Decentralized Organization

Module 4

What is a centralized organization?

- A centralized organization is a **hierarchical decision-making** structure where all processes and **decisions are made at the executive or top level**. Managers and employees in the organization's lower chain of command can rarely make or implement decisions without the executive's approval.
- In a centralized organization, all decisions, including those related to everyday operations, are made by the top-level executives or company owners. They create policies that team members can implement to improve their performances.
- Many organizations prefer using a centralized organization structure because the clear chain of command offers enhanced transparency, improves efficiency at work, reduces the cost of management and focuses on meeting the organization's goals.

Benefits of centralized organizations

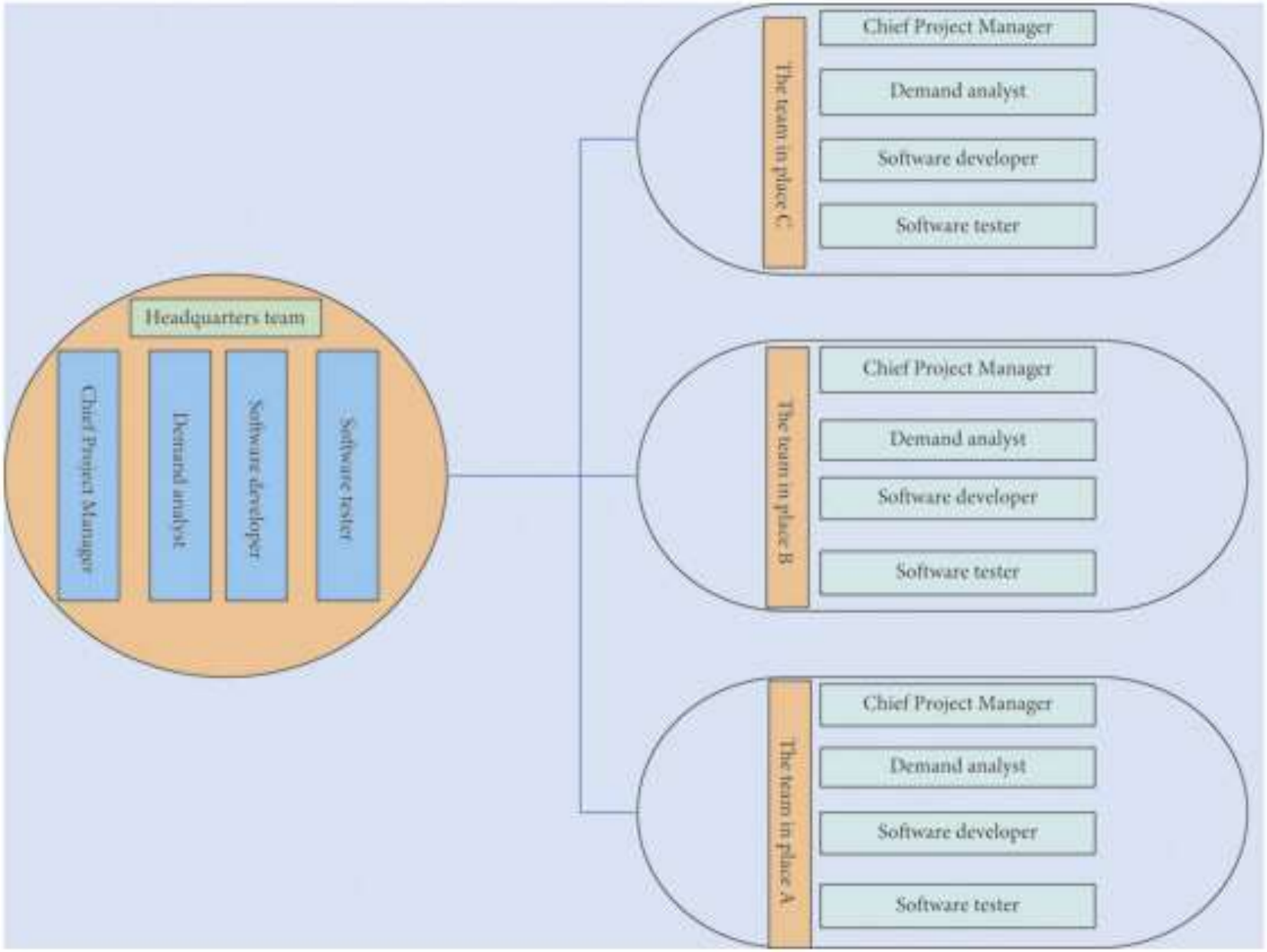
- Transparent chain of command
- Focused vision
- Reduced costs
- Quick execution of rational decisions
- Enhanced quality of work
- Expert guidance

Disadvantages of a centralized organization

- Bureaucratic leadership
- Remote control
- Delays at work
- Lower employee loyalty
- Lack of opportunity
- Slow implementation of decisions

Distributed Organization

- In its common usage, distributed organizing is used to describe an approach that also allows for **a degree of autonomy and local leadership around a common cause but is designed as a more centrally-directed system**, where the catalyzing **organization defines the framing and overall strategy** for the network and **local groups or individuals are free to take on and manage their own actions** within this framework.
- In a distributed organizing system, all local groups are in touch with the central organizing body and often connected with each other through planned convenings and coordination events.

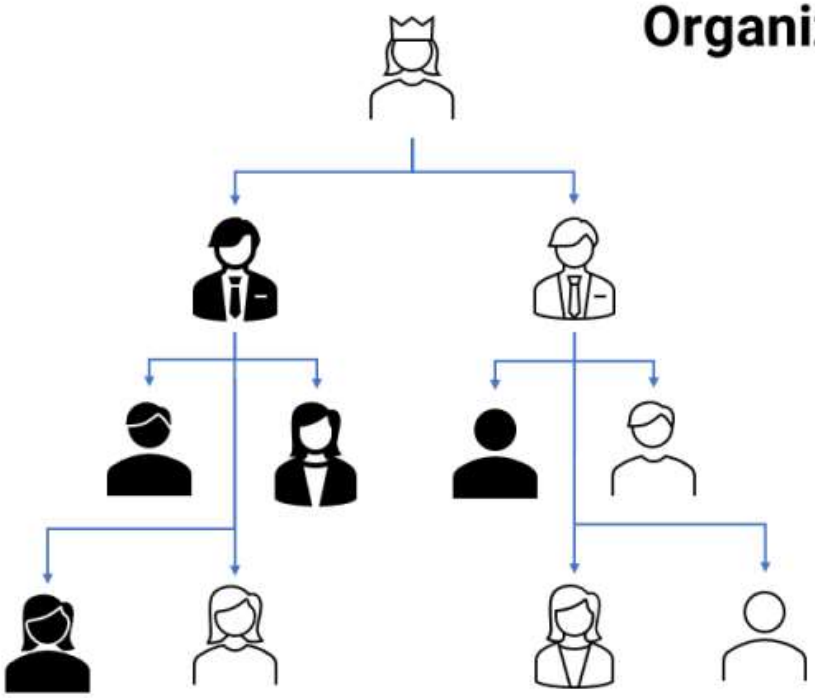

Headquarters team
Chief Project Manager
Demand analyst
Software developer
Software tester
The team in place C
Chief Project Manager
Demand analyst
Software developer
Software tester
The team in place B
Chief Project Manager
Demand analyst
Software developer
Software tester
The team in place A

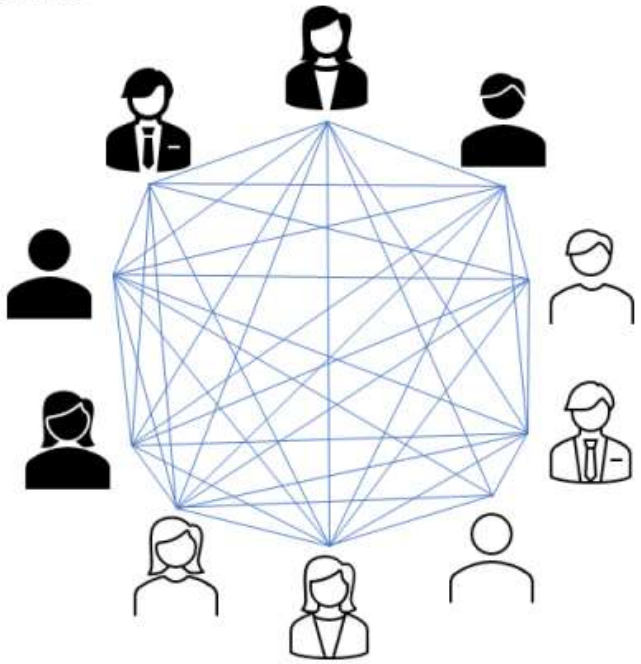
- A famous example to understand why distributed organizations/networks win in the long run is the rivalry in the 2000s between **Wikipedia** and its centralized competitors like **Microsoft's Encarta**.
- Theoretically speaking, If you used the two products in the 2000s, Encarta was a far better product, with better topic coverage and higher accuracy. Whereas Wikipedia was still a work in progress product.
- But something interesting started to happen, Wikipedia improved at a much faster rate as it had a very active community of volunteer contributors who aggressively started contributing to the platform, they were primarily attracted to Wikipedia for its decentralized, community-governed nature (DAOs sounds familiar?) and in 2005, within a few months of the community's active participation, Wikipedia became the most popular reference site on the internet and its major centralized rival Encarta was shut down in 2009.
- Another great example is the success story of **Github**, launched in 2007 and later acquired by Microsoft for $6 Billion. **Github** made coding social and also open-source. It remains to be the #1 destination for developers across the globe for a code repository.

Decentralized organization

- Decentralized organization involves **dispersing decision-making authority** among multiple employees or departmental teams. **Individuals at a lower level approve decisions** and then **report them to personnel in upper management**. Emerging industries often employ a decentralized organizational structure because it's **easier to move employees to different roles and change their responsibilities.**
- This structure is best for companies that want to respond quickly when confronting regionally differentiated customer needs or changing products based on individual markets.
- A decentralized organizational structure is also great for **empowering employees** to present new ideas and increasing career growth opportunities because it implements a more relaxed chain of command model. A decentralized structure is usually present in an alternative system that you may observe in newer companies and enterprise-level businesses.

Centralized Vs Decentralized Organizations





CRYPTO
SAFETY FIRST

	Centralized	**Decentralized**
Decision-making	One body of management decides procedures and develops strategies.	Management at every level makes decisions for their department and teams.
Communication	Information comes from the top down. Employees follow executive orders.	Departments communicate with each other more freely.
Responsibility	Overall responsibility belongs to a small group of executives. They execute the company's vision with little input	Managers and employees share responsibility. This allows for feedback and opinions from employees.
Size of organization	May be best for small organizations with fewer employees.	Often works well for large organizations with many levels.

DECENTRALIZED AUTONOMOUS ORGANIZATION (DAO)

DAO – using storytelling

- In the village of **Blockland**, the people had **one major problem**—their chief, **Lord Centralis**, controlled everything.
  - He **collected taxes unfairly** 💰.
  - He **decided village laws without asking anyone** 📜.
  - He **kept the village funds in a locked chest** that only he could open 🔐.
  - Worse, one night… he **disappeared with all the gold!** 😱
- The villagers were left helpless. They needed **a new system**—one where **no single person could ever control everything again**.

- A wise elder named **Satoshi** gathered the villagers and proposed an idea.
  “What if we don’t have a chief? What if we create an indestructible stone tablet (smart contract) that records all rules fairly?”
- The tablet would:
  Store all village funds safely (treasury)
  Let villagers propose and vote on decisions (governance)
  Automatically execute decisions without corruption (smart contract)
- This system was called the **DAO – Decentralized Autonomous Organization!**
- The villagers agreed and **chiseled the rules into the DAO Tablet**. **No one could change them once written**.

- The first proposal was made by **Builder Bob**: "Should we build a well in the village square?"
- **Villagers used voting tokens** to vote YES or NO.
- If the YES votes were greater, the **stone tablet automatically unlocked funds** for the well.
- **No human approval needed**—the rules were **unstoppable!**
- The well was built!
- **For the first time, the village thrived without a leader.**

- One day, **a group of thieves called ‘The Hackers’** found a **flaw in the tablet**. They tricked the system into **sending them all the village funds**! The villagers panicked! Could the DAO be broken?
- But then, they **worked together** to create an improved DAO. **A new rule was added:**
  “If a suspicious transaction happens, villagers can vote to stop it.”
- From then on, the DAO became **stronger and more secure**.

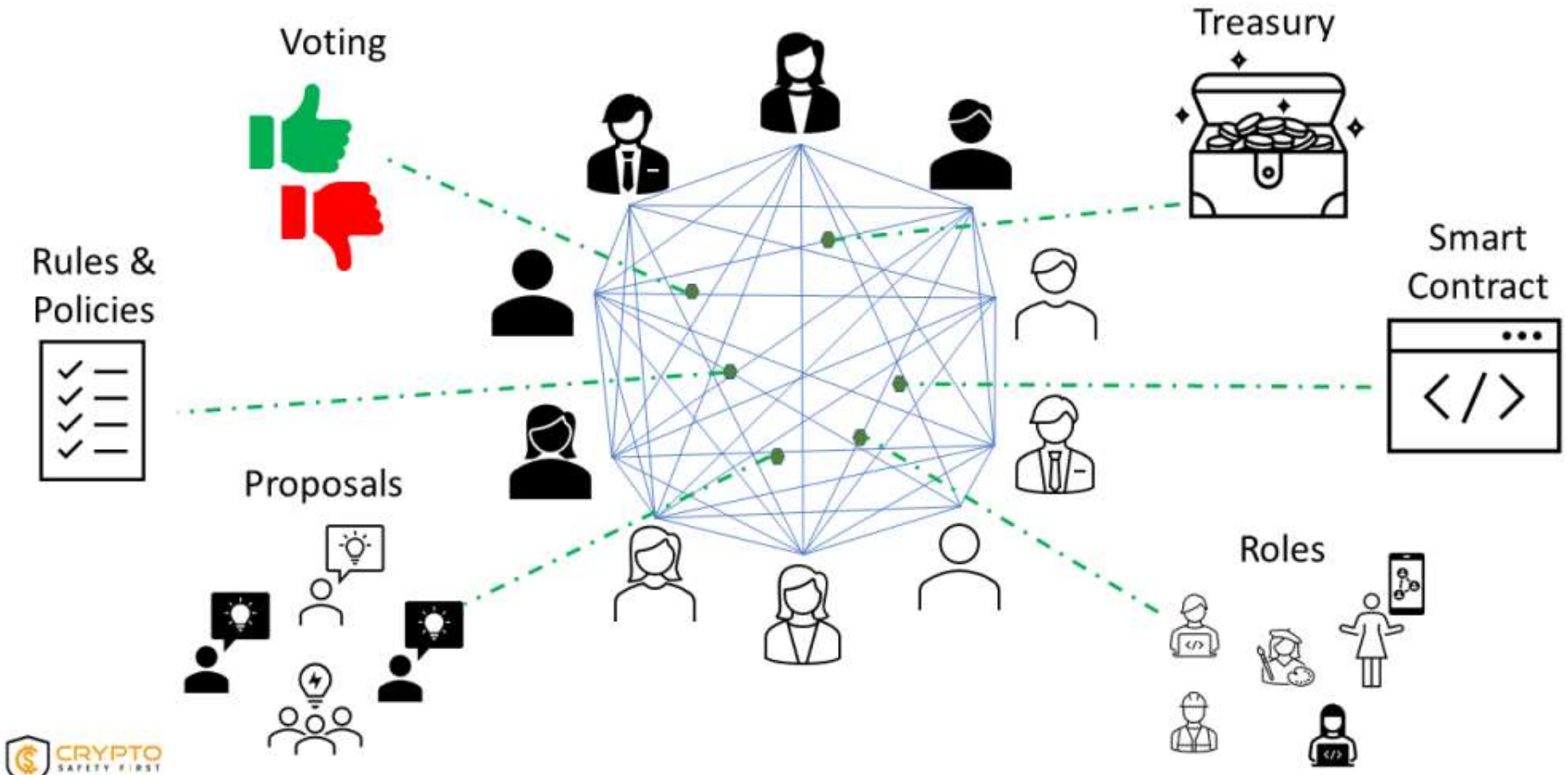
- With their success, **other villages copied the DAO system**. Soon, kingdoms and even empires started using DAOs to govern fairly.
- **“What if one day, entire nations were run as DAOs?”**
- And so, the village of **Blockland** became the first place to prove that:

  Communities can govern themselves
  Smart contracts make decisions fair & automatic
  Trust in code, not in rulers!

- Imagine a **smart vending machine** that can run its business independently. This machine:
  - Accepts payments (crypto)
  - Dispenses snacks automatically
  - Uses sensors to restock itself when supplies are low
  - Pays suppliers directly without a middleman
  - Even allows users to vote on which snacks to stock next
- This vending machine operates without a human manager—it follows rules written in its smart contract.
- A **Decentralized Autonomous Organization (DAO)** works similarly but on a larger scale. It is a self-governing entity run by smart contracts, where decisions are made collectively by token holders.

- **“What if companies had no CEOs?”** – DAOs replace traditional management with code and community governance.
- **“Can we create a self-running startup?”** – A DAO can fund itself, hire contributors, and operate without central control.
- **“Could governments run as DAOs?”** – Imagine a transparent, corruption-free governance system where policies are voted and enforced automatically.
- **“Would you trust a code more than a human boss?”** – DAOs eliminate human bias but are only as good as the code behind them.

Decentralized Autonomous Organization (DAO)



Decentralized Autonomous Organization (DAO)

- A decentralized autonomous organization is an entity structure in which **tokenholders participate in the management and decision-making** of an entity.
- There is **no central authority** of a DAO; instead, power is distributed across tokenholders who collectively cast votes.
- All votes and activity through the DAO are posted on a blockchain, making all actions of users publicly viewable.

- **A (DAO) is an organization run with smart contracts on a public blockchain.**
- Think of a DAO as an organization that uses smart contracts to facilitate and enforce decisions made by its members. **At its core, a DAO is a piece of software on the blockchain that manages permissions to other contracts and actors.**
- A DAO is **self-governing** in that it is managed directly by its members, as individuals or holders of assets, on the actions the organization takes.
- These actions occur **onchain**, meaning they are executed and recorded on the blockchain. Some actions include **managing funds, holding votes, or changing the DAO's governance parameters** itself.

DAO – the word

- **Decentralized = Built on top of trustless, permissionless infrastructure such as the Ethereum blockchain.** The infrastructure of Ethereum itself is decentralized, because it's maintained by a network of computers spread across the world. There's no centralized actor or entity that runs it. This means it can't be "turned off" by attacking or applying pressure on a single entity.

- **Autonomous = Self-governing via managing permissions and automatic execution of those actions, with trust placed in the smart contracts that make up the core of the organization rather than in human actors.** The smart contracts execute the decisions of the organization automatically. Permissions are managed autonomously by code, rather than by people.
- **Organization = Actors coordinating to reach shared objectives.** These actors can be humans or possibly intelligent non-human actors (for example, AI) in the future.

How are DAOs different from traditional organizations today?

- The main difference between DAOs and traditional organizations is that instead of enabling coordination by trusting central intermediaries to enforce decisions, **DAOs can remove this need for trust by making and enforcing decisions on the blockchain.**

- Using **blockchain as infrastructure** grants DAOs unique **properties** when compared to traditional organizations, including:
  - **Trustless execution of collective decisions:** Smart contracts automatically execute the results of collective decisions, such as those held through votes. The results of this decision-making process are recorded onchain, are not alterable and are reliably executed by smart contracts.
  - **Crypto-economic incentives create alignment between actors in the organization:** DAO technology allows you to program incentives directly into the governance of your organization through smart contracts.

- **Governance experimentation at the speed of software:** DAOs can experiment more freely than traditional large-scale organizations. Internet-native organizations can be more agile, creative, and dynamic, rather than slow and cumbersome (or potentially violent) like governance in nation states.
- **Blockchain as a source of truth:** The blockchain is a distributed ledger that anyone can write in but no one can change the entries of anyone else. So, this means all transactions the DAO makes can be universally verifiable by anyone. This includes onchain votes, asset transfers, token mints, and more.
- **Transparent and open by default:** DAOs operate open by default because the blockchain is transparent. Everything from treasury holdings to payment flows is visible to everyone globally. However, privacy-preserving technology, like Zero Knowledge Proofs, could be implemented to make certain aspects of DAO operations private.

S No.	DAO	Traditional Organizations
1.	Casting a ballot is needed by individuals for any progressions to be implemented.	Depending on the structure, changes can be requested from the sole party, or casting a ballot might be advertised.
2.	Votes were counted, and results were carried out consequently without a believed intermediary.	If casting a ballot is permitted, votes are counted inside, and the result of casting a ballot should be taken care of physically.
3.	Completely democratized.	Usually progressive.
4.	Administrations offered are taken care of consequently in a decentralized way.	Requires human taking care of, or halfway controlled mechanization, inclined to control.

How DAOs put organizational activities on the blockchain

- The smart contracts within the DAO framework (smart contracts that run the DAO) **facilitate, execute, and enforce actions without humans needing to intervene**.

- For example, the smart contracts dictate the conditions (such as passing a vote with a minimum quorum and level of support) to be met in order to withdraw funds. If those conditions are met, the smart contract will automatically send those funds.

- The actions a smart contract can take are determined up front by the actors who deploy the DAO. These can be changed later, following the initial conditions and parameters. For example, a smart contract with token-based voting couldn't just decide to change its parameters to implement wallet-based voting. You would need to follow the parameter changing process—maybe by holding a vote—to do this. If you can't, then you would need to deploy a whole new DAO that has the rules you want.

Here are a few key parts of an organization that are run onchain in a DAO framework:

- **Human organization activity:** membership is defined (ex.by paying monthly dues).
- ***On-chain equivalent:*** *purchase the DAO's governance token to get voting rights, similar to "membership," in the organization.*
- **Human organization activity:** a community makes decisions to achieve a common goal (ex. with a committee or board resolution).
- ***On-chain equivalent:*** *wallets cast governance tokens in votes, like voting chips.*
- **Human organization activity:** a shared pool of funds is managed (ex. a checking account at a bank)
- ***On-chain equivalent:*** *funds are withdrawn and deposited directly on the blockchain.*

Components of human organizations applied to DAOs

DAOs are another step forward in the evolution of human organizations. They use the same components that many organizations past and present have, such as:

- **Actors.** People or other agents participating in the DAO. In crypto, we use wallet addresses to represent humans or potential non-human actors (like a computer program or AI). These wallets hold tokens that allow them to participate in governance. However, one person can hold multiple wallets, so don't link these too tightly.
- **Assets.** DAOs have shared resources, such as fungible and nonfungible tokens.
- **Permissions.** Like rules, but enforced by code rather than by law.
- **Decision making.** Different mechanisms, like voting, that the DAO uses to decide what to do next.
- **Shared purpose and goals.** People organize because they're bound by a shared purpose or have goals in mind.

The five components of human organizations that apply to DAOs

- **Actors**
  - In DAOs, actors (people, organizations, or AI) use wallets and tokens to interact.
- **Wallets**
  A real-world wallet is where you hold your cash. In the web3 world, a crypto wallet is similar—except here, it's where you hold your tokens (also called crypto or cryptocurrencies).
  - **You can think of your crypto wallet as your access point to the blockchain.** You use your wallet to interact with the blockchain to buy, sell, move, and store tokens, and create and confirm actions in blockchain-enabled organizations like DAOs.
  - A wallet is necessary to interact with DAOs, because you'll need to hold tokens to participate. You can also use wallets for governance, because they can be put on an allowlist that grants your wallet access to vote. We'll go into detail in that below!

- **Tokens**
  - DAOs mint tokens to use in governance. The DAO might assign tokens to wallets or put them on a decentralized exchange for wallets to purchase.
  - You “hold” your tokens in your wallet. Really, your wallet just holds the keys to access your crypto on the blockchain. But you may hear that your tokens are “in” your wallet.
  - DAOs mint tokens that are unique to them. For example, at Aragon we use the Araon Network Token ($ANT) to cast votes and conduct governance.
  - But, it’s not necessary for a DAO to have a token at all! In fact, many DAOs choose to never mint tokens, using different governance models instead.

Assets

- Cryptocurrencies, including DAO governance tokens, are recorded on the blockchain. That means you need to make a transaction on the blockchain to move or interact with them. It's often said that crypto is simply "on" the blockchain. They're also called on-chain assets. This is opposed to off-chain assets, such as the local currency of your country, which is not stored on the blockchain.
- **A DAO treasury, sometimes called a vault, is similar to a wallet, except one person can't just sign the transaction and move funds. It's managed by a smart contract.** In order to move them, you must meet the conditions set in the smart contract, and then certain parameters must be met, such as number of votes (counted as tokens) and a certain percentage of tokens that need to be cast for "yes."

Permissions

- A permission management system is a set of smart contracts that determines **who and what can perform certain actions**.
- The "who" refers to the one able to execute the action. This can be any wallet or smart contract. The "what" refers to the action that gets triggered. This can be a token transfer, a transaction execution, or anything that you can write into a smart contract. In web3, this is all done trustlessly and automatically through smart contracts.
- **An onchain permission management system, or a DAO framework, is a set of smart contracts containing parameters that determine who can perform certain actions.**
- DAOs use **on-chain permission management systems to determine who is able to execute actions *and* under which circumstances.** For example, one person cannot withdraw the entire DAO treasury into their own wallet, because there are conditions and parameters, such as holding a vote that meets a certain quorum and level of support for proposals, required to gain access to the treasury.

Decision making

- DAO governance is *who* can make decisions and *how* those decisions are made. We use wallets and tokens to execute DAO governance.
- We are still in the early days of experimenting with DAO governance, but today it comes mostly in the form of voting, either wallet-based or token-based.
- For example:
- **In 1 authorized wallet = 1 vote governance, the voting power is the same for every wallet on the allowlist.** So, if one wallet holds 2% of tokens, and another wallet holds 5%, they still have the same voting power as long as they're both on the allowlist.
- **In 1 token = 1 vote governance, an individual's voting power is directly proportional to the number of tokens they hold.** So, if a token holder has 2% of tokens, they have 2% of the total available voting power.
- Fungible tokens (ERC-20s) and non-fungible tokens (ERC-721s) are two types of tokens your DAO might mint. Minting a governance token is like creating voting chips or issuing membership cards, except on the blockchain.
- DAO governance is still evolving. Right now, DAOs typically use voting to make decisions. We have both token voting and wallet voting available in the Aragon App. **But there are tons of governance mechanisms out there today. DAOs don't need to have voting at all to still be considered DAOs!**

Shared purpose and goals

- Actors organize to achieve a shared purpose and goals. The goal could be anything: raising funds for a community garden, starting a new protocol, or launching a sports fan club.
- Since you can join most DAOs permissionlessly by buying their governance token, many competing priorities and strategies can emerge. This can create more innovative spaces where DAO members propose and decide on the most compelling direction for the project to take. **This creates more dynamic and creative organizations, because there is a diversity of viewpoints and approaches amongst the organization's members.**
- Even though the members of a DAO may try to advance different strategies and goals, smart contracts can be used to align incentives. **DAO technology allows you to program incentives directly into the governance of your organization.** This can encourage certain behaviors, aligning incentives to reach certain goals.
- Smart contracts introduce the possibility for unique incentive structures. For example, if you need more participation in votes, you can program incentives into the contract that are given to token holders when they cast votes.
- **The programmability of smart contracts makes shared purpose not just intrinsic, but extrinsic as well.**

Components of DAO

- **No central legal entity:** In DAO, there is no central legal entity, this means that no single entity is responsible for regulating the project.
- **Self-enforcing code:** Smart contracts are created and extensively tested to make sure important details are not overlooked.
- **Token acts as an incentive for validators:** Tokens are used in DAO for validators to motivate them and to ensure active, fair, and quick participation.

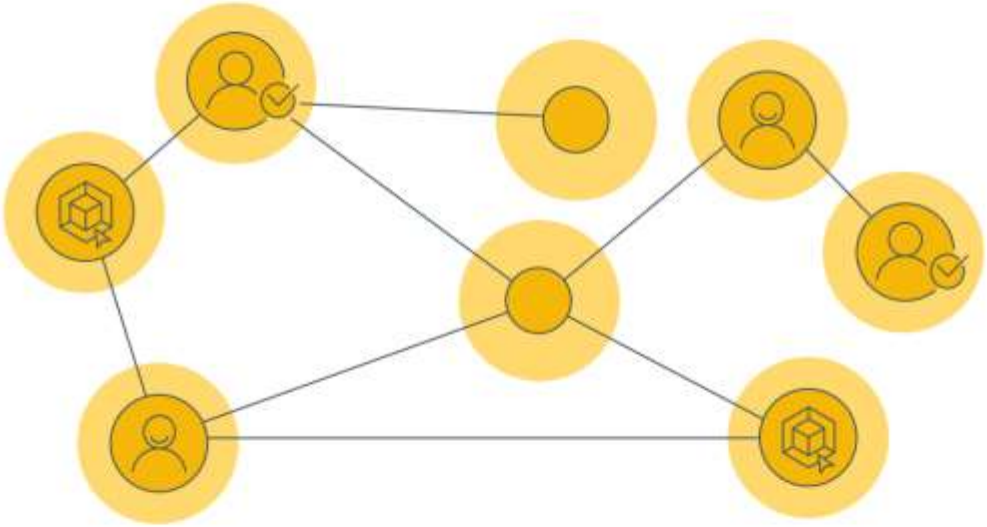
Components of DAOs

Machine consensus around token governance rulesets and smart contracts instead of legal employment contracts.

No centralized legal entity

Self-enforcing code (smart contracts)

Tokens act as incentive for validators

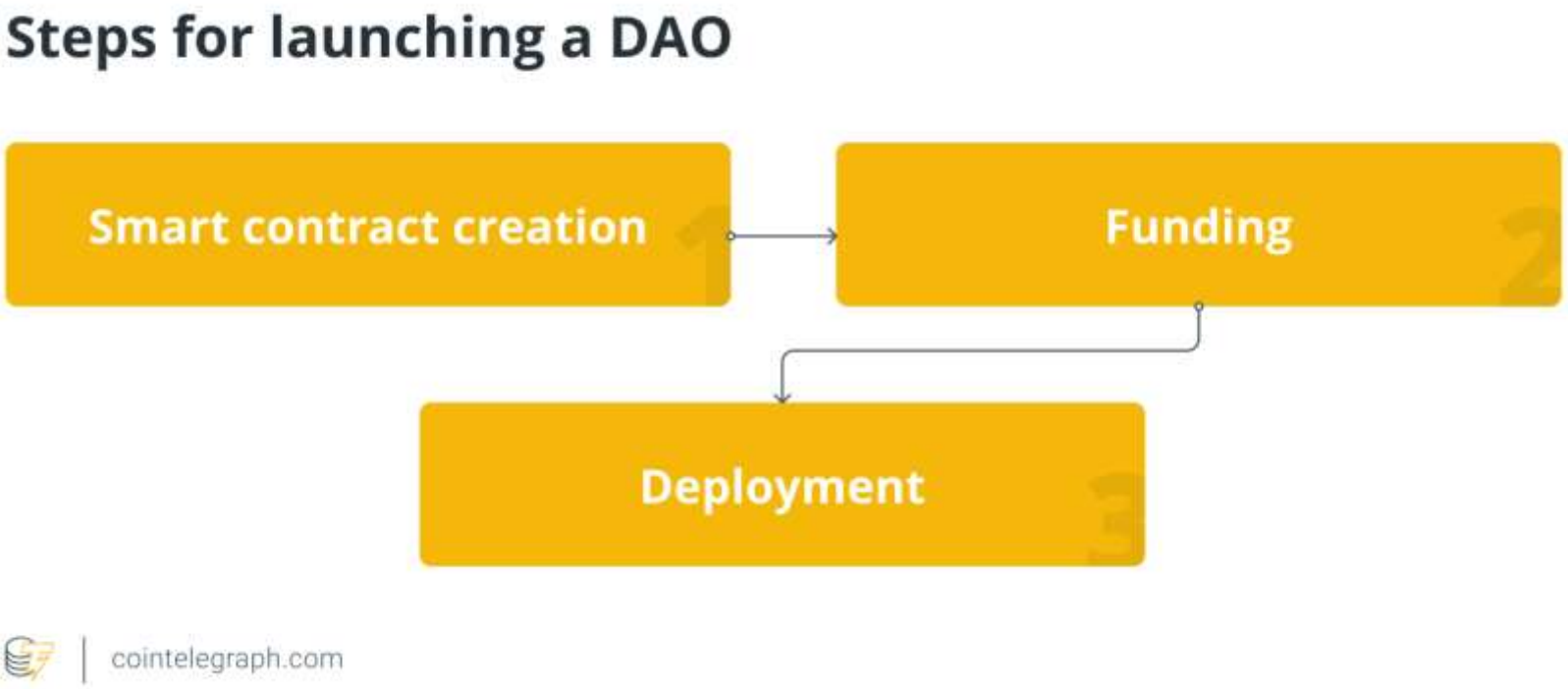


Distributed Network of Autonomous Stakeholders

Exchange | **Validator** | **User** | **Developer**

source: ***blockchainhub.net***

Steps For Launching a DAO



- **Smart contract creation**: First, a developer or group of developers must create the smart contract behind the DAO. After launch, they can only change the rules set by these contracts through the governance system. That means they must extensively test the contracts to ensure they don’t overlook important details.
- **Funding**: After the smart contracts have been created, the DAO needs to determine a way to receive funding and how to enact governance. More often than not, tokens are sold to raise funds; these tokens give holders voting rights.
- **Deployment**: Once everything is set up, the DAO needs to be deployed on the blockchain. From this point on, stakeholders decide on the future of the organization. The organization’s creators — those who wrote the smart contracts — no longer influence the project any more than other stakeholders.

DAO Examples

- Here are some examples illustrating how DAO can be utilized:
- **DASH:** The well-known computerized money Dash is an illustration of a decentralized independent association in light of the manner in which it is represented and the manner in which its planning framework is organized.
- **A cause:** One can acknowledge enrollment and gifts from anybody on the planet and the gathering can choose how they to spend gifts.
- **A consultant organization:** One can make an organization of workers for hire who pool their assets for office spaces and programming memberships.
- **Adventures and awards:** It is possible to make an endeavor store that pools speculation capital and decisions on dares to back. Reimbursed cash could later be rearranged among DAO individuals.

How Do DAOs Work?

- So far we are using people to “store” information instead. In order to know how much hiring a new person would cost? – There is a person answerable in the human resources department. Similarly, to get movement costs repaid?- There is a separate person responsible for this in the accounting.
  - In a DAO, there is a code for that. Computers will take over much of the decision-making and operations we see nowadays. The final control, however, is still with humans, the shareholders. Shareholders have voting rights just like in regular corporations. They dictate the general direction and accept or decline initiatives.
  - The general idea is to bring the benefits of blockchain technology to management.
  - The blockchain is immutable, precise, and consistent. It is also transparent and open so that anyone could review companies. Strong consistency makes DAOs reliable business partners.
  - Such an organization is also harder to put under pressure. It will be difficult to ban it from operating somewhere. As it is controlled by the organization members and not influenced by a central government authority to put under pressure

DAO Membership

- There are various models for DAO membership. Membership can decide how casting ballot functions and other key pieces of the DAO.
- **1. Token-based membership:** Normally completely permissionless, contingent upon the token utilized. For the most part, these administration tokens can be exchanged for permissionless on a decentralized trade. Others should be procured by giving liquidity or another 'evidence of work'. In any case, just holding the symbolic awards admittance to casting a ballot. Ordinarily used to administer expansive decentralized conventions as well as tokens themselves.
  - **Example:** MakerDAO's token MKR is generally accessible on decentralized trades. So anybody can become tied up with having cast a ballot power on the Maker convention's future.
- **2. Share-based membership:** Offer-based DAOs are more allowed, yet at the same time very open. Any imminent individuals can present a proposition to join the DAO, typically offering recognition of some worth as tokens or work. Offers to address direct democratic force and possession. Individuals can exit whenever with their proportionate portion of the depository. Regularly utilized for all the nearer sew, human-driven associations like foundations, laborer assemblages, and venture clubs. Can administer conventions and tokens too.
  - **Example:** MolochDAO is centered around financing Ethereum projects. They require a proposition for enrollment so the gathering can evaluate whether you have the important mastery and funding to make educated decisions about possible grantees. You can't simply purchase admittance to the DAO on the open market.

Ethereum and DAOs

- Ethereum is the ideal establishment for DAOs for various reasons:
- Ethereum's own agreement is conveyed and set up enough for associations to trust the organization.
- The agreement code can't be changed once live, even by its proprietors. This permits the DAO to run by the principles it was modified with.
- Agreements can send/get reserves. Without this, there is a need for a believed delegate to oversee a bunch of reserves.
- The Ethereum people group has demonstrated to be more synergistic than cutthroat, taking into consideration best practices and emotionally supportive networks to arise rapidly.

Advantages of DAO

- **Decentralization:** DAO emphasizes being driven by a collective rather than an individual. With DAO, participants have a much stronger say in the organization's direction.
- **Community Driven:** DAOs make it easy for communities worldwide to connect and build a prospering vision together. DAO is accessible to Individuals who may have had the opportunity in the past to connect and work together.
- **Principle-agent dilemma:** One of the main advantages of DAO is that it provides a solution for the principle-agent dilemma. This dilemma is a conflict in priorities between a person (principle) and the entities making decisions on their behalf (agent). One of the common examples of this is problems between Stakeholders and CEOs. DAO solves this problem through community governance. Here, principles don't have to trust agents who work on their behalf instead they work as a part of a group whose incentives are aligned.

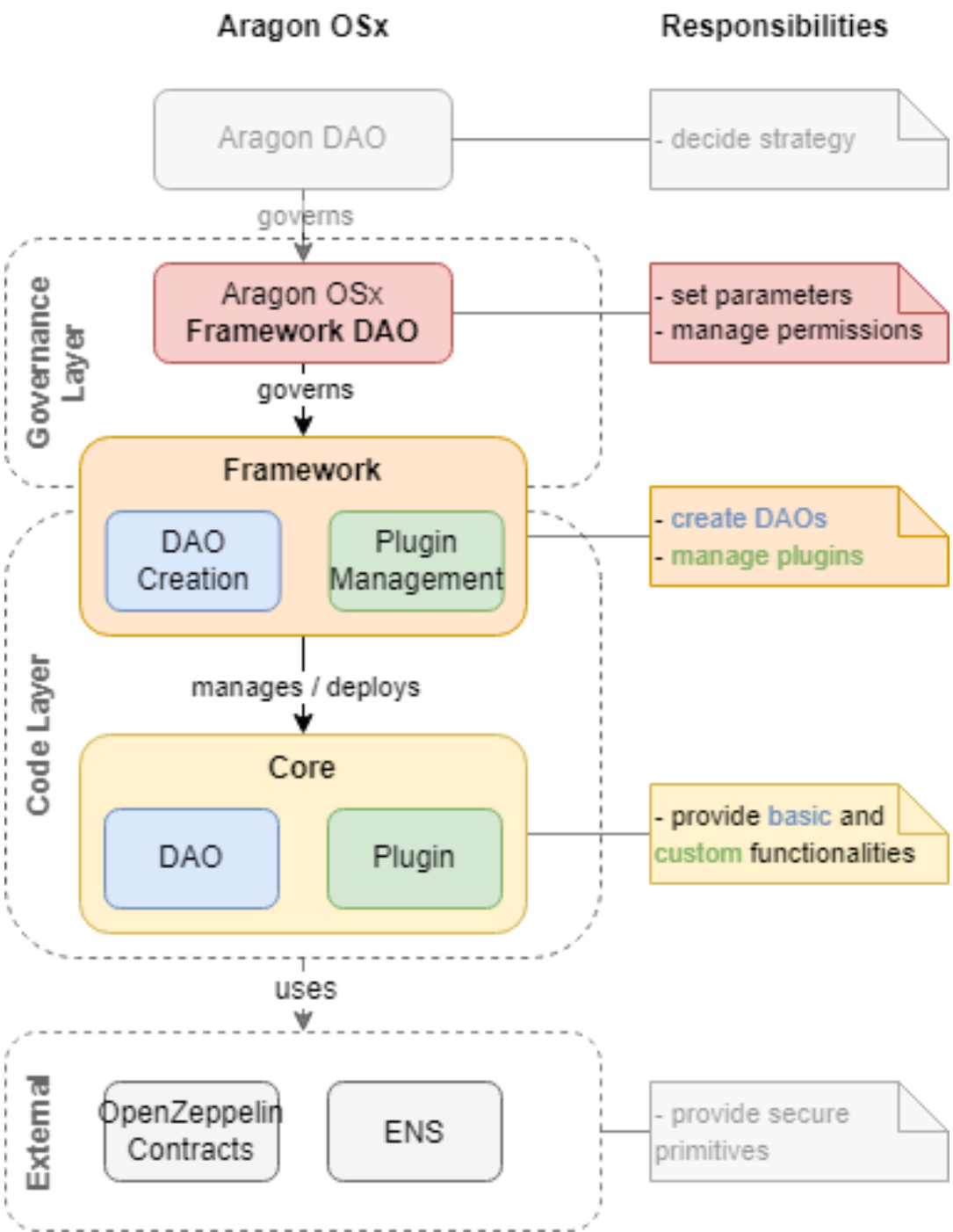
Disadvantages of DAO

- **Security:** DAO can be launched with just a few lines of code and given the immense tech stack a well-run DAO requires to operate effectively thus security remains a vulnerability as it requires significant technical expertise and it is **expensive** to keep best security practices implemented.
- **Slow Decision Making:** With DAO scaling there comes an issue of getting everyone to vote on proposals in a timely manner and with different time zones and investor priorities, keeping DAO participants up to date can be challenging.
- **The Bikeshedding Effect:** Parkinson's Law of Triviality states that the amount of time spent discussing an issue in an organization is inversely related to its importance in the scheme of things. This is also known as bike-shedding. It can have a negative impact on personal productivity as it causes inefficient management of time.
- **No legitimate structure for circulating DAOs:** DAOs can be circulated across different locales, and there's no legitimate structure for them. Any lawful issues that might emerge will probably require those required to manage various territorial laws in a convoluted fight in court. In July 2017, for instance, the United States Securities and Exchange Commission gave a report not really settled that the DAO sold protections as tokens on the Ethereum blockchain without approval, disregarding bits of protection law in the country.

ARAGON

Aragon - Overview

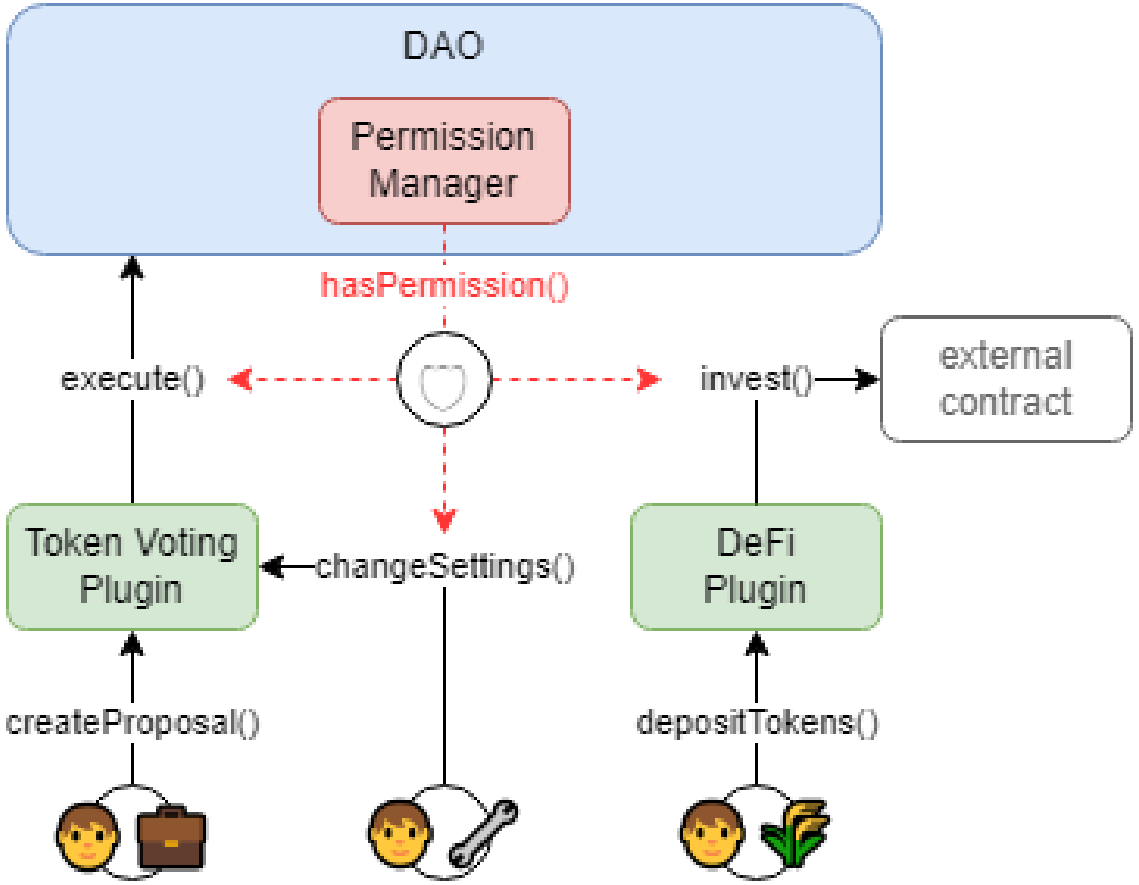
- Aragon is a blockchain-based platform that provides tools for building and managing Decentralized Autonomous Organizations (DAOs).
- It offers a **no-code platform** for launching and managing DAOs, along with a smart contract framework, Aragon OSx, for building custom DAOs.
- The platform facilitates **decentralized governance**, allowing users to create and participate in DAOs with ease and security.
- Aragon was founded in 2016 and allows users to manage organizations using blockchain technology, with the goal of creating more efficient infrastructures.
- The network is a "**digital jurisdiction**" governed by stakeholders that can vote to create or amend laws included in the network's "**constitution**".
- Operation of the network is supported by fees which are collected from organizations and allocated based on the underlying governance model.


Aragon OSx
Responsibilities
Aragon DAO
- decide strategy
governs
Governance Layer
Aragon OSx
Framework DAO
- set parameters
- manage permissions
governs
Framework
DAO Creation
Plugin Management
- create DAOs
- manage plugins
Code Layer
manages / deploys
Core
DAO
Plugin
- provide basic and custom functionalities
uses
External
OpenZeppelin Contracts
ENS
- provide secure primitives

- **Code Layer**
  - The foundation of the Aragon OSx protocol is the **code layer** constituted by the core and framework related contracts. The core contracts provide the core primitives intended to be used by users and implemented by developers of the DAO framework. The framework contracts provide the infrastructure to easily create and manage your DAOs and plugins easy. Both are built on top of external dependencies, most notably the OpenZeppelin and the Ethereum Name Service (ENS) contracts.
  - The core and framework contracts are free to use, and no additional fees are charged.
- **Governance Layer**
  - To govern the framework infrastructure, an Aragon OSx Framework DAO is deployed constituting the **governance layer** of the Aragon OSx protocol.

The Smart Contracts behind DAOs

- **The DAO contract:** The DAO contract is where the **core functionality** of the protocol lies. It is in charge of:
  - Representing the identity of the DAO (ENS name, logo, description, other metadata)
  - Keeping the treasury
  - Executing arbitrary actions to
    - Transfer assets
    - Call its own functions
    - Call functions in external contracts
  - Providing general technical utilities (signature validation, callback handling)
- **The Permission Manager:** The permission manager is part of the DAO contract and the center of our protocol architecture. It **manages permissions for your DAO** by specifying which addresses have permission to call distinct functions on contracts associated with your DAO.
- **Plugins:** Any custom functionality can be added or removed through plugins, allowing you to **fully customize your DAO**. These plugins can be related to
  - Governance (e.g., token voting, one-person one-vote)
  - Asset management (e.g., ERC-20 or NFT minting, token streaming, DeFi)
  - Membership (governing budget allowances, gating access, curating a member list)



An examplary DAO setup showing interactions between the three core contract pieces triggered by different user groups: The DAO contract in blue containing the PermissionManager in red, respectively, as well as two Plugin contracts in green. Function calls are visualized as black arrows and require permission checks (red, dashed arrow). In this example, the permission manager determines whether the token voting plugin can execute actions on the DAO, a member can change its settings, or if a DeFi-related plugin is allowed to invest in a certain, external contract.

six base functionalities

1. Execution of Arbitrary Actions

- The most important and basic functionality of your DAO is the **execution of arbitrary actions**, which allows you to execute the DAO's own functions as well as interacting with the rest of the world, i.e., calling methods in other contracts and sending assets to other addresses.
- Multiple Action structs can be put into one Action[] array and executed in a single transaction via the execute function.

- **2. Asset Management**
  - The DAO provides basic **asset management** functionality to deposit, withdraw, and keep track of
    - native
    - ERC-20 (Token Standard),
    - ERC-721 (NFT Standard), and
    - ERC-1155 (Multi Token Standard)
  - tokens in the treasury. In the future, more advanced asset management and finance functionality can be added to your DAO in the form of plugins.

3. Upgradeability

- Your DAO contract has the ability to be upgraded to a newer version (see Upgrade your DAO) if a new version of Aragon OSx is released in the future. These upgrades allow your DAO to smoothly transition to a new protocol version unlocking new features.

4. Callback Handling

- To interact with the DAO, external contracts might require certain callback functions to be present. Examples are the onERC721Received and onERC1155Received / onERC1155BatchReceived functions required by the ERC-721 (NFT Standard) and ERC-1155 (Multi Token Standard) tokens. Our CallbackHandler allows to register the required callback responses dynamically so that the DAO contract does not need to be upgraded.

5. Signature Validation

- Currently, externally owned accounts (EOAs) can sign messages with their associated private keys, but contracts cannot. An exemplary use case is a decentralized exchange with an off-chain order book, where buy/sell orders are signed messages. To accept such a request, both, the external service provider and caller need to follow a standard with which the signed message of the caller can be validated.
- By supporting the ERC-1271 standard, your DAO can validate signatures via its isValidSignature function that forwards the call to a signature validator contract.

6. Permission Management

- Lastly, it is essential that only the right entities (e.g., the DAO itself or trusted addresses) have permission to use the above-mentioned functionalities. This is why Aragon OSx DAOs contain a flexible and battle-tested **permission manager** being able to assign permissions for the above functionalities to specific addresses. Although possible, the permissions to execute arbitrary actions or upgrade the DAO should not be given to EOAs as this poses a security risk to the organization if the account is compromised or acts adversarial. Instead, the permissions for the above-mentioned functionalities are better restricted to the DAO contract itself and triggered through governance plugins that you can install on your DAO.

DAOSTACK

Overview

- DAOstack is an open-source platform built on the Ethereum blockchain, designed specifically for creating and managing DAOs.
- It offers a powerful suite of smart contracts and developer tools that enable the creation of scalable and customizable decentralized governance systems.
- DAOstack's core components include the **Arc framework, the ArcHives** protocol for decentralized data storage, and Alchemy, a user interface for managing DAOs.

- The DAOstack project was initiated in 2017 with an ambitious goal to build an infrastructure for global coordination. For that sake, DAOtack Limited, a non-profit in Gibraltar, has been founded.
- In May 2018, DAOstack Limited concluded fundraising via the GEN token sale, and has focused since then on developing technology and products for DAOs.
- In the 5 years since then, DAOstack Limited has developed several products, including the Arc modular framework for DAO governance, the Alchemy DAO platform which has been notably used by the DXdao to on-chain manage tens of millions of dollars, Common, a platform for open collaboration, and several other DAO products which did not reach the stage of production.

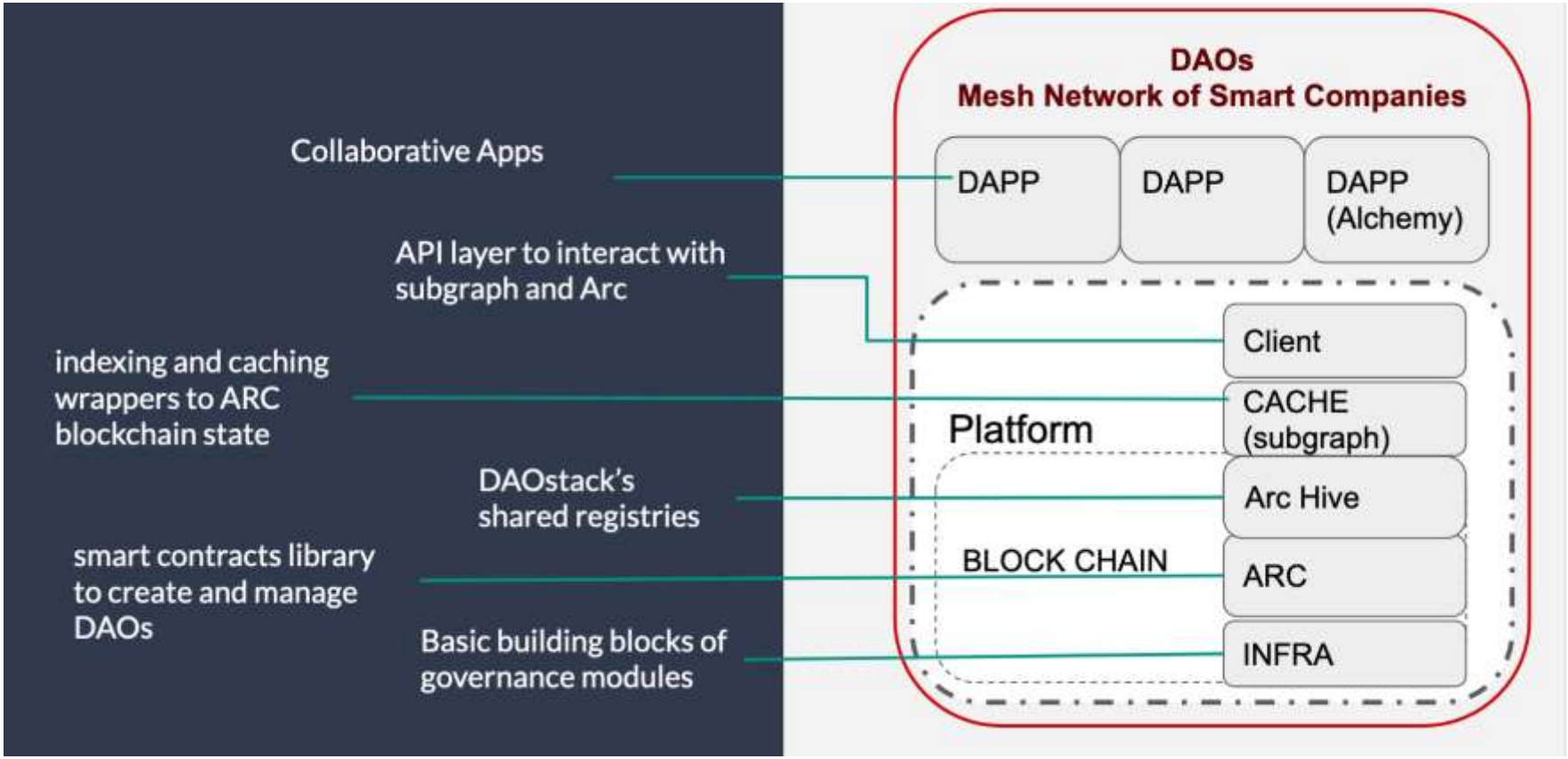
Key Features of DAOstack DAOs

- ***Modularity*:** DAOstack's Arc framework provides a set of modular smart contracts that can be combined and customized to create a unique governance system for each DAO.
- ***Scalability***: DAOstack is designed for large-scale collaboration, with the ability to handle thousands of participants and complex decision-making processes.
- ***Upgradability***: DAOs built on DAOstack can be upgraded over time to incorporate new features and improvements.
- ***Interoperability***: DAOstack's architecture supports integration with other blockchain platforms and technologies, such as decentralized finance (DeFi) protocols, NFTs, and more.
- ***User-friendly Interface***: Alchemy, DAOstack's native user interface, makes it easy for participants to interact with and manage DAOs.

DAOStack Architecture

The DAOstack ecosystem consists of a multitude of distinct but interoperable DAOs, interacting with each other to maximize the potential value of open and distributed collaboration.

- **Infra:** This is the base layer of the stack. It houses the most basic, decentralized governance backend software modules, such as **voting machines** and **voting permissions** systems such as DAOstack's Reputation System.
- **Arc:** This expands to create a complete library of modules to build the back ends of DAOs on Infra. Built also on Ethereum, Arc contains the **master contracts** that co-ordinate all the different DAO pieces together, as well as any number of flexible "**schemes**" that enable other DAO functions.
- **ArcGraph:** This is a **caching layer** based on The Graph which collects, stores, and organizes blockchain information, enabling DAOstack-based applications to achieve quick load times.
- **Application Layer:** Participants or creators of decentralized Organizations are to interact with the stack primarily through the **dApps** (decentralized applications) that use Infra, Arc, and ArcGraph for their backends.
- **Alchemy:** This is the first **production-ready application** built on the stack. It is an intuitive user interface for participating in decentralized Organizations governance. It lets projects seamlessly govern themselves, allocating shared resources and making effective decisions at scale.
- **ArcHives:** The service of the ArcHives is to organize and curate the modules within each layer



- public blockchain layer (Infra, Arc, Arc-Hive) which is the source of data
- caching layers (Subgraph) which allows fast access to the blockchain layer
- javascript library (Client) for application layer integration.

- **Infra** has two main components:
  - **Voting Machines** - A voting machine is a universal contract which can operate the voting process for any organization. Each voting machine follows its own predefined rules for the decision making and execution process. Rules for voting machines can be implemented for any voting process, from a simple protocol like an "Absolute Vote" (where 51% of the voting power should approve it in order for the decision to pass), or more sophisticated protocols like the Holographic Consensus voting protocol.
  - **Voting Rights Management** - A voting rights management system determines how voting rights are distributed. Any voting rights management system must have "balances" which represents the voting power each participant holds. There are 2 main approaches for managing voting rights: token-based voting and reputation-based voting. The main technical difference between the two is that tokens are transferable (i.e. tradable) while reputation is non-transferable. Another big difference which may appear (depending on implementation) is that a token is a property which cannot be taken while reputation may be redistributed by the organization itself. For most cases, we reccomend using the reputation-based voting model, however, Infra allows any voting right management system to be built.

- **Arc is a Solidity smart contract library for building DAOs.**
- DAOs built with Arc have a few basic contract components:
  - **Avatar** - The DAO's "account." This contract represents the address of the DAO and holds its assets.
  - **Reputation** - Voting in Arc is done mainly using Reputation. Reputation can be distributed and redistributed only by DAO decision, and it is generally given (via vote) to an agent according to their performance and contribution to a DAO.
  - Token - Each DAO may have its own token, which can be used in any way the DAO would like.
  - **Controller** - The controller is the "Access Control" of the DAO, managing who can interact with which DAO functions and enforcing the DAO's constraints.
  - **Schemes** - Schemes are a DAO's "actions": anything a DAO should act upon needs to be written and authorized by the controller as a scheme. Schemes might be used to help a DAO: propose and make investments, give reputation to agents, upgrade the DAO's contracts, register new schemes and constraints, etc.
  - **Global Constraints** - Global constraints are limitations on a DAO's actions. When executing a scheme, the controller checks the constraints to see if the action violates them, and blocks the execution if it does. Some examples for constraints might be: the token supply can't be increased over 1M tokens, the organization won't use more than 60% of its funds at once, etc.

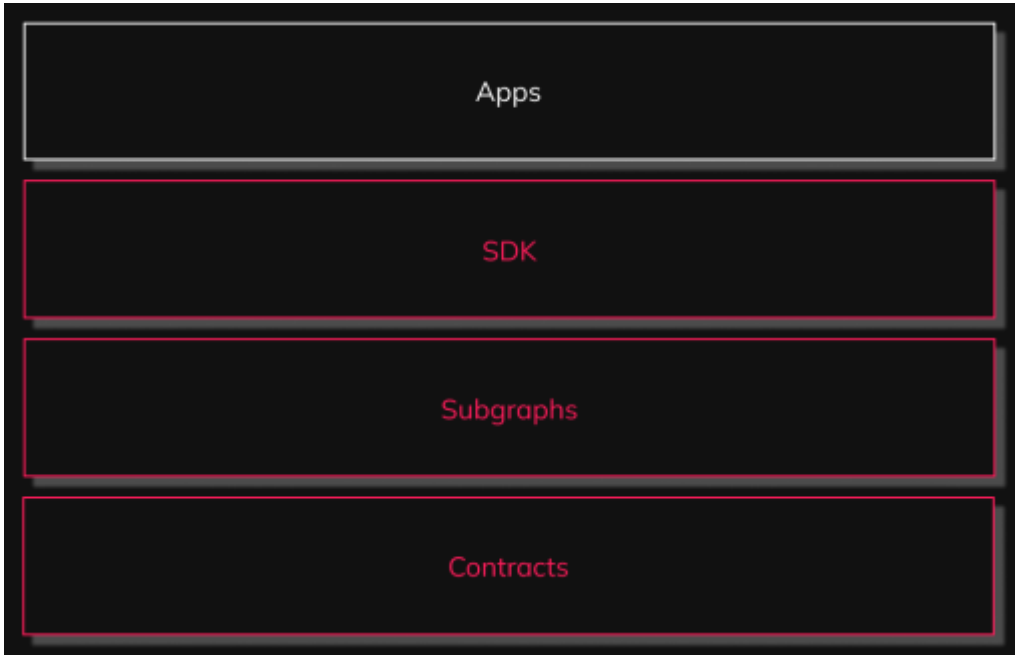
- **Client is a nodejs library that provides a helpful set of tools to interact with the DAOstack ecosystem.**
  - In particular, the library provides an interface to
  - DAOstack contracts and
  - DAOstack subgraph (an index of on-chain data).
- **Subgraph indexes the blockchain data and stores it in postgres database for easy and quick access.** The subgraph runs on a Graph Node which is a server that developers can run local or remote. The data store can be queried by GraphQL endpoints. DAOstack subgraph is based on graphprotocol.
- **DAOstack has built its own Dapp called Alchemy, a front-end interface for DAOs**, or more specifically, for budget management in decentralized organizations. Alchemy allows end users to make collaborative budgeting decisions and allocations using the Holographic Consensus protocol.

DAOHAUS

Overview

- DAOhaus facilitates the creation of DAOs using the Moloch v3 smart contracts.
- With Moloch DAOs, users can trustlessly share funds in safes and make decisions through proposals.
- Through DAOhaus, community actions are executed on-chain.

- The architecture of DAOhaus is built upon a layered approach to ensure flexibility, scalability, and ease of use.



- **SDK**
  - At the heart of the DAOhaus framework is the SDK. Designed for rapid development, it enables users to swiftly create a DAO app using our starter repo. The starter, crafted with TypeScript in React, offers a plethora of functionality like DAO overview, proposal forms, and members list, all while leveraging the Vite development environment for front-end optimization. It's an all-in-one toolkit that embeds both single and multi-DAO capabilities, wallet connections, and more.
- **SUBGRAPHS**
  - These provide a structured way to fetch data using a well-defined schema. The DAOhaus subgraph schema comprises entities such as DAOs, Proposals, Votes, and Members, among others. This structured representation ensures data consistency and accessibility across DAOhaus applications.
- **CONTRACTS**
  - Powering the DAOhaus apps is the robust Moloch DAO framework, a testament to our commitment to combating Coordination Failure. Moloch v3, our latest version, amalgamates the best of Moloch, Minion, and Compound frameworks. It's more than just a DAO template; it's a minimalistic yet modular governance layer that interfaces seamlessly with the Gnosis Zodiac standards, ensuring fluid transactions over a multisig treasury.

COLONY

Overview

- Colony is a platform that allows you to create and join DAOs that are organized around projects, tasks, and domains.
- A DAO is a group of people who coordinate their actions and decisions through a shared set of rules and incentives, without relying on a central authority or intermediary.
- Colony enables you to define your DAO's purpose, structure, and governance, and to assign roles and responsibilities to your members.
- You can also use Colony to reward your members for their contributions, using tokens and reputation.

How does Colony work?

- Colony works by using smart contracts on the Ethereum blockchain to automate and enforce the rules and processes of your DAO.
- You can create your own DAO using the Colony app, or join an existing one.
- Each DAO has its own native token, which represents the ownership and influence of its members.
- You can also use external tokens, such as ETH or DAI, to fund your DAO or pay for services.
- Each DAO is divided into domains, which are like departments or sub-projects.
- Each domain has its own budget, permissions, and reputation pool. You can create tasks within domains, and assign them to members or open them for bidding.
- Tasks have specifications, deadlines, and payouts. When a task is completed, the worker and the evaluator receive tokens and reputation, which reflect their skills and performance.

Why Colony?

- ***Powerful***
  - Colony has everything an organization needs to operate on-chain. Payments, teams, authority, governance, fundraising, revenue sharing, arbitration, the ability to interact with any other smart contract on Ethereum, and a whole lot more.
- ***Practical***
  - In a healthy organization, voting should be rare. That's why Colony operates a more agile system of 'lazy consensus'. It means organizations spend their time productively, not endlessly voting.
- ***Flexible***
  - Colony Core provides powerful primitives for finances, org structure, and authority. Extensions are plug-ins that build on those, shaping organizations according to their needs.
- ***Easy***
  - Colony makes DAOs easy. Unlike the complicated, clunky competition, Colony's application seamlessly integrates powerful DAO functionality into a polished, logical, and easy to use experience.

Key Features of Colony DAO

- Colony DAO offers a range of features that set it apart from other DAO platforms. Some of these features include:
  - **Reputation-based incentives**: Colony's reputation system rewards contributors with non-transferable tokens that represent their influence within the organization.
  - **Task management**: The platform allows users to create and assign tasks, manage deadlines, and set budgets.
  - **Budget allocation**: Colony enables organizations to allocate funds for tasks and projects based on the reputation of the contributors.
  - **Decentralized decision-making**: Colony's governance model promotes decentralized decision-making by allowing reputation-weighted voting.
  - **Open-source platform**: Colony's open-source nature allows developers to build custom applications and integrations.