

# Mycro: A Platform for Building Profitable Open-Source Projects

Version 1.0.1

June 10th 2018

Mycro Founding Team

[founders@mycrocoin.org](mailto:founders@mycrocoin.org)

## Abstract

The Internet has enabled [open-source software](#) to become the prevailing approach to writing production-grade tools and infrastructure for software development. Developers may contribute to open-source as part of an enjoyable hobby, to improve their skills, and immerse themselves in a larger developer community. Large corporations have the ability to offload part of the engineering costs involved in maintenance and development of their internal projects by making them open-source. Economic incentives don't exist for making profitable open-source consumer applications. We propose an organizational structure called a [DMOP](#) (DAO managed open-source project) which enables profitable open-source consumer applications. First, we make an argument for how DMOPs bring more efficiency to what we will define as [open-source micro economies](#). We then introduce [Mycro](#), a platform for building open-source consumer applications using DMOPs.

# Table of Contents

## [1. Introduction](#)

## [2. Background](#)

### [2.1 Open-Source Software](#)

### [2.2 Exchange](#)

### [2.3 Microeconomy](#)

### [2.4 Smart Contracts](#)

### [2.5 Decentralized Autonomous Organizations](#)

## [3. New Concepts and Definitions](#)

### [Microcurrency](#)

### [3.1 Open-Source Microeconomies](#)

### [3.2 DAO Managed Open-Source Projects](#)

### [3.3 Example Open-Source Project Use Case](#)

## [4. Mycro Platform](#)

### [4.1 DMOP Architecture](#)

### [4.2 MycroCoin](#)

## [5. DAO Actions](#)

### [5.1 Managing Source Code](#)

### [5.2 Selecting Authorized Servers](#)

### [5.3 Centralized Exchange Module](#)

## [6. DMOP Consensus](#)

### [6.1 Active Coins](#)

### [6.2 ASC approval process](#)

### [6.3 Vote Deferral](#)

## [7. Example Use Cases of Mycro DMOPs](#)

## [Conclusion](#)

## 1. Introduction

The Internet has enabled [open-source software](#) to become the prevailing approach to writing production-grade tools and infrastructure for software development. This has resulted in the adoption of grass roots open-source projects like Linux and Git by multi-billion-dollar corporations such as Google and Facebook. In addition, large corporations open-source some of their internal projects to offload part of the engineering costs involved in maintenance and development. Examples of this are Facebook releasing React and Google releasing Tensorflow. The open-source philosophy has proven to work well when the project has a sufficiently large user base whom have the ability to contribute code.

Economic incentives for consumer software applications do not naturally align between project owners, developers, and users well enough for open-source to flourish. Organizations often rely on a closed-source model in order to ensure they maintain a competitive advantage and can profit from the value produced by their consumer application. Recent developments in the cryptocurrency space, including turing complete blockchains such as Ethereum, enable open-source applications to achieve a level of profitability previously accessible only through proprietary licensing.

We propose an organizational structure called a [DMOP](#) (DAO managed open-source project) which enables profitable open-source consumer applications. First, we make an argument for how DMOPs bring more efficiency to what we will define as [open-source micro economies](#). We then introduce [Mycro](#), a platform for building these open-source consumer applications using DMOPs.

## 2. Background

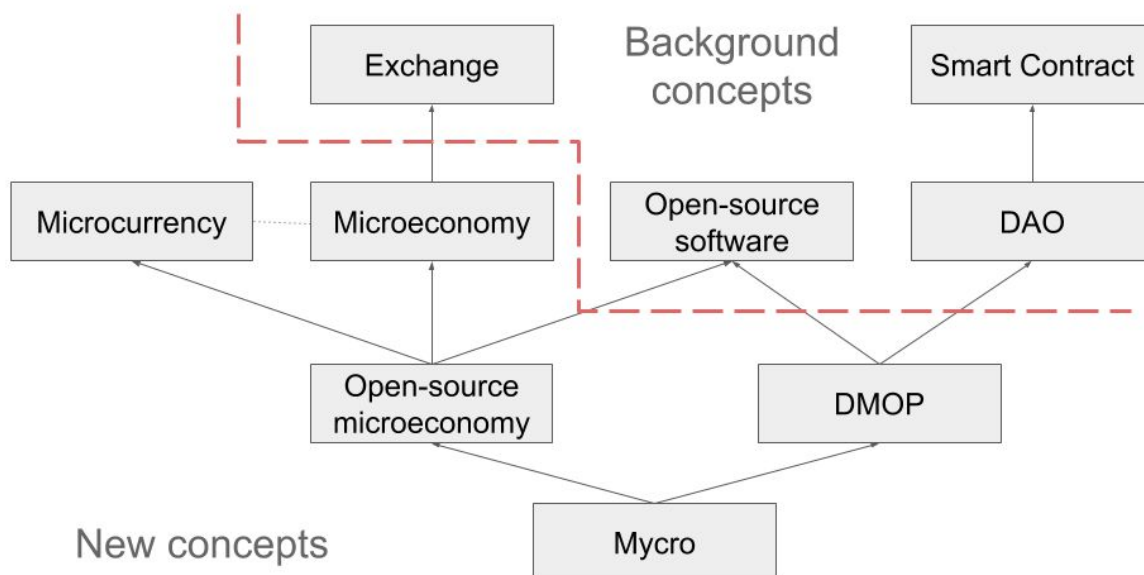


Figure1. Graph of concepts and their dependencies. To understand a DMOP you must first understand open-source software and DAOs.

## 2.1 Open-Source Software

Open-source software refers to software where the original source code used to create the software is made available to the public. Open-source allows software to be written in a style that Eric S Raymond famously referred to as a bazaar in his acclaimed essay "The Cathedral and the Bazaar".<sup>[1]</sup> By enabling any competent developer to contribute, software development begins to look like a busy bazaar: hectic, but full of energy and the exchange of ideas. Such an approach allows a distributed group of developers to efficiently contribute to one goal without wasting time reimplementing the same code.

Another valuable aspect of open-source is the idea of a benevolent dictatorship. This term comes from another essay by Raymond, "Homesteading the Noosphere."<sup>[2]</sup> Raymond proposes that leaders of an open-source project are forced to benevolence. If a leader is not benevolent, the community can copy the source code and choose new leadership. This process of copying the source code and choosing new leadership is called forking. The threat of forks forces open-source projects to move in a way that is beneficial to the community as a whole rather than leadership.

Projects like Linux have benefited greatly from open-source development. The reasons why developers contribute to open-source vary, some reasons are:

- I. To gain practical experience.
- II. To fix a problem they have with the project.
- III. Love of the community.

Raymond predicted that, entering the 21st century, open-source would spread into the application space citing nascent projects at the time such as Gimp. Almost two decades into the 21st century, the majority of consumer-applications are still closed-source. Open-source applications often face a few problems.

- I. They lack focus. Every developer wants to put their own use case within a project. Consequently, open-source projects often best cover the needs of the technical people who develop the application and leave other user demographics with little benefit.
- II. The user experience is poorly thought through and often added as an after-thought.
- III. They often lag behind commercial options in terms of features.

These issues can all be solved by setting appropriate economic incentives. Incentives for project leaders to better focus on vision, incentives to focus on the user experience, and incentives for developers to contribute more to open-source projects.

## 2.2 Exchange

An exchange is an abstract concept where parties trade goods and/or services with each other. Examples of exchanges include Uber, Walmart and even World of Warcraft. Exchanges are said to have a strong network effect because each new user provides value to all other users. Each new user on Craigslist makes the platform better because buyers have an easier time finding items they're interested in and sellers have an easier time finding buyers. The network effect of an exchange acts as protection against copycats.<sup>[3]</sup> Users gain more value from an exchange with one thousand participants than an exchange with ten.

## 2.3 Smart Contracts

Smart contracts are programs that run on blockchains. The original blockchain, Bitcoin, supported simple smart contracts. Ethereum popularized this idea by offering smart contracts with more advanced functionality.<sup>[4]</sup> Smart contracts are aptly named as they enable innovative and often complex agreements (contracts) to be created between an arbitrary number of parties. Once signed by all parties, smart contracts cannot be broken. When applicable, smart contracts provide a low-overhead alternative to traditional contracts. Actions defined in the smart contract are enforced by the blockchain and are resolved within minutes. In contrast, traditional contracts are enforced by a legal system that can take months or even years to resolve issues.

## 2.4 Decentralized Autonomous Organizations

Decentralized autonomous organizations (DAOs) are organizations that exist through [smart contracts](#). A company or organization can be expressed as series of legal contracts. A DAO is the smart contract analog of this. Like traditional companies, DAOs make it easier for a large group of people to work together as a single unit. However, a DAO also benefits from the lower overhead afforded with smart contracts.

# 3. New Concepts and Definitions

## 3.1 Microeconomy

One compelling use case of cryptocurrency is creating a "microeconomy". A microeconomy forms when a currency is used only for a limited set of transaction types (a market + a currency). For example, if a unique currency is used to buy and sell used cars, we can say there's a microeconomy for buying and selling used cars. We call the unique currency of a microeconomy a "microcurrency". A microeconomy can be modeled using the equation of exchange from standard monetary economics<sup>[5]</sup>:

$$MV = PT$$

Where M is money supply, V is velocity of money (how many times money changes hands within a time frame), P is the price of goods in the economy's currency, and T is the real value of the sum of all transactions within a time frame. The equation can be rewritten as

$$1/P = T/(MV)$$

1/P corresponds to the purchasing power of a coin. From here we can see that given a fixed velocity and money supply, an increase in the total value of transactions results in an increase in the value of the currency. Speculation is another way in which the value of the currency can increase artificially, but this issue can be mitigated by burning a small percentage of every coin used in a transaction<sup>[5]</sup>.

Because currency value is positively correlated with T, holders of currency in a microeconomy financially benefit if they can increase the real value of transactions within the microeconomy.

## 3.2 Open-Source Microeconomies

When paired with applications that can be expressed as an exchange, [microeconomies](#) offer an opportunity for a new type of profitable organization for open-source applications. When open-source software powers

a microeconomy, stakeholders (microcurrency holders) are incentivized to grow the microeconomy through software contributions. Microcurrency holders want their stake to become more valuable and this happens when the microeconomy grows. Traditional companies and open-source projects share a common pitfall: centralized governance. The central authority may act against the aggregate will of the stakeholders and can be a single point of failure. If a project is controlled by a single corporation and the corporation ceases to exist, stakeholders are at risk of losing money. The next section explains how managing open-source projects with DAOs eliminates these issues.

### 3.3 DAO Managed Open-Source Projects

A DAO Managed Open-Source Project (DMOP) is an [open-source project](#) managed by a [DAO](#). A DMOP has the following components:

- I. An open-source project .
- II. A DAO to manage the project.
- III. A DAO management token (DMT).

The DMT is how users can gain stake in and vote on the direction of the DMOP. The DMT will also be the [microcurrency](#) of any DMOP with a [microeconomy](#). It is important to note that not all DMOPs need to have microeconomies. For example, the Linux open-source project can be managed by a DAO and have an associated DMT (LinuxCoin) for voting purposes (creating the Linux DMOP). There exists no exchange of goods and services in the Linux open-source project which means that the Linux DMOP is not a microeconomy.

DAO Managed Open-Source Projects (DMOPs) reduce the barrier to competition and are self-incentivized to behave in the interest of users. A DMOP enables parties invested in a project's microeconomy to help guide the course of the application. Each stakeholder has a vote proportional to their stake in the microeconomy. The DMOP can outlive any corporation or person that started it because transfer of stake is easy. One concern raised with DAOs is that they are subject to hostile takeovers. A hostile takeover occurs when an attacker manages to acquire enough voting power in a DAO to exert full control over the organization. However, because DMOPs manage open-source projects, the source code can be forked by the stakeholders unhappy with a takeover.

## 4. Example Use Case

Suppose Alice is making a [microeconomy](#) competitor to Uber called Rebu. Rebu will use a microcurrency called RebuCoin. Alice mints 10 million RebuCoin. She holds on to 60% and distributes the remaining 40% through an Initial Coin Offering (ICO). Using the ICO profits, she hires drivers in her target city, and begins working on the platform. Alice holds a lot of RebuCoin, so her big payday comes if she can create a valuable microeconomy around Rebu. The platform is essential to the success of her app, so she [open-sources](#) it allowing more people to contribute. While this opens Alice up to the risk of competitors using her code, ride sharing has a strong network effect that should keep her safe. Any programmer who holds RebuCoin is now incentivized to contribute to the RebuCoin microeconomy. Alice can increase this incentive by structuring Rebu as a [DMOP](#) where everything from software contributions to issuance of new tokens can be managed by stakeholders. Awards can be given to contributors based on the discretion of stakeholders as well. Alice can still hold a majority share of Rebu as her company; but she is now opening her company to others and acts as one of many entities that profits from the success of Rebu.

## 5. Mycro Platform

Mycro will be a platform enabling the creation of [DMOPs](#) powered by Ethereum [smart contracts](#). Mycro itself will be a DMOP with a [DAO](#) called MycroDAO. Mycro will have a [DMT](#) called MycroCoin. Since Mycro is a [microeconomy](#), MycroCoin is also a microcurrency. Mycro DMOP DMTs will follow the ERC20 standard.

### 5.1 DMOP Architecture

A [DMOP](#) on Mycro will have three major components: the Kernel, Action Smart Contracts (ASCs), and Modules. The Kernel will contain the minimal amount of code required to manage a [DAO](#): [DMT](#) bookkeeping, voting and module registration. ASCs will be smart contracts describing actions the DAO can take, such as managing source code. In order for a DAO to execute an ASC, the ASC must first pass a vote. Each token in the DAO will have one vote. A threshold defined by the DAO must be exceeded in order for the vote to pass. If the proposal passes, the ASC will be authorized to invoke modules registered with the DAO. Module registration is done through ASCs. All features of a Mycro DAO not covered in the Kernel, such as managing source code and electing servers, will be handled in Modules.

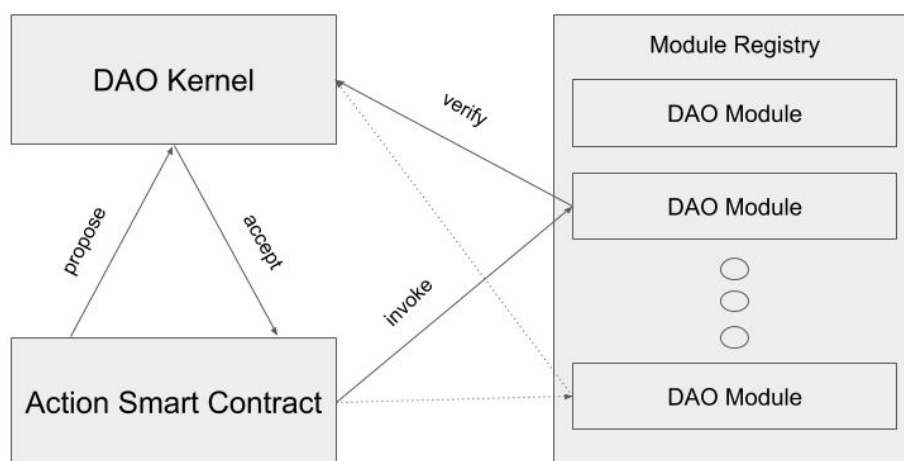


Figure 2. The structure of DMOPs on the Mycro platform.

The Module Registry simplifies versioning of individual modules. DAOs will be able to upgrade a Module in the registry through an ASC. Each Module in the Mycro ecosystem will need to define its own upgrade logic.

Upgrading the DAO Kernel is more complex. A separate smart contract, the Kernel Wrapper, will maintain a pointer to the current DAO Kernel. ASCs can be submitted which change this pointer to upgrade the DAO

kernel. Upgrading the DAO Kernel will require copying the state from the previous kernel's smart contract to the new kernel's smart contract.

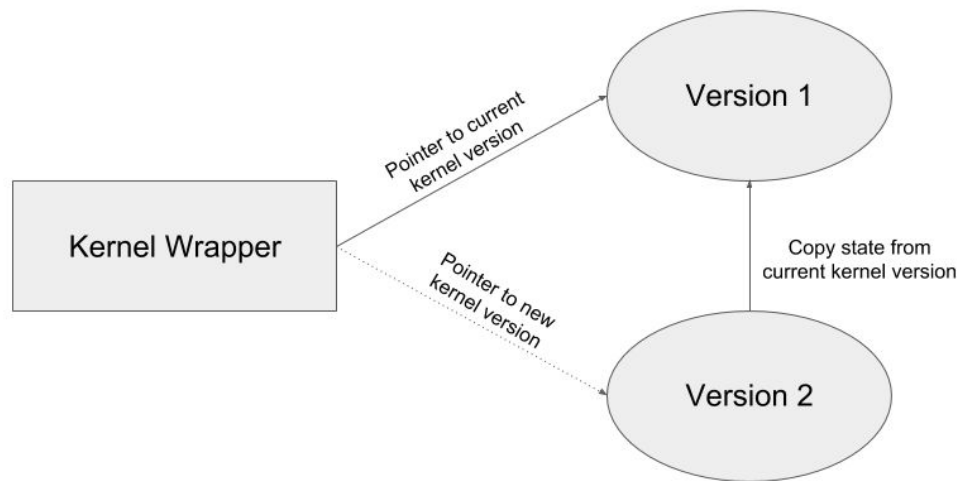


Figure 3. DAO Kernels can be updated by changing a pointer in the Kernel Wrapper.

## 5.2 MycroCoin

Mycro's [DMT](#) will be called MycroCoin. Every [DMOP](#) built on Mycro will be required to give 1% of DMTs minted to the Mycro DAO. The Mycro DAO will hold these coins, and Mycro stakeholders can decide how to best use them through ASCs submitted to the Mycro DAO. Stakeholders may decide to use the DMTs to fund further development of the platform. This ensures that stakeholders in Mycro are incentivized to develop the platform in a way that best benefits all DMOPs on the Mycro platform.

## 6. DAO Actions

A [DMOPs DAO](#) is the primary authority for the project. When an [ASC](#) is approved, it executes [Modules](#). When a Module is executed, it will authenticate and authorize its execution with the DAO. If authorized, the Module will publish a message to a public forum (an event on the Ethereum blockchain). Off-chain software will subscribe to the public forum for these messages and take action when messages of interest are seen.

### 6.1 Managing Source Code

[DMOP](#) source code will live on hosted git such as Github. Users can create proposals to change project source code and submit an [ASC](#) to the project's [DAO](#) requesting the proposal be accepted. If the ASC passes, a software layer owned by Mycro will translate this ASC approval to a Github approval. People who review these source code change proposals are known as code reviewers. Code reviewers will be elected by DMOP stakeholders through ASCs. Only elected code reviewers will be able to approve proposed source code



changes. Code reviewers will be compensated with DMTs. This compensation creates an incentive for quick project development.

## 6.2 Selecting Authorized Servers

Most [DMOPs](#) will require servers to run business logic, connect users, and offer whatever services are needed for their business. A project's [DAO](#) will appoint servers it trusts, such as Amazon EC2 instances, to run the project code. These servers will be elected through [ASCs](#) submitted by the DMOP community.

## 6.3 Centralized Exchange Module

One issue many [DMOPs](#) may face is the limited transaction rate of the Ethereum blockchain. As of 6/9/2018, Ethereum supports approximately 20 transactions per second and has approximately 90 seconds of latency per transaction. For the typical DMOP, this may prove to be an unreasonable wait time. To get around this limitation, many consumer transactions on DMOPs can be handled on a centralized exchange which is owned and maintained by Mycro.

A centralized exchange will improve transaction performance for Mycro DMOPs. This performance boost will be realized because multiple transactions can be preprocessed and batched before being posted to the slower Ethereum blockchain.

A centralized exchange enables Mycro DMOPs to operate with fiat currencies in addition to microcurrencies. Use of microcurrencies can be completely transparent to the end user if the centralized exchange is sufficiently efficient. Payers and payees can use fiat while the centralized exchanges performs exchanges between fiat and microcurrencies.

# 7. DMOP Consensus

[DMOPs](#) require consensus in order to approve and execute [ASCs](#). In order for consensus to be reached and an [ASC](#) to be approved, a percentage of active coins must vote in favor of the ASC. DMOPs will define their own pass threshold. [ASCs](#) can be submitted to change the pass threshold as the DMOP evolves.

## 7.1 Active Coins

Cryptocurrencies can suffer from dead coins. Users may lose access to their wallet and [DMTs](#) may become inaccessible. To ensure that only active coins are considered when [ASCs](#) are being voted on, DMT holders must "activate" their tokens on some regular interval. Each [DMOP](#) will define its own activation interval. Mycro will offer stakeholders a mechanism to activate their coins.

Bad actors may purposefully leave their coins deactivated until a moment of their choosing to activate the coins and attempt a hostile takeover of a DMOP. This can be mitigated by adding a delay before coins are activated. Additionally, because of the open-source nature of DMOPs, the rest of the community can fork the project if they disagree with the actions imposed by the bad actor.

## 7.2 ASC Approval Process

When an [ASC](#) is submitted, it has a limited time to pass a vote (exact time period is determined by the [DMOP](#)). If consensus is not reached by the end of that time period, the vote fails. Otherwise, if the percentage of "yes" votes exceeds the DAOs specified pass threshold, the vote passes and the ASC is

executed. An ASC can be resolved early if enough no or yes votes are reached to guarantee an outcome given the pool of active coins.

### 7.3 Vote Deferral

Organizations often benefit from direct leadership. Not all stakeholders in a [DMOP](#) will be regular voters. Some stakeholders may feel unqualified to vote. Mycro will allow stakeholders to defer their voting power to other stakeholders of the same DMOP. Deferred coins will be activated by the stakeholder receiving the voting rights. This will enable stakeholders to defer their votes to a CEO or other leader who will be able to execute a clear vision of the project without significant delay on [ASCs](#).

## 8. Example Use Cases of Mycro DMOPs

[Mycro DMOPs](#) can be applied as an organizational structure to projects in many different markets. To help illustrate the organizational flexibility of [DMOPs](#) some example use cases for profitable [open-source](#) consumer applications are given below.

- Children's Education Platform
  - A DMOP can be used to incentivize children to study by offering an exchange around school projects.
- Online Game
  - An online multiplayer role playing game can be built on Mycro. The open-source nature of DMOPs allows game assets, levels, and code to be contributed by both fans and professionals. Weekly backups of the game database can be stored on IPFS and, should something go wrong with the DMOP, a new DMOP can be created allowing players to pick-up where they left off with their character.
- Ad Platform
  - A platform for ad exchanges like adsense can be built on Mycro.
- Social Media Site
  - Social media applications have a network effect and will be a prime candidate for a DMOP.
- Classified listings exchange
  - A classifieds listings app like Craigslist will also function well as a DMOP.

## Conclusion

We proposed an organizational structure called a [DMOP](#) (DAO managed open-source project) which enables profitable open-source consumer applications using [microeconomies](#). DMOPs bring more efficiency to [open-source microeconomies](#) and align economic incentives between creators, contributors, consumers and investors. We then introduced [Mycro](#), a platform for building open-source consumer applications using DMOPs.

## References

- [1] <http://www.catb.org/esr/writings/cathedral-bazaar/>
- [2] <https://pdfs.semanticscholar.org/98a6/c566a7e28a48facb664c8689607a52c57a5d.pdf>
- [3] <https://builtbykrit.com/blog/network-effects-defensible-product>
- [4] <https://github.com/ethereum/wiki/wiki/White-Paper>
- [5] <https://vitalik.ca/general/2017/10/17/moe.html>



