

Apocryph.Network Overview

Background

The Apocryph Network was conceptualized and initially developed by members of the Comrade Cooperative - a cooperatively owned entity registered in Sofia, Bulgaria. The entire project is open-sourced under MIT license and is currently developed by an open community of contributors, led by the core team which initially shaped the project.

The need for developing the Apocryph Network in the Comrade Cooperative came after we realized that none of the existing blockchain networks was capable of handling the two projects for which we established our cooperative - Wetonomy for building autonomous organizations, and ScyNet for distributed automated machine learning.

Apocryph is built on top of Perper - an MIT licensed, open-source project developed separately by members of our team. It's a serverless compute platform that runs any language and framework, utilizing industry standard technologies like Kubernetes, KEDA, and Apache Ignite.

Apocryph Network Features

Developer productivity

Our main goal with Apocryph is to make it easy to write blockchain applications. Currently these applications are an exotic genre with a steep learning curve to get into. Apocryph utilizes existing tools, practices, and programming languages to allow ordinary developers from the software industry to build blockchain applications with the same ease and productivity as they build standard business software.

New class of applications

Together with removing the hurdles for the developers to write blockchain applications, the Apocryph Network also introduces new programming models that are not possible with traditional smart-contract based networks. The agents in Apocryph are not just passive transaction validators but can act proactively, even without the need for a user interaction as a trigger. This unlocks a vast range of automation scenarios, where the agents can react to custom events, the actions of other agents or execute on regular intervals. To enable this, the execution fees are not paid directly by the users, but by the agents themselves, which might consequently charge the users in a variety of ways,

including monthly subscriptions and white-listing. Allowing the users to execute transactions free of charge enables yet another major set of use cases like voting and collecting user feedback that might be hindered in networks where all transactions have to be paid directly by the users who trigger them.

Decentralized cloud

Apocryph is designed to become a decentralized cloud that can host all kinds of applications, not just smart contracts. In addition to the public agents which have their state validated by the network validators, developers can also deploy private agents as containers which have private state and are hosted on-demand by individual nodes. Utilizing this mechanism, developers can implement a wide range of functionalities - training Machine Learning models (like what we do with ScyNet), private communication channels, IoT hubs, interoperability with external systems, personal cloud solutions or just pay a node to host your web app.

Unlimited scalability

The Apocryph Network is not a single blockchain, but each group of agents (dApp) runs on a separate chain with separate consensus and a set of validators. These dApp chains run independently and in parallel to one another, thus theoretically there is no limit on the transactions per second that the entire network can process. Moreover, since we build on top of modern technologies like Kubernetes, each node in the Apocryph Network can be a cluster of computers, not just a single computer. This allows the nodes to utilize professional hardware and infrastructure to scale together with the demands of the network.

Fast transactions

With Apocryph we introduce a novel leaderless Byzantine fault tolerance consensus protocol building on top of Avalanche's Snowball which provides blazing fast transaction propagation and finality. To resolve Snowball's cold start problem, we use it together with King of the Hill key mining, where network nodes compete to win a validator slot and participate in the consensus to get rewards. This mining is conceptually different from the Proof of Work block mining typical for the Satoshi consensus family, providing a variety of useful properties and utility to the mining process. Most notably, miners can not predict for which dApp chain their key will fit, thus ensuring random and equal distribution of the validators to the chains.

Interoperable by design

Since it consists of multiple dApp chains that need to communicate with each other without relying on a central bottleneck, the Inter Blockchain Communication (IBC) mechanisms are a major component of Apocryph's architecture. Even though the agents are running in different chains, they can natively communicate through event subscriptions, following the principles of reactive programming. Using similar mechanisms we can also implement an efficient IBC with practically any modern blockchain network with smart contract functionality, including Ethereum. Furthermore, Apocryph utilizes IPFS as a networking layer which provides even higher interoperability and extensibility, in addition to a reliable and mature infrastructure.

Your dApp, your consensus

By having a developer-first approach, Apocryph's architecture is designed to be extremely modular and flexible, allowing multiple extension points across all of its components. Thus, developers have full control over the parameters of the chain where their dApp is running. They can choose the number of validator slots, the complexity params of the slot mining and even decide to mix it with Proof of Stake mechanisms to enhance the token economics of their project. Moreover, dApp chains can further boost their adoption by rewarding their validators with their own token, in addition to the CRYPH token they get as transaction fees.

Token Utility

The CRYPH token is the utility token of the Apocryph Network - an open-source, distributed, and trustless system designed to serve as a consensus network for autonomous agents. The programming code of the agents is executed physically on the anonymous computers that have joined the Apocryph Network and that are running free and open-source software to participate in the network.

As every open-source project needs a community to be successful, the blockchain based projects have the advantage of using token economy models to drive their community adoption. In that regard, below we outline the main use cases and driving forces for the CRYPH token that we hope to enable the Apocryph Network growth and further development.

Execution fees

The Apocryph Network requires the agents executing within its infrastructure to pay the computer nodes of the network for their execution using CRYPH tokens. Each agent will

provide meta information about the time and resource requirements of its methods which will be used for pricing their execution by the network validators. In addition, private agents can offer additional incentives to the network nodes for hosting them.

Mainnet inflation and deflation

While the network doesn't require a central communication hub, it needs a set of registries that we collectively call Agent Zero. Since this is the root dApp, it doesn't have a token balance itself and that's why it mints the tokens needed for its execution, thus creating an inflation of the CRYPH token.

On the other hand, inactive agents should be penalized since validators keep allocating resources for them, even though there are no transactions to generate validator rewards. The network defines a minimum amount of transactions that should be reached over a period of time and if an agent can not reach it, the difference between this threshold and the agent's actual consumption is automatically burned, thus creating a deflation of the CRYPH token. Agents with empty balances will be archived, allowing them to perform a final hibernation operation so they can resume work if additional balance is provided.

Network validators have a mechanism for establishing all network parameters by adjusting them slightly with each validated block. This mechanism is used also for setting the parameters required for the inflation and deflation mechanisms.

Token locking prior mainnet

The CRYPH token will exist initially as an ERC20 contract on the Ethereum network prior to the launch of the Apocryph mainnet. Even though an IBC mechanism with Ethereum will exist when the mainnet is operational, the users will be incentivized to transfer their tokens prior to the launch by locking them on the ERC20 contract so they can be included in Agent Zero's genesis block. For this purpose an interest rate will be offered to the users who decide to lock their tokens early and this will also generate inflation of the CRYPH tokens.

Governance

Even though the Comrade Cooperative was the organization where Apocryph first took form, the network is not supposed to be shaped by a single entity, even if it's member-owned and decentralized. That's why we propose the establishment of three Decentralized Autonomous Organizations (DAOs) which operate entirely on-chain, utilizing first the Ethereum network and later moving to the Apocryph mainnet once it's

operational. These DAOs will gradually receive part of the CRYPH tokens which they can further distribute based on their decision making mechanisms, taking into account the token balances of the DAO members.

We propose the creation of the following DAOs with their respective responsibilities:

- Core Team DAO
 - Consists of the core team members who can vote proportionately to their token balances and accept new members of the DAO.
 - Focused mainly on the development of the Apocryph Network.
 - Allocated tokens will be used primarily to engage core team members by offering them fixed amounts of tokens, vested over a period of time.
 - This DAO is supposed to be the quickest to make a decision and thus it is supposed to be most active in establishing the initial community by offering bounties and hourly rates in tokens for early contributors.
- Comrade Cooperative DAO
 - Consists of the members of the Comrade Cooperative who can accept new members according to the coop's rules.
 - Focused mainly on its existing projects, Wetonomy and ScyNet, which were the reason for the development of Apocryph and will be among its first major dApp use cases.
 - Allocated tokens will be used for bounties on the current and future projects of the cooperative which utilize Apocryph, as well as the further development and popularization of Apocryph itself.
 - Since this DAO is also a legal entity it will be used to host all relationships that require a more traditional framework to operate, such as taking grants and venture capital investments, or funding with grants and investments startup companies that are willing to use Apocryph or any of the coop projects which utilize it.
 - The Comrade Cooperative will also use part of the tokens allocated for it to repay the debt it has accumulated to its contributors and creditors using the automated debt concept introduced by Wetonomy.

- Community DAO
 - Consists of all CRYPH token holders who can vote proportionately to their token balances
 - This DAO will be slow in taking decisions so it's not supposed to provide bounties, but to allocate funding to the other existing DAOs or establish new ones
 - Since the Community DAO will represent the entire community of token holders it will have some special responsibilities including:
 - Approving the IBC contracts that will have the permissions to transfer tokens between the ERC20 contract on Ethereum and the Apocryph mainnet
 - Release and utilize the accumulated funds from the bonding curve contract once it reaches its maximum price and holds it for a reasonable period of time

Token Distribution

Initially the ERC20 contract will distribute one billion CRYPH tokens that will later undergo both inflation and deflation based on the dynamics of the Apocryph mainnet. Half of the tokens will be allocated to a bonding curve contract that will drive the price of the token from 0.03 to 1 DAI. The rest will be distributed within four years among the three DAOs governing the project in five equal installments - one when deploying the contract and one each year after that until the full amount is allocated. As a result, all team rewards will be also vested within four years.

