DAOs, DACs, DAs and More: An Incomplete Terminology Guide

One of the most popular topics in the digital consensus space (a new term for cryptocurrency 2.0 that I'm beta-testing) is the concept of decentralized autonomous entities. There are now a number of groups rapidly getting involved in the space, including Bitshares (also known as Invictus Innovations) developing "decentralized autonomous companies", BitAngels' David Johnston with decentralized applications, our own concept of decentralized autonomous corporations which has since transformed into the much more general and not necessarily financial "decentralized autonomous organizations" (DAOs); all in all, it is safe to say that "DAOism" is well on its way to becoming a quasi-cyber-religion. However, one of the hidden problems lurking beneath the space is a rather blatant one: no one even knows what all of these invididual terms mean. What exactly is a decentralized organization, what is the difference between an organization and an application, and what even makes something autonomous in the first place? Many of us have been frustrated by the lack of coherent terminology here; as Bitshares' Daniel Larimer points out, "everyone thinks a DAC is just a way of IPOing your centralized company." The intent of this article will be to delve into some of these concepts, and see if we can come up with at least the beginnings of a coherent understanding of what all of these things actually are.

Smart contracts

A smart contract is the simplest form of decentralized automation, and is most easily and accurately defined as follows: a smart contract is a mechanism involving digital assets and two or more parties, where some or all of the parties put assets in and assets are automatically redistributed among those parties according to a formula based on certain data that is not known at the time the contract is initiated.

One example of a smart contract would be an employment agreement: A wants to pay $500 to B to build a website. The contract would work as follows: A puts $500 into the contract, and the funds are locked up. When B finishes the website, B can send a message to the contract asking to unlock the funds. If A agrees, the funds are released. If B decides not to finish the website, B can quit by sending a message to relinquish the funds. If B claims that he finished the website, but A does not agree, then after a 7-day waiting period it's up to judge J to provide a verdict in A or B's favor.

The key property of a smart contract is simple: there is only a fixed number of parties. The parties do not all have to be known at initialization-time; a sell order, where A offers to sell 50 units of asset A to anyone who can provide 10 units of asset B, is also a smart contract. Smart contracts can run on forever; hedging contracts and escrow contracts are good examples there. However, smart contracts that run on forever should still have a fixed number of parties (eg. an entire decentralized exchange is not a smart contract), and contracts that are not intended to exist forever are smart contracts because existing for a finite time necessarily implies the involvement of a finite number of parties.

Note that there is one gray area here: contracts which are finite on one side, but infinite on the other side. For example, if I want to hedge the value of my digital assets, I might want to create a contract where anyone can freely enter and leave. Hence, the other side of the contract, the parties that are speculating on the asset at 2x leverage, has an unbounded number of parties, but my side of the contract does not. Here, I propose the following divide: if the side with a bounded number of parties is the side that intends to receive a specific service (ie. is a consumer), then it is a smart contract; however, if the side with a bounded number of parties is just in it for profit (ie. is a producer), then it is not.

Autonomous Agents

Autonomous agents are on the other side of the automation spectrum; in an autonomous agent, there is no necessary specific human involvement at all; that is to say, while some degree of human effort might be necessary to build the hardware that the agent runs on, there is no need for any humans to exist that are aware of the agent's existence. One example of an autonomous agent that already exists today would be a computer virus; the virus survives by replicating itself from machine to machine without deliberate human action, and exists almost as a biological organism. A more benign entity would be a decentralized self-replicating cloud computing service; such a system would start off running an automated business on one virtual private server, and then once its profits increase it would rent other servers and install its own software on them, adding them to its network.

A full autonomous agent, or a full artificial intelligence, is the dream of science fiction; such an entity would be able to adjust to arbitrary changes in circumstances, and even expand to manufacture the hardware needed for its own sustainability in theory. Between that, and single purpose agents like computer viruses, is a large range of possibilities, on a scale which can alternatively be described as intelligence or versatility. For example, the self-replicating cloud service, in its simplest form, would only be able to rent servers from a specific set of providers (eg. Amazon, Microtronix and Namecheap). A more complex version, however, should be able to figure out how to rent a server from any provider given only a link to its website, and then use any search engine to locate new websites (and, of course, new search engines in case Google fails). The next level from there would involve upgrading its own software, perhaps using evolutionary algorithms, or being able to adapt to new paradigms of server rental (eg. make offers for ordinary users to install its software and earn funds with their desktops), and then the penultimate step consists of being able to discover and enter new industries (the ultimate step, of course, is generalizing completely into a full AI).

Autonomous agents are some of the hardest things to create, because in order to be successful they need to be able to navigate in an environment that is not just complicated and rapidly changing, but also hostile. If a web hosting provider wants to be unscrupulous, they might specifically locate all instances of the service, and then replace them with nodes that cheat in some fashion; an autonomous agent must be able to detect such cheating and remove or at least neutralize cheating nodes from the system.

Decentralized Applications

A decentralized application is similar to a smart contract, but different in two key ways. First of all, a decentralized application has an unbounded number of participants on all sides of the market. Second, a decentralized application need not be necessarily financial. Because of this second requirement, decentralized applications are actually some of the easiest things to write (or at least, were the easiest before generalized digital consensus platforms came along). For example, BitTorrent qualifies as a decentralized application, as do Popcorn Time, BitMessage, Tor and Maidsafe (note that Maidsafe is also itself a platform for other decentralized applications).

Generally, decentralized applications fall into two classes, likely with a substantial gray area between the two. The first class is a fully anonymous decentralized application. Here, it does not matter who the nodes are; every participant is essentially anonymous and the system is made up of a series of instant atomic interactions. BitTorrent and BitMessage are examples of this. The second class is a reputation-based decentralized application, where the system (or at least nodes in the system) keep track of nodes, and nodes maintain status inside of the application with a mechanism that is purely maintained for the purpose of ensuring trust. Status should not be transferable or have de-facto monetary value. Maidsafe is an example of this. Of course, purity is impossible – even a BitTorrent-like system needs to have peers maintain reputation-like statistics of other peers for anti-DDoS purposes; however, the role that these statistics play is purely in the background and very limited in scope.

An interesting gray area between decentralized applications and “something else” is applications like Bitcoin and Namecoin; these differ from traditional applications because they create ecosystems and there is a concept of virtual property that has value inside the context of this ecosystem, in Bitcoin's case bitcoins and in Namecoin's case namecoins and domain names. As we'll see below, my classification of decentralized autonomous organizations touches on such concepts, and it is not quite clear exactly where they sit.

Decentralized Organizations

In general, a human organization can be defined as combination of two things: a set of property, and a protocol for a set of individuals, which may or may not be divided into certain classes with different conditions for entering or leaving the set, to interact with each other including rules for under what circumstances the individuals may use certain parts of the property. For example, consider a simple corporation running a chain of stores. The corporation has three classes of members: investors, employees and customers. The membership rule for investors is that of a fixed-size (or optionally quorum-adjustable size) slice of virtual property; you buy some virtual property to get in, and you become an investor until you sell your shares. Employees need to be hired by either investors or other employees specifically authorized by investors (or other employees authorized by other employees authorized by investors, and so on recursively) to participate, and can also be fired in the same way, and customers are an open-membership system where anyone can freely interact with the store in the obvious officially sanctioned way for any time. Suppliers, in this model, are equivalent to employees. A nonprofit charity has a somewhat different structure, involving donors and members (charity recipients may or may not be considered members; the alternative view sees the positive increments in the recipients' welfare as being the charity's "product").

The idea of a decentralized organization takes the same concept of an organization, and decentralizes it. Instead of a hierarchical structure managed by a set of humans interacting in person and controlling property via the legal system, a decentralized organization involves a set of humans interacting with each other according to a protocol specified in code, and enforced on the blockchain. A DO may or may not make use of the legal system for some protection of its physical property, but even there such usage is secondary. For example, one can take the shareholder-owned corporation above, and transplant it entirely on the blockchain; a long-running blockchain-based contract maintains a record of each individual's holdings of their shares, and on-blockchain voting would allow the shareholders to select the positions of the board of directors and the employees. Smart property systems can also be integrated into the blockchain directly, potentially allowing DOs to control vehicles, safety deposit boxes and buildings.

Decentralized Autonomous Organizations

Here, we get into what is perhaps the holy grail, the thing that has the murkiest definition of all: decentralized autonomous organizations, and their corporate subclass, decentralized autonomous corporations (or, more recently, "companies"). The ideal of a decentralized autonomous organization is easy to describe: it is an entity that lives on the internet and exists autonomously, but also heavily relies on hiring individuals to perform certain tasks that the automaton itself cannot do.

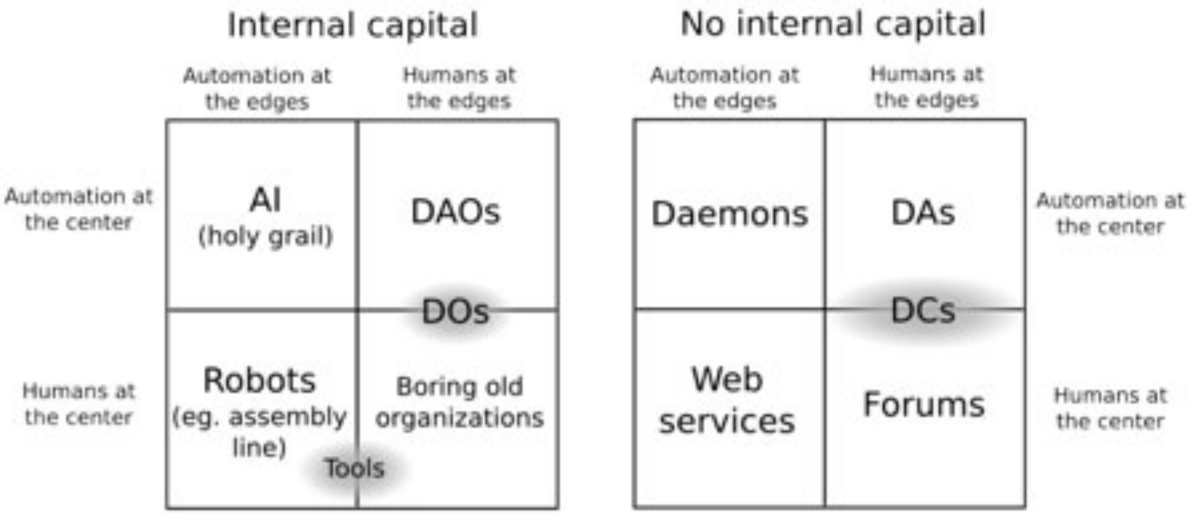
Given the above, the important part of the definition is actually to focus on what a DAO is not, and what is not a DAO and is instead either a DO, a DA or an automated agent/AI. First of all, let's consider DAs. The main difference between a DA and a DAO is that a DAO has *internal capital*; that is, a DAO contains some kind of internal property that is valuable in some way, and it has the ability to use that property as a mechanism for rewarding certain activities. BitTorrent has no internal property, and Bitcloud/Maidsafe-like systems have reputation but that reputation is not a saleable asset. Bitcoin and Namecoin, on the other hand, do. However, plain old DOs also have internal capital, as do autonomous agents.

Second, we can look at DOs. The obvious difference between a DO and a DAO, and the one inherent in the language, is the word "autonomous"; that is, in a DO the humans are the ones making the decisions, and a DAO is something that, in some fashion, makes decisions for itself. This is a surprisingly tricky distinction to define because, as dictatorships are always keen to point out, there is really no difference between a certain set of actors making decisions directly and that set of actors controlling all of the information through which decisions are made. In Bitcoin, a 51% attack between a small number of mining pools can make the blockchain reverse transactions, and in a hypothetical decentralized autonomous corporation the providers of the data inputs can all collude to make the DAC think that sending all of its money to1FxkfJQLJTXpW6QmxGT6oF43ZH959ns8Cq constitutes paying for a million nodes' worth of computing power for ten years. However, there is obviously a meaningful distinction between the two, and so we do need to define it.

My own effort at defining the difference is as follows. DOs and DAOs are both vulnerable to collusion attacks, where (in the best case) a majority or (in worse cases) a significant percentage of a certain type of members collude to specifically direct the D*O's activity. However, the difference is this: in a DAO collusion attacks are treated as a bug, whereas in a DO they are a feature. In a democracy, for example, the whole point is that a plurality of members choose what they like best and that solution gets executed; in Bitcoin's on the other hand, the "default" behavior that happens when everyone acts according to individual interest without any desire for a specific outcome is the intent, and a 51% attack to favor a specific blockchain is an aberration. This appeal to social consensus is similar to the definition of a government: if a local gang starts charging a property tax to all shopowners, it may even get away with it in certain parts of the world, but no significant portion of the population will treat it as legitimate, whereas if a government starts doing the same the public response will be tilted in the other direction.

Bitcoin is an interesting case here. In general, it seems to be much closer to a DAO than a DO. However, there was one incident in 2013 where the reality proved to be rather different. What happened was that an exceptional block was (at least we hope) accidentally produced, which was treated as valid according to the BitcoinQt 0.8 clients, but invalid according to the rules of BitcoinQt 0.7. The blockchain forked, with some nodes following the blockchain after this exceptional block (we'll call this chain B1), and the other nodes that saw that block as invalid working on a separate blockchain (which we'll call B2). Most mining pools had upgraded to BitcoinQt 0.8, so they followed B1, but most users were still on 0.7 and so followed B2. The mining pool operators came together on IRC chat, and agreed to switch their pools to mining on B2, since that outcome would be simpler for users because it would not require them to upgrade, and after six hours the B2 chain overtook B1 as a result of this deliberate action, and B1 fell away. Thus, in this case, there was a deliberate 51% attack which was seen by the community as legitimate, making Bitcoin a DO rather than a DAO. In most cases, however, this does not happen, so the best way to classify Bitcoin would be as a DAO with an imperfection in its implementation of autonomy.

However, others are not content to classify Bitcoin as a DAO, because it is not really smart enough. Bitcoin does not think, it does not go out and "hire" people with the exception of the mining protocol, and it follows simple rules the upgrading process for which is more DO-like than DAO-like. People with this view would see a DAO as something that has a large degree of autonomous intelligence of its own. However, the issue with this view is that there must be a distinction made between a DAO and an AA/AI. The distinction here is arguably this: an AI is completely autonomous, whereas a DAO still requires heavy involvement from humans specifically interacting according to a protocol defined by the DAO in order to operate. We can classify DAOs, DOs (and plain old Os), AIs and a fourth category, plain old robots, according to a good old quadrant chart, with another quadrant chart to classify entities that do not have internal capital thus altogether making a cube:



DAOs == automation at the center, humans at the edges. Thus, on the whole, it makes most sense to see Bitcoin and Namecoin as DAOs, albeit ones that barely cross the threshold from the DA mark. The other important distinction is internal capital; a DAO without internal capital is a DA and an organization without internal capital is a forum; the G8, for example, would qualify as a forum. DCs in the graph above are "decentralized communities"; an example of that might be something like a decentralized Reddit, where there is a decentralized platform, but there is also a community around that platform, and it is somewhat ambiguous whether the community or the protocol is truly "in charge".

Decentralized Autonomous Corporations

Decentralized autonomous corporations/companies are a smaller topic, because they are basically a subclass of DAOs, but they are worth mentioning. Since the main exponent of DAC as terminology is Daniel Larimer, we will borrow as a definition the point that he himself consistently promotes: a DAC pays dividends. That is, there is a concept of shares in a DAC which are purchaseable and tradeable in some fashion, and those shares potentially entitle their holders to continual receipts based on the DAC's success. A DAO is non-profit; though you can make money in a DAO, the way to do that is by participating in its ecosystem and not by providing investment into the DAO itself. Obviously, this distinction is a murky one; all DAOs contain internal capital that can be owned, and the value of that internal capital can easily go up as the DAO becomes more powerful/popular, so a large portion of DAOs are inevitably going to be DAC-like to some extent.

Thus, the distinction is more of a fluid one and hinges on emphasis: to what extent are dividends the main point, and to what extent is it about earning tokens by participation? Also, to what extent does the concept of a "share" exist as opposed to simple virtual property? For example, a membership on a nonprofit board is not really a share, because membership frequently gets granted and confiscated at will, something which would be unacceptable for something classified as investable property, and a bitcoin is not a share because a bitcoin does not entitle you to any claim on profits or decision-making ability inside the system, whereas a share in a corporation definitely is a share. In the end, perhaps the distinction might ultimately be the surprisingly obscure point of whether or not the profit mechanism and the consensus mechanism are the same thing.

The above definitions are still not close to complete; there will likely be gray areas and holes in them, and exactly what kind of automation a DO must have before it becomes a DAO is a very hard question to answer. Additionally, there is also the question of how all of these things should be built. An AI, for example, should likely exist as a network of private servers, each one running often proprietary local code, whereas a DO should be fully open source and blockchain-based. Between those two extremes, there is a large number of different paradigms to pursue. How much of the intelligence should be in the core code? Should genetic algorithms be used for updating code, or should it be futarchy or some voting or vetting mechanism based on individuals? Should membership be corporate-style, with sellable and transferable shares, or nonprofit-style, where members can vote other members in and out? Should blockchains be proof of work, proof of stake, or reputation-based? Should DAOs try to maintain balances in other currencies, or should they only reward behavior by issuing their own internal token? These are all hard problems and we have only just begun scratching the surface of them.