# Birdchain Whitepaper

# Birdchain - Business white paper
## The app that rewards


1. [Abstract](#abstract)
2. [Established business](#established-business)
3. [Birdchain. The app that rewards](#birdchain-the-app-that-rewards)
   1. [Raising value of the Token](#raising-value-of-the-token)
      1. [Streams of the Token](#streams-of-the-token)
      2. [GBI - Global Basic Income](#gbi---global-basic-income)
   2. [What is A2P Messaging and Why it is Important?](#what-is-A2P-messaging-and-why-it-is-important)
   3. [Issues to Solve](#issues-to-solve)
   3. [Birdchain App](#birdchain-app)
   4. [Expected Messages a.k.a. Transactional messages](#expected-messages-a.k.a.-transactional-messages)
      1. [Overall Description (Expected Messages)](#overall-description-expected-messages)
      2. [Specific Requirements (Expected Messages)](#specific-requirements-expected-messages)
      3. [Functional Requirements and Use Cases](#functional-requirements-and-use-cases)
      4. [Prioritization and Release Plan](#prioritization-and-release-plan)
   5. [Unexpected Messages a.k.a. Promotional messages](#unexpected-messages-a.k.a.-promotional-messages)
      1. [Overall Description (Unexpected Messages)](#overall-description-unexpected-messages)
      2. [User Flow](#user-flow)
      3. [Specific Requirements (Unexpected Messages)](#specific-requirements-unexpected-messages)
      4. [Functional Requirements](#functional-requirements)
      5. [Server Setup](#server-setup)
      6. [Prioritization and Release Plan (Unexpected Messages)](#prioritization-and-release-plan-unexpected-messages)
4. [Token sale](#token-sale)
   1. [Soft and Hard cap's](#soft-and-hard-caps)
   2. [Purpose of the Token](#purpose-of-the-token)
   3. [Fund Allocation](#fund-allocation)
   3. [Are Birdchain tokens Securities?](#are-birdchain-tokens-securities)
   4. [Please Note](#please-note)
5. [Exchanges](#exchanges)
6. [Appendix](#appendix)


# Abstract


The team of the Birdchain is not a start-up. It is part of a parent company, named Vertex. Vertex is a profitable company with a team of experienced professionals, whose professionalism and skills were tested in action. The founding team has successfully co-launched several businesses.


  <img alt="Vertex"
src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/BirdChain-Whitepaper-Vertex.png">

Birdchain is a decentralized application (Dapp) similar to instant messenger. The big difference is Birdchain will allow its users to make a monthly passive and active income. Simultaneously, Birdchain will provide companies with higher quality and better-priced services: A2P SMS distribution, content engagement, and others.



CRYP2NOVA

Archived copy · cryp2nova.com

# Birdchain Whitepaper

One of the key features of Birdchain is that the app will allow its users to "sell" their unused SMS for BIRD tokens. These texts will be used by companies like Google, WesternUnion, Facebook, and many others, to deliver transactional messages. Transactional messages are expected by the end receiver. As these messages will be send automatically in the background of the app, it will offer app users passive income.

The other key feature - P2P advertising. The idea is quite simple: users get rewarded in BIRD tokens for receiving rich media messages from businesses. The receiver can get additional, bigger reward when they share it with people who might be interested in it. This feature will allow Birdchain exploit possibilities of promotional (unexpected) messages to increase usability and demand for BIRD token. This service will require some user interaction, thus it will help creating active mean of earnings.

All features of the Birdchain create strong BIRD token. It stands out with low initial cost and token flexibility as a unique currency with favourable viability.

The official app presentation is planned at the Mobile World Congress in Barcelona in 2019.

# Established business

  <img alt="Flags" src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/Flags.png">

The heart of Birdchain is [the European Union (Lithuania) company JSC Vertex](https://rekvizitai.vz.lt/en/company/vertex/). In 2009, Vertex started our A2P SMS aggregation business, called [VertexSMS](https://vertexsms.com/?lang=en). Since then, [VertexSMS](https://vertexsms.com/?lang=en) has delivered more than 2.6 billion SMS messages globally for various international brands, including Western Union, SalesForce, TransferGo, and many others.

  <img alt="Clients" src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/BirdChain-Whitepaper-Clients2.jpg">

Vertex is a profitable company. In 2015, Vertex received an award for being the fastest growing ITT company in Lithuania. Vertex continued to grow in 2016. In 2012, Vertex was recognized as one of the fastest growing companies in Latvia.

In early 2015, Vertex was certified in information security management and received an ISO/IEC 27001 certificate. The certificate is adapted to all our businesses held in Vertex headquarters: dating sites, bulk SMS sending systems, and a rent-to-own company. Now, all these procedures are being applied to Birdchain.

The combined per country turnover and EBITDA of Vertex is growing every year.

| | Turnover | EBITDA |
| --- | --- | --- |
| 2017* | 4-5M euro* | 20%* | * Estimations |
| 2016 | 3-4M euro | 19% |
| 2015 | 3-4M euro | 17% |
| 2014 | 2-3M euro | 18% |

Archived copy · cryp2nova.com

# Birdchain Whitepaper

2013 | 2-3M euro | 17% |
2012 | 2-3M euro | 16% |

Vertex owns several brands and VertexSMS is one of them. Company founders gathered Birdchain team members from other brands, meaning the members of the Birdchain team know each other well and have been working together for years. This decision eliminates possibilities of the team not being able to work together and creates a healthy, creative, and productive working atmosphere.

Birdchain team members are individuals with a set of unique experiences to make Birdchain fly. Most of them have experience working in a global A2P SMS market and share extended know-how about regulations, key players, technologies, and other aspects. Other team members are experts in marketing, business development, law, and software development.

The founding team has successfully co-launched several businesses:

**Social network** with more than 1 000 000 unique users (flirtas.lt, flirts.lv, flirtak.pl, teidi.ee, and others). This business allowed our team to design and test new ways of advertisement. It exposed several unique user experience patterns to us, which led to developing very efficient campaigns. We learned a lot about software and app development in the user communication field. We will adopt this knowledge to the Birdchain app and service.

In 2013, we invested in **advanced microlaser manufacturer** (integratedoptics.com). Today, we manufacture the smallest lasers in the world, and our company is considered one of the most promising Eastern European start-ups. We have more orders for our lasers than we can manufacture.

**Retail**. In 2013, we launched the first in the region rent-to-buy service (bekredito.lt).

Birdchain team is led by [Ernestas Petkevicius](https://www.linkedin.com/in/epetkevicius), [Audrius Vrubliauskas](https://www.linkedin.com/in/outrum), [Aurimas Sabaitis](https://www.linkedin.com/in/sabaitis), [Marius Petrauskas](https://www.linkedin.com/in/mariuspetrauskas), [Joao Martins](https://www.linkedin.com/in/joao-martins-porto), [Arminas Keraitis](https://www.linkedin.com/in/arminas-keraitis-460a4a40), [Tomas Vitkauskas](https://www.linkedin.com/in/tomasvit6/), [Aiste Surutkoviciute](https://www.linkedin.com/in/aistesur), and [Zygimantas Vanagas](https://www.linkedin.com/in/zygimantas-vanagas).

  <img alt="Team" src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/Team%202.png">

### Advisors

**[Anatoly Ressin](https://www.linkedin.com/in/anatolyressin)**

![Anatoly](https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/200xanatoly.png)

A founder of Fastr, a reading-training application development firm, and AssistUnion, which builds intelligent and performing web-enabled systems. Has been a team member of Civic as a contractor, senior software developer at C.T.Co and #Decent as a core developer. At the moment, he is a CEO of BlockVis, a company involved in organizing blockchain competence and education opportunities and structuring the blockchain market in Latvia and abroad.

**[Andreas Petrow](https://www.linkedin.com/in/andreypetrow)**

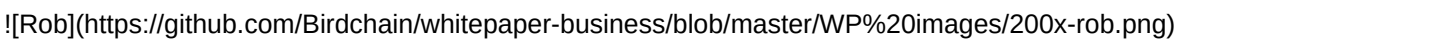
Archived copy · cryp2nova.com

# Birdchain Whitepaper




7+ years of experience working in financial services & data providing business. He is a leading Relationship Manager within Standard & Poors  improving client relations and consulting Asset Managers in Northern Europe. Andrey is developing and cultivating strategic relationships, maintaining and developing existing business at CEO-Level. By helping customers develop their businesses using advanced analytics, standing up financial-data analytical tools, creating and maintaining financial models, and onboarding compelling new datasets  growth is always a top priority. The focus lies on Portfolio Analytics and Quantitative investment approach.

**[Rob Eijgenraam](https://www.linkedin.com/in/robeijgenraam)**



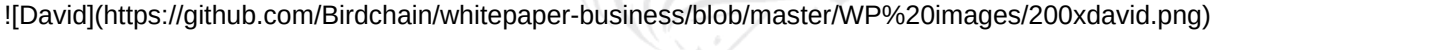

Rob Eijgenraam is an online marketing expert, founder of ICO Marketing company CryptoLeads.io, and a co-founder of Purple Minds. He is a speaker about cryptocurrency marketing and an organizer of crypto educational events. His passion for Bitcoin and Blockchain comes from a belief in open systems and a permissionless society and market place.

**[Paulius Petretis](https://www.linkedin.com/in/sauga)**




Paulius Petretis is a Data Privacy and Information Security expert, guest speaker at various conferences and seminars, Trainer at exclusive training courses. Paulius manages VORAS Consulting which offers consultancy services including data privacy and information security management, social engineering and IT service management. Certification in CISSP-ISSMP, CISM, CGEIT, CRISC, CISA, PMP, CIW-SA!

**[David A. Cohen](https://www.linkedin.com/in/davidallencohen)**




David A. Cohen is internationally renowned for his pioneering work in the Decentralized Software industry and recently in the Digital Currency and Blockchain industry.  David is working on next generation blockchain architectures such as Hashgraph, and is currently advisor to Hashgraph. David is a former IOTA Foundation member.  He is also working with FarmaTrust, WePower, Wireline.io, and other blockchain companies.  He is Founder of Dcntral.ai, a blockchain-based cybersecurity software company.

He was the founder and CEO of Infotility where he pioneered the Grid Edge unlocking multi-billion-dollar software markets and creating the industries first artificial intelligence-based software platform - GridAgents.

David is a thought leader and has spoken at many venues such as MIT, TEDx, Consensus, Blockchain 100X and Singularity University.  He is an expert on Blockchain, Artificial Intelligence, Cybersecurity, and Edge Computing.

# Birdchain. The app that rewards

There are two main types of A2P messages: transactional (expected) and promotional (unexpected). Each type has unique purpose, problems, and motives to be used by businesses. Team of Birdchain employs almost 10 years of experience in the market to upgrade these services for businesses. At the same time we create oportunity for Birdchain app users to benefit from both types of A2P messages.

Archived copy · cryp2nova.com

# Birdchain Whitepaper

In the sections bellow you will find detailed information about what is A2P messaging market, how it works, what issues delivery of expected and unexpected messages have, and how people will get rewards for using our app. Considerable portion of this white paper is focused on technical implementation of the idea of the Birdchain as it is already has been started to develop after reaching softcap.

# Raising value of the token

**Please note**: this is not a financial advice. Sole intent of this section is to introduce posible possitive effects on BIRD token value which lies in the fundamentals of our idea.

Our goal is to create a strong token that would be useful for our contributors and could circulate in the world easily. Here are key points that, in our opinion, makes BIRD token to stand out:
1. **Low initial cost**. During the ICO, the BIRD was sold at the lowest price to create a higher incentive for contributors.
2. **Token flexibility**. When the BIRD value and demand increases, it can be fragmented into 18 decimals to serve market needs.
3. **Unique currency**. Exclusively, BIRD tokens are used for purchasing Birdchain A2P messaging services ensuring the BIRD token has a real demand.
4. **Favorable viability**. To have a sustainable business of delivering expected (transactional) messages, it needs only 100 000 app installs. Each phone can sell about 100 SMS per day on a conservative approach.
5. **High desirability**. Providing the possibility of an income to its users, the Birdchain app is highly appealing to the public. The more users will install the app, the more SMS flux it will create, meaning a faster BIRD rotation, demand, and appreciation.
6. New features and higher **BIRD circulation**. We will drive demand and circulation of BIRDs with new Birdchain features and services, which will be developed after launch of the app.
7. **Embracing the competition**. Both of our key features will be appealing to all companies that look for lower price, high-quality service, and new channels to reach their customers.
8. Detailed **financial forecast**. Check it out here: [Fund Allocation](#fund-allocation)

## GBI - Global Basic Income

The main goal of Birdchain is to disrupt the A2P messaging market with blockchain technology. We aim to make it better, faster, more transparent, and fair-priced.

However, we believe that, soon it can become obvious that Birdchain is much more than just a (r)evolutionary idea. Birdchain can become the first (in the history of the world!) non-governmental initiative to create Global Basic Income.

Basically, all it takes for user to start earning BIRD tokens is to install the Birdchain app. Depending on a country and market activity, Birdchain app users should be able to earn up to 150 USD per month. Even in rich countries such additional income is welcomed as it might cover some monthly expences. It might have significantly higher impact in the less developed regions.

We see this feature as a huge marketing opportunity and a chance to stand out in the global market of messaging apps. However, Birdchain guarantees no revenue to Birdchain token holders or Birdchain app users.

# What is A2P Messaging and Why it is Important?

An Application To Person (A2P) SMS is sent from an application  typically a web app  to a mobile subscriber. There are several types of the most popular A2P SMS:
* SMS notifications (transactional);
* SMS-based two-factor authentications (transactional);

Archived copy · cryp2nova.com

* Automatic booking confirmations (transactional);
* One-time passwords (transactional);
* Loyalty programs and marketing messages (promotional).

Transactional SMS service is mostly used by social networks and companies to provide services of e-commerce and online retail, travel and transportation, financial services and payments, and gaming. Because of their effectiveness, promotional messages are sent by companies operating in almost all markets.

Some examples of a transactional SMSs:
* G-123456 is your verification code.
* Hair Salon appointment reminder: haircut tomorrow at 10:30AM with Charles. If you have questions, please call 0161 55.

  <img alt="SMS stats" src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/07.png">

Why modern companies deliver transactional SMS:
* SMS is platform-independent;
* SMS is network-independent;
* SMS is geographically independent;
* SMS is non-proprietary;
* SMS is app-independent (no need to have an app to communicate);
* SMS requires no data subscription (roaming scenarios for example);
* SMS is by far the most standardized messaging technology.

  <img alt="Phone users" src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/06.png">

A2P market is expected to grow by around 5% per year and reach $84 billion by 2024.

![A2P SMS market growth](https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/04.png)

The reasons for A2P growth:
* SMS is the most effective and the cheapest way to reach customers. 97 percent of people read an SMS within three minutes from receiving it, and SMS has the highest engagement rate compared to emails and OTT apps.

| User Count | SMS Cost | Email Cost | AdWords Remarketing* | Facebook |
| :--- | :---: | :---: | :---: | :---: |
| CPM** | 30$ | 15$ | 30$ | 52.5$ |
| True View | 98% | 22% | 80% | 60% |
| Real CPM*** | 31$ | 66$ | 37$ | 87.5$ |

* An increase in two-factor authentication and security via SMS. Today, personal data is transferred over the web more often than ever before. Seeking to add a security layer, two-factor authentication has become the go-to method for companies that aim to protect their users. Sending an SMS as an authentication has become very popular because customers usually have a mobile device on them at almost any moment.
* A 2-way authentication has a staggering 99.9% reachability rate across all mobile users, as long as the number provided is correct. It does not require an internet connection. Mobile bandwidth is enough to receive a two-factor

Archived copy · cryp2nova.com

authentication or marketing message.
* No need to install additional apps. SMS messages can be received and read on any mobile phone in any network, almost anywhere in the world.

**In the ideal world**, when a business wants to conduct an SMS marketing campaign, it must contact aggregators. Aggregators are companies that mediate between businesses that WANT to send SMS and operators that CAN send them.

  <img alt="Ideal SMS path"
src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/IdeaWorld.png">

**In the real world**, it is not that straightforward.
![Real-world path](https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/09.png?raw=true)

As an independent aggregator, we see market flaws as opportunities for Birdchain. Birdchain makes the path of SMS sending as straight as possible.
![Path of the Birdchain](https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/10.png)

# Issues to Solve

As Birdchain intends to solve two different groups of A2P messaging issues it takes two tailored approaches. Due to technical nuances our solution for Expected messages is Android exclusive and solution to Unexpected messages will work on both iOS and Android. This is why we separate both approaches. Nonetheless, each of them supports and complements another.

  <img alt="Birdchain Scheme"
src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/Asset%203.png">

### Expected messages

**Issues:**

* Even though A2P messaging technology is longstanding, there are no reliable tools to measure deliverability. Without it, there is no transparency.
* A2P messaging is extremely expensive. Single A2P message in Belgium costs 0,1 USD, in Japan  0,08 USD, in New Zealand  0,1 USD.

**Solution:**

Birdchain will employ unused SMS of regular people and reward them for every single used SMS. This background process is completely automatic.

This approach will grant us access to cheap and reliable route of A2P message delivery. We will track deliverability reports and log them on blockchain  provide unprecedented level of transparency to the market.

For carriers we will present Birdchain solution as loyalty program that will help them to onboard more network users and

CRYP2NOVA

Archived copy · cryp2nova.com

get bigger market share.

This solution will be Android exclusive. Covering 87,7 percent of the market.

### Unexpected messages

**Issues:**

* It is getting harder and more expensive for businesses to get in contact with their customers. Effectiveness of communication channels has decreased drastically. A2P messaging could be the solution, but it is too expensive for companies to use daily.
* The technology is moldy. It is impossible to send images, videos, or formatted text  attributes which are necessary for every modern marketing campaign.
* Recent events revealed that people do not like how Facebook and Google exploit their personal information. These companies make billions in revenue and actual owners of this data are not rewarded in any way.

**Solution:**

Birdchain will allow businesses to send direct rich media messages for Birdchain users. Users are rewarded to engage with the message. They can get additional  bigger  reward for sharing the message with people who might be interested in it. Hence the name  P2P marketing.

Birdchain app will prevent delivery of the same message to a person more than once. Even from different phones.

Rich media messages will allow businesses to send their clients images, videos, formatted text, and call-to-action buttons.

This feature will be Android and iOS compatible. Covering 99,8 of the market.

# Birdchain app

Architecture, performance management, and capacity planning of applications depends on the ability to model these distributed applications at design time as well as during normal operations. This paper specifies overall description and some details on the functional requirements of performance and architecture modeling of the Birdchain Network. These specifications include the need to model transactions consisting of multiple, nested, synchronous and concurrent client-server interactions using tractable techniques and best practice. Example of application model with a centralized backend is presented based on the past experience and as a result of the current research. Suggestions for future work concludes the paper.

Design planning of application architecture and performance becomes more demanding as the client / server paradigm is used in commercial enterprise applications now scales with the ability to decentralize the dataflow and implement cryptographically secured payment gateways based on battle tested blockchain technologies. Modelling of the distributed applications with a centralized core node is crucial for every new service before the development starts. This paper motivates the need for modeling capabilities for these applications, and specifies the functional requirements of performance modeling of commercial distributed-client / centralized-server applications based on available technology solutions and required centralized servers and databases.

# Expected Messages a.k.a. Transactional messages

The Birdchain network is a mobile centered distributed application with a centralized server core which helps people to

Archived copy · cryp2nova.com

make monthly passive income and help companies to distribute content. The client side application should be available exclusively on Android devices and on all future-released open source operational systems with the ability to use related devices in the Birdchain network.

Based on a given details we decide to set a client side application as a center of the business model, and a centralized server infrastructure as a data and communication support provider. Such a mobile-centric architecture make it really easy to scale in future or even to replace a centralized architecture on a decentralized network with a custom consensus. Taking back to our research of existing solutions the closest architecture for the future decentralization is the directed acyclic graph ledger or a lightning network protocol in future. But to make any available in future distributed architectures we have to be sure its possible to provide minimum or zero fee for the transaction, required consensus of nodes to be sure users are safe from a double spend issue, and to make the transaction as fast as possible(meaning less than 4-5 seconds). Unfortunately there is still no any such available distributed ledgers including IOTA Tangle.

## Definitions, acronyms, and abbreviations

* **Commodities** - Available message quote provided via android application by a mobile phone owner.
* **Obligations** - Required message quote asked by a company or individual for a messages distribution in a specific region
* **Provider** - A person who interacts with the client side android application to provide commodities to the network
* **Consumer** - Someone who interacts with the client side application to provide obligations to the network.
Android Miner Application Application downloaded from Play Store. This application should be used by a person who is willing to provide a bulk of text messages in a specific region.
* **Consumer Application** - Application downloaded from Play Store, App Store or a browser interface. This application should be used by a person or a company who is willing to request a bulk of text messages in a specific region.
* **Web Portal** - Browser interface for consumers.
* **Administration Application** - Application that provides a possibility for the moderating staff to avoid illegal campaigns and to provide tokens on a new campaign requests. Also it provides a statistical information on users activities. Should be available from browser, iOS and Android devices.
* **Monitoring Application** - Application the provides a possibility for the development team to collect statistical information on Birdchain network and ledger(database) health. Monitoring Application should be linked to all available microservices, also mobile statistics should be well integrated into it.
* **Birdchain network** - A set of microservices and different kind of databases to provide the core of the network and various caches to the final network clients** - mobile applications, smart contracts and web portals. Birdchain network contains all required data inside to be fully functional and completely separated from any infrastructural components and administrational user interfaces.
* **Ethereum Smart Contracts** - Back end services provided inside the Ethereum ledger. All of them are custom services created by Birdchain developers to give a possibility to the final user to get both in-chain tokens and BIRD crypto. Also smart contracts should provide all required business logic around token escrows.
* **Public Interfaces** - Both Birdchain Network and Ethereum Smart Contracts have public interfaces to provide the data for anyone who is willing to make an audit actions and to the third party software providers.
* **Private Interfaces** - All provided by Birdchain and Edenlab teams infrastructure that gives a possibility to the final users - commodity providers, obligation providers, administrators - to operate within the Birdchain network and to act as middle mans between Birdchain economy and outside world.

## Overall Description (Expected Messages)

This section will give an overview of the whole system. The system will be explained in its context to show how it interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

Archived copy · cryp2nova.com

# Birdchain Whitepaper

### Product Perspective

This system will consist of three parts: one mobile application, one web portal and a backend server with a database. The mobile application will have two parts:
* to find commodity providers and schedule message delivery
* to share commodity within consumers network

While the web portal will be used for managing user specific information on profile and payment details and administrative access for selected users. Backend server will contain current Birdchain network state including analytics, statistical information, administrative actions and various caches.

### Tokenization

Birdchain is a two-party system:
* **Commodity providers** - plain users with an available quota for message transactions.
* **Obligation providers** - message content distributors.

We propose to refer to the Birdchain network as to the closed economy model which gives a possibility to exchange user provided commodities for a digital token. Token should be substitutable - any party can make a one-to-one exchange; should be singular - parties should not exchange one token for any amount of goods or services. Meaning Birdchain token has all characteristics of the utility token and should be used with the single purpose - exchange between provider and consumer parties. For the token minting service we will take Ethereum network. Token should be minted with a smart contract by a request from the client application and attached to the telephone number as a unique identifier and address. Each token should be baked by a single text message. Consumer party acts as a token provider changing currency(ETH, USDT, BTC or any other fiat currency available for the exchange in Birdchain network in the future).

Since its a data-centric product it will need to store the data. For that, a centralized database will be used with a potential to transform into a custom distributed ledger. Both the mobile application and web portal will communicate with the database, however in a different ways. The mobile application will only use the database to receive a newly appeared obligation, its status and initial requirements for requested commodity; then send a Proof-of-Work to the database. Also mobile application should have an access to the current accounts balance via wallet and to the token-to-ETH exchange functionality. All of the database communicate will go over Internet, all actions for the Proof-of-Work will go through local telecom networks.

The mobile application has some restrictions about the resource allocation. To avoid problems with overloading the operating system the application should not contain any data unrelated to accounts actions.

### Product Functions

With the mobile application users will be able to provide their devices and mobile operator text message quota to third party individuals and companies anonymously. The only information provided by every account should be mobile phone number. Using phone number we can provide user with the token balance. Accounts should be picked anonymously and

Archived copy · cryp2nova.com

randomly without sharing phone number information by selecting the required geographical location and linked selected accounts in a mining pool for every linked to the consumer. Pickup criteria is a request from the consumer for specific number of text message transactions within one region. Providers dont have the ability to get information about the consumer or the amount of initial requested quota. System should not use the full amount of the provided message quota but pick the amount of transactions individually and randomly based on accounts rating and number of providers in a selected region. Every provider should have a rating. Rating should scale based on the amount of successfully done transactions.

Rating should be normalized within each region to avoid abnormal outliers and to involve more newcomers into the selection. In future we propose to use Markovs chain Monte Carlo algorithm to pick accounts within the probability space of each region using their transactions history chain and the transaction history chain of the consumer. Provider mobile application should work in a background mode as a mining entity. Also it should have a minimalistic user interface to provide current balance, current amount of proposed obligations and a history of completed transactions and spent commodities. Currently the only mobile platform available for this is Android.

Consumer should be able to settle new obligations in a required region within web portal or within mobile interface. There is no problems at all to provide a fully featured applications for Android platform. The only interaction with the outside world is happening via Internet to the server(or to the distributed ledger in future releases). Both application should be linked to notification microservices deployed within back end bundle.

Web portal is available for both for consumers and for providers to give the access to current balance, to setup new obligations and for other statistical information.

Web portal should communicate directly to the data handling microservice(or distributed ledger) and all available cache and notification services services.

### User Characteristics

There are three types of users that interact with the system: providers, consumers, administrators. Each of these three types of users has different use of the system so each of them has their own requirements.

Providers can only use the mobile application to provide commodities and view their token balance. This means that the user have to be able to install the application and after providing required permissions(send text messages and internet access) application should be able to act as message sender. Application will automatically link user to the region and select user into the mining pool. User cant change the region because its linked to the telephone number and user dont have any influence on the rating system and mining pool initial selection. In case the phone has two SIM cards app should be able to act using both SIMs as two different accounts. Accounts cant be merged, or splitted, or telephone number changed, all accounts actions are linked to the telephone number.

Consumers can only request a commodity and select a region. Consumers should have a balance and a token exchanger access.

Administrators acts only to handle statistical information, report bugs and respond to users. Administrators cant change mining pool configurations, cancel transactions etc.

### Constraints

The mobile application is constrained by the system interface to the SMS and Internet Access within the Android Operating System. Since the interface is the same and is available on all phones. Since application doesnt work offline, it is crucial that there is an Internet connection for the application to function.

Archived copy · cryp2nova.com

Both the web portal and mobile application should not be constrained by the capacity of the database. Since the database is shared between both applications, it may be forced to queue incoming requests and therefore increase the time it takes to fetch the data. In future this should be solved by a distributed database. We dont see any additional significant efforts to transfer the data while changing between distributed and centralized database architecture and agree such architecture as enough for the minimum viable product and next architecture upgrades.

## Specific Requirements (Expected Messages)

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

### User Interface

**Provider Android Application and Web Portal**

A first-time user of the provider application should see the log-in screen when he/she opens the application. Login screen should fetch current telephone number and by a request create a new account or authenticate the user with the previously created one.

New account registration is the wallet creation process. Birdchain wallet has three available types of tokens in it. The first one is the application token itself, the second one is ERC223 BIRD token represented as a cryptocurrency and available on different crypto exchanges. The third one is Ethereum crypto - added into Birdchain wallet to provide a possibility to the user to exchange both in-chain token, BIRD token onto ETH manually or automatically during mining process or later after token will be received by the user. Also its necessary to provide the ability to exchange singular in-chain token into available outside of the Birdchain network BIRD token.

If the user isnt a first-time user, he/she should be able to see the history of transactions done within Proof-of-Work consensus, achieved reward as a current token balance and and access to the wallet-exchanger screen.

User interface should be minimized to make signup/signin flow as short as possible and all user actions have to be simple across the app. All application logic should be handled automatically by the backend - ledger (or database - centralized or decentralized), backend server(several separated microservices) and queue managers.

User should be able to login into the web interface with the telephone number he/she uses for providing commodities, email, passcode or fingerprint. Web interface functionality should be similar to the android app interface except the ability to start making the Proof-of-Work.

Provider should be able to pass through a detailed registration to get access to a multi account user interface. Multi account user interface should contain statistical information from all available accounts belonging to the user.

All account validations should be done via SMS code verification. We suggest to use two-way authentication for all multi accounts settled into the network.

Application dashboard should give an ability to access different mining pools opened into the region where user operates. Mining pool is available to all miners, but contains several restrictions - rating system acts as a Proof-of-Stake; number of miners into the pool should be selected randomly from available users who have joined the mining pool manually. User selection happens randomly based on the probability calculated within users previous activity, his rating and several other factors. More details on users selection is explained in the section about the Birdchain Network.

**Consumer Android Application**

Archived copy · cryp2nova.com

This requirements should be met by Android application. A first-time user of the consumer should be able to sign up and provide a basic information like email and password. Also application should ask user about the passcode, fingerprint, SMS authentication etc. All available hardware and software techniques should be used to make users privacy and transaction security as strong as possible. Next screen should provide a list of active, finished and draft campaigns created by the user previously.

User should have an access to the token wallet/exchanger. Wallet works absolutely the same as an Android commodity provider application do. User should be able to exchange in-chain token into off-chain BIRD and ETH crypto. User should be able to transfer off-chain cryptos from the account the any external address. Birdchain application doesnt verify the address - user should be warned previously before the transaction begin. More cryptos should be provided as an options to ETH in future.

When user starts a new campaign, available correspondent tokens should be locked by a smart contract as an escrow for commodity providers. Tokens should be released proportionally to the Proof-of-Work provided by the user. Amount of tokens locked is an automatic decision based on the price listed by all available providers in a region. Released amount of tokens should be calculated proportionally to the stake in the mining pool for every provider independently. Tokens are not eligible for the chargeback. Tokens outside the escrow can be exchanged to ETH or any other available currency on a request.

Security flow is the same as provider application has, beside that there should be obvious email validation(miners application doesnt force user to provide an email address, but asked for it as an optional security flow for the wallet account restore).

  <img alt="Authentication Mockup"
src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/BIRD-Authentication-Mockup.png">

  <img alt="Workflow Mockup"
src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/BIRD-WorkflowMockup.png">

### Minimum Hardware Requirements Statement

Any recommendations regarding centralized server configuration, cloud architecture and client-side applications which may be implemented using Birdchain software are based on Edenlab's experience and understanding of the customers intended use of a future product.

Individual requirements for the future setup, scalability and performance may vary. The optimal configuration of software and hardware may depend on a variety of factors, such as the  outbound message activity, complexity of applications and databases deployed and monitored, the size and complexity of the user options available, the amount and rate of delivery of the data within the Birdchain network, and many other factors. Hardware that does not meet the minimum requirements may result in varying degrees of performance degradation. Edenlab may require bringing the hardware platform into compliance with stated specifications before production deployment will be provided.

### Software Interfaces

The mobile application communicates with the server gateway, Ethereum ledger and telecom provider. In future

Archived copy · cryp2nova.com

releases there will be also API for the distributed ledger. The communication between the database and the web portal consists of operation concerning both reading and modifying the data, while communication between the database and the mobile application consists of only reading data, except provide Proof-of-Work statements.

**TBD**

### Communication Interfaces
The communication between the different parts of the system is important since they depend on each other.

**TBD**

## Functional Requirements and Use Cases

This section provides all details on user interaction with the system. All user stories and possible use cases provided below with the reference to interface mockups.

### Back End Microservices Details and Principal Scheme

  <img alt="Microservice Principal Scheme"
src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/Microservice%20Principal%20Scheme.png">

Here we provide back end decomposition and microservices overview. Core microservices included into Birdchain Network solution are:

** Operational Service
Operational Service represents the core logic of the entire back end. This is an obvious component and should be done in the first milestone. It handles all campaign infrastructural tasks:

* New campaign creation
* Campaign cancellation
* Campaign finalization
* Queries Ethereum network to deploy new campaign instance smart contracts
* Listens to Ethereum events
* Handles Ethereum exceptions and react on them
* Deploy and Dismiss mining pool instances
* Contains internal data storage with all current campaign states
* Update Birdchain Ledger with completed, failed and cancelled campaigns.

Operational Service consists of next components:
* **Task Manager** - contains all business logic of the Operational Service. Reacts on incoming messages from other microservices, handles internal data storage and changes a current state appropriately to users actions. Operational Service can update Birdchain Ledger (Database). Task Manager send tasks to Campaign Creation Service and handles all the response from Mining Pool Broker. Task Manager has a direct connection with Network Administrators Service in order to monitor the status of all ongoing operations and suspend, cancel and resume all tasks on any emergency.
* **Campaign Creation Service** - serves to send new campaign creational requests to Ethereum Smart Contracts. Interoperate with Ethereum Manager Service; listens to Ethereum Manager on a successful campaign event. Sends a creational request to Mining Pool Broker in order of new Mining Pool creation.

Archived copy · cryp2nova.com

* **Pool Broker and Queue** - contains a queue of Mining Pool action tasks. Receives action tasks from Campaign Creation Service(on-submitted campaign); receives action tasks from Task Manager(on-suspend and on-cancel campaign); receives tasks from Mining Pools(on-finish campaign). Pool Broker and Queue service is linked with the Mining Pool Process Registry.
* **Mining Pool Process Registry** - contains a key-value mapper with ongoing campaigns, serves as a service of zero-time delay access to any created mining pool instance.

Operational Service always starts with initial state and contains current state that describes all available temporary data that helps to communicate between internal services.

Initial State should contain:
**TBD**

Current State should contain:
**TBD**

Operational Service has next communication interfaces:
* **Instance API** - handles all communications between microservices inside Birdchain Backend. Instance API never talks to users and to any third party services outside Birdchain Network. It serves message passing with all available SMS Mining Pools and with SMS Consumer Service.
* **Management API** - handles all communication between entire microservice and Administrative Service(not described on the scheme above). Serves for microservice configuration and monitoring during its work. Task Manager is a single instance that uses this interface.
* **Ethereum Proxy** - handles all requests to Ethereum Smart Contracts and listens to all specified events from Ethereum network. Campaign Creation Service is a single instance that uses this interface.

**Mining Pool Server**

Mining Pool Server contains all business model related to mining pool operations. This is an obvious component and should be done in the first milestone. As a central unit inside the service is a Task Manager. It contains internal data storage represented by a Repository and Hot and Cold key-value storages. There are two queues inside the microservices - workers queue and waiting workers queue.

It handles next infrastructural tasks:
* Request new bulk of workers from Phones On-Demand Producer microservice.
* Listen for the response of Phones On-Demand Producer with a batch of new agents.
* Place agents into a working priority queue and additionally into a waiting priority queue. One agent should be represented in a queue only once. After agent successfully send an SMS it should be removed from both queues or added to waiting priority queue.

Mining Pool Server consists of next components:
* **Task Manager** - handles all communication inside the microservice and receives incoming messages from Phone Agents, also it has all available functionality to remove and to send creational tasks to Phone Agents. Create tasks into SMS Broker & Queue service. Takes tasks from SMS Broker & Queue and send tasks to Phone Agents. It has Initial State and Current State. Updates Repository saving the current state to Storages. Takes users from Repositorys Working Queue. Store agents address with an assigned SMS task in SMS Broker & Queue.
* **SMS Broker & Queue** - contains all current tasks assigned to agents.
* **Phone Agents Manager** - listens to Task Manager and request new agents from Phone On-Demand Producer. Split agents into two priority queues - working queue and waiting queue based on users rating. Saves agent queues into Repository.

CRYP2NOVA

Archived copy · cryp2nova.com

* **Repository** - middleware between current state and two storages:
 * In-memory hot key-value storage
 * On-disk cold key-value storage

Both storages should contain identical information in case they are synced. Hot storage can be ahead of the cold storage, but it always should be synced later. Cold storage should be used when there is an emergency and hot storage become inconsistent.

Operational Service always starts with initial state and contains current state that describes all available temporary data that helps to communicate between internal services.

Initial State should contain:
**TBD**
Current State should contain:
**TBD**

Operational Service has next communication interfaces:
* **Instance API** - it serves message passing with all available Phone Agents and with Operational Service.
* **Monitoring API** - monitors all communication between entire microservice and Administrative Service(not described on the scheme above). Also deliver statistical information to SMS Consumer Microservice.

**Phone Agent**

Phone Agent contains all logic of miners operation and its related states. This is an obvious component and should be done in the first milestone. It handles next tasks:
* Sends requests to the linked mobile application with an SMS sending task;
* Waiting for the response from the mobile unit on message sent;
* Sends a receipt to the mining pool with a success / failure;
* One agent can work with multiple pool units;
* All tasks should be stored into a bag and taking from the bag randomly;
* Has initial state and current state;
* Has a hot key-value storage with all current state of the microservice in it.

Phone Agent consists of next components:
* **Task Manager** - receive tasks from Mining Pools, place tasks inside the bag. Serially takes tasks from the bag and send it to the mobile unit. Receives confirmation messages from the mobile unit and transfer them to the Mining Pool.

Phone Agent always starts with initial state and contains current state that describes all available temporary data that helps to communicate between internal services. This state should be stored inside Hot key-value Storage,
Initial State should contain:
**TBD**
Current State should contain:
**TBD**

Phone Agent has next communication interfaces:
* **Instance API** - serves message passing with all correspondent Mining Pools.
* **User API** - message gateway for Android device. One agent should have only one correspondent mobile unit.

**SMS Consumer Service**

Archived copy · cryp2nova.com

Service provided to buyers in order to create new campaigns. This is an obvious component and should be done in the first milestone. It handles the business model of a new campaign creation, campaign state actions and monitoring. It handles next tasks:
* Receives requests from SMS consumers to start a new campaign or extended existing one;
* Requests in-chain token and notify on a successful campaign creation in Ethereum Smart Contract;
* Listen on a successful / fail event response from Operational Manager;
* Reads current campaign state from Birdchain Ledger.

SMS Consumer Service contains of next services:
* **Task Manager** - handles all tasks inside SMS Consumer Service, send new campaigns on approval, provides statistical information on current and finished campaigns;
* **Campaign Listener Service** - listens to Ethereum Ledger events related to any of the current campaigns, transform it into a statistical information and provide to consumer applications;
* **Monitor Service** - provides information from consumers corresponded mining pools, transform it and send a final statistical information to the user.

SMS Consumer Service always starts with initial state and contains current state that describes all available temporary data that helps to communicate between internal services.

Initial State should contain:
**TBD**
Current State should contain:
**TBD**

SMS Consumer Service has next communication interfaces:
* **Instance API** - serves message passing with Operational Service;
* **Consumer API** - message gateway for mobile devices and web portal for SMS consumers;
* **Monitor API** - handles statistical information from Mining Pools;
* **Ethereum Proxy** - listens to new campaign creational event from Ethereum Smart Contracts.

**Birdchain Ledger**
Microservice provides a data storage functionality and contains all transactional information of Birdchain. This is an obvious component and should be done in the first milestone.

Birdchain Ledger consists of the next components:
* **Task Manager** - control unit of the ledger, saves and fetche transactions from the data storage;
* **Data Storage** - ledger with all transactional data;
* **Monitor Service** - logging unit with statistical abilities.

Birdchain Ledger has next communication interfaces:
* Sockets for microservice communications. Ledger is connected to SMS Consumer Service, Operational Service and Android Miner Applications
* JSON-RPC for third party services and mobile applications.

**Ethereum Operator**
Ethereum Operator is a transaction processing microservice. It sends requests to Ethereum ledger via Parity full Ethereum node laid inside the microservice. Also it listends events from it.

Ethereum Operator microservice contains the next components:
* **Parity Node** - full Ethereum client node.

Archived copy · cryp2nova.com

* **Sender Server** - sends requests to Ethereum Smart Contracts.
* **Listener Server** - listens to Ethereum events and transfer them outside.

Ethereum Operator has next communication interfaces:
* **Parity Node** - incoming and outgoing Ethereum connections;
* **Instance API** - access to Ethereum from outside of the microservice;
* **Event Generator** - event transferring gate. Microservices should connect via the socket to Event Generator and receive all incoming events.

**Ethereum Manager**
Ethereum Manager is a microservice contains specifications and their implementation in order to work with Ethereum Smart Contracts created specifically for Birdchain Network. Ethereum manager from the one side listens to Ethereum Operator for specified Ethereum Events, and from the other side sends requests via Ethereum Operator to Ethereum ledger.

Ethereum Manager microservice contains the next components:
* **Deque Manager & Contract Listener** - the service created with the reason to listen to events coming from Ethereum Operator and to enqueue them into the hot storage. Also it deque events from the hot storage sending them correspondent Birdchain microservices.
* **Contract Caller** - listens to incoming requests from Birdchain Network and pass them to Ethereum Operator. Listens for the OK response from Ethereum Operator.

Ethereum Manager has next communication interfaces:
* **Instance API** - access to Ethereum Manager inside Birdchain Network.
* **Socket Provider** - socket connections pool available in order to listen to incoming events.

**Wallets Service**
Wallet Service handles all users requests related to their ethereum addresses and exchanging abilities. Its a unified service both for SMS commodities providers(sellers) and SMS obligation providers(buyers). Wallets Service is similar to Ethereum Manager but handles only wallets business model of the Birdchain Network.

Wallet Service contains the next components:
* **Balances Manager** - contains all functionality related to Birdchain wallets, also contains local cache of all balances inside the networking including in-chain token, BIRD public token and local ETH balances. Tightly coupled with in-memory Hot Storage.
* **Contract Caller** - listens to incoming wallets coming from the service context (Balances Manager), then passing them through to Ethereum Operator.
* **Contract Listener** - similarly to Ethereum Managers Contract Listener handles all incoming wallet events from Ethereum Operator and passing them through to the Balance manager.

Wallets Service has next communication interfaces:
* **Instance API** - links the service with Android app. Handles all incoming user requests to the wallet service;
* **Socket Provider** - links all infrastructural microservices like Push Notifications and Email Notifications. Also handles incoming and outgoing requests to Ethereum Operator microservice.

### Smart Contracts Details and Principal Scheme

**TBD**
<p align="center">
  <img alt="Backend Principal Scheme"

Archived copy · cryp2nova.com

src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/Ethereum%20Smart%20Contracts%20Functional%20Scheme%20rev.2.png">
</p>
**TBD**

## Prioritization and Release Plan

### Prioritization

| Requirement ID | Title | Requirement Type |
|:---|:---|:---|
| | Provider Android App - Authentication/Authorization scheme - Telephone number and region registration within the system.| Function |
| | Provider Android App - Proof-of-Work - SMS gateway with cryptographical confirmation with the receipt. | Function |
| | Consumer Web Portal with a database - Authentication/Authorization scheme - two-factor authentication within email and text message confirmation. | Function |
| | Consumer Web Portal - Obligation creation process - new message campaign | Function |
| | Both applications - wallet/exchanger | Function |
| | Smart Contracts - glue between Provider Android App and a database | Function |
| | System Availability | Quality |
| | Communication Security | Quality |
| | System reliability and response time | Quality |

### Release Plan
The requirements were divided into three releases based on the prioritization and their dependencies. The three different releases were assembled so that each would work as a fully functional application.

In the first release the requirements that build up the foundation of the application were included, together with the most highly prioritized requirements and their dependencies.

# Unexpected Messages a.k.a. Promotional messages

## Definitions, acronyms, and abbreviations

* **Provider** - Ads content provider.
* **Receiver** - Ads content receiver.
* **Android & iOS Mobile Applications** - Application for mobile devices that acts as a messaging channels and Ethereum wallet.
* **Web Portal** - Browser interface for senders to provide a way to start ads campaign, escrow funds for it and to monitor the campaign status.
* **Administration Application** - Application that provides a possibility for the administrative staff to avoid illegal campaigns. Also it provides a statistical information on users activities. Should be available from browser.
* **Birdchain network** - A set of microservices and different kind of databases to provide the core of the network and various caches to the final network clients - mobile applications, smart contracts and web portals. Birdchain network contains all required data inside to be fully functional and completely separated from any infrastructural components and administrational user interfaces.
* **Ethereum Smart Contracts** - Backend services provided inside the Ethereum ledger. All of them are custom services created by Birdchain developers to give a possibility to get rewards and exchange tokens and provide escrows.

CRYP2NOVA

Archived copy · cryp2nova.com

# Birdchain Whitepaper

## MVP Overview
The Birdchain MVP is a messenger application with the ability to share ads content between content providers and content receivers. The client side application is divided into two parts.

* Mobile applications - iOS and Android apps - used by ads viewers
* Browser front end application - used by content providers
* Backend microservices - different services to provide viewers-to-providers communications and various infrastructural services like push notifications, ethereum services etc.

# Overall Description (Unexpected Messages)
This section will give an overview of the whole system. The system will be explained in its context to show how it interacts with other systems and introduce the basic functionality of it. It will also describe what type of users that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

## Product perspective
This system will consist of three parts: mobile applications - Android and iOS, one web portal for ads providers and a backend server - various microservices - with a database(s) and caches.

The web portal will be used for managing user specific information on profile and payment details and administrative access for selected users. Mobile applications has minimum amount of information provided to Birdchain and super-fast sign up consisted from only one step - user phone number should be provided.

Backend server will contain current Birdchain network state including analytics, statistical information, administrative actions and various caches.

## Tokenization
BIRD token is an initial Birdchain token created with a purpose to provide an initial coin offering and to use the token into an application as a supply commodity. This is a plain Ethereum ERC20 token with the ability to use onto crypto exchanges and can be stored into any Ethereum wallet.

In Birdchain environment users can exchange token into obligations with Birdchain itself  to spread ads into receiver channels - all existing users of mobile applications.

## Product Functions
Mobile application is a gate to Birdchain network. It acts as a messenger where user can subscribe to channels, every channel is a filtered topic of ads available in the network. Filtering by the content and by the location gives a possibility not just to earn by reviewing ads content but, but also to be subscribed to a useful content stream.

Ads content is a rich media with images and sound attached, all text is represented via HTML markup. User can view ads and get a reward. Also user can share it within Birdchain network or outside of Birdchain to Facebook, Twitter etc. Naturally every sharing should be paid to the user in BIRD token.

*E.x. Im a crypto evangelist and Im interested in earning some BIRD tokens by viewing ads or sharing it, but also my interests are about new computer hardware prices on the market and in visiting blockchain related events. Thats why I have a separate channel in Birdchain messenger with all related ads beside all other. Probably I have a lot of blockchain related friends and I will be happy to share just posted crypto event ads with them.

<p align="center">
  <img alt="Mobile App"

Archived copy · cryp2nova.com

src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/BIRD-Mobile-App-Mocks.png?raw=true">

Once user gets his BIRD token via viewing or sharing ads, the wallet inside the application can help to exchange BIRD into ETH and to withdraw it to any other Ethereum address. Based on the requirement of EIP20 token standard, users wallet is pretty straightforward and well known to all Ethereum users around the globe.

Thinking of future its natural to add a possibility to all users create its own ads right from the mobile application. Thats how a plain user can spend its BIRD tokens. This idea is well suited for a small business owners and public persons.

## Constraints
Since the interface is the same and is available on all phones - iOS and Android - there is no constraint in messenger usage. Since application doesnt fetch new data offline, it is crucial that there is an Internet connection for the application to function.

Both the web portal and mobile application should not be constrained by the capacity of the database. Meaning all content should be fetched when app is online and stored inside the database for the future offline review.

# User Flow
There are two types of the user - ads subscribers and ads producers. Both of them acting as a separate entities. But we specify that ads subscriber will be available to make actions belonging to ads producers in future releases.

## Ads Subscriber

  <img alt="User Flow. Ads Subscriber"
src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/Vector-UserFlow-AdsSubscriber.png">

Subscriber user flow starts from the sign up into the network. The only required field for sign up is a phone number. Birdchain will verify the phone number with SMS in future releases. Other options like email, gender, should be added optionally and have to be used for more targeted ads suggestion.

Next app should ask a user for a passcode or fingerprint if available. Its necessary to save valuable user information inside crypto secured storages - Keychain on iOS, Android Keystore System on Android.

Next user should select preferences for precisely targeted ads. Backend should provide ads content based on user preferences, his profile information.

Backend should create new public address for ETH and BIRD tokens. Both balances are available in settings for the work with them. User can withdraw BIRD or exchange BIRD into ETH and then withdraw ETH. User should be able to recover his account or create a new one with a mnemonic phrase. Note: mnemonic phrase never stored inside the application, this is a security reason and user should be warned.

Once all information has been entered, app started a socket connection with the server. Socket connection is only available when messenger feed is on screen. Otherwise connection will drain phone battery. Ads coming into the feed in the same fashion messages come into plain messenger applications.

Archived copy · cryp2nova.com

User can preview ads, for every preview user get a reward on his BIRD balance. User can respond to the ads, response will be available to ads provider. User can share ads within his social accounts - Facebook, Twitter, etc. For every ads sharing user should get a reward.
User can withdraw received tokens or change them into Ethereum token in a wallet section of the application.

Settings section should provide user with all required inputs to change feed preferences, to filter incoming push notifications, to view a help information, terms of services and privacy policy of the application, to change user profile information, link social networking accounts. In future releases it should be possible to create several accounts and link them under the super user.

2-factors authentication nowadays is a must and should be provided to the user later. When available, all user information should be changed using it. For example to use a combination of pin code and a code sent to another users device.

## Ads Provider

  <img alt="User Flow. Ads Provider"
src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/Vector-UserFlow-AdsProvider.png">

Ads provider is a user who is willing to show ads in a selected region for users in a selected targeting scope. Provider should pay to Birdchain system with BIRD tokens or ETH tokens in order to submit a campaign. Campaign should fit into Birdchains Terms of Services.
First step to become a birdchain ads provider is to sign up on browser application and provide a user related information like email, password, first and last name etc. On sign up Birdchain shows a mnemonic phrase or uses a mnemonic phrase provided by the user to identify and link Ethereum address to specified account. In case when user doesnt have a mnemonic phrase and accepts a new one, Birdchain should ask user to write it into a safe place in order to restore the account later.

After user have successfully signed in, he can start new campaign creation process. To begin a new campaign user should provide campaign details including transaction volume of the campaign. User should escrow BIRD tokens to his account, from this tokens receivers will get their reward for ads viewing or sharing. **There is no real transactions inside Birdchain network, but netting is used. Ethereum transactions happen only on withdrawal from Birdchain to external address.** This is how Birdchain can make transactions a way faster than in Ethereum ledger and avoid issues of paying the gas for every transaction inside the network.

All campaign ads should be provided as a rich media content consisted from HTML, image or even video and audio content.
Once the balance is less than one viewing fee, campaign stops. User can load more money to his address and proceed with the campaign, or to finish a campaign successfully.

User can withdraw BIRD or ETH tokens left after the finish of the campaign or after campaign has been stopped. In case user have been restored his account, all amount available in Birdchain is also available in his external wallets.

Additionally Birdchain can provide various cryptocurrencies as exchange functionality. Unfortunately there is no direct possibilities to exchange EIP20 tokens into other types of crypto beside connection to the external exchange API like Kraken or manually by system administrator.

# Specific Requirements (Unexpected Messages)

Archived copy · cryp2nova.com

## Netting
There are 3 types of token transactions inside the system:
* User escrow BIRD token in order to create a new campaign. This transaction requires ETH token to be converted inside BIRD and BIRD token is escrowed inside correspondent smart contract. As the other options ETH escrowed without exchanging it into BIRD. And as the third option Birdchain can escrow the amount user wants to use in Birdchain network and assign virtual BIRD tokens which will be used inside the system. Every payment to the developer is a netting operation where there is no transactions happen inside Ethereum ledger but all calculations lay inside of Birdchain backend. Either way should be chosen but the third one will drastically reduce the amount of gas spent on transactions. In such a case transactional gas will be used only when user escrow the amount and withdraw the spend or the reward.
* The second type is a payment to ads viewers and sharers. In order to reduce the amount of gas we propose to change Ethereum transactions onto netting operations between providers and viewers.
* The third one is a withdraw operation. For viewers it happens when they have collected enough BIRD tokens to withdraw or exchange into ETH token. For providers it happens when there is a spend after campaign has been finished, or campaign has been cancelled and provider wants to get his money back.

## Escrow

Ads provider should be able to begin the campaign and in order to make ads viewers sure their actions will be paid, provider escrow the required amount to the corresponded smart contract on the full length of the campaign.

## Battery Life

Battery lifetime is critical when we use sockets in mobile applications. Its very easy to drain the battery and force user to uninstall the application. Birdchain app should use socket connections only in the case of foreground mode with ads feed visible on the screen. All other communication should be provided as a plain REST queries. Also app should use push notifications to receive all valuable information requested by the user (all notification options are in Settings part of the app).

# Functional Requirements

This section provides all details on user interaction with the system. All user stories and possible use cases provided below with the reference to interface mockups.

## Backend Microservices Details and Functional Scheme

Here we provide backend decomposition and microservices overview. Core microservices included into Birdchain Network solution are:

  <img alt="Microservices"
src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/Vector-Backend-Functional-Scheme.png">

### Operational Service

**Warning: operational service should be architectured with a mind that it contains three different microservices - authentication, mobile messaging channels, campaign logic. This microservices should act separately and have to be splitted into three in future releases.

Archived copy · cryp2nova.com

Operational Service represents the core logic of the entire backend. *This is an obvious component and should be done in the first milestone.* It handles all campaign and mobile communication infrastructure tasks:
* New campaign creation
* Campaign cancellation
* Campaign finalization
* Queries Ethereum network to deploy new campaign instance smart contracts
* Listens to Ethereum events
* Handles Ethereum exceptions and react on them
* Contains internal data storage with all current campaign states
* Update Birdchain database with completed, failed and cancelled campaigns
* Handles mobile messaging channels.

Operational Service consists of next components:
* **Task Manager** - contains all business logic of the Operational Service. Reacts on incoming messages from other microservices, handles internal data storage and changes a current state appropriately to users actions. Operational Service can update database. Task Manager send tasks to Campaign Creation Service. Task Manager has a direct connection with future Network Administrators Service in order to monitor the status of all ongoing operations and suspend, cancel and resume all tasks on any emergency.
* **Campaign Manager** - serves to send new campaign creational requests to Ethereum Smart Contracts. Interoperate with Ethereum Manager Service; listens to Ethereum Manager on a successful campaign event.
Query Broker - a queue manager to avoid concurrent requests overloading. It can be any of battle-tested brokers - RabbitMQ, ZeroMQ, SQS etc.
* **Hot Storage** - plain in-memory key-value storage. All microservice state should stored in it and taken from it on the request. Because we use Elixir in a stack, its natural to use ETS (Erlang Terms Storage).
* **Cold Storage** - disk-based key-value storage. It should be used on process / service restarts to save data from Hot Storage and on data restore to Hot Storage. Elixir and Erlang stack provides DETS for it.
* **Repository API** - is a microservice that handles all incoming connections to Hot Storage. It manages communication between microservice and Hot Storage, it saves Hot Storage state to Cold Storage on a request or regularly within some discrete time.

Operational Service always starts with initial state and contains current state that describes all available temporary data that helps to communicate between internal services.

Operational Service has next communication interfaces:
* **Instance API** - handles all communications between microservices inside Birdchain Backend. Instance API never talks to users and to any third party services outside Birdchain Network.
* **Management API** - handles all communication between entire microservice and Administrative Service(not described on the scheme above). Serves for microservice configuration and monitoring during its work. Task Manager is a single instance that uses this interface.
* **Ethereum Proxy** - handles all requests to Ethereum Smart Contracts and listens to all specified events from Ethereum network. Campaign Creation Service is a single instance that uses this interface.

### OAuth 2.0 Authentication

Birdchain authentication built on top of OAuth 2.0 protocol. We decide to use OAuth as a battle-tested technology available and well known for all platforms. More details on OAuth protocol can be found on the link below:
[OAuth Specification](https://oauth.net/2/)

### Proxy Service

Archived copy · cryp2nova.com

Proxy Service represents a set of small microservices used primarily to handle incoming connections:
* **Router** - should be used to navigate incoming queries to corresponded microservices, each microservice has its own API. There is quite many APIs exposed to receive incoming connections thats why one routing point becomes very useful.
* **Logger** - its a good practice to log everything is coming and going out, collect statistical information and Birdchain should do this at one point instead of on every API endpoint.
* **Rate limiter** - rate limiter should be used to guard possible attach vector related to DDoS attack.
* **Load balancer** - we have to use load balancers on every cluster of hardware servers in the network to be sure we can spawn new instances on warnings related to the lack of resources.
* **TLS terminator** - we propose to terminate HTTPS traffic before Birdchain network. We define all servers as private and there is no possibility for the man-in-the-middle attack or any other chances to take out valuable user information.

Proxy Service should be used for all incoming connections from client side applications excluding direct Ethereum connections.

### Channel Builder

Every user communicates with Birdchain network with its own ads channel. We propose to link users mobile application with a socket connection when user is online - than we can send ads like in all other messaging apps - immediately on ads appear.

Channel builder represents a cluster of nodes linked to each other with a process registry. Every node runs thousands of processes that contain every users state - ads feed. Every user is always connected to such a process while hes online and application is in a foreground mode. On application turned into background or stopped modes channel should persist for a short time and self-destroy saving its state into cold storage. On new user session new channel should be created restoring its state from the cold storage.

The best choice for such functionality is a Phoenix framework built on top of Erlang Virtual Machine. EVM is designed to handle millions of concurrent connections and mainly used in messengers and telecoms.

For interprocess communication its a good option to use Process Groups 2 (PG2) library from Erlang environment.

### Cold Storage

To provide the best consistency of the data and fastest response, we propose to use relational PostgreSQL database for the cold storage. Beside its a battle tested database with marvelous amount of great features and extensions, there is a great support of it in Elixir community and it has libraries which are great in usage.

Database should be splitted into schemes according to microservices to provide multi-tenancy and future distribution and sharding - key features of the scale process.

### Hot Storage

As a hot storage for every microservice we propose to use Erlang native key-value in-memory storage ETS (Erlang Terms Storage) and its companion DETS as a filesystem backup of ETS. It has a native Erlang-Elixir support and provides super fast turnaround on reads and writes.

### Ethereum operator

CRYP2NOVA

Archived copy · cryp2nova.com

Ethereum Operator is a transaction processing microservice. It sends requests to Ethereum ledger via Parity full Ethereum node laid inside the microservice. Also it listends events from it.

Ethereum Operator microservice contains the next components:
* **Parity Node** - full Ethereum client node.
* **Sender Server** - sends requests to Ethereum Smart Contracts.
* **Listener Server** - listens to Ethereum events and transfer them outside.

Ethereum Operator has the following communication interfaces:
* **Parity Node**  - incoming and outgoing Ethereum connections;
* **Instance API** - access to Ethereum from outside of the microservice;
* **Event Generator** - event transferring gate. Microservices should connect via the socket to Event Generator and receive all incoming events.

### Ethereum Manager

Ethereum Manager is a microservice contains specifications and their implementation in order to work with Ethereum Smart Contracts created specifically for Birdchain Network. Ethereum manager from the one side listens to Ethereum Operator for specified Ethereum Events, and from the other side sends requests via Ethereum Operator to Ethereum ledger.

Ethereum Manager microservice contains the next components:
* **Deque Manager & Contract Listener** - the service created with the reason to listen to events coming from Ethereum Operator and to enqueue them into the hot storage. Also it deque events from the hot storage sending them correspondent Birdchain microservices.
* **Contract Caller** - listens to incoming requests from Birdchain Network and pass them to Ethereum Operator. Listens for the OK response from Ethereum Operator.

Ethereum Manager has the following communication interfaces:
* **Instance API** - access to Ethereum Manager inside Birdchain Network.
* **Socket Provider** - socket connections pool available in order to listen to incoming events.

### Administrative Service

Birdchain network should be monitored by administrative staff to avoid various illegal content. It should be done automatically by various filters and manually by accepting new campaigns and monitoring of their status. Also Administrative Service should contain all required tools to watch the health of servers and databases, receive information from Logger and statistical information from all other services. Administrator Service should use automatic tools to filter and cancel various malicious and illegal content sent by ads providers.

Administrative Service is an Elixir application running on a single node and providing REST API for administrative front end application. Administrative Service is communicating via REST with all other services, including database health checks and database content check.

### Front End Applications

Front End Applications are ReactJS browser apps linked via REST API to Birdchain API. On the first stage of the development only ads providers will use browser front end to create new ads, get ads statistics and transfer their funds.

There are two different front end applications in the network:

Archived copy · cryp2nova.com

* Ads provider application
* Administrator application

Both applications linked to their own API endpoints via REST protocol. For the local cache in browser we assume the usage of Redux and Local Storage. Same as mobile applications browser apps will use OAuth 2.0 protocol to establish stateless sessions with the server. In order to update information reactively we propose to use socket connections in future releases.

Design mockups should be provided before starting to build any of them, were following a convention - design first. In order to provide the best user experience Birdchain should have its own set of user interface React components library. All of them will be used in container components.

Administrator Application should contain features required for monitoring and handling all cold storages available in Birdchain. Every storage should have its own component and interface to manage users data.

Administrative Application should have an interface to confirm new campaigns. Campaign should be confirmed only in a case when there is enough BIRD or ETH token sent to Escrow Contract.

Administrative Application should have all statistical information from microservices on user load, ip locations and others.

### Mobile Applications

There are two types of mobile applications - for Android devices and for iOS devices. We propose to use reactive architecture and Model-View-View-Model architecture for both of them in order to provide real time updates fetched via socket connection and plain HTTP REST endpoints. Socket connection should be used for ads stream only in order to save users battery lifetime.

iOS technology stack should be based on top of Swift programming language and contain the following technologies:
* ReactiveX (RxSwift) for the reactive architecture
* Core Data (SQLite3) for the local cache and permanent storage
* Core Location to determine users location for a better ads targeting
* Alamofire for plain HTTP requests
* Facebooks Socket Rocket for socket connections.

Android technology stack should be based on top of Kotlin programming language and contain the following technologies:
* Architecture Components to provide MVVM architecture into Android environment
* Room for the local cache and permanent storage
* Volley for HTTP requests
* Socket.io for socket connections.

## Smart Contracts Details and Functional Scheme

Birdchain network uses Ethereum ledger to provide users with a default payment gate based on ETH and BIRD tokens done on ERC20 (EIP20) standard. Birdchain is linked with Ethereum ledger via Operational Service and User Wallets. Smart contracts development should be provided by Solidity developer and fully tested manually and by providing automated unit testing use cases. Its a requirement for any contract deployed into a blockchain ledger to be sure that even stochastic bugs are not presented in a code. Once contract has been deployed, there is no possibility to change or update it. Its possible only to deploy a new contract with a new address and all state variables are empty. Such

Archived copy · cryp2nova.com

upgrades is a usual case of issues related to the next balance airdrops, balance lost or funds locks. Smart contract development should be done only by a highly skilled developer with obvious code review.

  <img alt="Smart Contract"
src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/Vector-Eth-Smart-Contracts.png">

### Balance Storage Contract

Balance storage contract is a business model and its solidity realization that handles all current balances inside Birdchain network. Usually its a simple mapper between addresses and values linked to addresses. All balance related operations - transactions and getters should be settled into the contract too.

### Campaign Creation Factory Contract

Campaign creation factory is a contract which handles all required actions to deploy a new campaign contract instances into Ethereum network escrow contract instances. Every campaign has two separate smart contracts for handling all of transactions.

### Ads Campaign Contract

All campaign related functionality including related information data structures and various getters. Campaign contract is a parent contract for the Escrow contract and is able to act on users behalf and destroy both Escrow contract and itself.

### Escrow Contract

All escrow related functionality including withdraw, balance storage and various getters. Escrow contract is a mapper between campaign address and deposited tokens.

### Proxy Contract

Proxy contract acts as a middleware between all Birdchain contracts and incoming requests. Proxy contract is a safety action in order to not to change initial addresses for backend and frontend queries to Ethereum. Once Birdchain dev team decide to upgrade the contract, there is no possibility to leave initial contract address and simply swap the contract on a newer version. This is why we have a mapping between addresses available for the public and internal Birdchain contracts addresses.

### Token Contract

Token contract is a standard EIP20 (ERC20) contract created by Birdchain for the ICO with a BIRD symbol. Birdchain network will use BIRD for all transactions, BIRD is available in any Ethereum wallet.

## Operations on Smart Contracts and required toolset

The decision to use standardized ERC20 token gives a possibility to work with plain Ethereum wallets and with a toolset created for Ethereum. Its straightforward to start using Web3JS library just from the browser using Metamask app (available on Chrome and Firefox) to directly communicate with Ethereum ledger and send transactions.

Various Ethereum browsers like Mist and Parity are also available to communicate with Ethereum, they have all required

Archived copy · cryp2nova.com

toolset from the scratch, including Ethereum wallet and a full Ethereum node.

Android applications can communicate with Ethereum directly using Web3J library, its similar to Web3JS but is created on top of Java.
Development toolset should include Truffle framework, all contracts should be unit tested using JS testing environment - Mocha and Chai. All testing use cases and user stories for acceptance tests have to be described before smart contracts development started.

# Server Setup

In this section we describe rough setup of the server environment. This environment includes microservice clusters of nodes, required load balancing and various endpoints.

**TBD**

# Prioritization and Release Plan (Unexpected Messages)

## Prioritization

| Requirement ID | Title | Requirement Type |
|:---|:---|:---|
| | Android Messaging Application | Function |
| | iOS Messaging Application | Function |
| | Ads Provider Web Portal | Function |
| | Operational Service | Function |
| | Ethereum Smart Contracts | Function |
| | System Availability | Quality |
| | Communication Security | Quality |
| | System reliability and response time | Quality |

## Release Plan
The requirements were divided into three releases based on the prioritization and their dependencies. The three different releases were assembled so that each would work as a fully functional application.

In the first release the requirements that build up the foundation of the application were included, together with the most highly prioritized requirements and their dependencies.

# Token sale
Token sale will be implemented in several steps:
* Closed pre-sale (2018.01.01-2018.01.18)
* Open pre-sale (starting 2018.01.18)
* Main ICO event (2018.03.01-2018.05.31)

## Soft and Hard cap's

Soft cap of the ICO - **1,500 ETH**
Hard cap of the ICO - **10,500 ETH**

## Token distribution

Archived copy · cryp2nova.com

* Maximum coin supply **580,263,158**
* Total ICO distribution **441,000,000**
* Team and founders **52,223,684**
* Involving corporate contributors and other expenses **87,039,474**

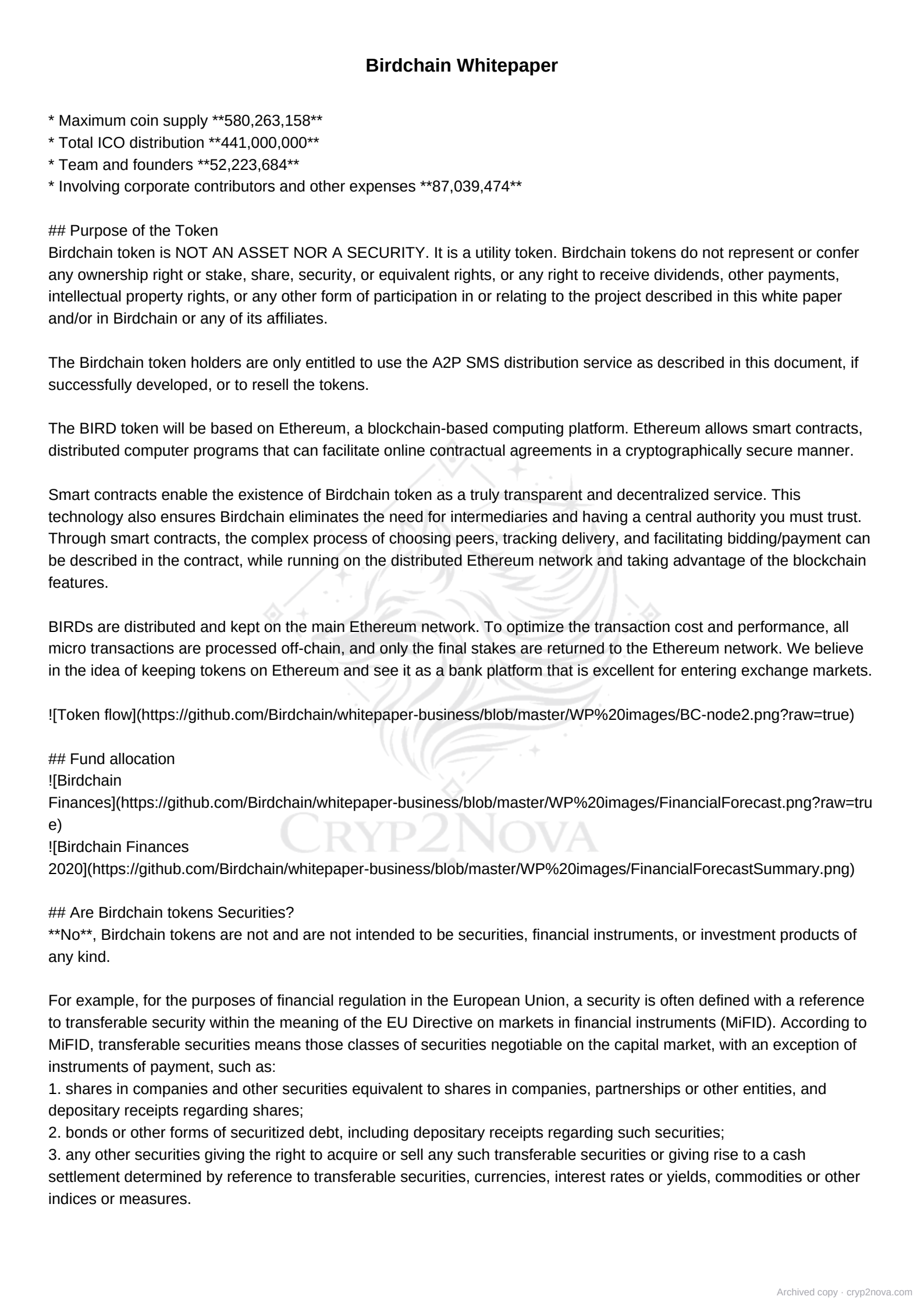
## Purpose of the Token
Birdchain token is NOT AN ASSET NOR A SECURITY. It is a utility token. Birdchain tokens do not represent or confer any ownership right or stake, share, security, or equivalent rights, or any right to receive dividends, other payments, intellectual property rights, or any other form of participation in or relating to the project described in this white paper and/or in Birdchain or any of its affiliates.

The Birdchain token holders are only entitled to use the A2P SMS distribution service as described in this document, if successfully developed, or to resell the tokens.

The BIRD token will be based on Ethereum, a blockchain-based computing platform. Ethereum allows smart contracts, distributed computer programs that can facilitate online contractual agreements in a cryptographically secure manner.

Smart contracts enable the existence of Birdchain token as a truly transparent and decentralized service. This technology also ensures Birdchain eliminates the need for intermediaries and having a central authority you must trust. Through smart contracts, the complex process of choosing peers, tracking delivery, and facilitating bidding/payment can be described in the contract, while running on the distributed Ethereum network and taking advantage of the blockchain features.

BIRDs are distributed and kept on the main Ethereum network. To optimize the transaction cost and performance, all micro transactions are processed off-chain, and only the final stakes are returned to the Ethereum network. We believe in the idea of keeping tokens on Ethereum and see it as a bank platform that is excellent for entering exchange markets.




## Fund allocation





## Are Birdchain tokens Securities?
**No**, Birdchain tokens are not and are not intended to be securities, financial instruments, or investment products of any kind.

For example, for the purposes of financial regulation in the European Union, a security is often defined with a reference to transferable security within the meaning of the EU Directive on markets in financial instruments (MiFID). According to MiFID, transferable securities means those classes of securities negotiable on the capital market, with an exception of instruments of payment, such as:
1. shares in companies and other securities equivalent to shares in companies, partnerships or other entities, and depositary receipts regarding shares;
2. bonds or other forms of securitized debt, including depositary receipts regarding such securities;
3. any other securities giving the right to acquire or sell any such transferable securities or giving rise to a cash settlement determined by reference to transferable securities, currencies, interest rates or yields, commodities or other indices or measures.

Archived copy · cryp2nova.com

BIRD does not qualify as transferable security for the above purposes.

BIRD, as a utility token, will only provide opportunity to access Birdchain services. The aim of the BIRD is to make it possible to exchange SMSs between the people wishing to access them and the people wishing to dispose extra SMSs.

The BIRD has a specific utility use case. It serves a crucial role in settlement-making with a potentially very large number of mobile phones on the Birdchain network by allowing a fast and open blockchain-based transaction protocol that would not be possible using fiat currency. If the BIRD was not based on blockchain technology, it would be similar to purchasing a ticket or any other document that proves the person may use a particular service.

## Please Note
In this white paper, references to Birdchain, we, us or our are to [Birdchain OÜ](https://www.e-krediidiinfo.ee/14362929-BIRDCHAIN%20O%C3%9C?lang=en), a company incorporated under the laws of Estonia with registry code 14362929, and references to BIRD or Birdchain token are to blockchain tokens created by Birdchain in connection with the project described in this white paper. Sometimes, the term Birdchain has been used to refer to the service proposed to be developed by us.

We will announce the date of BIRD sale and all related details on our official website: www.birdchain.io. The rights and obligations of the holders of BIRDS and Birdchain as the issuer will be governed by the terms and conditions of the Birdchain token, which will be available on the same website, and the Estonian law. Please carefully follow the instructions stated on the website to avoid being scammed. Birdchain tokens are not securities or any other form of investment in any jurisdiction. The sale of tokens should not be considered offering of securities or as investment advice, solicitation, or an endorsement of any kind. Any persons deciding to acquire Birdchain token must do this at its own risk and discretion following a thorough analysis of the risks associated.

Nothing in this whitepaper or in any other materials put forth by Birdchain or any other third person acting on behalf of Birdchain should be considered a guarantee that the proposed business ideas will be a success nor are there any guarantees the tokens could be sold. Specifically, no promises on the value of the token are made.

All forward looking statements presented in the whitepaper or in any other materials put forth by Birdchain or any other third person acting on behalf of Birdchain are subject to risks and uncertainty and the outcomes might differ from the forward looking statements made. Birdchain does not take any responsibility for updating any forward looking statements made, should any uncertainties materialize.

Any citizens and residents of and persons in Canada, or the United States of America (USA), Puerto Rico and the Virgin Islands, and any other USA possessions, or any countries where the sale of Birdchain tokens would constitute a public offering of securities or the sale of tokens would be restricted are prohibited from participating in the sales process of Birdchain tokens and from receiving the tokens. Representatives and individuals acting in the interests of legal entities registered in the referred jurisdictions are prohibited from participating in the sales process of Birdchain tokens and from receiving the tokens. Each potential acquirer of a Birdchain token must ensure itself that the acquisition, possession, and use of the Birdchain tokens comply with all applicable laws and regulations, including any requirements for the sale and acquisition of tokens applicable in the relevant jurisdiction.

Birdchain does not make and hereby disclaims, any representation, warranty, or undertaking in any form to any entity or person, including any representation, warranty, or undertaking in relation to the accuracy and completeness of any information set out in this whitepaper. Birdchain reserves the right to amend the white paper until the ICO starts.

<p align="center">
  <img alt="Roadmap"

Archived copy · cryp2nova.com

# Birdchain Whitepaper

src="https://github.com/Birdchain/whitepaper-business/blob/master/WP%20images/Roadmap.png?raw=true">
</p>

# Appendix
* How off-chain trading will work by Martin Koeppelmann
* Ethereum Lightning Network and Beyond by Robert McCone
* State Channels by Jeff Coleman
* Raiden: Scaling Out With Off-chain State Networks by Heiko Hees
* Epicenter Bitcoin: State Networks by Jeff Coleman
* Universal Payment Channels by Jehan Tremback
* Scalability via State Channels by Martin Koeppelmann
* Ethereum: Platform Review (page 30) by Vitalik Buterin
* Sparky: A Lightning Network in Two Pages of Solidity by Dennis Peterson
* An Introduction to State Channels in Depth by Ameen Soleimani
* State Channels Wiki by Jeff Coleman
* Gamble Channels: Fast Verifiable Off-Chain Gambling by Denis Peterson
* Toy State Channels Jeff Coleman
* Raiden Network by Brainbot
* Machinomy by Sergey Ukustov
* http://plasma.io/plasma.pdf
* https://iota.org/IOTA_Whitepaper.pdf
* And many MANY others.

CRYP2NOVA

Archived copy · cryp2nova.com