MY WISH

Making Contracts Really Smart. For Life.

Author: Vladimir Tikhomirov
Revision: 1.3.02

Contents

Executive summary

Currently, there are over 15 millions of Bitcoin wallets from which more than a quarter of all existing bitcoins were lost forever. In the world of cryptocurrencies there are no mechanisms for managing the funds in various life circumstances, such as in the case of lifelong annuities, marriage contracts, death or the loss of key for the wallet.

MyWish platform is dedicated to solving the issue. The platform allows creating and executing the rules for distribution of funds using smart contracts and decentralized environment for calling them.

Our mission is to bring common money-related practices into the crypto world: automatic regular payouts, wedding agreements, wills, lost wallet keys protection etc.

We make using crypto assets safe and convenient.

From the user's point of view, MyWish is a mobile application and a website where anyone can easily create a smart contract and then manage it. The simplest contract consists of three parts: wallet (source), conditions, and wallet (destination). The conditions determine, when the transfer of funds must be done. The platform supports full life cycle of a contract: creation, starting, monitoring and completion. It provides the tools for management and customization of personalized contracts.

Unlike other projects dedicated to smart contracts in general, MyWish platform focuses on the crypto funds management under various life circumstances. Main features of the platform are:

- the platform implements the decentralized environment for calling smart contracts, ensuring performance of the contract even in the absence of the platform;
- User-friendly interface, multiplatform
- integration of external developers contracts is implemented;
- transfer of crypto assets to heirs' traditional banking accounts integration.

The service is available at: https://contracts.MyWish.io/

Introduction

Gaining access to a family member's bank account after their death is reasonably easy: every jurisdiction has clear rules governing probate of assets and unless a dispute arises (usually between the family members), the process is straightforward.

Token wallets are not bank accounts and they are regulated by digital media rules: in accordance with the law, individuals can leave instructions for their intellectual property and digital media in case of death. For most people, emails, music, photos, Facebook and Instagram accounts will constitute the majority of our digital legacy; but if accounts are locked, problem of passing the tokens to our dears or friends in case of death or serious injury does not have a convenient solution.

Let's consider how family members can receive tokens. If a bitcoin wallet was left "open" and if family members know about it, they can transfer the money out with just a few clicks. However, this works only in the case of the family being aware of its existence. Spouses or relatives may inspect smartphones or computers for valuable data, but very few people will look for bitcoin wallets. What's more, bitcoin wallets can be anywhere.

Moreover, passwords that allow direct access to valuable information such as bitcoin wallets should not be kept in wills, but that does not mean that this cannot be done indirectly. Passwords can be stored somewhere safe, while the will can be used to pass down instructions. [1]

Smart Contracts Background

Starting with Bitcoin (BTC), digital currencies have allowed money to be moved over the Internet without the need for intermediaries. This is possible by the use of public registers of information about transactions that are copied many times and updated to the blockchain.

The special computer software on the blockchain, which will be executed by a network of computers, is called smart contracts. Smart contracts are the key to unlock the world, where any computer-oriented task can be performed completely autonomously and correctly, without fear of external manipulation or imitation.

The main problem for many users when working with services such as PayPal is that the service administrators have too much authority to manage the account or asset stored in their system, which allows them to freeze accounts and cancel transactions for any reason. But, using smart contract technology, these small financial tasks become available to all who can use it in a fully automated, independent way, which writes these tasks down directly to the blockchain. This technology excludes the possibility of closing banks or financial institutions and changing, deleting transactions, and also ensures that each payment will be made exactly as indicated, without any risk of misuse or malicious interference.

Although the ways of setting up simple smart contracts exist for many digital currencies, after the launch of the Ethereum in 2015, it became possible to program the Turing complete smart contracts. The Ethereum is a computer network based on the blockchain, in combination with the platform currency (ether) implemented a powerful distributed computing environment called the Ethereum Virtual Machine (EVM). Within the framework of the platform, the programming language for writing smart contracts, focused on EVM, called Solidity, implements the usual high-level programming concepts that allow you to write complex smart contracts. However, these smart contracts are supposed to be written by programmers, and that makes them available to a limited number of users who understand the code. [2]

Although smart contracts are unambiguous in their meaning, only those who are familiar with the programming language can understand them, so, it makes it difficult to reach an agreement with any party that does not have technical knowledge.

There are also other problems with smart contracts: even when they are written by capable developers with good intentions, they can contain errors that lead to problems in work and even to loss of assets. For example, DAO Hack led to a direct loss of 3.6 million ETH. A well-tested smart contract would not have such vulnerability and would not allow an attacker to withdraw someone else's funds.

Companies strive to use smart contracts in their work, correcting their various faults. Without easy-to-use, proven and reliable methods to effectively create accurate smart contracts, their widespread adoption will never happen. Simply put, smart contracts will allow you to use advanced locking options for the widest audience.

Mission Statement

Our mission: to provide a service for funds management in case of different life events, in safe and convenient way.

Platform description

Development of the project started from LastWill project, and in course of investigating the task of management of savings, the necessity for a broader approach to the problem became obvious.

Logically, the platform may be represented by the following parts:

1. A set of customizable smart contracts
2. Interface for smart contract creation, customization, and managing
3. Joule decentralized system for regular calling of smart contracts and payment of bonuses for the call
4. Integration module for operation with non-Ethereum cryptocurrencies
5. Interface for downloading and verification of smart contracts from external developers and the system of payments for usage of third-party smart contracts.

Let us consider each part of the platform in more detail.

Customizable smart contracts.

Customizable smart contracts are the basis of the platform enabling creation and execution of user instructions for managing of the funds.

Smart contract in MyWish may have the following states:

- **Created** - the user's parameters (wallet address, amount, etc.) are specified in the contract template;
- **Paid**, the internal state of the platform meaning that this contract can be deployed in Ethereum
- **Started**, the contract was started in Ethereum network and waits for calls for checking conditions
- **Completed**, as a result of a call of the contract, the conditions have been confirmed and the transfer of funds was performed
- **Cancelled**, the user cancelled the execution of the contract, the funds have been remitted to his/her account
- **Overdue**, the term of monitoring the contract has expired, the conditions of the contract were not fulfilled, the user may either return the funds or extend the contract

According to the project roadmap, the first contract will be Lastwill (will), which distributes funds in the absence of activity of the wallet, and confirmation by the trustees to the user of the occurrence of the event. Let us consider this contract in more detail.

Types of "Lastwill contracts"

This section provides conditions and actions for «Lastwill contracts".

Condition type «A».

Contracts initiated due to activity / inactivity of the user.

- User's wallet has no activity for more than N* days
- User's wallet has no outcomes for more than N days
- User does not confirm activity within N days
- The user does not visit certain resources for N days (for example, in social networks and profiles there is no activity)

*N – amount of days, set by user while contact creation

Condition type «B».

When creating a contract, the user has the option to choose confidant(s) who can confirm / deny the occurrence of certain events. In this case, the user can configure a different number of necessary confirmations / no response / denials (for example, a contract is considered confirmed if at least 50% of the confidants confirmed the fact of the event and no more than 10% disproved).

Type «B» can be combined with type «A». For example, in the case of user inactivity, a survey of trustees occurs, which confirm that the event has occurred.

Type of action #1.

The main action is the translation of the crypto currency. One or more wallets specified by the user.

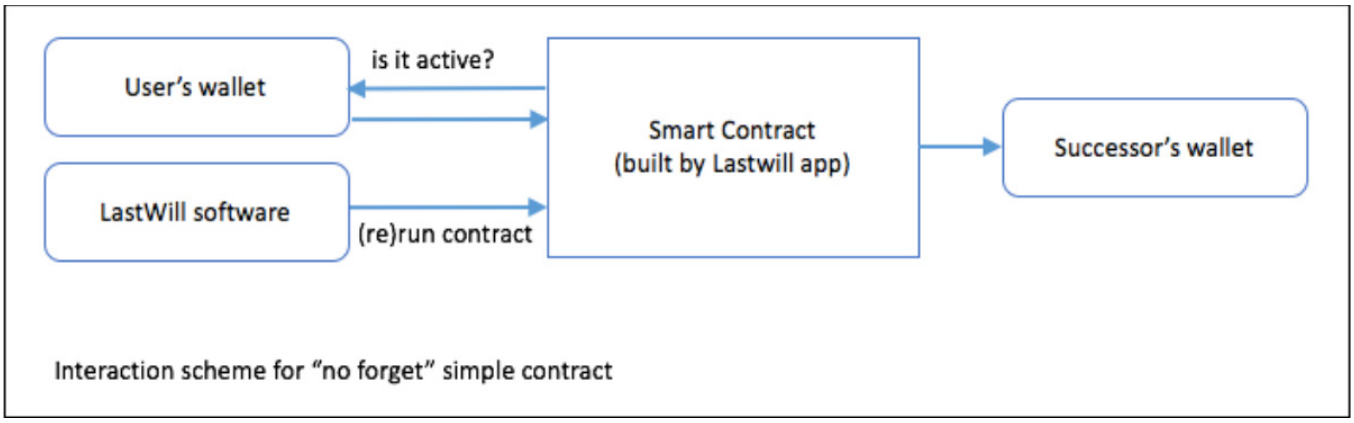
Type of action #2.

Sending of encrypted information, which can be opened only by the recipient. For example, a pin code from a user's card. This mechanism still requires elaboration.

Summarizing this section, it should be noted that any combinations of types of conditions and types of actions are allowed. For example, within one will contract there may be transfers of funds and information on various conditions.

Lastwill contracts implementation

We worked out several options for implementing contracts of last will. In this section we will consider several of them.



Interaction scheme for "no forget" simple contract

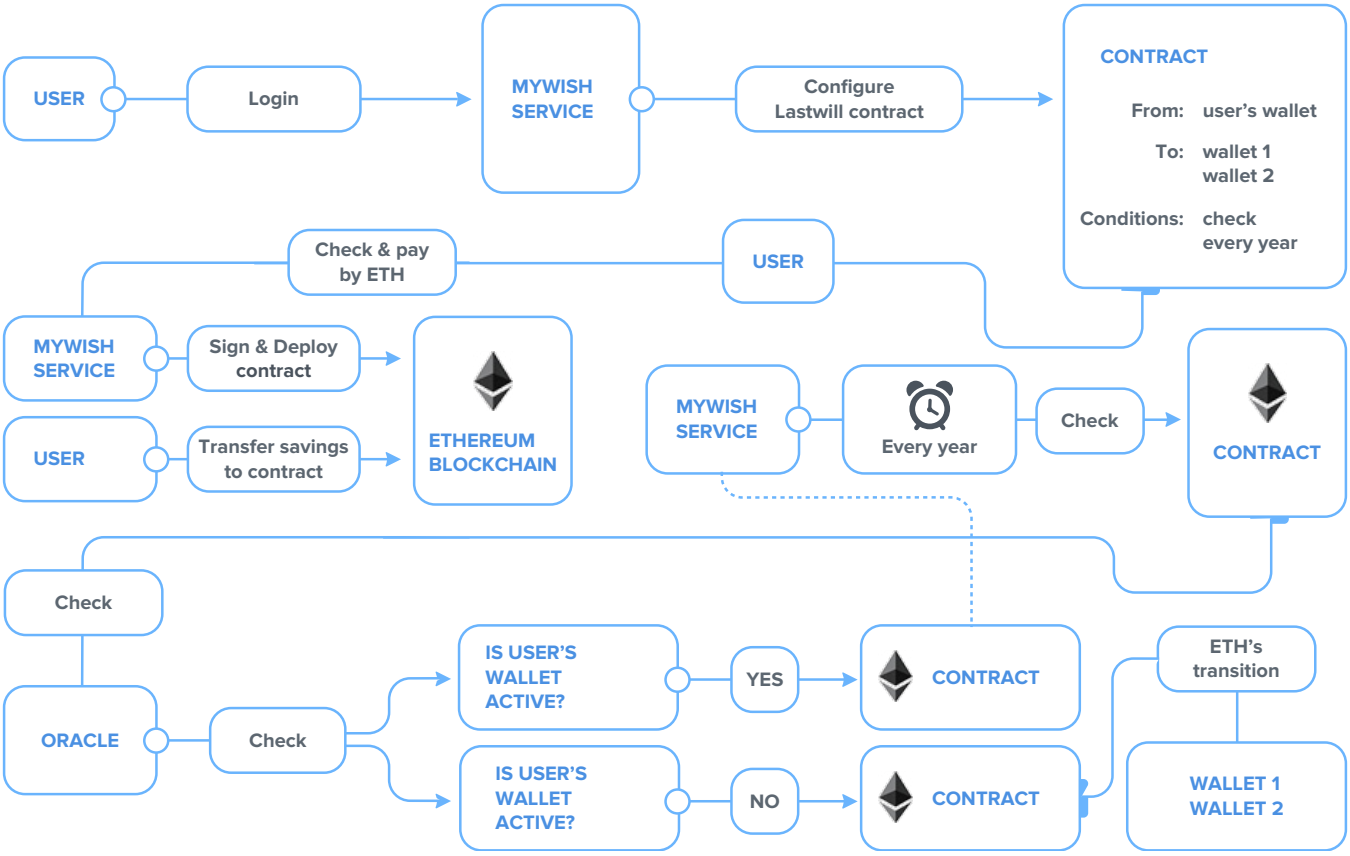
At the moment, the priority in the development is given to the concept #1, but, in the future, other concepts can be implemented also.

Concept #1. Common wallet.

In this concept, to use the last service, the user needs a wallet (it's address).

Briefly the scheme of interaction with the service can be presented in the form: Fill the wallet address, conditions, recipients - the generation of the contract - the transfer of funds to the contract – contract execution.

More detailed scheme for contract with activity analysis is shown below.



Let's consider transitions in more detail:

1. The user enters our service (by login / email).
2. Enters the address of his wallet, the addressees of the transfer and the amount (or %), configures the terms of the contract
3. Make payment for the creation of a contract in WISH tokens
4. The service generates a contract (source code - .sol file). The user can additionally verify the contract before launch.
5. The contract is signed by MyWish, while the function of stopping the contract can only be called by this user (MyWish can not stop it).

6. After this, the contract will be executed in Ethereum

7. According to the schedule specified at contact's terms, service MyWish calls method of the contract for conditions check.

8. The contract uses oracles for checking wallet's activity and transfers money, if the conditions are met.

Or, for example, user confirms his availability by himself (call the contract method) and thereby prolong the waiting time for the contract.

The disadvantage of this solution (as in all solutions with oracles) is dependency on external services to verify the conditions. In the future, no less than three sources will be used for activity of the wallet monitoring.

Concept # 2. Contract-wallet.

For example, on the basis of the MIST browser [4] we make the code changes by adding functions that implement the lastwill of the user.

Thus the user in a usual mode uses a wallet-contract, connecting it to the standard interface. And also has the ability to configure the transfer of funds when events occur.

The advantages of this solution is that to control the wallet activity, there is no need for external services, since all payments go through the same contract.

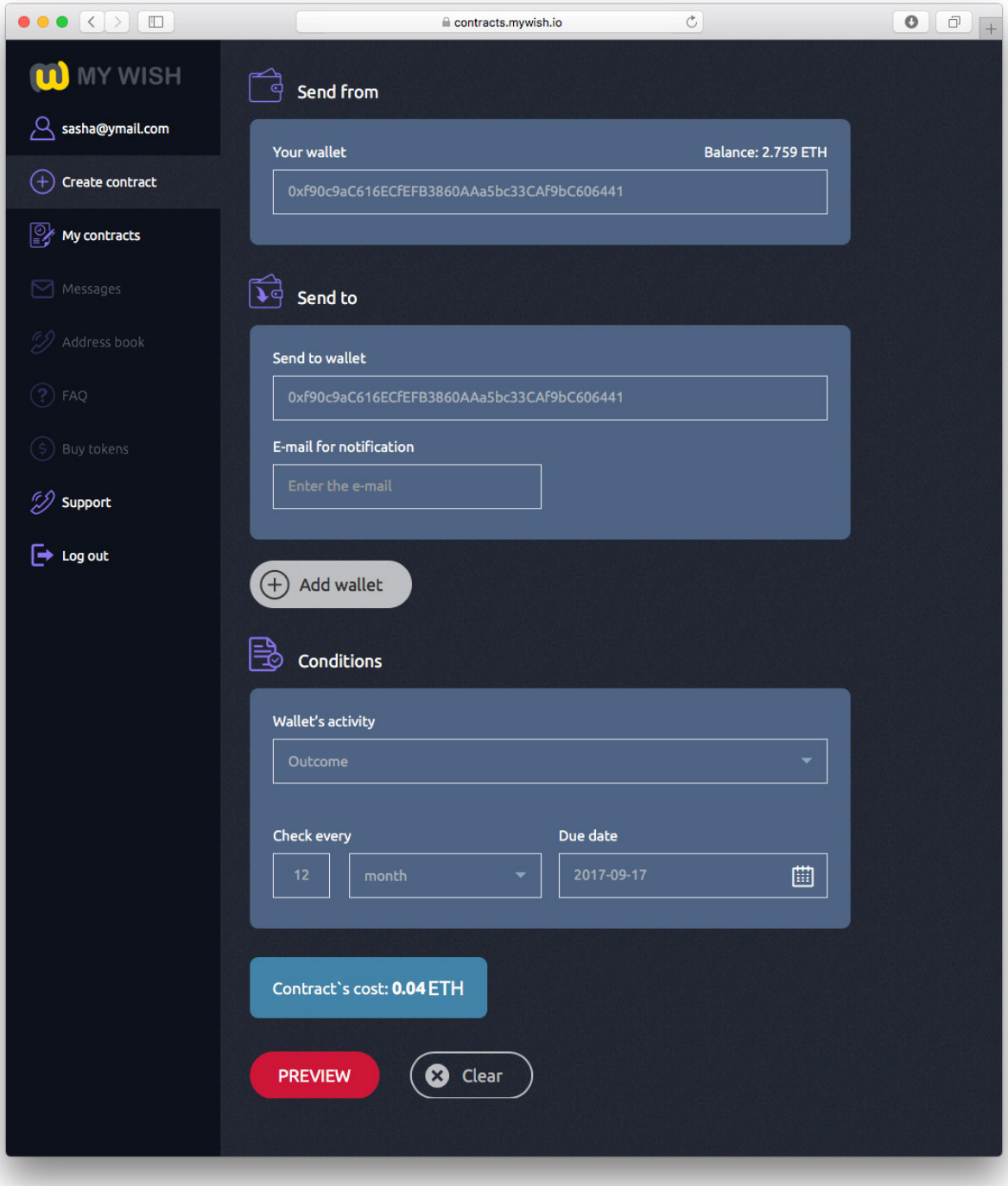
The disadvantages of this solution the relevance of the versions (MIST & Lastwill code) of the contract and the small distribution of wallets-contracts among users.

Interface

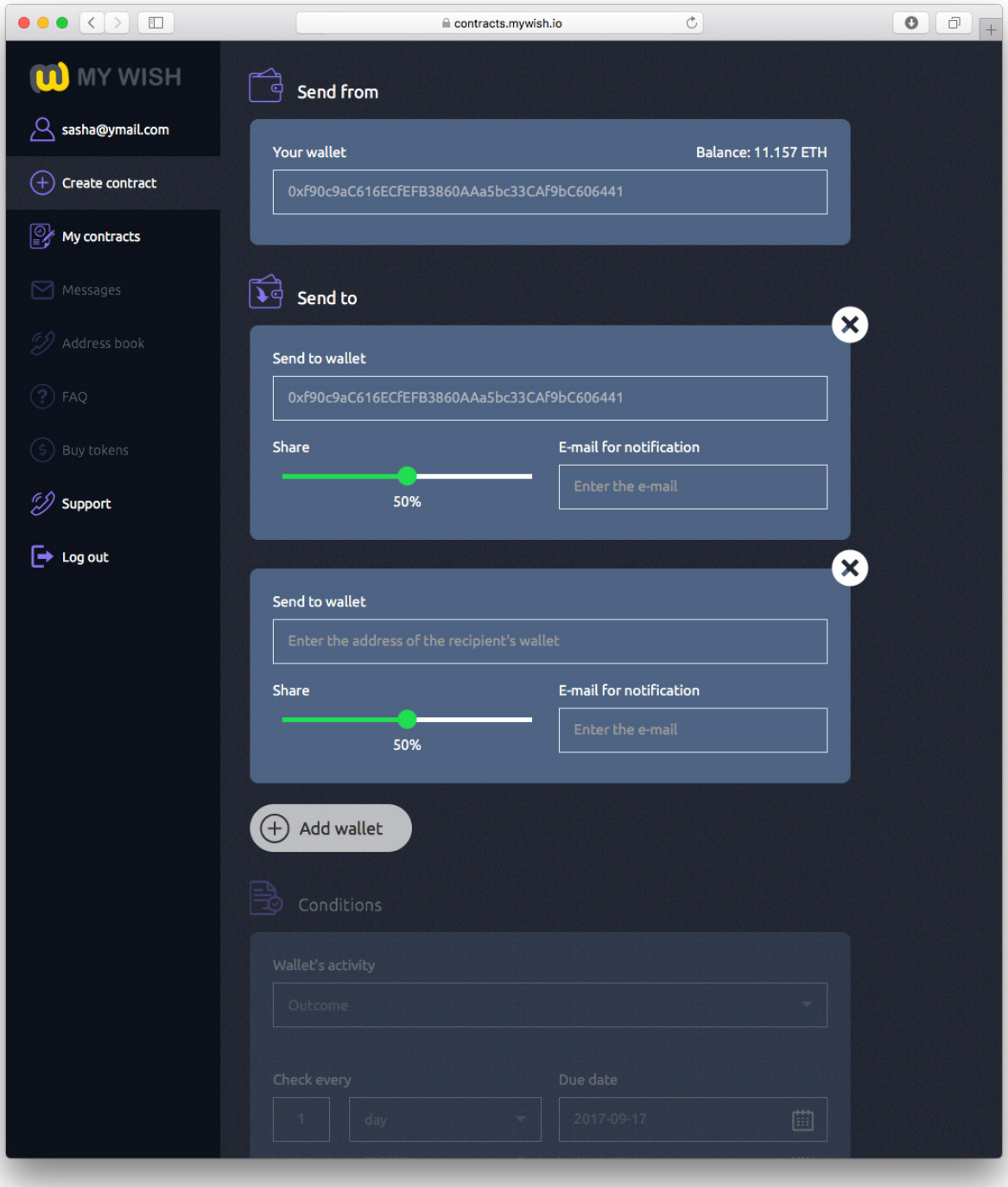
There are three planned applications for users. Both of them are mobile applications (iOS & Android), the third – web application (will be available on lastwill.io portal).

Alfa version is available on: https://contracts.MyWish.io/

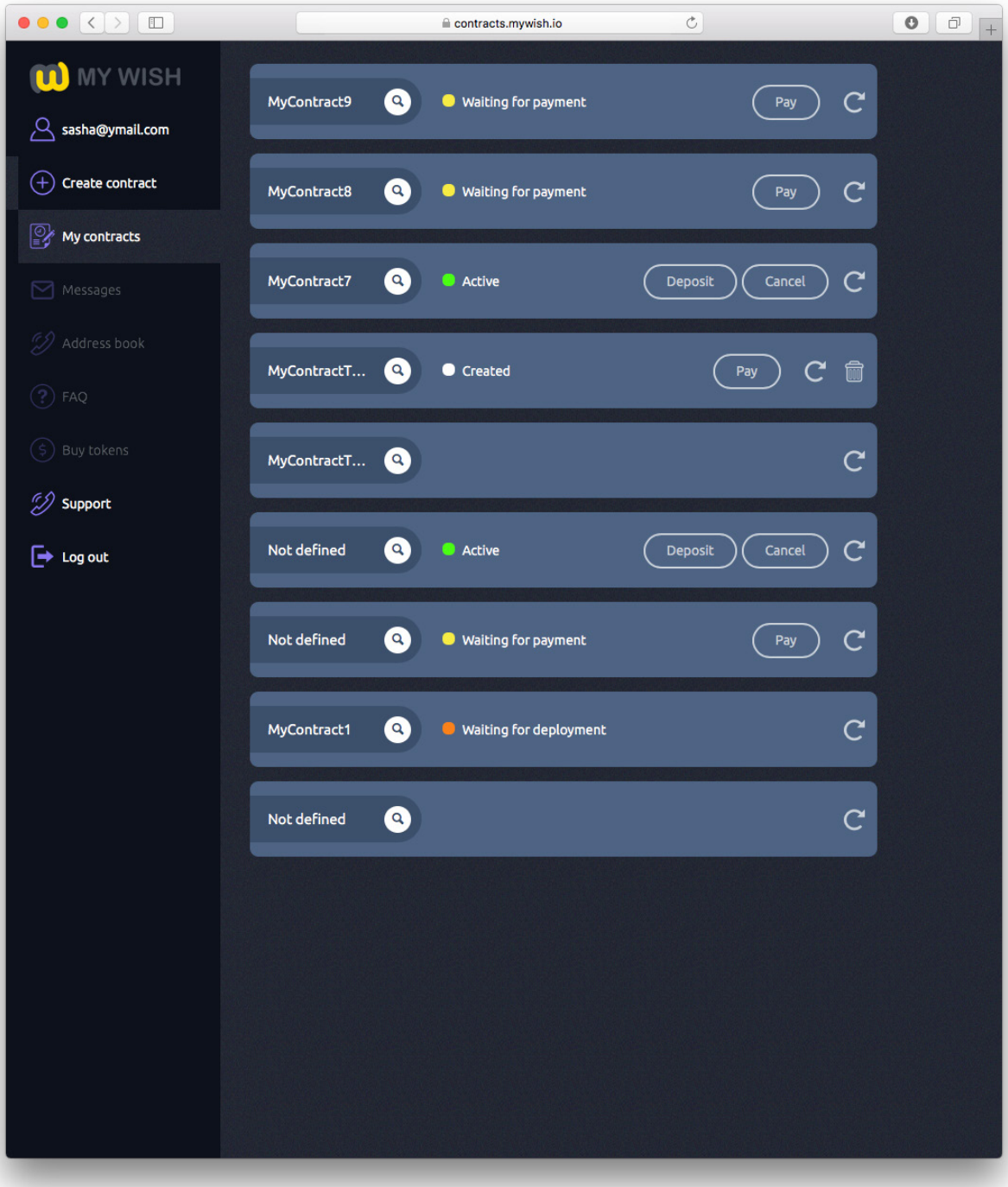
The interfaces for simple contract creation are as follows (for community review):



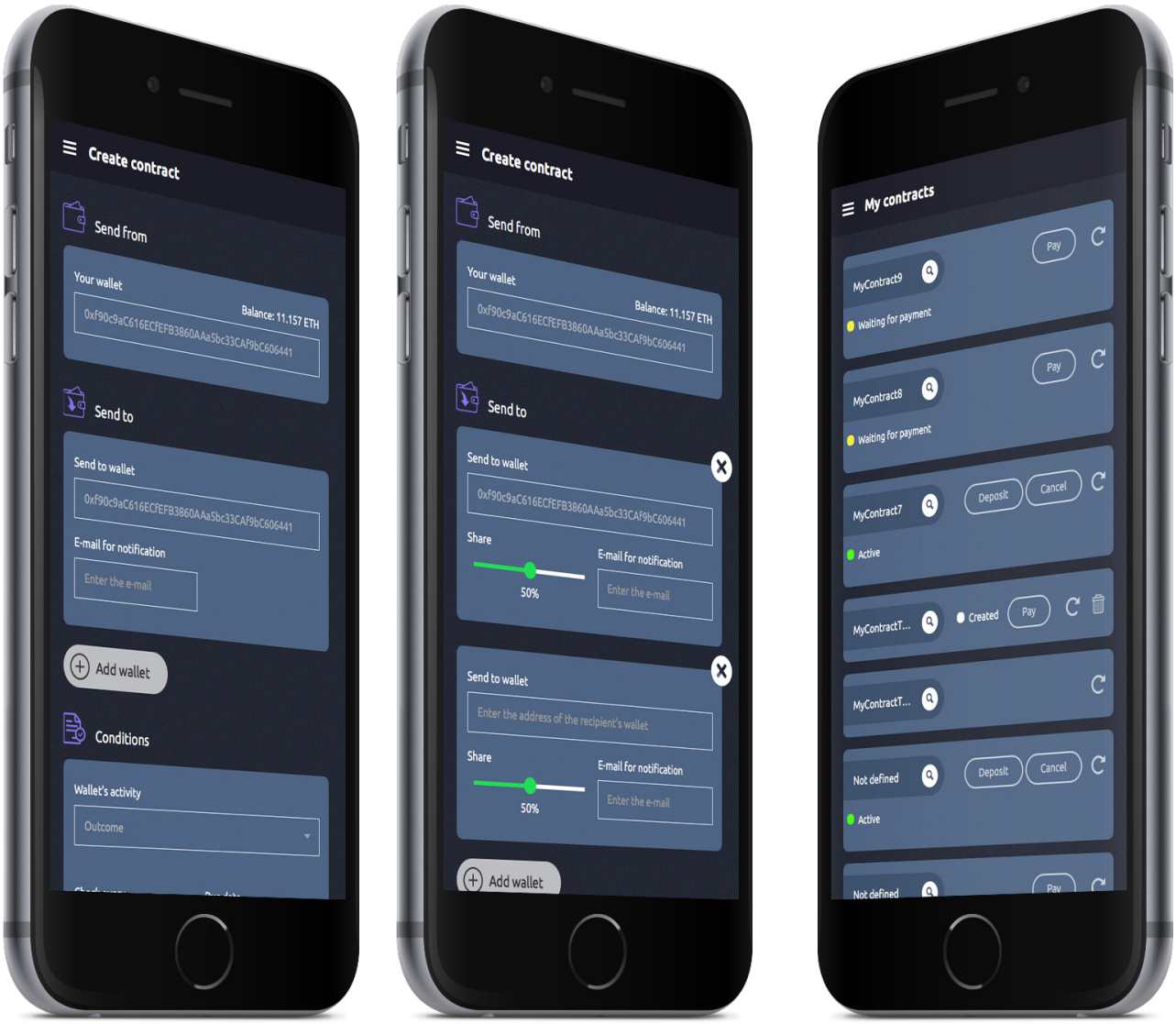
Step1. Choosing the wallet (we are not showing the process of wallet connection).



Step2. Choosing the recipients and amounts.

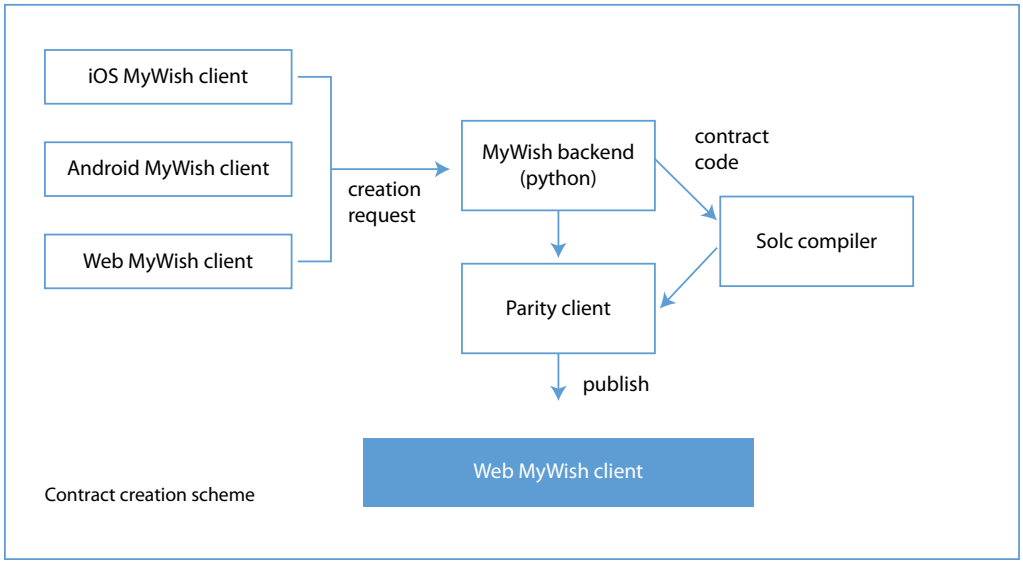


Step3. List of the contracts and management.



Mobile application interface.

In view of the fact that we have several client applications (at least three: web application, iOS application & Android application), we plan to develop a back-end server that will provide API for creating contracts for client applications and third-party developers.

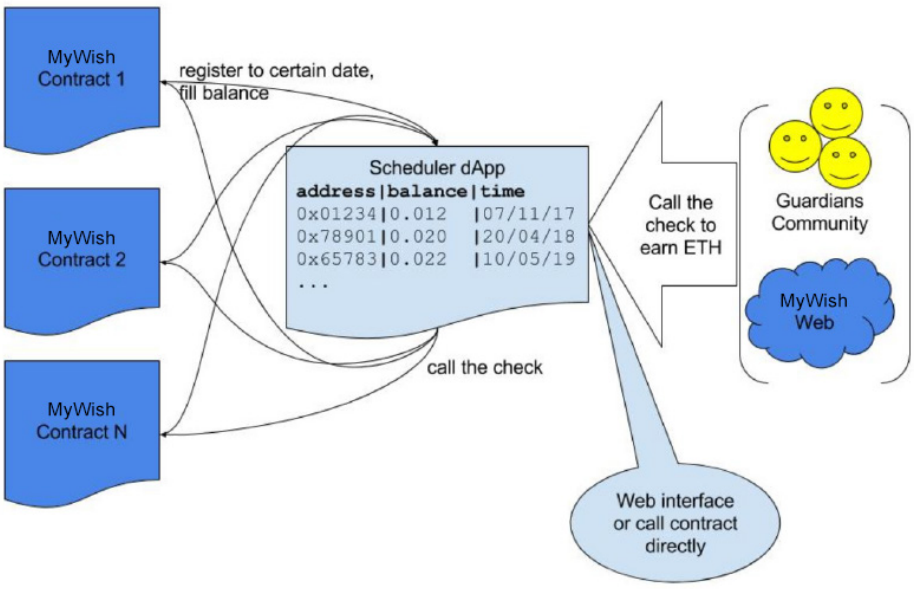


Joule – the decentralized calls of smart contracts

In Ethereum environment, smart contracts are useless entities if there are no calls for them. In other words, a contract will never be able to initiate the check of conditions and subsequent transfer of funds without an external call. Therefore, a call of a smart contract is an event not less important than contract creation.

Unlike other projects dedicated to the creation of smart contracts, MyWish implements a system, named Joule, that allows calling contracts using a decentralized solution. The proposed approach ensures calls of contracts during the whole contract work cycle and is platform-independent.

The implementation is built on the basis of the controlling smart contract which is located in Ethereum network, has all the information related to scheduling of calls of smart contracts, and initiates calls of user contracts. The controlling contract is called by MyWish system or by any other network user. To motivate calling the controlling contract, remuneration is paid in ETH/WISH to the first user that made the call within a certain time interval (or reward distribution among calls within a single block). Note that this approach is a proven solution for Ethereum [5].



Operating scheme of the controlling smart contract

When a contract is created, it registers itself with the controlling smart contract, specifies the amount of remuneration for the call and the call date.

The controlling contract, after it is called by MyWish platform or by any other network user, initiates a call of all contracts that match the current time.

Interaction with other currencies

The aim of the project is to develop a mechanism that will create lastwill contracts not only for ETH owners, but also for other crypto-currencies. First of all, for BTC. At the moment, the development focus is on ETH, but work is underway to connect BTC.

To solve this problem, it is planned to use the Ginger network from RSK [6]. RSK solution essentially allows creation of smart contracts for bitcoin, which is a necessary requirement for MyWish project. This solution is most preferable because of the low complexity of implementation.

The second approach, allowing to approach multi-currency support is based on the interchain blockchain. Among the most notable works in this direction are Cosmos [7] and polkadot [8]. Both projects will allow transactions in any currency, but the dependence on smart contracts (on ETH) is retained. Also we'd like to mention the MelonPort project [9], which is based on polkadot that will allow our project to be used for all the currencies after its launch.

The BTCRelay project [10] will allow payments to be received by BTC, while the contract can control the activity of the BTC wallet, but with this approach there will be a problem with the transfer of BTC to the recipients.

According to the project roadmap, integration of BTC into the system is planned on March 2018.

The priority is given to RSK direction.

Interaction with external developers

MyWish platform is a complete solution from the view point of the contract life cycle, starting from contract creation till its completion, either due to its execution or expiry.

Over time, as confidence in the platform will grow, the demand for implementation of new scenarios and related contracts will arise. MyWish developers will be primarily focused on the development of the platform; hence, writing of smart contracts by third-party developers is a mutually beneficial cooperation both for the platform and for the developers and, ultimately, for the users of the platform.

Usage of contracts from third-party developers is not a difficulty and does not require rebuilding of the project architecture. According to the roadmap, opening of the platform for third-party developers is scheduled for spring 2018.

Let us consider the steps for adding a contract from third-party developers to the platform:

1. Sending the code of the contract and its description to the experts of MyWish platform for reviewing

2. In case of successful review, start of integration into the platform (programming the adjustment of the contract (including UI) for its subsequent integration)

3. Testing with involvement of industry experts

4. Release of the contract into the product version of the service

5. Programming of a smart contract for payment of a fee to the developer for the usage of his contract by MyWish users

When adding the contract, the priority will be given to proper testing and security.

Payment of commission to the developers will be made in tokens of the project (WISH).

The MyWish platform economy

WISH token is the fuel required for the platform to function. No contract can be created or executed without WISH tokens payments. After receiving the payment in the form of a token, part of the funds remains in the MyWish platform for subsequent payments within the Joule system and for paying to third-party developers for their contracts. The other part is exchanged for ETH to pay for gas for the creation of contracts.

The growth of the token rate is ensured by the accumulation of funds in the platform for deferred payments, which are necessary during the contract's existence in the system. Thus, the number of WISH tokens on the market will gradually decrease. Also, the growth of the WISH token rate will be facilitated by an increase in the number of users of the system.

Compensation for a contract call within the Joule system is made by the following formula: compensation in ETH for the use of gas to call the verification function and compensation in WISH tokens as motivation.

To pay for the contract, the user makes an independent purchase of WISH tokens on the exchange or inside the MyWish platform, which purchases WISH for the user's currency (ETH, BTC, LTC) using the integration with the exchange.

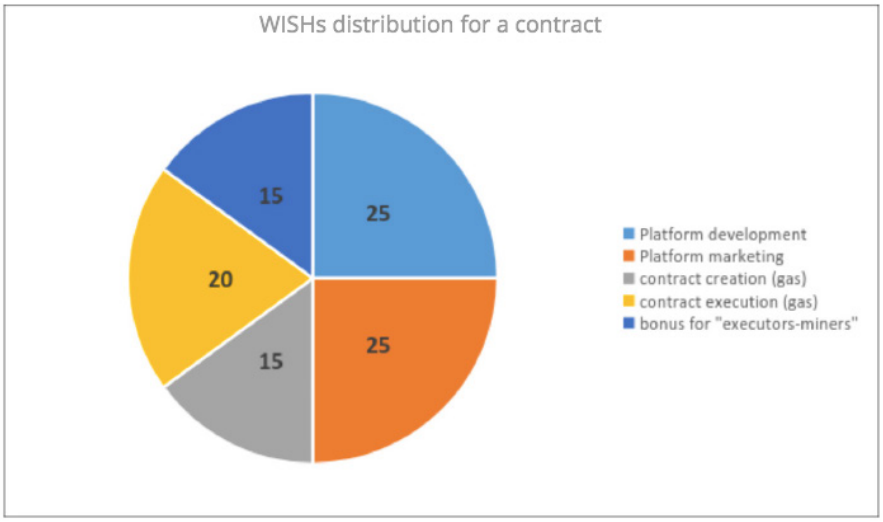
Lets consider the basic scenario of the user of the platform, who wants to issue a contract for 3 years. The average cost of such a contract will be 750 WISH (0.5 ETH at the rate at the time of Token Sale).

After the payment:

~ 15% of paid tokens (112.5 WISH) are sold to purchase the gas needed to create and deploy the user's contract;

~ 50% (375 WISH) are used by the platform for maintenance and development, including marketing and can be spent at any time;

~ 35% (262.5 WISH) remain in the platform and are used as the smart contract is checked and executed for 3 years.



WISH Token usage

Token WISH is built into the architecture of the project to maintain its value. Without buying the required number of tokens, the user will not be able to create a smart contract. After setting up the desired contract, you must pay to create the contract.

Thus, the distribution of the service will lead to the increase in demand for tokens, thereby the certain market to ensure the fair price will be reached.

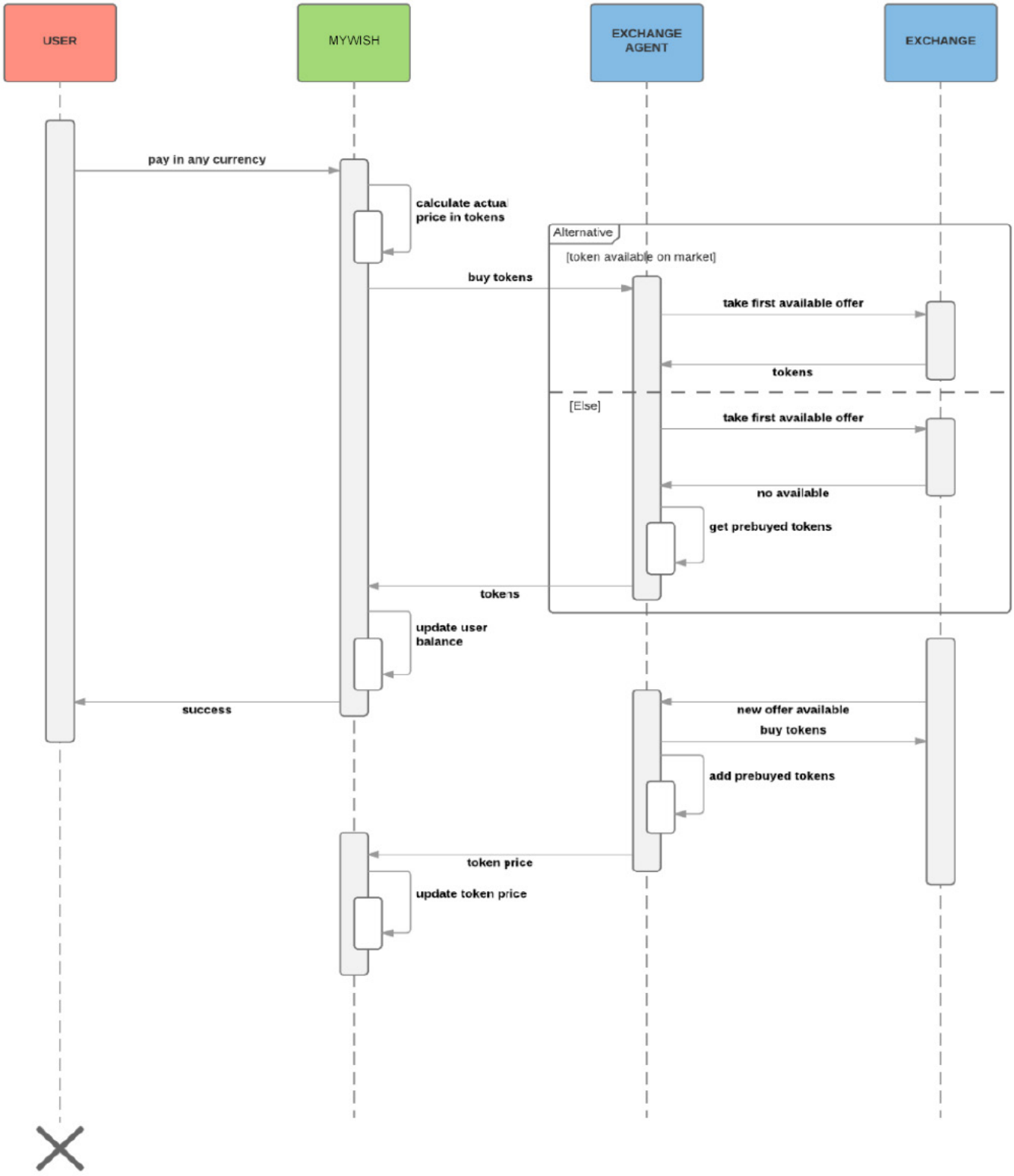
Let's consider in more detail the cost of using the service.

The cost of creating a MyWish contract (and its implementation) is dynamically calculated and depends on:

- Terms of the contract
- Frequency of verification of contract conditions
- Number of wallets (recipients)
- Contract terms (including external services and oracles)
- MyWish Service Commissions

After transferring funds to the service account, the user gets access to the source code of the contract to verify it. After confirming the contract, MyWish creates a contract in Ethereum blockchain from user's account. All the necessary costs for ethereum gas to create and call contracts the service takes care of. The service for these costs is obtained by converting the received WISH to ETH.

Below you can find a diagram of the user, platform and exchange interactions when buying tokens.


USER
MYWISH
EXCHANGE AGENT
EXCHANGE
pay in any currency
calculate actual price in tokens
Alternative
[token available on market]
buy tokens
take first available offer
tokens
[Else]
take first available offer
no available
get prebuyed tokens
tokens
update user balance
success
new offer available
buy tokens
add prebuyed tokens
token price
update token price

MyWish service commission is main source for financing the MyWish project for further development and marketing promotion. It is planned that the commission will be up to 100% of the cost of all other (necessary) costs.

Estimated value of the basic contract (valid for 3 years, activities inspections every six months) will be 0.5 ETH, including the service fee.

Besides, WISH tokens will be used to pay for third-party developers, if their contract undergoes system verification and is posted at the platform.

Thus, the market price for WISH tokens directly depends on the number of users of the service, the cost of the service and the demand for services. The team, in view of the fact that it has 16% of all issued WISH tokens, is most interested in observing the balance of the popularity of the service and the cost of the services provided.

Our competitors

We receive many questions about our service: is it unique? What are the competitors? Why are you better? What are your main advantages?

These questions inspired us to write this article.

Among competitors of MyWish platform one can distinguish two different classes: the projects oriented on creating smart-contracts and the ones dedicated to solving a certain real-world problem (for example, crypto assets inheritance)

First let's consider BlockCAT [3] – a project concerning the first group, which ICO campaign was successfully finished in September. BlockCAT is a smart contract constructor with a possibility to deploy it in Ethereum. It allows to decrease the gas spending significantly by means of reusing the whole functional by several contracts.

Common features of the project:

- smart-contract library
- gas optimization
- comfortable interface
- open platform for third-party developers

One can distinguish two major differences between MyWish and BlockCAT:

1. BlockCAT is not concerned about contract invocation; after creating, the contract stays in Ethereum and awaits to be called for execution. MyWish guarantees every created contract will be called and executed even if our platform no longer exists in the future. Thus, MyWish team implemented complete contract lifecycle from its creation to either successful execution or cancel demanded by user.

2. BlockCAT is not supposed to realise specific life case contracts, but it basically is a marketplace for all types of contracts. MyWish gives priority to certain tasks and develops contracts in order to find solutions for the tasks. In other words, MyWish's main idea comes from the users' need to solve a certain problem: to make up a wedding contract or a testament, redeem assets from a wallet when the private key is lost etc. But after implementing basic contracts the platform will look through external contracts very thoroughly before adding them into the library.

Another competitor class are the projects dedicated to solving certain problems.

Now let's consider the DigiPulse project [11], which, unfortunately, has not reached the soft cap during crowdfunding. The project was about to implement a mechanism that allows anyone to distribute crypto assets in case of user's death. The Digipulse received support of The Coinbase and it's main bet was concerned with the member's death confirmation by standard, "non-crypto" methods of real world.

Among the main differences between platform MyWish and DigiPulse project (ver1.0.) we'd like to mention:

1. MyWish - The implementation of contracts library for various life situations, not only wills.
2. MyWish is based on the Ethereum platform and does not require integration with external resources, such as exchanges or wallets (coinbase).

Token WISH - Terms and Conditions of Sale

During the Token Sale, the WISH token will be used to raise the project's capital. After the Sale (and the achievement of the soft cap), the token will be placed on the crypto-exchange exchanges for free trade. The exchange rate of the token will be regulated by the market.

Emission details:

- 22'000'000 ERC20 tokens
- Ethereum blockchain
- Date of emission: The 22nd of October

MyWish will begin its Token Sale on October 2th, selling 62.72% of the total supply to crowd sale participants. It will last for a period of 37 days or until all WISHs are sold, whichever period of time is shorter.

Each WISH will be sold for .00067 ETH, approximately $.2 USD, meaning the effective hard cap for the crowd sale is approximately $2,500,000 USD. The soft cap is not set due to successful pre-sale (project will be continued anyway).

Pre-Sale

Pre-Sale lasted until August 17, and it collected 759 ETH, with a target of 500 ETH.
Before Token Sale all PRLs (pre-sale tokens) will be converted to WISHs (1 PRL = 1 WISH).

Token sale Bonus

Token Sale also provides bonus program:

- Get 30% discount during the first 14.5% of total tokens
- Get 20% discount during the second 14.5% of total tokens
- Get 10% discount during the third 14.5% of total tokens
- All others tokens (19%) will be sold without discounts

Finally, Token distribution will be the following:

- Bounty & Bonus: 9 % tokens
- Team: 14.3 % tokens
- Pre-Sale: 13.8 % tokens
- Token sale: 62.72 % tokens

References

1. https://www.coindesk.com/will-become-bitcoins-die/
2. https://medium.freecodecamp.org/a-hacker-stole-31m-of-ether-how-it-happened-and-what-it-means-for-ethereum-9e5dc29e33ce
3. https://blockcat.io/wp-content/uploads/whitepaper.pdf
4. https://github.com/ethereum/mist/
5. http://www.ethereum-alarm-clock.com/
6. https://github.com/rsksmart/rskj
7. https://cosmos.network/
8. https://polkadot.io/
9. https://melonport.com/
10. http://btcrelay.org/
11. https://www.digipulse.io/