Decentralized autonomous Community – (Fundamental) Whitepaper [Alpha Release]

Embedding all social networks in one easy to use sandbox framework with lightweight consensus algorithms

Pre-Scriptum:

The term “community” shall be defined in this context, since a Community centralizes a certain group around a specific category per definition.

On the following pages, using the term “Community”, we are talking about the blockchain as governing entity and the possibility of adapting any existing group of interest, business, social network on the same blockchain, with specific, but lightweight rules.

Since each group of interest, (…), acts as an independent entity in the manner of speciality and functionality, the term Community is used for generalizing and simplifying reasons, representing the sum of all possible Communitys interacting with each other in order to reach Consensus and generate value on the same Blockchain.

Introduction:

We live in a world, where every internet user is registered on at least 3 different networks and messengers to connect, inform, research and entertain himself. There is just a small amount of people, who actually use different social networks equally. In reality, the first network to meet your desires will become your main source for informations. Usually, there has to happen something very bad with the network in order for the user to change his preferred network. The results are centralized, controlled and manipulated social networks (introducing the term “social islands”), with no possibilities of gathering relevant informations from other social networks.

What if your best friend is waiting for you on twitter, but you are still on facebook, because you simply use it more often?

Currently, the only solution is to get another account, another app and adapt yourself to the different rules of the different network, just to extend the informations you are currently happy with on your preferred social island.

So this project is not simply another social network, which people wouldn´t get used to anyways. We aim to provide you a tool, a social framework, which helps you extend your social media experience across all available social islands, communitys and messengers, limiting the exposure of confidential user informations, while staying flexible on future innovations and technologies.

The DOMO Project

The DOMO Project consists of two main parts, which are essential for the idea to succeed. The technical realization of the “social framework” and an autonomous community, which acts as representative of the open source product and guarantees immutable and decentralized development of the framework.

This is a non-profit, organizational proposal for anyone who may read this manifesto. I am well aware of the impact of a social network, which embeds all existing social media communications into one open source solution. This is not a proposal for another open source social network like mastodon or friendica, this is a solution for THE social network. By just putting together, what is split across the big companys like facebook, twitter and snapchat, we are not only presenting the all-in-one solution for all social network communications, but also get the opportunity to decide on what information we share with each social network. This is the beginning of a war with centralized social networks, which begins on their playgrounds and can only be fought with a ever growing community supporting the idea of a non-profit “OPEN SOURCE SOCIAL NETWORK EMBEDDING THE WORLD”

1.0 The open-source Community

Before raising money and spending ressources on features and functionalities, which may sound great from a technical point of view, it is way more important to make the future users part of the development. There are lots of great open-source projects with the potential of disrupting old structures, but mostly fail adoption by anyone not familiar with the code. Since developing functional code takes a lot of effort and knowledge, it may be too much to ask for from the average users point of view.

1.1 Educating the Crowd

In order to achieve the state of a contributing user, one can not expect every user to bring knowledge, skill and effort to the Community. The Community itself has to provide necessary tools and content for every user to educate themselves to a certain degree of general consensus in terms of functionality, usability and the importance of liberty of voice within the Community. The user needs to understand, that an initial effort has to be done, which may vary greatly, based on the Community he is joining or task he is trying to achieve.

1.2 User oriented development

Since this project is about connecting various types of users from various networks with similar, but still different functions and priorities, it would be naïve to think, that defined functionalities from the very beginning would fit every possible group of users. By introducing a different approach, we call it the “broken product” approach, users will have to deal with imperfect functionalities to trigger “miserable moments”. As this approach is anything else than intuitive or user friendly, a loss of a certain amount of users is unavoidable in the early stages of the project. However early adopters, who contribute directly to the code or indirectly through trial and error, research or marketing will be rewarded and ranked from the Community in future development stages. This kind of approach is an experiment and has yet to be approved and may be replaced by other development techniques. The aim of the “broken product” approach is to engage early adopters in the developing process, while providing developers a helping hand and instant feedback on new implementations or ideas.

Everyone united by the same vision, contributing with his individual expertise.

1.3 Open-source development from the very beginning

This project is about developing a product, which fits the needs of many, if not all social media users. Deciding to pass the lead to the Community to put all necessary parts together and add features, which a fixed development team might not see as viable, is the only way this idea might work out. The reasons are simple:

- No singe point of failure, as the code still exists, even in case of organisational or abusive missbehaviour
- Implementing as many features as possible, it´s pointless to hide the code from potential masterminds providing code and functionalities not imagined before
- Different opinions are free to fork from each other, instead of leaving the development
- No single entity to be liable for potential regulatory attempts by centralized social networks

In fact, the goal is to distribute the code as much as possible to reach a truly immutable development

1.4 Adapting available technologies

There are many open-source projects out there providing very impressive Community functionalities, which thankfully don´t have to be developed from scratch anymore. The main goal of the development process is to gather as much existing and relevant code, to minimize workload of unique lines of code and stay open minded for new technologies.

2.0 The Blockchain consensus

The basic understanding of delegated proof of stake and masternodes as blockchain backbone is essential for the introduction of the following concept of the social media blockchain.

As you know, the dPoS concept basically consists of candidates, elected by the network, competing with each other for the right to produce blocks on behalf of the network. This competition will be the main driving factor to guarantee proper behaviour and development towards a greater good of the network. Similar to mastodon, the blockchain will consist of different servers (we will call them masternodes), which will implement the same consensus in order to participate in the network. Each masternode owner will be the administrator of the server and will become a candidate for the dPoS algorithm.

2.1 Different Communitys, different rules

Although all masternodes will implement the same code in order to participate in the network, it is up to the administrator and the users, joining the masternode, to setup rules, moderation and appropriate content in terms of the rules defined by the masternode owner/administrator. The blockchain will only provide the networking layer for standardized communications between the servers, no censoring or moderating functionalities.

2.2 User data storage

Since every user is free to switch to any server and thus every Community, a second function is applied to the blockchain. In order to avoid multiple accounts and user data collection, which would result in lack of adoption, sensitive user data have to be stored on the blockchain.

The idea of storing user credentials on the blockchain is still very young and possible implementations still have to be researched, although the basic idea is to store new user credentials in newly generated blocks in addition to the transactions made on the blockchain.

2.3 dPoS candidates

Masternode owners and thus the administrators of the community servers are eligible candidates for one of the limited free slots of a block producer. The block producer is choosen by the users within the network. Ideally, the masternodes providing the best content and new functionalities for the users and thus attracting the biggest userbase, will receive most votes and become participants in the dPoS algorithm. It is up to the candidates how the staked coins will be used or distributed, however with growing user registrations the servers need to scale and pay for more server ressources. The process of voting has yet to be researched on available technologies to minimize manipulation and abusive behaviour.

3.0 The Social Network Framework

As introduced, all servers will be using the framework as basis to deliver valuable content for and from the community. It´s the main goal of the project to develop a working product, which can be promoted as content management system for social media, run by the users and maintained by the server owners.

3.1 Connecting social networks

The main feature of the framework is the ability to integrate any social network or community. Every user has his own favourites, which can be embedded with all functionalities within the framework. The goal is to present the user a visualized and simple overview of all connected accounts with customizing features. News feeds, charts, chat rooms, private channels, facebook, twitter, Instagram, telegramm, discord, slack everything at one place. Administrators will decide which networks will be connectable and every user will have to run through an initial, standardized configuration process, in order to be registered on the blockchain. Since many api integrations are intentionally limited by the providers of the networks, we are currently investigating possible alternatives to avoid future bans or bottlenecks with account integrations, any help is appreciated!

The framework itself is very lightweight, as user credentials are stored on the blockchain and processing of user content is managed by the masternode entitys.

3.2 Intercommunication features

In addition to visualizing all accounts at one place, another feature of the framework is to chat across different social networks. Users within the same network or already using the framework do have this ability by default, however we are also researching on anonymous entitys to enable user chat across different networks with intermediaries or temporary shared accounts.

3.3 Rest API and Browser features

Depending on the backend configuration of the server, we will also add browsing functionalities to our features collection. Realized via browser extensions for firefox, chrome and Tor users will be able to display content within the framework which doesn´t provide api access. As this feature needs computing ressources from the client side, this will be a very instable and difficult to implement feature, which relies on the expertise of the server administrator. In addition, we will also develop a powerful rest APi to provide developers with as many tools as possible to implement new connections to the framework.

Additional Points (not categorized yet):

User login/registration

The most important and security critical part is to register new users in the network and broadcast the data across all network participants in a secure and scalable manner. In order to avoid network congestion, users will have to pay transaction fees for initial registrations and account changes. The elected dPoS candidates will provide the necessary processing ressources to make these changes on the network. Users will get a public key, private key, wallet and reserved username in exchange for the initial fee. It is critical for the user to only use the username in the public, as the disclosure of more details may result in loosing anonymity or access to the account.

UDNS – User domain name server

Since users will always have the ability to connect with each other, independ of the server they are connected with, a unified DNS will be created. Depending on the privacy settings of the user, the username, registering server and connected social media accounts will be stored and publicly available. The main purpose of the UDNS is to create a public profile of each user to distinguish and locate users spread across multiple servers. Users will also have the ability to sign messages with a unique hash pointing to the UDNS entry to avoid impersonators and perpetrators. Another goal of the UDNS is to provide a standardized verification process for social media accounts without providing personal informations.

Friendlist and Followers

Organizing and visualizing friends and followers from different networks is a very difficult task. While some social networks provide api functionalities to access followers and friends, we cannot rely on those functionalities, especially on privacy oriented networks. There are basically three possible scenarios on storing these informations:

- Public: All followers and connected users will be stored in a publicly accessible library
- Local: Users will have to store a copy of the friendslists locally or on a private cloud, which can be accessed by the servers running the framework (on demand)
- Central: Every server will provide a database to store lists. When changing the server, the user will have to initiate a migration of the informations

These scenarios can also be combined, depending on the configurations of the administrator and user preferences. Storing lists locally opens up whole new possibilities, where users don´t have to rely on main accounts in order to stay connected with their friends and can provide a signed message to be distinguished from scammers.