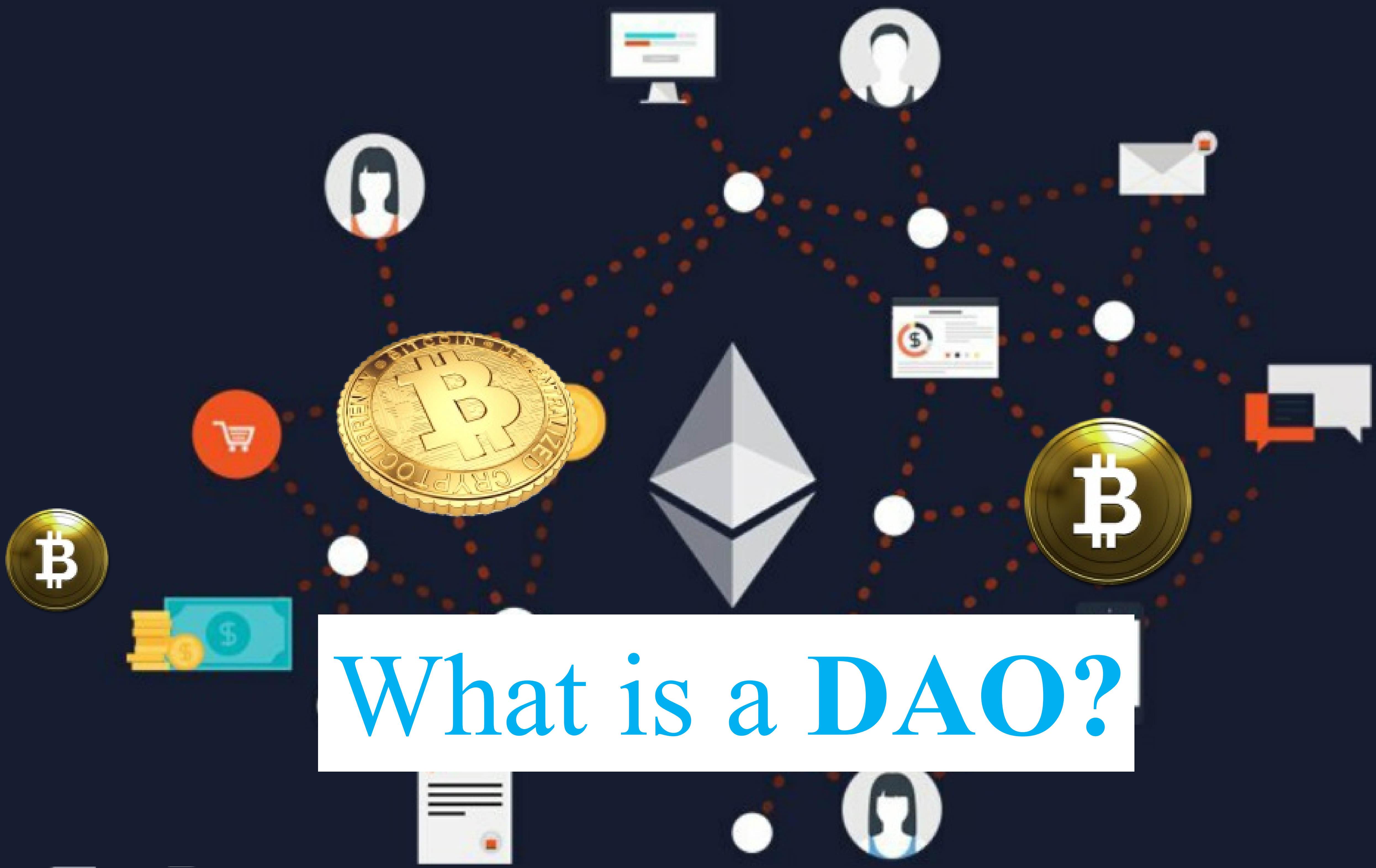
# What is a DAO?



MorCryp

Imagine this: a driverless car cruises around in search of passengers.

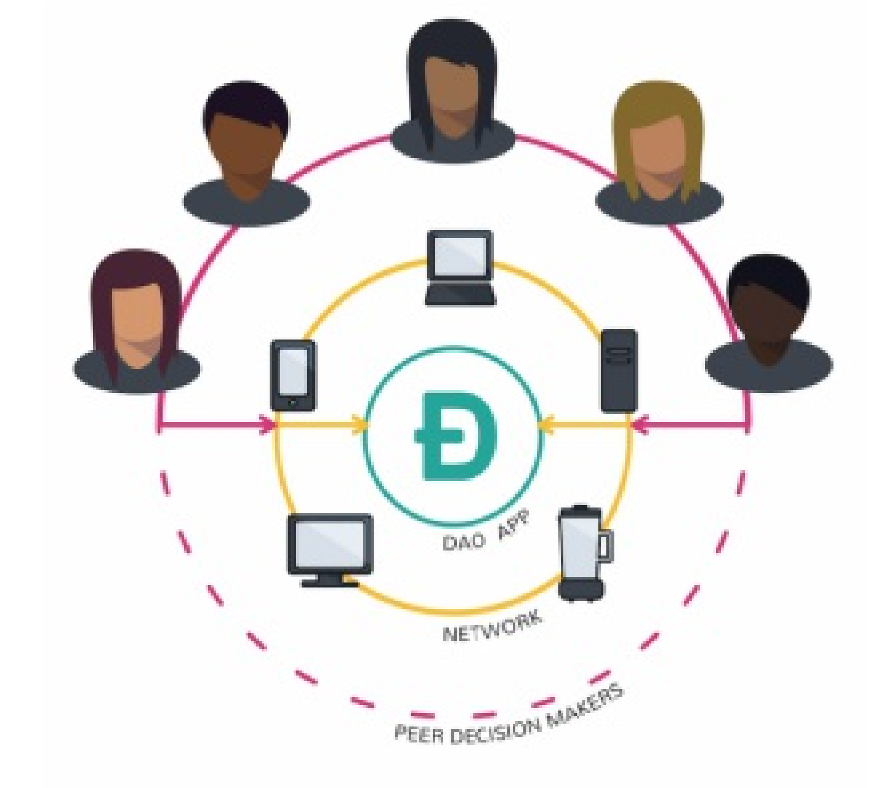
After dropping someone off, the car uses its profits for a trip to a charging station. Except for it's initial programming, the car doesn't need outside help to determine how to carry out its mission.

That's one "thought experiment" brought to you by former bitcoin contributor Mike Hearn in which he describes how bitcoin could help power leaderless organizations 30-or-so years into the future.

What Hearn described is one dream use case for a decentralized autonomous organization, or a DAO, an idea that swirled through the community not long after bitcoin was released in 2009. The thought is that if bitcoin can do away with financial middlemen, then maybe companies and other organizations can one day operate without hierarchical management.

In short, DAOs aim to hard-code certain rules that a company would from the get-go. This could be setting aside a certain percentage of earnings for a cause or determining a process by which such a rule could be changed.

In the abstract, this is similar to how a normal company works. The big difference is that the rules of normal companies are not enforced digitally.



**The DAO**

The best-known attempt at creating such an organization was called "The DAO." Launched in 2016, the project failed in a matter of months, but it's a good example of what people have in mind when they talk about the technology.

The plan was for participants to receive DAO tokens, then vote for which projects to fund. For selecting projects to invest in, it relied on the "wisdom of crowds."

There are a few ways that The DAO intended to improve on the governance of today's organizations:

•Anyone with internet access could hold DAO tokens or buy them

•DAO creators could set whatever rules they voted on.

In abstract, DAOs function similarly. They rely on smart contracts, or pre-programmed rules that describe what can happen in the system.

These smart contracts can be programmed to carry out a variety of tasks, such as doling out funds after a certain date or when a certain percentage of voters agree to fund a project.
Some proponents say it can work for an organization where any sort of decision needs to be made, not just those related to money.
Essentially, they see it as a way to cryptographically guarantee democracy, where stakeholders can vote on adding new rules, changing the rules, or ousting a member, to name a few examples.

**SECURITY**
It's easy to see why "unstoppable code" could pose a security problem.
Today, it's difficult to change a DAO, or the smart contracts underpinning it once it's deployed to the ethereum blockchain. This is "good," because one person or entity can't change the rules.
But it's also potentially a huge disadvantage. If someone spots a bug in a running DAO, developers can't necessarily change the code.
That was the problem with The DAO. Observers watched the attacker slowly drain of funds, but they couldn't do anything to stop it. (Technically, the hacker was following the rules as they were deployed).
Ethereum's lead coders reversed the transaction history to return funds to their owners, which was a controversial decision leading to a rift in the community.
The best way to handle a similar future situation is still up to debate.