JavaScript API), with which any functionality of the Arc framework can be called directly from within the JavaScript environment, without knowledge of the Solidity programming language.

We expect this additional layer to make the Arc framework (and the rest of the DAO stack) extensively more accessible for the open-source development community, and third parties, and thus greatly promote the early adoption of the DAO stack and the growth of its ecosystem.

## The ArcHives

Arc is the basis operating layer for the DAOstack ecosystem, and the ArcHives are its shared, curated records. Shared, curated databases can be very powerful, providing a central locus for data integrity and data quality; but more so, they are the locus for the ecosystem network effect, which is also the source for an open-ecosystem monetization (i.e. business model). We will describe below the basic ArcHives of DAOstack: Compendium, Mosaic and the Hive Registries, and will explain the monetization model behind them. Beyond these, each agency or DAO can have their own independent registries, and in particular DAOstack can also have many other registries in the future.



## Compendium

Arc's open framework makes it possible for anyone to deploy new governance *elements* —schemes and constraints. For the purpose of security, DAOs will likely decide to limit themselves to schemes that have gone through exhaustive and professional audit, and more so, have been heavily battle tested. For that purpose, DAOstack has implemented its own “app store” for elements, the Compendium, which records all governance modules that have been approved by the DAOstack expert community.

Compendium provides an additional utility to the Stacks. On one hand, it would cost Stacks to deploy a governance module onto the registry. On the other hand, independent developers who develop and register their own elements on Compendium can have a universal business model, collecting a particular amount of Stacks (voted on by the DAOstack community) upon every subscription of new agency to that module. It is an app store for governance modules. All of the elements developed by the DAOstack founding team will remain free of charge for the community.

Once there is a network effect of users using the Compendium for the elements they subscribe to, the value proposition of this registry will appreciate and accordingly its monetization capability. It will be more profitable for developers to publish their elements on this registry, and thus to pay the registration fee. The registration fee is also useful to filter spam publications, as it will only be profitable if the element will be approved and sufficiently used by the DAOstack ecosystem.

## Mosaic

DAOstack also implements an organization registry that records a list and metadata of all agencies and DAOs deployed through the Arc framework. This database is critical for the interoperability of all collaborative DApps and organizations in the DAOstack ecosystem. Mosaic will be curated by the DAOstack community, which also implements a search engine to facilitate discovery. Just like in the case of the Compendium, it will cost Stacks to register or promote an organization on the Mosaic registry.

## The Hive

The Hive is where innovators, professionals and stakeholders can meet each other within multiple organizations. It is an open billboard that everyone can use to post requests or offers, which are registered and can be promoted by spending Stacks. The Hive registry is curated by the DAOstack community, which, again, also implements a search engine to facilitate discovery. While each collaborative DApp may have its own (non-blockchain) database, the Hive is where all collaborative DApps can maintain data interoperability and enhance the network effect.



## Alchemy

The DAO stack is designed to make it easy for independent developers and third parties to develop their own collaborative application on top of Arc and to be integrated with the DAO stack and its community. At the same time, DAOstack itself develops its native collaborative DApp which makes it easy to interface with Arc: open agencies and DAOs, configure their governance protocol, make proposals and vote on existing ones, collaborate with others on projects and much more. It is intended as a first interface to this new world of work and organization.

## 5. The Stack Economy

### Circular Token Economy

Agencies, DAOs and DApps (Decentralized Applications) are all based on circular token economies. The general idea is that tokens are being distributed to contributors of value —for value that is contributed to the network. On the other hand, the same tokens gain benefit from the value created by the network. The more value is being created the higher the value of the tokens, the more the organization can incentivize the contribution of more value, and so on and so forth.

### The DApp Model

In the *DApp model*, the utility —and thus value— of the tokens is directly related to the usage of a network application. Tokens are being distributed to contributors of value to the network: builders, early adopters and maintainers of the DApp. On the other side, the usage of the DApp requires the spend of those same tokens. For example, Ether was distributed to founders and builders of the Ethereum blockchain, and is now being distributed to miners that maintain the network. On the other hand, Ether needs to be spent by users in order to process autonomous computations on the Ethereum blockchain or use its storage.

Success of the DApp drives up the demand for its usage, thus the demand for its token and thus the token's value. The more people run computations on the Ethereum blockchain the more Ether they would need to spend, and the higher would be its value in the open market (given its limited supply).



## DAO Economy

The added feature of a DAO with respect to a DApp is its ability to process and execute general-purpose collective decision making. There are various domains of decisions a DAO can engage with, and accordingly a various categories of DAOs (and the possible hybrid of those).

### Decentralized Collaboration

When the governance system of the DAO is used for decisions about the incentivization and reward of contributions to the DAO itself, we would call it a *decentralized collaboration*.

As already seen from the blockchain DApps, the token-distribution incentive model works very well in driving engagement and adoption. In the Bitcoin network, miners adoption rate looks like this:



Total hashing power in the Bitcoin network has grown exponentially and multiplied about 650 millions times over the past 7 years, due to the built-in economic incentive in mining.

However, in the blockchain case, miners are only the maintainers of the network, and no direct reward is awarded for the developers of the network or its early adopters. In newer projects tokens are commonly distributed to founders of the project, and by the founders as bounties to other contributors of value. But this incentivization model cannot be systematized and scaled up without the governance system of the DAO.

With DAO stack, and, for example, the relative-majority decision-making protocol described [above](#), tokens can be easily distributed to contributors of value, including developers, promoters and early adopters of a network. There is no limit (unless chosen so by the DAO itself) on the purpose and rate of reward to contributions, and it



allows for crowd engagement and coordination at unprecedented levels.

The token of a decentralized collaboration is tied to the result of that collaboration. For example, if the result is a DApp, then the tokens distributed in the decentralized collaboration are the DApp tokens.

### **Decentralized Cooperative**

In the DApp model the purpose of the network is to create and support an application, and the value of the token is tied to the usage of that application. A *share model* fits the situation where the purpose of the agency, or network, is simply to gain external revenue. It can be an agency that is developing inside a bigger agency, or a DAO, and receives its tokens in return; or it can be an investment DAO that invests in other projects and is intended to make a return on investment. In either case, the value of the token is tied with the external revenue of the organization, and this relationship can be implemented in various ways. One way is to implement a redemption scheme whereby the agency (or DAO) tokens are made redeemable against the funds existing in the agency wallet, pro rata, and thus the value of the token is derived from the revenue made by the agency. Other implementations can involve dividend or other mechanisms. In this model the DAO is a *decentralized cooperative*, and its token resembles its share.

### **Decentralized Curation Network**

DAOs can also organize around the curation of content. The value of the network is then manifested in having a shared locus of information, thus shared locus of attention, on the DAO registries. Once there is a network effect of attention looking at the same point, that point can be monetized by the network itself. The network curate the objects posted on the its registries, and promoters of content would pay with the network's token to post and promote content. Posting low-quality content would result in negative curation, and thus reduction of the poster's reputation. Promotion of content is tied both with boosting fees and the endorsing reputation; thus, the better the content of a promoter is, as curated by the network, the cheaper it would be for him to post again in the same network, and vice versa.

The mechanics of this process, as well as its monetization, is a variation on the decision-making protocol described [above](#). Let us explain it via an example.

Say that a very specific network is about the curation of vegan restaurants in Paris. Anyone can post a restaurant to this network's registry, perhaps by paying some of its tokens (or feely if decided so by the network). Generally restaurants goes to a queue and are rated by the network's reputation holders, out of the entire reputation in the network. At each point in time a few restaurant posts can be active for a *relative curation* —meaning, the curation of the network's reputation holders calculated out of only the participating curators. The relative





curation process is open for a finite time period, say, of at least two weeks. Every time a relative curation is closed, the restaurant from the queue with the highest score enters the active relative-curation stack, and thus gains both more highlighted attention from the community, and a rank based on relative rather than absolute reputation. Score of restaurants on the queue is set by the amount of endorsing reputation thus far and the boosting token fee, similar to the mechanism described above. The score formula can be anything; but if, for example, the evaluation of restaurants is given with five-star ranking,<sup>10</sup> then the score of a restaurant can be something like:  $(A - 3)^2 \cdot B$ , where  $A$  is the reputation-weighted average of stars (i.e. each evaluation is weighted with the reputation of its evaluator, and then one takes the weighted average over all evaluations), and  $B$  is the amount of boosting tokens.

## The Stack Model

DAOstack is a DAO that involves the three above categories. It incentivizes contributors to participate and develop the DAO stack and its ecosystem. In that sense it is a *decentralized collaboration*. It intends to use part of its raised funds (collected via its token sale) for investments in projects that would come on top of the DAO stack (as well as in projects which will build and enhance the DAO stack), and accelerate the ecosystem, while also expecting return on investments. In that sense it is a *decentralized cooperative*. Finally, it will also be a *decentralized curation network*, in particular for governance elements and DAOs, but possibly for many other things. In all contexts, the Stack is the token that buys the collective attention of the DAOstack network, just as ETH buys the collective attention of computers on the Ethereum blockchain. For local decision-making processes inside DAOs, or inside a DApp, a local DAO or DApp token might be used to call for the local collective attention. But for any action, promotion or decision-making process beyond DAOs and between DApps, the Stack will be required to call for the ecosystem collective attention. In analogy to the Internet, imagine that every time you use your local intranet network you would pay with your local token; while, every time you make a call through HTTP to reach out the bigger Internet, you would “pay the HTTP token”. Thus, we call Stack the “collective attention token”.

The DAOstack code is fully open source, backend and frontend, and is found [here on Github](#).

---

<sup>10</sup> When 1-star is given for “horrible restaurant”, 2-stars given for “bad restaurant”, 3-stars given for “average restaurant”, 4-stars given for “good restaurant” and 5-stars for “excellent restaurant”.