WHITE PAPER

Liberdyne messenger

Liberdyne is a P2P secure messenger powered by the **Dynemix** blockchain platform, which is described in a separate paper. **Liberdyne** and **Dynemix** share the account base and transport protocols, which greatly improve the messenger's capabilities.

Liberdyne is designed to compete with popular centralized messaging solutions on equal footing while at the same time offering users a superior level of security and privacy.

Liberdyne is designed to scale up to a size comparable to that of popular centralized solutions.

Liberdyne is endowed with a powerful marketing feature: by being a channel for the distribution of the helicopter coins of the **Dynemix** blockchain, **Liberdyne** becomes an application that grants users financial rewards for the usage of the app.

The following white paper contains an informal description of the project. The provided specifications are not final and are subject to amendments, if necessary, up to the mainnet launch. Any updates to this white paper can be made without prior notice.

The following white paper does not contain any coin purchase offerings or any other information on the initial coin distribution.

www.liberdyne.com

Contents

I. Background

1. Introduction

When we came up with the idea of creating the first widely adopted cryptocurrency, we quickly realized that even if we provided all the required features on the blockchain layer, we would still face the issue of the entry threshold, which would severely obstruct the expansion of the platform.

To solve the issue, we needed to wrap the platform into a product that was convenient and familiar to the overwhelming majority of common users. A p2p messenger was chosen out of possible options as a solution that checked all the boxes and was less resource-demanding to build and maintain.

During the development, it became clear that not only the messenger could provide required solutions for a number of the blockchain's issues, but the underlying blockchain technology also allowed to introduce groundbreaking features to the IM market, thus making Liberdyne a competitive product on its own.

2. Advantages of Integrating P2P Messenger with Blockchain

Today, the most popular IM applications are centralized messengers, which use servers to process most of the user's data (WhatsApp, WeChat, Telegram etc.).

After the revelations of Edward Snowden, the problem of insufficient user privacy has begun to be actively discussed. To provide a solution to this problem, the developers of messengers have implemented end2end encryption.

This solves the issue of data privacy only partially, however. The main problem of popular messengers lies in their centralized architecture and administration, which engenders several problems concerning privacy:

a) **End2end encryption does not necessarily prevent third parties from reading encrypted conversations.**

Firstly, in most cases we cannot be sure that developers do not have an opportunity to obtain private keys. Most messengers' source code is not publicly available, and those who claim to be fully open source and free might turn out not to be so open and free. Considering that client apps transfer a lot of metadata to servers, nothing stops developers from secretly obtaining private keys.

Secondly, even if all the transferred data is securely encrypted and no keys are sent to the developers, it is still possible to remotely read the whole chat history stored on the device by different means. There were multiple reports from users who claimed to be getting targeted ads on Facebook regarding keywords that they typed only in WhatsApp chats. Data sharing between apps can be technically achieved by the usage of shared containers and other technologies that allow to circumvent encryption.

b) **End2end encryption is not necessarily applied to all user interactions.**

For example, in the Telegram messenger, end2end encryption is applied only to specially created secret chats, while other interactions stay unencrypted (only client-server encryption is applied). Most users simply do not investigate these peculiarities and use the app in the default mode. Since all user data, including transferred messages, is stored permanently on servers, nothing prevents developers from using it at their discretion.

Other messengers persistently suggest creating a backup of all data, which is then stored in the cloud, where it can be accessed by third parties.

c) **Even with end2end encryption, a lot of metadata still can be stored on servers.**

Analysis of the metadata can provide a lot of information about users: much more, in fact, than meets the eye.

The problem of metadata collection and analysis is also recognized on the highest levels. Here is an extract from the report of the UN High Commissioner for Human Rights (A/HRC/27/37):

«The aggregation of information commonly referred to as "metadata" may give an insight into an individual's behavior, social relationships, private preferences and identity that go beyond even that conveyed by accessing the content of a private communication. As the European Union Court of Justice recently observed, communications metadata "taken as a whole may allow very precise conclusions to be drawn concerning the private lives of the persons whose data has been retained." Recognition of this evolution has prompted initiatives to reform existing policies and practices to ensure stronger protection of privacy.»

The first two issues can be solved by making a fully free and open source software. Some developers are already following this path (for example, the Signal messenger), so even though the most popular solutions on the market do not fit this requirement, we cannot say that these issues are completely unresolved.

The metadata issue, however, is a problem inherent in any centralized solution. The only way to solve it is to use peer-to-peer architecture.

Complete P2P design offers several benefits to a messaging service:

- Client and server functions are merged in one app, making it easier to audit the code and ensure that there are no backdoors.
- Users do not have to trust any particular party, since the whole system is run by users themselves.
- The developers have no extended access to system operations or user data; therefore, there is no sense in applying pressure on the developers by any parties who wish to influence the system or obtain user data.
- Using an appropriate system of relays, it is possible to hide user interactions, making the app relatively anonymous and secure.

This means that if we want to take the next major step in instant messaging evolution, we should develop a decentralized messenger working via a peer-to-peer protocol with an open source code.

To date, there is no truly fast, stable and convenient P2P messenger. All products are currently in a very rudimentary state and have unclear prospects, since there are some issues that are very hard to solve:

a) **There are numerous popular solutions available, and the market has long been formed.**

If we come up with just another messenger, how do we motivate people to download it if it is not used by a large number of their contacts? It is quite hard to promote a new product in such a crowded market even though we can offer several major improvements. To obtain a large userbase, we need both to conduct a serious marketing campaign and to introduce some killer features capable of attracting the attention of a major audience.

By merging a messenger with a cryptocurrency, we can introduce several new stunning features that were not available to developers earlier, thus greatly improving its marketing potential.

For a more detailed description of our marketing strategy and Liberdyne's killer features, please refer to the Marketing chapter.

b) **People are not willing to share their resources for free.**

During a system's infancy stage, P2P messengers do not consume any noticeable amount of resources, so most users simply pay no attention to this matter. With the growth of the userbase and the increasing percentage of mobile clients, however, full nodes will begin to consume more resources, and at some point users who are not willing to constantly share their bandwidth for free will start shutting down their full nodes, which in turn will shift the load to the remaining users, thereby increasing their overhead. This will lead to a chain reaction that creates the risk of bringing the system back to something resembling client-server architecture, where developers' servers will be the only full nodes left.

To overcome this stage, we need to offer users incentives to keep running full nodes, and that is when a cryptocurrency comes into play. Since we are merging two technologies in one product, a full node of the messenger is at the same time a full node of the blockchain. This allows rewards to be issued for supporting both components of the system, thus completely solving the problem of user motivation.

It is worth mentioning that a number of developers have already tried to develop a messenger based on a cryptocurrency platform, but to date, in our opinion, no one has been able to present a complete, high-quality product. One of the main problems of the unsuccessful attempts is that messengers are built on the basis of already working blockchain platforms, which are not optimized for such applications. To create an adequate product, both components must be developed in coordination.

II. The Liberdyne Messenger

1. Decentralized Architecture

Liberdyne is a decentralized P2P messenger with an emphasis on security and privacy.

Liberdyne uses the account base and transport protocols of the Dynemix blockchain system, which greatly improve the messenger's capabilities.

Liberdyne is designed to withstand a significant load created by a userbase of a size comparable to that of popular centralized solutions.

Liberdyne is endowed with a powerful marketing feature: by being a channel for the distribution of the helicopter coins of the Dynemix blockchain, Liberdyne becomes an application that grants users financial rewards for the usage of the app.

1) Liberdyne and Dynemix blockchain

Unlike the typical approach, we did not aim to use technology for the sake of technology. Blockchain is used only to the extent that helps build a better system and improve the user experience. Technically, we cannot call Liberdyne a blockchain messenger, considering the actual degree of blockchain involvement in the messenger's architecture.

There are several existing blockchain messenger projects, but blockchain technology itself is not optimal for storing and transferring data of low value, such as ordinary correspondence. Any messenger that stores transferred data directly in the blockchain is a very niche product that cannot compete with conventional client-server apps from the point of view of efficiency and scalability.

Liberdyne is closer to more common P2P messengers (e.g. Tox) in its design, but the Dynemix blockchain still plays an important role in providing a solution to several problems emerging from P2P architecture.

a) **Accounts**

Secure communication requires the ability to reliably identify interlocutors, which is why in a decentralized peering system, management of the account base can become a serious challenge. Blockchain technology offers an optimal solution providing secure decentralized consistent key storage, which is highly resistant to MITM attacks.

b) **Rewards for system support**

When it comes to system support, most P2P systems rely on altruistic behavior by the participants, who share hardware resources for the sake of community. This model may work to a certain extent, but in the case of a significant hardware load, altruistic intentions may not suffice. Dynemix helps Liberdyne build a fair system of economic incentives, which ensures smooth operation.

c) **Dynemix wallet**

An embedded cryptowallet is a quite standard feature of any blockchain messenger and often the only reason a messenger is actually called as such. Liberdyne allows payments to be made from Dynemix accounts or unregistered key pairs, which can be added if the user wishes to engage multiple wallets.

d) **Transport protocols**

Liberdyne uses the low-level transport protocols of Dynemix. This helps build the mentioned reward distribution system with the help of blockchain transactions and also provides for traffic obfuscation for user-generated content, making meta-data analysis significantly more difficult.

2. Accounts in Liberdyne

Liberdyne employs Dynemix accounts for its operation. To create an account in Dynemix, a master key should be generated that is then secretly stored by the user and engaged to log in. The Dynemix protocol has no requirements for key-generation methods; the only thing that matters is format matching.

There can be different approaches to key generation, however, which significantly affects the user experience. Since Liberdyne is a complete product, it should feature particular methods to meet the needs of most users.

During the account-creation process, users will be offered three different methods of key generation, each of which will be more suitable for certain types of system usage.

1) Random generation

For users who will store significant amounts of dynes in the account or transfer important information

If the user chooses this option, a master key is generated randomly. This approach makes the key uncrackable, but requires storing the key and entering it on each logon. Since a 256-bit randomly generated key is unmemorizable, users will have to engage some kind of a storage (from a simple sheet of paper to a special hardware wallet). Although a random generated key is immune to brute-force attacks, the wrong choice of storage may compromise the key itself and cause the loss of the account.

This method is standard for most blockchain systems, but since we are creating a product for mass audience, many common users may find this approach inconvenient.

2) Password hashing

Security: average | Convinience: high | Privacy: high

For users who will store small numbers of dynes in their accounts or use the application for everyday communication

To improve the user experience, another approach will be offered – hashing a master key from a user-generated password. Though we will use a secure key derivation function (Argon2 or any other stronger solution that may be developed in future), this method is still vulnerable to brute-force attacks if the password set by the user is not strong enough.

The positive side is that the user will be able to remember the password (or the passphrase) and interact with the system in a much more convenient way. Since this method is familiar to most users, we believe that it will prevail.

3) Custodial account management

Security: high | Convinience: high | Privacy: low

For users who do not care about privacy and prefer the most convenient and familiar option

In currently available centralized solutions, owners of the app administrate user accounts. This provides the highest level of convenience and an opportunity to restore the account at the cost of a possible account suspension and/or access of users' private data by the administrator.

Practice shows that many users accept this trade-off, and we should consider this option despite that it renders some of the breakthrough features of Liberdyne inefficient.

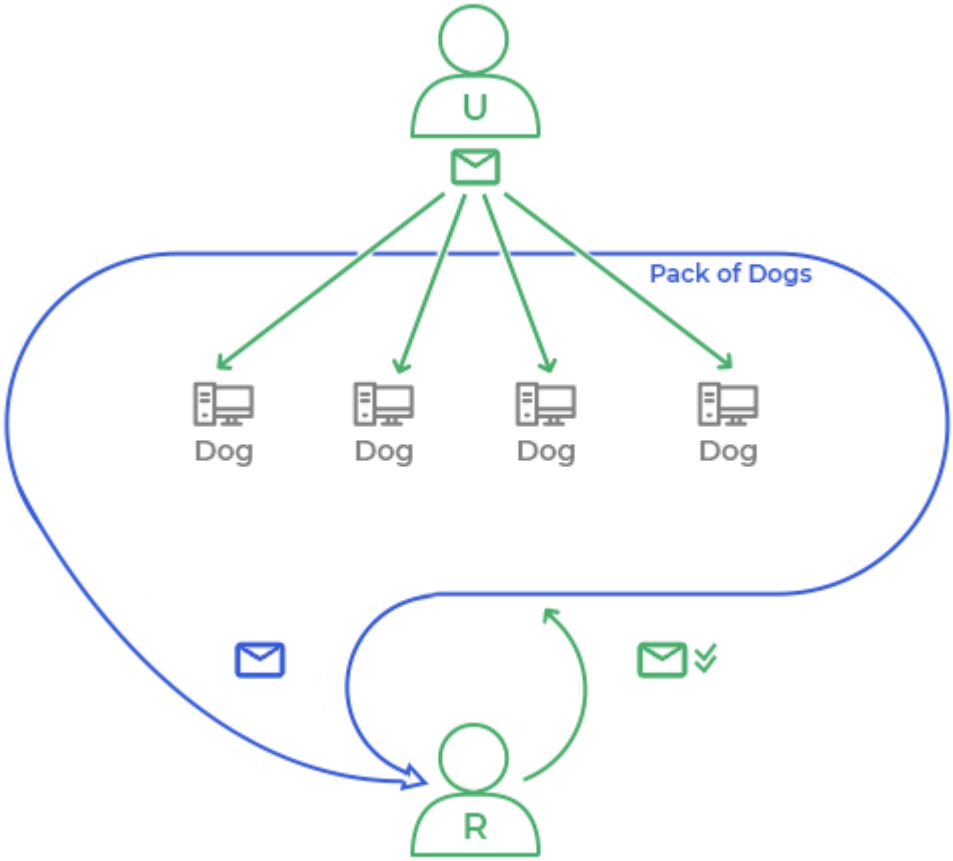
3. Delivery to Offline Guys (DOG)

P2P architecture raises a serious problem – delivery of content when direct P2P interaction is impossible. If the recipient is offline, the message cannot be delivered. The sender's client may try to connect intermittently and deliver the message when the recipient reconnects to the system, but what if the sender himself goes offline?

In centralized messengers, servers are used as delivery relays, so the message is stored on a server, which obviously never goes offline until the recipient connects to the network, and thus the problem is solved. In a P2P network, peers perform the relay functions, which is why such a solution will not work.

To solve this problem, we implemented a special protocol of delivery to offline guys – DOG.

1) Delivery protocol



If user U tries to send a message to recipient R but fails to connect to the node of R, he finds several DOG nodes (or simply *dogs*) and commits the delivery to them, thus forming a *pack*. U informs each dog about all other members of the pack so that they will be able to maintain the size of the pack.

If U goes offline, the replicated messages will still be stored on the packs' nodes, each of which will intermittently try to deliver the message and check whether the entire pack is still online. If one or more nodes of the pack goes offline, the remaining dogs will find substituting nodes, thus keeping the pack size constant.

When R finally connects to the network, one of the dogs delivers the message. The others will attempt the delivery as well, but R will respond that he has already received the message.

After all dogs get delivery confirmation, the pack is considered disbanded.

It is worth noting that DOG is not strictly a Liberdyne feature but a low-level transport protocol of the Dynemix system, which means that any type of service data can be delivered via DOG if needed.

2) Rewards for dogs

Dogs will be rewarded with newly issued coins. This is a feature that makes Dynemix significantly different from any other blockchain system. While in other cryptocurrencies all issued coins are distributed among the block producers, Dynemix distributes rewards among nodes that perform different functions, thus creating an absolutely new economic framework.

For a more detailed description of the role of the DOG reward system in the economy of Dynemix, please refer to the economics paper.

Rewards will be distributed by minters as the result of pseudorandom spot checks.

3) DOG as a solution to the reward distribution issue

A very important feature of DOG is that, unlike all other system support functions in Dynemix, DOG can be performed by even light clients (mobile devices). It serves the following purposes:

a) **Removing load from full nodes**

Full nodes perform various functions in the context of Dynemix/Liberdyne support. Unfortunately, due to the hardware and bandwidth limitations, we can delegate almost none of them to light nodes.

DOG, however, is a function that does not require powerful hardware or low latency. Considering that mobile devices will presumably form the majority of the Dynemix nodes, the load required by DOG will be distributed enough to significantly reduce the resource consumption of each dog. This also means that the load will not increase with the growth of the userbase, since the number of dogs will scale proportionally.

b) **Achieving greater reward distribution**

In the beginning of the Dynemix white paper, we stated that one of our goals on the way to creation of a next-generation decentralized payment system was the following:

- **Newly issued coins should be distributed among as wide an audience as possible.**

The DOG reward system helps us achieve this goal. Since DOG is easily performed by any type of device, it fits the concept perfectly. If we make DOG rewards a large portion of the issued coins, this may allow the distribution of the issued dynes among almost all nodes of the network.

4. Secure Anonymous Tunneling Across Network (SATAN)

Direct P2P connections between users cannot provide full anonymity. Even if registration does not require any personal user data and all transferred messages are end2end encrypted, it is still possible to acquire the IP address of the sender and the recipient of any message and keep a record of user interactions, which in itself provides a lot of information.

This problem is also relevant for centralized messengers, since all connections are relayed through servers, which make it extremely easy for a person who has access to the servers to gather the information and use it for any purpose, including providing it to third parties.

To ensure complete communication safety, we must apply a solution to hide the interaction between the users.

To accomplish this, we have implemented a secure anonymous tunneling across the network (SATAN) protocol that allows users to send and receive messages through a chain of relays, thus completely hiding their IP addresses behind the relay nodes. Furthermore, the relay nodes cannot know for sure which particular node is the sender or the recipient of the message.

The only fact that can be seen by a possible intruder is that the recipient's user account received some sort of message from an unknown account. Essentially, SATAN uses the principles of onion routing.

If the user wishes to hide all his activity, he may pay a fee to the system and turn on the SATAN protocol.

When node N run by user U is connected to the network, it chooses two chains of 3–6 relay nodes for incoming and outgoing messages.

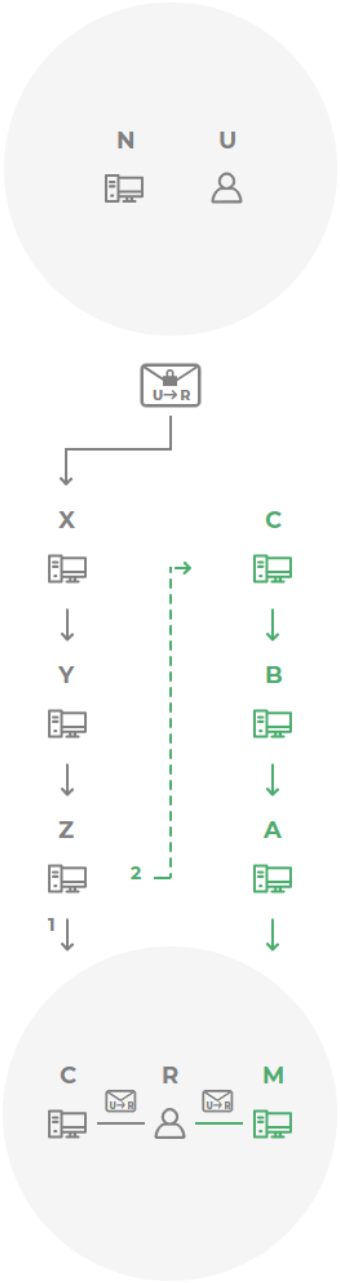
1) Sending a message

a) When user U on node N wants to send a message to recipient account R, he gets information about which node is associated with user R from DHT. Say it is node C. U chooses three relay nodes for an output chain (X, Y, Z), sequentially encrypts the message, along with the metadata about the route, with the public keys of user R and of nodes C, Z, Y, X, and sends the encrypted data to node X.

b) X receives the message and decrypts it with its private key. It sees that the message contains some encrypted data and instructions to send the data to node Y, which it does.

c) Nodes Y and Z do the same as X.

d) C receives the message from Z and decrypts it with its private key. It learns that this is a message for account R. Since the owner of the node has the private key of account R, he decrypts the message and reads it.



None of the relaying nodes knows which particular nodes and which user accounts are the sender or the recipient of the message.

The owner of X may suppose that node N is the sender node; therefore, user U is the sender of the message, but it may also be just a relaying node, like X itself.

The owner of Z may suppose that C is the recipient, but it may also be just a relaying node, like Z itself.

Since nobody knows how many nodes U has actually chosen as relays, even if the adversary runs all nodes in the chain (X, Y, Z), he still cannot be sure that U is the sender of the message.

2) Receiving a message

If user R, along with user U, is using SATAN, then after exiting U's output chain, a message does not go directly to the node of R but enters his input chain instead.

In this case, R puts into DHT the information that he can be found not on the node on which he actually is (M) but on the first node of his input chain (C). Then, using his output chain, user R sends the following messages:

a) He tells node C that he can be found on B.

b) He tells node B that he can be found on A.

c) He tells node A that he can be found on M.

The message travels through his input chain and is finally delivered to node M, where R decrypts it with his private key.

Relay nodes from R's input chain know only that they are delivering some data to R, but they do not know the sender of the message or what node R is actually on. Node A may suppose that node M is the recipient node, but it may also be just a relaying node, like A itself.

Therefore, with the help of the SATAN protocol, any user can stay fully anonymous regardless of whether his interlocutor also uses the SATAN protocol.

3) Rewards for relays

Since SATAN is a paid feature, relay nodes will be rewarded for their effort.

Whenever a user wishes to enable SATAN, he sends an arbitrary number of dynes to the special system account. After minters add this transaction to a block, they update the user state in blockchain, setting a period during which the subscription will be activated.

The special system account accumulates all fees, and each block a portion of the accumulated dynes is distributed by the minters as the result of pseudorandom spot checks in a manner similar to DOG reward distribution, but unlike DOG, SATAN rewards are not coin-based and are withdrawn from the account mentioned.

5. Modes of Operation: Anonymous or Social

Liberdyne was conceived as the ultimate means of ensuring the right to freedom of speech and of unlocking the potential of decentralized technologies, and that is where its name originates.

One of the key goals of the project is to provide the possibility of fully private system usage and to secure personal data. The problem is that many features that consumers are used to are not compatible with such standards of security and privacy, so it is necessary to sacrifice one for the sake of the other.

At the same time, we understand that the majority of users do not care much about privacy and may be ready to trade it for better functionality. This means that we either have to compromise or offer two separate solutions – one for better privacy and one for a better user experience.

We have chosen to follow the second path and develop an application that can function as a tool for secure and anonymous communication or a social platform, depending on the user's needs.

A very important feature is that users of both modes will stay in the same single ecosystem and will be able to interact without any difficulties. Anonymous users can communicate with socialized users without any threat to their privacy.

As the features and preferences of these modes differ drastically, instead of just offering to let users tweak everything themselves, which can turn out to be a challenging task, Liberdyne will feature a simple switch that activates one of the modes.

When a user runs the app for the first time, he is offered the choice of either anonymous mode or social mode. After the user makes a choice, the app is automatically configured, and the interface is painted with the corresponding color to let the user know which mode is active at any given time.

1) Private mode (blue)

A dark blue interface will indicate that the app is running in the private mode, which provides maximum security and privacy.

Once the user tries to access a feature that can threaten his privacy, the app will show a warning. If the user proceeds with the feature, the color will switch to green, indicating that the user has quit the private mode. If the user wishes to return to the private mode, he can activate the switch from the preferences pane, which will instantly reconfigure the app.

Certain features, however, can deanonymize the user irreversibly. For example, if the user binds a phone number with the account, this information may be instantly collected by malicious actors, and removing the binding will not help him obtain privacy again. In case such risky features are activated, Liberdyne will block switching to the private mode to show the user that he cannot be completely safe further on.

The private mode will feature the following specs:

a) **SATAN is always on.**

Considering that the SATAN protocol is a paid feature, the private mode will consume dynes, and users with zero balance will not be able to communicate with others in this mode. Transactions can still be issued free of charge, however (using the same chain of relays and keeping the user anonymous), so payment features will be available at any time.

b) **Features that require a streaming P2P connection are disabled.**

This applies to features that require data streaming, such as voice and video calls or any other features of this type that can be implemented in the future. Using relays to retranslate such types of data does not make sense, since the resulting delays may turn these features unusable.

c) **System support except relaying is disabled.**

System support functions require low latency, a good internet connection and stable transport routes. Providing system support while using a chain of random relays is technically almost impossible. In addition, it will put an unreasonably excessive load on relay nodes.

Performing relay functions in this mode, however, is not only acceptable but also enhances privacy, as it creates additional noise in the user's traffic. Besides, turning the relay functions off makes the user more vulnerable to certain deanonymizing attacks, which is why performing relay functions in this mode is recommended.

d) **Binding a phone number to the account is disabled.**

Phone-number binding and using custodial account management in general is a feature that poses the highest threat to privacy. We actually do not like the idea of implementing this feature at all, but we understand that many users would feel uncomfortable without it, and rejecting this feature can greatly reduce the potential of Liberdyne's distribution.

2) Social mode (green)

A green interface will indicate that the app is running in the social mode, which provides maximum functionality.

The social mode is designed to provide the level of user experience close to that of existing centralized solutions. Though it does not allow the full use of the potential of decentralized technologies, if we count on the mass adoption of Liberdyne and Dynemix, it will be necessary to offer familiar experience for the general audience.

Using Liberdyne in the social mode does not provide the same level of privacy as in the private mode, but it may still seem sufficient for most users. All transferred data is end2end encrypted, and traffic obfuscation methods are applied in both modes, which means that going social does not make a user highly vulnerable to data analysis attempts, although metadata collection becomes less resource-demanding.

In addition, if the user performs relay functions, it creates a lot of traffic noise, making it extremely hard to analyze his connections. Finally, since Liberdyne employ standard Dynemix transport protocols, service data flows, which are hardly distinguishable from user-generated content transmissions, also increase privacy in the same manner.

6. System Support Tweaking

Liberdyne will be the default client for a Dynemix node and will include everything required to use the system and perform the full node functions. No special minting software or wallets will be

needed (although third-party developers may come up with all kinds of specific software for different purposes).

Liberdyne versions for mobile devices (Android, iOS) will fully support only one service function – DOG – and may have limited support for relaying in the case of meeting technical requirements, while desktop versions will support the full capabilities of the system, including:

- Minting;
- Relaying;
- DOG;
- DCS.

This approach allows the improvement of the user experience and the creation of an interesting marketing concept.

According to the conventional approach, which is used in most cryptocurrencies, the user should download a light client (a wallet) if he wishes to use the system only for payments, and a full node client if he wishes to participate in system support.

Liberdyne, on the other hand, is a complete solution in itself. The user does not need to think of choices – he just downloads the product and starts using it while everything is set up automatically according to the user's hardware capabilities.

At a first glance this does not seem to be much of a deal, but from the user's point of view the system starts to look totally different – it looks as if the messenger is paying users simply for using it.

The app will be configured so that users will not feel any significant inconvenience and at the same time, they will receive rewards for background system support. This may serve as a powerful marketing feature that will help us attract audience.

For a more detailed description of our marketing strategy and the role of rewards for system support in it, please refer to the Marketing chapter.

We will enable the support functions to be tweaked through Liberdyne's preferences. If the user wishes to share more resources to earn more rewards, he will be able to simply drag the slider and the app will be instantly reconfigured. It is especially convenient for mobile clients since the user will be able to precisely specify the amount of traffic and battery life that he is ready to share.

The user will be also able to prioritize certain functions or disable those in which he is not interested. If the user engages in minting, the app will automatically stop all other activities during the block-creation process to ensure its smooth operation.

7. Centralized Services

Despite our goal to create a completely decentralized system, there are certain services that have a centralized nature and cannot be decentralized.

Since those services are not crucial for the system and do not affect the security and privacy of those who do not use them, we decided to implement them to improve the user experience.

1) Verified namespace

To verify accounts, a centralized trusted authority is needed, because verification requires establishing off-chain facts (like ownership of an account by a specific individual or legal entity).

In the early stages, this service will be administrated by the developers, but later we may consider transferring these authorities to a specially registered non-profit organization.

2) Custodial account management

Phone number verification requires the confirmation of ownership of the phone number by the account holder. Keeping the entire phone number base in the public domain is also not a good idea, which is why a centralized solution is required.

Custodial account management allows to recover the account in case of a password/key loss, which comes at a cost of the opportunity to suspend the account at the discretion of the administrator.

We do not like the idea of implementing this service since it can irreversibly deanonymize any user and creates the opportunities for third-party censorship but at the same time, we understand that many people are already used to such functionality, and we must take into account the interests of the majority.

8. Decentralized Cloud Storage (DCS)

Certain functions of the messenger will require some sort of external storage of the user's data. Since we are making a decentralized system, we should not fully rely on third-party centralized solutions, and that is why a built-in decentralized storage protocol is required.

DCS will be able to provide users with confidence that their sensitive data is stored persistently and that access to the data cannot be arbitrarily restricted.

Liberdyne requires external storage for the following types of data:

a) **Contact lists**

Centralized messengers either store user contact lists on servers (Telegram) or use contacts stored on the user device (WhatsApp and others), which in turn are backed up in the clouds of OS developers and can be restored from user's Google/Apple account.

Keeping such a type of information in blockchain is not a good idea, since it will greatly increase the weight of the account states, which is why we can either offer users the ability to employ third-party centralized clouds or use the DCS solution.

b) **Generated content backup**

Centralized messengers either store user-generated content (messages, files, pictures etc.) on servers (Telegram) or use third-party cloud storage (WhatsApp and others).

Since dialogs may also be important to users, we should provide secure storage, which can also be achieved by keeping backups in DCS or third-party clouds.

ⓘ The Dynemix DCS protocol is still in development. More details about the protocol will be provided when ready.

III. Personal data, censorship and interaction with authorities

1. Unbiased Content Filtering

We stand for the protection of the substantive human rights, with freedom of expression among them. Unfortunately, human rights can be also exercised solely for malicious purposes. This is why freedom of speech is usually not considered absolute and is subjected to certain limitations.

The problem is that, once we appoint a censor, we cannot be sure that such limitations will be applied objectively and impartially. In real life, it often happens that governments use their censorship abilities to oppress any opposing opinions and deprive people of their ability to express their disapproval of government actions.

Since we are utilizing a blockhain protocol with embedded strong censorship resilience, it would be unwise not to use the emerging opportunities for censorship-free interaction. At the same time, we should secure certain ways of preventing users from abusing this opportunity for malicious activities.

To solve both problems at the same time, we decided to apply a novel approach of content filtering via guardian oracles, which will have the following specs:

a) **Blacklists of banned users will be administrated by a specially appointed entity or by any third parties.**

   By default, when a user downloads any version of the Liberdyne messenger, it will contain an active blacklist of user accounts that have been accused of breaking the rules by an appointed entity. In the early stages, the blacklist will be fed by an oracle controlled by developers. Later, the functions of maintaining the oracle may be transferred to a specially created organization.

   There will also be an opportunity to apply any other blacklists, including those administrated by third parties through personal preferences, if the user wishes. For example, there can be organizations monitoring certain types of undesired content or parental-control services that the user wishes to apply.

   It will be also possible to create decentralized oracles that will be controlled by cumulative voting or any other possible types of oracles that the community may come up with.

b) **Filtering exists only on the messaging-application layer.**

   Filtering will be applied only toward the messaging functions; blacklisted users will not be able to send messages, calls, files or requests to be added to contact lists or perform any other messaging functions toward other users.

   At the same time, no censorship of any kind should be ever applied on the blockchain layer, since it is a straightforward violation of decentralization.

   If necessary, it will be possible to create a decentralized oracle that will manage censorship on the blockchain layer (i.e. provide a set of default rules for the Guess My Block game). We are uncertain whether this approach is appropriate, however. The final decision is up to the community.

c) **Filtering can be deactivated by the user at any time.**

   Once the user decides that he does not need this kind of protection, he can easily disable filtering through the app's preferences and get a censorship-free messaging app.

   We cannot guarantee that such an opportunity will be available for all platforms, since it may violate the platform administrator's policy.

   The function of enabling/disabling blacklists will be likely available on the *nix, Windows and Android versions, since these platforms do not restrict the installation of apps downloaded from an arbitrary source.

   As for the versions distributed through the App Store and Google Play, this opportunity will depend on Apple's and Google's attitude toward it. In the case that we get any claims form Apple or Google, we may have to remove the blacklist-disabling feature from the relevant version.

The combination of these features ensures an unbiased approach to filtering and at the same time provides sufficient protection from malicious content distribution. Even If the developers (or

other entities controlling the default guardian oracle) try to apply any prejudiced filtering rules, users are free to choose another filtering solution, which renders such attempts inefficient.

In general, filtering will be mostly used to protect users from spam and certain kinds of undesired content, but in the case that the user wishes to access any content, he can easily disable censorship. We believe that it is up to the user to decide how he wants to use the possibilities provided by the app, unless his actions violate another user's rights.

We do not support any darknet activities, but we also understand that it is impossible to fully restrict them. For example, we can restrict users from disabling the blacklist, but, since it works on the application layer and not on the protocol layer, any third party can modify the code (as we go open source) and create an alternative client that has no censorship. Therefore, such actions are pointless. On the other hand, if we try to apply censorship to the protocol layer, this will instantly ruin decentralization, which is unacceptable.

2. Liberdyne and Personal Data

Many countries have adopted comprehensive data-protection laws. The most commonly known is GDPR, which regulates the processing of personal data within the European Union. Since the problem has been widely discussed, we also took it into account while building the system architecture.

Our approach is simple – due to its distributed architecture, the Liberdyne messenger will not collect or send any user data, either to us or to any other parties, unless it is required by the transport protocols (for example, if the user enables relay chains for anonymity purposes, all his correspondence will be routed through a chain of nodes in encrypted form). This can be easily verified by anyone, since the source code of the app will be publicly available. Our access to user data is limited to the information held in the public domain; therefore, we do not have to comply with any data-protection regulations.

There may be, however, certain situations when users provide us with their personal data by choice, for example, while contacting the support or abuse departments. In such cases, we will process the personal data according to the regulations, and all of the collected data will be instantly deleted after the matter of the appeal is resolved (unless the regulations require temporary storage).

3. Interaction with Authorities

Since we do not run any servers that process user data or participate in the system operations in any other way, apart from running our own peer nodes, we cannot collect any user-generated content that may be of interest to governments, corporations or other parties.

If any information about the user's personal data, communications, interactions with other users etc. is requested by authorities, we will not be technically able to satisfy such a request.

This may cause pressure from the authorities of certain states, where developers of a communication app are obliged to implement means of obtaining any user's information (including transmitted content, like private messages, files etc.) and providing it to the authorities upon request.

We believe that such laws go far beyond reasonable human-rights limitations, and we do not support this paradigm. We have intentionally developed an architecture that fully prevents any possibility of imperceptible user data obtainment, either by ourselves or by any third parties.

Our approach is fully consistent with the ideas outlined in the International Covenant on Civil and Political Rights, the UNGA Resolution 68/167 and the subsequent report of the United Nations High Commissioner for Human Rights A/HRC/27/37.

Nevertheless, we understand that our product's release may cause a negative reaction from certain autocratic governments that intend to fully control the private lives of their citizens, and that we will face attempts to block Liberdyne in a number of countries.

These attempts will most likely fail, considering that Liberdyne is designed to resist blocking attempts much better than any centralized messaging app possibly can.

On the other hand, the majority of democratic regimes around the world stand for the protection of substantive human rights and the freedom of expression among them, hence, considering that we apply reasonable limited censorship to restrain the abuse of those rights, we expect to avoid confrontations with the authorities of most countries.

IV. Open source

From the very start, the crypto-community has relied on certain principles that form the basis of crypto-philosophy. One of these principles is to make cryptocurrency software fully free and open source to ensure the possibility of the decentralized and transparent development of the platform.

To prove the proclaimed security and privacy of a messenger app, its source code should also be open to public examination and independent audit.

Considering the importance of making both components of our app open, we intend to make Liberdyne fully free and open source, distributed under the GNU GPL (other open licenses may also be considered).

If we remove copyright straight from the start, however, we may provoke the creation of numerous scam and copycat projects, which will try to confuse users. Without any legal protection, it will be challenging to deal with such behavior, and this can cause a lot of problems both for us and for users, who will find it difficult to distinguish the original Liberdyne/Dynemix.

At the same time, we do not want to register any patents for our solutions, for we intend eventually to make the platform fully free and open, and we should not obstruct the process with our own actions.

To solve this matter, we have decided to distribute the software under different licenses during two periods.

a) **The proprietary period** – from the start of the public testing to a short period after the mainnet launch.

   From the first publication of the software, it will be distributed with an open source code but under a proprietary license, which will cover the source code by copyright and restrict modification of the code. This is especially important for the test period, as the creation of modified versions and forks by users may cause turmoil and negatively affect the development process.

b) **The free period** – from a short period after the mainnet launch and onward.

   Shortly after the mainnet launch, Liberdyne will be distributed under the GNU GPL, thus removing any copyright protection and making the software fully open and free.

V. Competition

It should be noted that the project actually has no direct competitors since the information about other cryptocurrency platforms created in a symbiotic relationship with the basic application for distribution has not been published even in the form of a project, and we simply have nothing to directly compare the Liberdyne/Dynemix project with.

Nevertheless, as separate component, Liberdyne can be compared to other solutions on the market. Despite there being several messengers that use blockchain technology, Liberdyne is unique due to its development in conjunction with Dynemix.

As a rule, blockchain platforms are created separately as a tool for the development of a wide range of applications, and decentralized messengers are created by other developers as applications on top of existing platforms, although there are a few exceptions. For example, Adamant Messenger uses its own blockchain, but is actually based on Lisk, and its main purpose is message delivery; hence, it differs from our concept.

On the other hand, the Dynemix/Liberdyne project was developed as a whole, and both components are interdependent and mutually beneficial. This fact puts Liberdyne ahead of other P2P messengers since Liberdyne features a well-designed reward system that encourages users' interest in supporting system operations without relying solely on their altruistic intentions.

Another noteworthy feature of Liberdyne is its optimal use of blockchain technology. After the cryptocurrency boom in 2017, a number of developers introduced concepts of decentralized messengers based on blockchain technology. The problem was that many of these projects used blockchain technology for building more hype, not intending to create an optimized solution that could help improve the user experience and create an ultimate product in the first place.

As a result, to date no one has been able to present a product capable of competing with popular centralized solutions.

On the other hand, Liberdyne uses blockchain technology only to the extent that allows it to achieve the proclaimed goals and present a product that can withstand significant loads, providing a user experience that is not substantially inferior to centralized solutions (although we have to admit that we can hardly manage to reach the same exact level of speed and reliability as popular centralized solutions due to the limitations of P2P architecture).

VI. Marketing strategy

We understand that, though we are developing a breakthrough project that may let the whole industry take a huge step forward, only those few who are deeply involved in cryptocurrency technology will be able to fully evaluate the potential of the project.

Since our main goal is not just to create a new advanced platform for the cryptocommunity in its current state but to spread blockchain technology all over the world via the creation of a mass-market product, it is absolutely clear that about 99% of our target audience has a very limited basic understanding of the technology. While we can convince some portion of the cryptocommunity members of the strong potential of Liberdyne/Dynemix simply by explaining all the advantages of the platform, it will not work with people who are not much into blockchain technology.

There is no way we can capture a major audience's attention long enough to explain all features of the Dynemix platform and benefits it can bring to people's everyday lives, which is why we need to introduce another approach that will help us raise interest in the platform even among those who do not plan to join the cryptocommunity.

To accomplish this, we need some simple, and at the same time striking, “killer features” that can be easily explained and understood, thus driving public attention to the project by themselves. Fortunately, we have a couple of those.

1. Liberdyne – Next-Generation Decentralized Private Messenger

As mentioned, the revelations of Edward Snowden, according to which the US special services had been actively collecting people’s private information on a global scale, led to the rise of discussions about creating a secure means of communication that could guarantee users’ privacy.

To solve this problem, the developers of instant messengers implemented end2end encryption. Telegram was the first popular app to do so, and it was its major marketing feature. In fact, it was the main distinguishing feature of the Telegram messenger back at its launch, while, from the point of view of functionality, it was inferior to its competitors (it did not even feature voice calls). Using that single feature as the basis of the app’s marketing helped Telegram rapidly gain a significant userbase and forced other competitors to quickly implement end2end encryption as well.

This example shows us that the question of user privacy can be a strong basis for a marketing campaign. Even though end2end encryption has not solved the problem of user privacy, since users still have to trust the developers, who operate servers through which all user data passes (which opens up an opportunity for the meta-data analysis), it helped promote a product to the level of its competitors, which were years ahead of it at the time.

Liberdyne makes a much larger step toward user privacy – it is a next-generation product that works fully P2P without any servers involved. In combination with the peer relaying technology (SATAN), which provides a whole new level of privacy and security to every user, it can finally give users confidence that their communication is reliably safe.

Due to the excellent censorship resilience of the Dynemix blockchain, we managed to implement a new content filtering solution into Liberdyne, which can allow for decentralized unbiased user-selectable filtering. This approach will put an end to biased censorship, which has been applied by major social platforms during the past years.

Given that Liberdyne is not just offering a slight protocol improvement but is a generation ahead of all major centralized instant messaging software, while at the same time keeping most current functionality and parameters available, it may provide strong incentives for users to download and try it.

In addition, we should note that the major centralized messengers will not be able to offer the same quickly, since it requires a complete architecture redesign. That fact should keep the project ahead of any competitors for a long time.

2. Liberdyne – Messenger That Pays You to Use It

By default, the application will be configured to perform a number of functions to support the system (namely minting, DOG, SATAN relays, and storage nodes), for which each user will automatically receive a small reward, becoming a sort of “miner.” At the same time, the functionality of the mobile versions will be configured so that the user will not feel that there is a significant waste of resources (i.e. battery and Internet traffic). If desired, through the settings, the user can both increase the amount of resources spent (thereby increasing the reward) or reduce or even disable the feature entirely.

As a result, users will receive a reward in the Dynemix cryptocurrency for the mere act of using the application (as “usage” includes system support, like in all P2P systems). At the same time, the algorithm of reward distribution will be such that the fewer users who are online, the greater the reward each of them will receive. According to our idea, this will encourage users to install the application as quickly as possible in pursuit of a larger reward, which should lead to an explosive

growth of the userbase. When the userbase becomes large enough, people will already be interested in using the application for reasons other than receiving a reward.

Getting a small reward and having the ability to spend it should incentivize a huge number of new users to become involved in the cryptocurrency world, allowing them to overcome the entry threshold. Thus, we will gain a loyal userbase that has experience using the Dynemix cryptocurrency.

Having learned to use the cryptocurrency on a virtually free basis and realizing its advantages, people will be able to gradually exchange their fiat money for it and come up with other scenarios for its use by themselves (for example, for payments between users).

Businesses may also be interested in using Dynemix for payments, given the size of the loyal userbase that Liberdyne may obtain, and the possibility of using Liberdyne as a distribution channel that can repeatedly exceed the capabilities of any other cryptocurrency platform.

The inclusion of more users in the Dynemix economy should stabilize its volatility, and the eventual implementation of the economic model of a decentralized algorithmic central bank will make it a convenient tool for any kind of payment.

Thus, through the use of this simple slogan and the concept behind it, the project has the potential to take the leading position in the cryptocurrency market.

This feature will also help overcome users' initial reluctance to use a new P2P messenger (which means a lack of available contacts within the app in the beginning), as people will be incentivized by receiving more rewards with a smaller userbase.

Combining these two "killer features," we can conduct a marketing campaign that will hit both the instant messaging market and the cryptocurrency market, reaching the largest potential audience and helping the platform to gain a significant userbase. We expect that this approach has the potential to secure leading positions for the project in both markets.

VII. Monetization and revenue

1. Monetizing Liberdyne

We do not intend to monetize the Liberdyne messenger directly. Liberdyne is conceived as a basic platform for communication and payments and as the foundation of the entire Dynemix ecosystem, which is why we intend to keep it completely non-commercial. All paid services integrated into Liberdyne (e.g. SATAN, cloud storage) will be directly P2P and will bring no revenue to the developers.

Unlike most products with a centralized architecture (e.g. WhatsApp, WeChat), in our case, the cost of supporting the application will be much less.

This is due to the fact that the operation of the system is secured by the users themselves (with P2P architecture), which eliminates the need to maintain powerful data centers to process huge flows of information.

When it comes to the technical support of the infrastructure, developers play the same role as any other users and only during the starting period will the involvement of the developers be necessary. According to our estimates, we will need to run about 100 master-nodes to guarantee stable system performance.

After attracting a sufficient number of users, if additional income is necessary to support the system, we may consider ways of monetization:

- Adding additional services to the application that will be paid for by the users in dynes.
- Charging a fee to service providers for their content distributed via the messenger.

We see monetization, however, as a last resort and hope that we will be able to avoid it.

2. Other Projects in Dynemix/Liberdyne Ecosystem

Many developers of cryptocurrency platforms are not involved in building the ecosystem around the platform. As a rule, they create basic state-transition software and delegate everything else to third parties.

We may take another approach, and we do not exclude the possibility of participating in the development of the platform's ecosystem, which refers to both the direct development of different services and the support of third-party projects. At this stage, however, we cannot reliably confirm our intentions, as it is not currently clear how large a set resources we will possess and how our priorities will be set.