WHITE PAPER

# TAKE CHARGE OF THE FINANCIAL REVOLUTION

- EDITION, 2022 -

# TABLE OF CONTENTS

5. Interface Specification

5.1. Smart Contracts
5.1.1. ERC-20 Contract
5.1.2. Extended Interface Methods
5.1.3. Faucet Contract
5.1.4. Sales Contract
5.1.5. Trader Contract
5.2. Application Interfaces
1. Account creation
2. Read Operations
1. Reading the coin balance of a user
2. Reading the gas balance of a user
3. Retrieving transaction info
6.3.3. Write operations
1. BPO Format
2. The /action endpoint
3. Sending ether to account
4. Sending coins to account
5. Paying to a seller with a conditional payment
6. Confirming or cancelling conditional payments
7. Exchanging Tokens

6. Errors

5.2.1. Error Result Structure
5.2.2. Validation Errors
*Validation Errors*
6.4.3. Network Errors
6.4.4. Blockchain Errors

7. Migration Strategy

7.1. ReplayingTransactions
7.2. Migration of Balances
7.3. Account Identifier Migration

8. Guidelines for Implementation

8.1. Work Environment
8.2. Technological Stack and Dependencies
8.2.1. Web Services Middleware
8.2.2. Database
8.2.3. Blockchain Connector
8.2.4. Operating System
8.3. Configuration Parameters
e. The roadmap
f. Compliance Program
g. Legal Considerations & Risks
h. Disclaimer

*That decade has been one of important achievements. Most notably, macroprudential policy for banks has developed from an idea into a reality. But this morning I would like to look forward and ask: how can we prepare for the next decade?*

*There are known threats to financial stability which have always been on the ESRB's radar, such as unbridled risk-taking, stretched asset valuations and excessive credit growth. But today we are also facing new types of threats which could affect financial stability profoundly.*

**CHRISTINE LAGARDE**, President of the ECB,
at the fifth annual conference of the ESRB
Frankfurt am Main, 8 December 2021

*We have faced a series of crises in the euro's short life, ranging from the great financial crisis to the sovereign debt crisis, and to the coronavirus pandemic. We have faced periods of too-high inflation and too-low inflation. But throughout, our actions have been guided by delivering on our mandate.*

*Yet the global economy never stands still – and central banks have to adapt if they are to stay faithful to their mandates in changing times.*

*As Goethe, a son of Frankfurt, once wrote: "Life belongs to the living, and whoever lives must be prepared for change.*

**CHRISTINE LAGARDE**, President of the ECB,
at a virtual ceremony marking the change of office of the President of the Bundesbank
Frankfurt am Main, 11 January 2022

*Nearly 50% of Europeans say they have worked from home during the pandemic. E-commerce increased by almost one-fifth during the first lockdown, and it stayed at that level even after restrictions were lifted. We have seen a surge in online payments and a shift towards contactless: about 40% of respondents to a recent survey say they have reduced their use of cash.*

*Such shifts are neither unusual nor unwelcome. Major technological advances have been recurring features in our history.*

**CHRISTINE LAGARDE**, President of the ECB,
at the European Banking Congress.
Frankfurt am Main, 20 November 2020

# Executive Summary

One Ecosystem is a company whose main activity is the sale of its online educational program, One Academy. The Company's focus is on direct sale through multi-level marketing. Our network of Independent Marketing Associates create the expansion of our One Ecosystem with the focus on developing educated miners. One Ecosystem, seeks to increase users' quality of life, providing equal worldwide access to financial education, thereby encouraging the creation of new economic opportunities. These online educational services are distributed by Independent Marketing Associates (IMAs). The objective is to present and provide intermediation services regarding training products. The educational program allows IMAs and users to trade in crypto-currencies or tokens to mine units of ONE, following the receipt of free optional promotional tokens with the purchase of any Education Module. The online educational program is an introduction to the financial products and trading, and provides the trainees with online reading materials, video presentations, tests, and a certificate to begin exploring the One Ecosystem Vision. One Ecosystem is dedicated to creating a coin suitable for mass-market usage in line with its vision to provide access to financial services for everyone.


To fulfil the commitment to the growing community mining and using the ONE Cryptocurrency, an ECOSYSTEM was created around the ONE to unify all the elements in the environment and provide a universal, uniform currency for all the users in the system. One Ecosystem promotes usability, functionality, and reliability of its Networking platform, its Educational Platform (One Academy), an e-Commerce platform (DealShaker), as a result of the software used by the company to distribute educational programs and allow the exchange of units of the virtual currency ONE and tokens. One Ecosystem in turn is computer-generated virtual currency created by using mathematical and cryptographic algorithms based on tokens. A growing number of online and offline venues are now accepting ONE as a form of payment via the DealShaker. The values are determined by the principle of supply and demand in the market.

An important part of the One Ecosystem is the blockchain network. It stores and processes all transactions that are coming via the REST services layer. The blockchain ensures the integrity of the payment process and stores immutable information about the history of transactions and balances of user accounts. Both the Payment Services Layer and the Blockchain.

The solution requirements are separated into two main groups - functional and non-functional. The functional requirements are the known operational activities and processes that are implemented and consumed by the rest of the system. The non-functional requirements are describing qualitative characteristics and necessary parameters that have to be covered by the solution in order to satisfy the business case and the overall functional operation of the system that utilizes the solution.
Whenever a payment transaction takes place, we are talking about the transfer of ownership of a fungible token managed by a smart contract on the blockchain. This action of value transfer is expressed and translated into a blockchain transaction that invokes a smart contract which manages the balances of users and specifically their accounts regarding the number of coins each balance is holding before and after the execution of the requested transaction.

The rules by which a token is created, transferred, and destroyed is governed by a smart contract that is deployed on the blockchain component of the solution. These rules are enforced by the virtual machine (VM) that executes the smart contract code, without any difference on which validator node the code is executed.

There are two types of payment processes that the system utilizes - atomic payments and conditional payments. Initially, the solution uses a private setup of a blockchain network that is managed and monitored by the owner of the system or other authorized organization(s). At a later stage, the transactions must be moved onto a public blockchain. Thus, the need for a complete history of transaction requests and their statuses. The selected blockchain network is based on the Ethereum software and its established codebase.

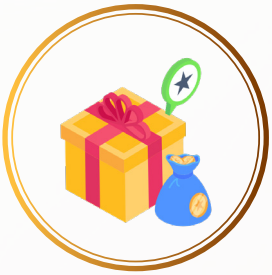
# a. INTRODUCTION

From the perspective of personal development and self-improvement theories, the Company has focused on an introduction to financial products, with the impact of providing financial education to guide its students in the maintenance of a "healthy" level of net worth and determining the rationale for getting to know every component of their net worth in detail since:

## *Net Worth = Financial Assets - Financial Liabilities*

Before getting to know about financial products, one should be familiar with the significance of banking institutions - from Central Banks to Commercial Banks, and how they intervene in peoples' financial and money-making decisions and how you can benefit from them. Banks are complex institutions, which provide not only loans and deposits, but also security and regulations. Banks are essential partners in our lives – the more you know and understand how they operate the better you would understand how you can utilise their services to your benefit.
If you want to increase your net worth you should work towards increasing your assets and decreasing your liabilities. You can increase assets a number of ways including holding some, or all, of following:

Cash and cash equivalents



Bonds

Stocks

**The One EcoSystem** delves into different investment products such as commodities, forex, structured products, derivatives to highlight a few.

Students of One Academy are guided to minimize the side of liabilities by wisely evaluating the difference between a "bad debt" and a "good debt". In the long term a "bad debt" will not bring any benefits and in many cases, it is just used to satisfy some short-term consumption needs. On the other hand, when you are accumulating a "good debt", it is for the sake of purchasing something that will bring returns your initial investment in the long term. The best-case scenario of having a "good debt" is that the return will cover the initial debt in the short term. By decreasing to a minimum level, the amount of "bad debt" within your net worth calculation, would enhance the possibility of quicker accumulation of wealth.

The final and the most innovative part of **One EcoSystem** covers cryptocurrency. The Company explicitly demonstrates exactly what cryptocurrency is how benefits can be derived from it.
What is the beneficial difference of holding a cryptocurrency, rather than just holding a currency? What is the difference between a cryptocurrency and a currency? Years after the abolishment of "gold standards", the US dollar is not backed up by any tangible asset. This pure fact is valid for all global currencies. What does this mean? It means that the money which you hold in your hands is just pieces of printed paper or numbers on your screen. It has no real value, no equivalence - there is no tangible asset behind it.

Cryptocurrency is based on cryptology, which is securing or encrypting, the real currency value behind it. The initial purpose for the creation of a cryptocurrency was the introduction of a new currency unit, which will not be affected by any governmental or economic local player. Cryptocurrencies are not tied to any worldwide currencies and for this reason their price will not fluctuate due to any governmental policies.

Furthermore, cryptocurrencies are a perfect investment alternative, either for the short-term or for the long term. You can use cryptocurrencies also for instant exchange all around the globe or as a payment tool for various products and services. The most significant difference which you should consider between a "normal" fiat currency and a cryptocurrency is the security which "cryptos" provide.

> ***One Ecosystem** puts you one step ahead of your regular life and reminds you once again - you are your only obstacle to achieving financial independence!*

# b. DESCRIPTION OF THE MARKET AND THE PROBLEM

Virtual currencies are a relatively new technological phenomenon, the coming into existence of which is to a large extent facilitated by the deeper and deeper penetration of information and communication technologies in our daily lives and the increasing use of the Internet. According to an Opinion of the EBA dated 4 July 2014 on virtual currencies, it is stated that at the time of publishing the document over 200 different virtual currency systems existed. Other unconfirmed sources claim that as at 2021 there were over 6000 types cryptocurrencies. The large amount of virtual currency systems and the technological complexities related to their generation and the closing of transactions involving them (sale, transfer actions, use as a means of payment, etc.) and their possible applications create a significant problem for the implementation of a systematic methodological approach to categorizing them. Nevertheless, in view of the existence of some common features of individual types, the application of certain criteria relating to the convertibility of units of virtual currency, their liquidity, the technologies used, etc., when considered in a broader context, the following categories can be conditionally distinguished:

- Depending on whether it is possible to purchase virtual units in exchange for official currencies and to purchase goods or services:

1. **Closed virtual currency systems** - these are systems that have no or almost no link to the real economy. They are offered in closed (software) environments and can be used to obtain virtual goods or services, only in the context of the corresponding environment and, by definition, cannot be used in the turnover outside such environment. The most common examples illustrating closed virtual currency systems are associated with computer games, where the virtual currency is generated depending on the performance of the user in the game;

2. **Virtual currency systems with unidirectional flows** - these are systems in which units of the virtual currency can be purchased using official currency but cannot be **exchanged back** (hence the definition of 'unidirectional flows'). The terms and conditions for this are defined by the creator of the corresponding system. Regardless of the fact that it is not possible to translate the virtual currency into an official currency, it is possible to purchase goods and services using units of the virtual currency;

3. **Virtual currency systems with bi-directional flows** - in these systems users can buy and sell units of the virtual currency using or in exchange for an official currency. In view of their economic significance and popularity, these systems are the subject of greatest interest both by users and by regulators, supervisory and legislative government bodies.

- Depending on the technologies used:

1. **Virtual currency systems using cryptographic algorithms** - protect users and authenticate the generation of and transaction in units of the virtual currency, and also limits the number of units that can be generated over a certain period of time; they are also referred to as cryptographic currencies or crypto currencies. In the general case, the technology could be described as similar or relating to the 'public key infrastructure' used to create advanced and qualified electronic signatures within the meaning of the Electronic Document and Electronic Signature Act. In this connection, however, it shall be noted that with cryptocurrencies the figure of the supplier of identification services is non-existent and in this sense a comparison could continue to be valid only win respect of

the technological aspects and not to the structural and organisational aspects.

2 **Virtual currency systems not using cryptographic algorithms** - in the operations with the virtual currency.

Depending on whether the system is controlled by a known operator or operators:

1 **Decentralized virtual currency systems** - in these systems the software required for generating units of the virtual currency and operating with them is usually not under the control of a specific operator. In these cases, a 'peer-to-peer' communication technology is usually used, where communication between users is direct and individual devices are used as both server and client. The applications required are distributed under open licenses allowing free use of the source code for processing of existing software applications or developing new ones. Crypto currencies are usually based exactly on such decentralized systems.

2 **Centralized virtual currency systems** - in these systems, unlike in the decentralized systems, the owner and/or operator (usually this is the developer of the system) of the software systems used to generate and operate with the virtual currency are known. While in decentralized systems the generation of new units of the virtual currency is determined by mechanisms built into the software system (software protocols, also referred to as 'controls') and terms and conditions based on solving cryptographic tasks, in centralized systems this issue depends on and is sometimes entirely under the control of the owner and/or operator. On most occasions, closed virtual currency systems and systems with unidirectional flows are centralized.

What is common to all virtual currencies is the fact that they all function as a substitute for official currencies and monetary units, on the one hand, and, on the other hand, exist only in cyberspace, i.e., in a computer-generated environment, without a tangible, physical sign of their existence other than an electronic record in an information system. In view of this and to make a clearer characterization of virtual currencies we need to compare and clarify their relationship with the traditional notions of "currency" and 'money'. According to some authors, money are bearer securities and materialize rights, are issued by the state or an issuing institution designated thereby. They are securities only in cash and always materialize only value. According to other authors, based on the assertion that securities are private dispositive documents, money are official documents. According to the latter authors, after the abolition of the gold standard money does not materialize rights, but only value. Next, a third category of authors notes that the identification of money with a document (either private or official) is fundamentally flawed, and such statements should not be shared. According to them, a significant deficiency in the drafting of the definitions above is that they do not recognize the fact that money is both materialized and dematerialized, existing in bank accounts as liabilities of banks. These groups define money as a specific type of official certifying sign that denotes different nominal values, represents a legal tender issued by an authorized institution, serves to make payments, accumulate wealth and as a reporting unit, and can have different physical form or be dematerialized - expressed in numerical form in accounts on the liabilities side of banks. In this last definition we can see the influence of the economic theory, according to which, in functional aspect, money serves as:

-**Means of circulation, medium of exchange** - money is a convenient substitute for all goods and services, which greatly facilitates exchange.

-**Unit of value, unit of reporting** - by dividing the money into convenient units, goods and services are represented as a certain amount of money. In this sense, money is a measure which facilitates the expressing of a certain amount of goods or services provided or received.

-**Means to preserve value** - money is used to accumulate and preserve wealth, to accumulate material resources.

The fact that virtual currencies are not legal tender is undoubted, but that does not affect their legality and does not preclude the possibility to qualify them as contractual means of payment. This fact underlies the approach taken by some countries and jurisdictions to make virtual currencies subject of the legal regime relating to barter exchange. The contents of the concept
'Legal Tender' includes the following main points: where a consideration is negotiated in a particular currency, the creditor is obliged to accept the payment, regardless of the type or nominal value of the money, as long as they are in the same currency; obligations of the issuer to the currency; and minimum regulatory threshold for the use of the currency.
It is also beyond dispute that virtual currencies are not issued by a central bank and are not subject to regulation by such bank. In this sense, they are not an official monetary unit or currency.

It is for this reason, that virtual currencies are classified as 'alternative currencies' in the sense that they are an alternative to official monetary units. This difference, however, also does not affect the legality of virtual currency systems.
The third difference is the fact that virtual currencies are not normally characterized by the existence of a tangible medium. This, however, is relative, as far as the information providing access to and ability to operate in a virtual currency could be reproduced on a physical medium. In the context of crypto currencies, for example, this would be the information about the public and private key of the user used to sign transactions involving available units of the corresponding virtual currency.
We can also see, however, significant similarities between money and virtual currencies in terms of main functions according to established monetary theories in economics - both money and units of a virtual currency can serve as a means of circulation, unit of value, and means to preserve value. In this sense the existence of equivalence in terms of functionality between traditionally established understanding of money and virtual currencies could be justified.

The positions and opinions of official European financial authorities define virtual currencies differently and therefore we cannot establish the existence of a single accepted definition. For example, the EBA defines virtual currencies as **digital representation of value that is neither issued by a central bank or a public authority, nor necessarily attached to a fiat currency, but is accepted by natural or legal persons as a means of payment and can be transferred, stored or traded electronically**. Next, ECB gives the following definition: **virtual currency can be defined as a type of unregulated, digital money, which is issued and usually controlled by its developers, and used and accepted among the members of a specific virtual community.** On the other hand, in an opinion discussing matters relating to the VAT regime and taxation applicable to Bitcoin, the Group on the Future of VAT at Directorate-General 'Taxation And
Customs Union' of the European Commission (EC) refers directly to the definitions of the EBA and ECB when characterizing the concept 'virtual currency', regardless of the differences between them.

What has been said so far about the categories and characteristics of virtual currencies allows ONE to be defined as ***a special type of closed centralized cryptographic virtual currency with bidirectional flows, which is a digital representation of monetary value used by individuals and organizations - members of a specific virtual community - as a means of payment and can be transferred, stored or traded electronically.***

**ONE** is defined as a **centralized virtual currency,** since the owner who is at the same time the operator of the information system used to generate and operate with the virtual currency is known, and the generation of new units of the virtual currency is under its control.
**ONE** is defined as a **cryptographic virtual currency,** since within the information system cryptographic algorithms are used to protect users and authenticate the generation of and

transaction in units of the virtual currency. At the same time, in the specific case the cryptographic technology is not used to limit the number of units of the virtual currency that can be generated.

ONE is defined as a closed **virtual currency,** since at present the units of the virtual currency cannot be used outside the information system, and their generation is directly related to the performance of users.

ONE is defined as a special type of **virtual currency with bi-directional flows** as users can buy and sell units of the virtual currency using or in exchange for an official currency. The peculiarity stems from the fact that users do not have the opportunity to directly purchase units of virtual currency, and the fact that payments of any cash generated are made not by other users but by the operator and owner of the information system.

ONE has some of the fundamental economic characteristics of money - **unit of value and means to preserve value, it is used as a means of circulation,** because now there is possibility to purchase goods and services through the platform DealShaker.

An important part of the One Ecosystem is the blockchain network. It stores and processes all transactions that are coming via the REST services layer. The blockchain ensures the integrity of the payment process and stores immutable information about the his6tory of transactions and balances of user accounts. Both the Payment Services Layer and the Blockchain.
The solution requirements are separated into two main groups - functional and non-functional. The functional requirements are the known operational activities and processes that are implemented and consumed by the rest of the system. The non-functional requirements are describing qualitative characteristics and necessary parameters that have to be covered by the solution in order to satisfy the business case and the overall functional operation of the system that utilizes the solution.
Whenever a payment transaction takes place, we are talking about the transfer of ownership of a fungible token managed by a smart contract on the blockchain. This action of value transfer is expressed and translated into a blockchain transaction that invokes a smart contract that manages the balances of users and specifically their accounts with regard to the number of coins each balance is holding before and after the execution of the requested transaction.
The rules by which a token is created, transferred and destroyed is governed by a smart contract that is deployed on the blockchain component of the solution. These rules are enforced by the virtual machine (VM) that executes the smart contract code, without any difference on which validator node the code is executed.

There are two types of payment processes that the system utilizes - atomic payments and conditional payments.
In view of the absence of specific statutory regulations, it should be considered that these activities are inherently, along with everything else, also information society services within the meaning of the Electronic Commerce Act (EC Act) and Directive 2000/31/EC of 8 June 2000 on certain legal aspects of information society services, in particular electronic commerce, in the Internal Market (Directive 2000/31/EC). Pursuant to Article 1, Paragraphs (2) and (3) of the EC Act, electronic commerce shall mean providing services for the information society, which are defined as such services, including transmission of commercial messages, which are usually onerous and are provided from a distance by electronic means upon an explicit declaration of the recipient of the service.

As far as the activities relating to One Ecosystem do not fall within the exceptions to the scope of the law contained in Article 1, Paragraph (4) of the Act, the owner and operator of the information system will have to be defined as a provider of information society services, and the services and products provided by it — as direct e-commerce, which, according to the established theory, as opposed to indirect e-commerce, constitutes e-commerce where all elements of the legal relationship

are performed remotely and electronically - both the declarations of intent between the parties for conclusion, modification and termination, and the actual performance and implementation of the outcome of the transaction. Pursuant to the provisions of Article 4, Paragraph (1) of the EC Act, a provider of information society services shall be under the obligation to provide a certain range of data and information, including: its name or title; its permanent address or its seat and registered office; contact information, including telephone number and e-mail address; data for registration in a commercial or any other public register; etc. Furthermore, pursuant to Article 9 service providers shall be under the obligation to place at recipient's disposal the general terms and conditions and the content of the contract concluded electronically in a way that allows him to store and reproduce them.

The basic paradox that bedevils cryptocurrency which will be harder to surmount is - the more popular they become, the more regulation and government scrutiny they are likely to attract, which erodes the fundamental premise for their existence.

While the number of merchants who accept cryptocurrencies has steadily increased, they are still very much in the minority. For cryptocurrencies to become more widely used, they must first gain widespread acceptance among consumers. However, their relative complexity compared to conventional currencies will probably deter most people except for the technologically adept.
A cryptocurrency that aspires to become part of the mainstream financial system may have to satisfy widely divergent criteria. It would need to be mathematically complex (to avoid fraud and hacker attacks), but easy for consumers to understand; decentralized but with adequate consumer safeguards and protection; and preserve user anonymity without being a conduit for tax evasion, money laundering and other nefarious activities. Since these are formidable criteria to satisfy, is it possible that the most popular cryptocurrency in a few years' time could have attributes that fall in between heavily regulated fiat currencies and today's cryptocurrencies. While that possibility looks remote, there is little doubt that as the leading cryptocurrency at present, Bitcoin's success (or lack

thereof) in dealing with the challenges it faces may determine the fortunes of other cryptocurrencies in the years ahead.

# c. DESCRIPTION OF THE PRODUCT AND HOW IT'S GOING TO SOLVE SAID PROBLEM

Virtual currencies are a relatively new technological phenomenon, the coming into existence One Ecosystem is a company whose main activity is the sale of its online educational program, One Academy. The Company's focus is on direct sale through multi-level marketing. Our network of Independent Marketing Associates creates the expansion of our One Ecosystem with the focus on developing educated miners. One Ecosystem, seeks to increase users' quality of life, providing equal worldwide access to financial education, thereby encouraging the creation of new economic opportunities. These online educational services are distributed by Independent Marketing Associates (IMAs). At the beginning of their marketing activities, each IMA receives an "Activation Module" free of charge. It contains basic information about the main product of the Company - OneAcademy. The online educational program is an introduction to the financial products and trading, and provides the trainees with online reading materials, video presentations, tests, and a certificate to begin exploring the One Ecosystem Vision. Opportunities are provided for up to six (6) levels of training, as IMAs can purchase additionally "higher level" educational modules. Each module includes components of the program at a specified level of training, including information regarding the general provisions, processing of and trade in virtual currencies, crypto-currencies and ONE. In addition, each module contains a specified number of promotional tokens which can be submitted for 'mining', therefore generating units of the virtual currency ONE.

There is no obligation to purchase the product after registration. IMAs can work to realize retail sales and earn commissions from them. Depending on their personal objectives, IMAs may choose the training courses needed and the extent to which they wish to participate in generating and trading of the digital currency. Both training materials and promotional tokens are offered, notwithstanding the role in the system of distribution. A "Rookie" account requires no investment at all. In this position, an IMA can already realize sales activities by distributing educational modules comprising free promotional tokens and training units. For this a commission will be received. In addition, IMAs can expand their own teams based on recruiting new IMAs. The IMAs are also not given any false impression in respect of reaching exceptionally great and fast economic success without the respective work effort within the remuneration system. As part of its training program it is expressly brought to the IMA's attention that the level of income decisively depends on the personal work effort and individual skills.

The study of the services provided by the Company in connection with the educational product OneAcademy allows the following conclusions to be drawn:

- The virtual currency ONE does not constitute a separate service, but rather can be characterized as an instrument through which the principles of the educational materials provided can be applied in practice;

- The virtual currency ONE is generated in a centralized manner, within the system developed and operated by the Company. Cryptographic algorithms are used in the generation of new units of the virtual currency ONE

- New units of the virtual currency ONE can be generated only using promotional tokens provided by the Company. Promotional tokens, like the virtual currency, exist only in the form of electronic records within the system, and neither they nor the rights that can be realized through them are objectified in tangible form.

- Operations involving the virtual currency ONES can be performed only within the system provided by the Company.

- DealShaker - is an online deals marketplace and advertising service provider with a membership-based customer base. The platform enables business-to-customer and customer-to-customer deal promotions in combination of cash (EUR) and the new-age, mass cryptocurrency-ONEs. Ads are grouped based on geographic area, business category and type.

The focus of One Ecosystem is on sales and customer acquisition through independent marketing agents, IMAs. The company applies internal ethical rules that comply with the regulation to build clear and fair rules. The following statements are basically underlying the relationship between the company and the network:

- Direct selling companies should deal fairly and honourably with direct sellers and prospective recruits and should not abuse their trust or exploit their possible lack of business experience. Payments and withholdings should be made in a commercially reasonable manner.

- Direct sellers should be given adequate education and training to enable them to operate ethically. Direct selling companies should communicate the contents of the Code to all direct sellers. They should be required, as a condition of membership in the company's distribution system, to comply with the standards of the company.

- Direct sellers should be fully informed by direct selling companies as to the characteristics of the goods or services offered, to enable the direct sellers to give the consumer all necessary information.

- Misleading, deceptive, or otherwise unfair recruiting practices should not be used. Unverifiable factual representations or false promises should not be made to prospective recruits. The advantages of the selling opportunity should be presented truthfully and should not be exaggerated.

- Entrance fees, training fees, franchise fees, fees for promotional materials or other fees related solely to the right to participate in the business should not be unreasonably high. Any fees charged to become a direct seller should relate directly to the value of materials or products provided in return.

- Actual or potential sales or earnings of direct sellers should not be misrepresented. Any sales or earnings representations made should be based upon documented facts.

***These rules aim to create fair competition that is not misleading and / or impair the ability to make reasonable/ sound decisions.***

# d. MOTIVATION

A new system that manages customer interactions is being built. As part of its available functionality to its users and customers, it provides features for the transfer and accumulation of value in the form of digital tokens. For this purpose, the envisioned system architecture foresees a service layer that provides the necessary means for token management in the form of REST services. The description of the requirements of this service layer is the object of this document.

Figure 1 presents the overall layout context of the solution. An important part of the solution is the blockchain network. It stores and processes all transactions that are coming via the REST services layer. The blockchain ensures the integrity of the payment process and stores immutable information about the history of transactions and balances of user accounts. Both the Payment Services Layer and the Blockchain, along with its infrastructure and deployed code, are referred to in the rest of the document as "***solution***".

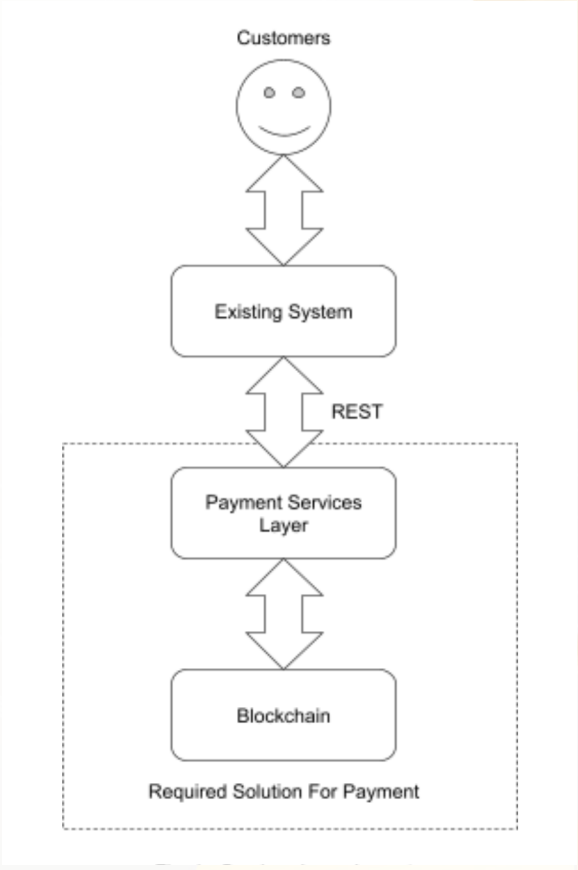



*Fig. 1 - Services Layer Layout*

The solution requirements are separated into two main groups - functional and non-functional. The functional requirements are the known operational activities and processes that are implemented and consumed by the rest of the system. The non-functional requirements are describing qualitative characteristics and necessary parameters that must be covered by the solution in order to satisfy the business case and the overall functional operation of the system that utilizes the solution.

## 2. Functional requirements

The solution is aimed at providing a convenient, secure, but at the same time transparent way of handling payment transactions between users of the utilizing system. Having in mind the nature of the blockchain component as a required part of the solution, most of the functional requirements are derived from the specifics of its operation.

***All functional elements described in this paragraph are implemented as REST services reachable over HTTP or HTTPS protocol. Every service requires a private key passed as a parameter or as a header field with which the user authenticates him/herself.***

Whenever a payment transaction takes place, we are talking about the transfer of ownership of a fungible token managed by a smart contract on the blockchain. This action of value transfer is expressed and translated into a blockchain transaction that invokes a smart contract that manages the balances of users and specifically their accounts regarding the number of coins each balance is holding before and after the execution of the requested transaction. The notion of a fungible token depicts several important characteristics:

- a token has no differentiator against other tokens
- a token can be always exchanged between accounts
- a token cannot be treated differently with regards to any of the associated transactions that took place throughout the period of its existence.
- a token has no specific identification

The token is perceived by the system as a 1:1 pegged digital asset to the chosen by the system currency or other asset that the users of the systems are able to exchange.

### 2.1. Token Management

The rules by which a token is created, transferred, and destroyed is governed by a smart contract that is deployed on the blockchain component of the solution. These rules are enforced by the virtual machine (VM) that executes the smart contract code, without any difference on which validator node the code is executed. The operations which can be executed on a token are:

- Minting
- Transferring
- Burning
- Balance Reset

Each of these operations are available or not depending on the role of the transaction executor.

| Action | Available for Roles |
|---|---|
| Minting | Owners |
| Transferring | Owners, Everyone |
| Balance Reset | Owners |
| Balance Reading | Owners, Everyone |

*Table 1. Coin Operations Roles and Permissions*

The role of Owners is fulfilled by a set of accounts that need to execute the respective action; a process commonly known as multi-sig. A multi-sig action is an action that is allowed to be executed only when a certain threshold of signatures is reached from a pre-selected list of accounts entitled for the execution of the action.
The need for multi-sig ic is a bit extended in this case, as there should be groups of keys each group representing a role (role group). For a role group to be considered a signing party it must reach its own threshold of signatures. Then, a threshold of role group signatures is needed to count a signature complete.

***Example:***

***Role Group 1 has 5 keys and threshold 2.***
***Role Group 2 has 10 keys and threshold 3.***
***Role Group 3 has 1 key and threshold 1.***

To achieve complete signature Role Group 1 has to provide 2 signatures, Role Group 2 has to provide 3 signatures and Role Group 3 has to provide its only signatures. In total the multisig will require 6 signatures for a specific action to be taken as a result of the signature.

The role of Everyone is anyone else who is entitled to hold the managed token.

## *2.2. Payment Processes*

There are two types of payment processes that the system utilizes - atomic payments and conditional payments.

### *2.2.1 Atomic Payment Process*

An atomic payment process is initiated by a user with the intent to transfer value represented by a number of tokens from one of his accounts to another account. Technically, this process has to go through several stages of execution and yet be atomic with either a positive result indicating successful transfer or failure status indicating that the transaction has either been cancelled or there are other factors preventing the execution of the process.

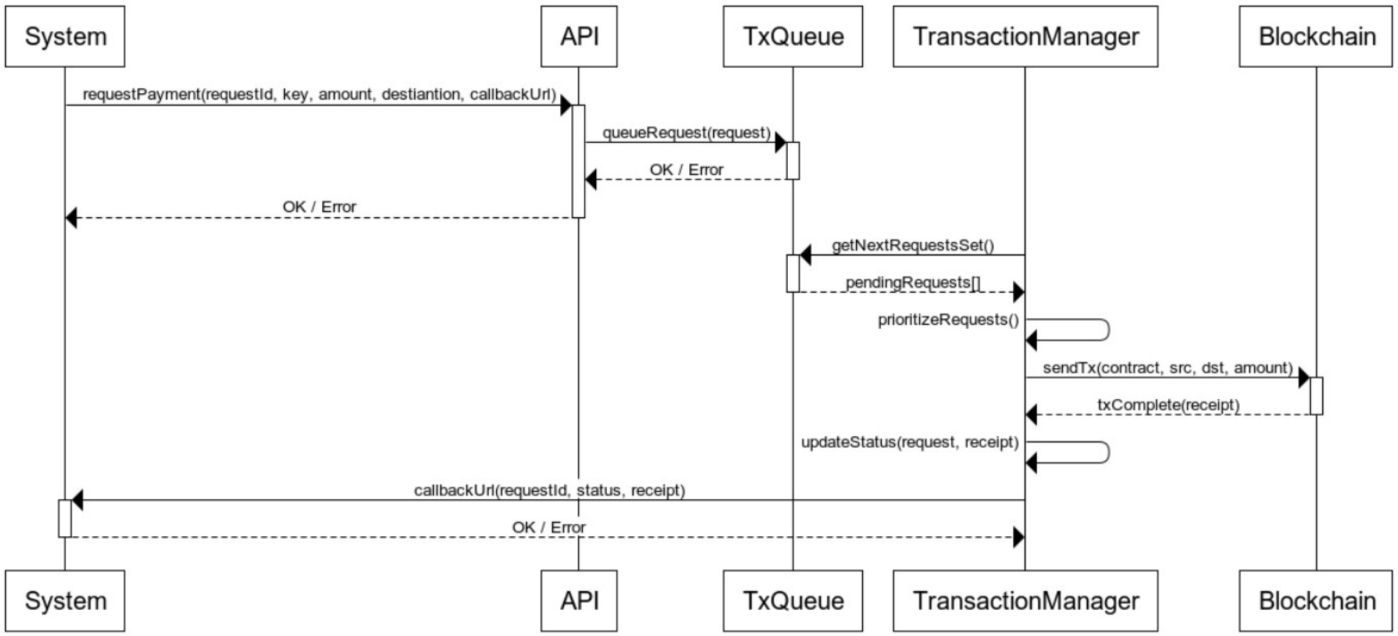



*Figure 2. - Payment process execution*

The process starts when the user requests from the system payment with a specified amount and destination account.

With that information, the system is preparing a request by creating a unique identifier for that request, providing a key for the execution of a blockchain transaction and providing a callback URL that will be used to deliver.

The request information is sent as an HTTP/S request to the solution service endpoint.

Internally the request is queued for execution along the passed from the system information to the Transaction Manager which handles the prioritization, execution and status management of every payment request. This is the first step for which the system may receive an affirmative status or an error object. In case of an error object, the system receives an error code and message which explains the reason for the failure.

It is up to the transaction manager to schedule the execution of the request along with a respective smart contract transaction call. The Transaction Manager places a transaction on the blockchain by addressing the smart contract that manages the coin balances and starts polling for the status of the transaction. At this point, it is unknown if the transaction will be successful or not. There is a period of several blocks, respectively seconds, during which the transaction will be picked by a blockchain validator and will be sealed after executing the smart contract that has been requested. After a transaction has been processed, the Transaction Manager receives a transaction receipt which carries the status of the transaction and more information about the blockchain parameters involved in its execution block number, transaction hash, block hash, etc., but most importantly it indicates if the transaction and the smart contract have been executed successfully.

Independently from the blockchain transaction status, the Transaction Manager updates its internal record for the request and executes an HTTP/S request to the specified callback URL. The payload contains the receipt, error code and error message. The response of the callback request should indicate either success or error.
This call can be implemented on the system side as a synchronous awaiting call or if the users are given

an asynchronous way of interacting with the payment process, asynchronous.

### 2.2.2 Conditional Payments

The system supports a payment process in which a buyer and a provider are concluding a sale which has the following flow illustrated in Fig. 3

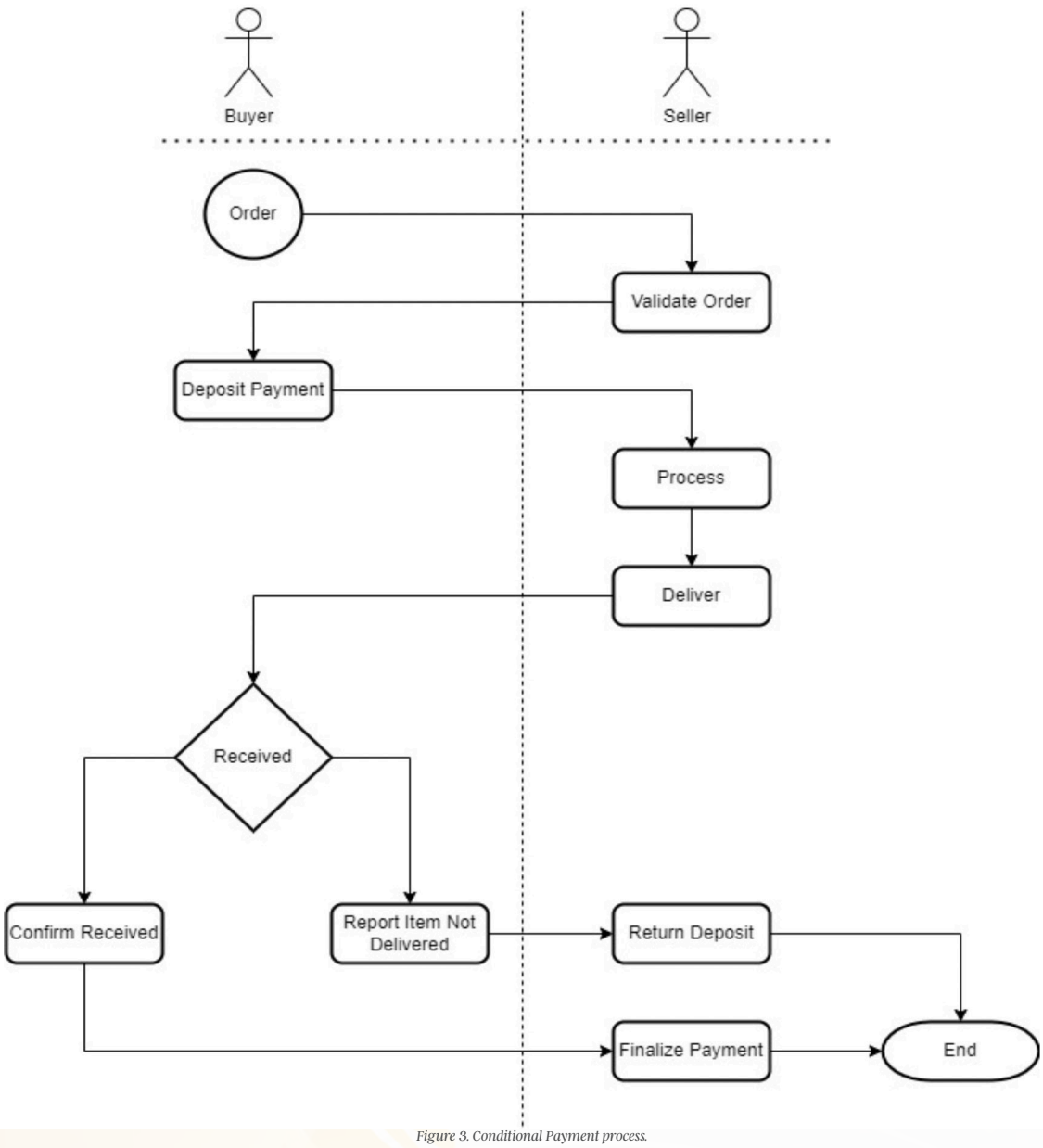



*Figure 3. Conditional Payment process.*

Figure 3 shows that the buyer and the seller are interacting for the delivery of an order. The outcome of the process is either confirmed payment or cancelled payment, depending on the fact that the delivery is respectively successful or not.

The solution should provide services to enable the payment to be broken down into two steps. The first step is the deposit step in which the user is committing to the payment by depositing the amount. The balance of his account is reduced with the amount deposited. The step of confirmation or cancellation should happen on the side of the system which depends on the outcome of the delivery step and the overall agreement between the parties decides to confirm the payment or to cancel it. In the case of confirmation, the seller is receiving the deposited amount. In the case of cancellation, the buyer receives back his deposit and his original balance on the account is restored.

### 2.2.3 Exchange of Value Tokens

A required flow is the possibility to exchange different representations of value (tokens), the exchange rate for which is defined by the system. Such a flow will facilitate a voucher-based stimulation of the users who can get vouchers whenever eligible. Collected vouchers can then be exchanged for the system currency token.

The vouchers themselves can be created the very same way as the main currency token but have a different currency symbol encoded in the smart contract that governs the voucher token. The process of exchange must be governed by another smart contract (trader contract) which will interact with the voucher and with the main currency smart contracts to update the respective balances at the request of the user. The trader contract will hold its own balances of system currency and voucher tokens. An admin (contract owner or entitled role) should be able to load or withdraw currency tokens and vouchers into the trader contract whenever needed. In the process of conversion, currency tokens are not burned but moved between the accounts of the contract. This will allow for refined control on the number of vouchers circulating in the network.

## 2.3. Required Interfaces and Endpoints

The interaction with the solution is required to be based on REST as a set of interfaces, which means that the payment process can be separated as a standalone set of functional elements. The REST interface implies the usage of standard payload formats such as JSON and XML, preferably JSON for increased and lightweight interoperability and integration with modern internet browsers.

The blockchain nodes provide generic transaction-oriented REST services for executing smart contracts or primitive value transfer transactions. These are available and should remain untouched in the solution as a standard interface.

There are two other classes of REST services that form the functional aspect of the solution and they are System Facing Interface and Smart Contract Interface.

Fig 4. illustrates the dependencies and direction of the call flow.

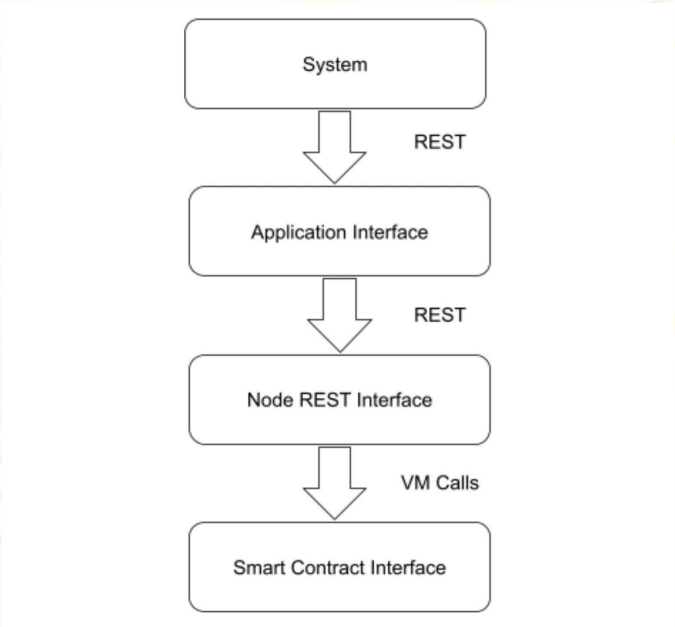



*Figure 4. Interface Dependencies*

The Camel case practice for the naming of endpoints is be adopted on both application and smart contract levels.

### 2.3.1 System Facing (Application) Interface

These are the services that provide high-level functionality such as the already mentioned minting, value transfer, burning, balance reset and balance reading. These services of the solution will be used by the system.

The interface should remain as much as possible neutral to the underlying blockchain type and provide stateless interaction. This means that two consequential calls to the same or different endpoint from the interface should be bound to an internal state, thus making the second call dependent on the output of the first call as an internal constraint.

### 2.3.2 Smart contract Interfaces

The high-level services defined in the application interface requires the same or lower level of granularity on the level of interface definition of the smart contracts. The smart contract language will enable appropriate interface definition so that different implementations of the same interface.

The interface on the smart contract level must declare parameters as close as possible similar types and naming conventions used in the application interface to facilitate easier management and support.

## 2.4. Blockchain Network

Initially the solution uses a private setup of a blockchain network that is managed and monitored by the owner of the system or other authorized organization(s). At a later stage, the transactions must be moved onto a public blockchain. Thus, the need for a complete history of transaction requests and their statuses. The selected blockchain network is based on the Ethereum software and its established codebase.

### 2.4.1 Consensus Algorithm

The necessary consensus algorithm must be energy-efficient and lightweight at the same time. Such an algorithm must not rely on Proof of Work (PoW) for its inefficiency in many aspects. A very strong candidate for such an algorithm is the Proof of Authority (PoA) which allows for faster block times and does not rely on heavy resources.

### 2.4.2 Block Time

The desired block time must be not higher than 10 seconds. Preferably it should be 5 seconds. This block time will create comfortable transaction execution times that will not confuse the users.

### 2.4.3 Software and Consensus Implementation

The blockchain network is implemented with a solid codebase and is supportable in the long term. Having this requirement in mind, the official Ethereum software code base is a good choice for its proven track record of network availability, support for critical updates and support for different consensus protocols. The official Ethereum distribution has a PoA implementation called Clique, which satisfies all requirements for the desired consensus algorithm.\

## 2.5. Supported Standard for Token Payments

During the last five years blockchain-based tokens gained popularity and proved to be a sustainable (technically seen) approach to transfer of value and digital assets on the blockchain. Using the solution,

the users of the system will be empowered to earn, get and transfer tokens (or coins) with an extremely reliable mechanism.

The standard that has emerged as most used and stable is the ERC-20 standard. It defines the interface for implementation of smart contracts that govern the processes of minting, burning, balance management and transfer of tokens. It is foreseen that the token standard that the solution should support must be on par with ERC-20 and it should implement it fully as defined by the Ethereum community.

The standard for Ethereum-based custom coins/currencies is defined under the following URL:

*https://ethereum.org/en/developers/docs/standards/tokens/erc-20/*

The token standard defined the minimal interface calls that should exist for all other systems and users to be able to execute.

# Non-functional Requirements

The system is currently serving thousands of customers and need to provide the payment functionality to all of them. This means that the solution must meet certain criteria for input throughput and processing throughput so that the user experience remains acceptable and yet the processed payments remain secure.

## *3.1. Input throughput (transactions/sec)*

The blockchain technology has evolved during the last few years in respect to its reliability and scalability. The proposed solution should aim at the highest possible throughput at the level of the request of the user. This means that when a user is creating a request for payment through the system, this request must be handled quickly enough to not cause waiting or feeling of uncertainty on the client side. Then, the request must be converted to a transaction and processed by the blockchain.

This mechanism will assure that the user requests never get processed directly by the blockchain, rather will be put into a transaction queue prior to that. Such a queue will also provide the ability to track transaction status by querying the queue instead of submitting direct requests to blockchain gateways or other open blockchain APIs.

Figure 5 illustrates the flow of requesting a payment transaction.

The input transaction processing is handled by the payment request queue or a manager associated with it and bound to an API. The requirements for this step of the payment

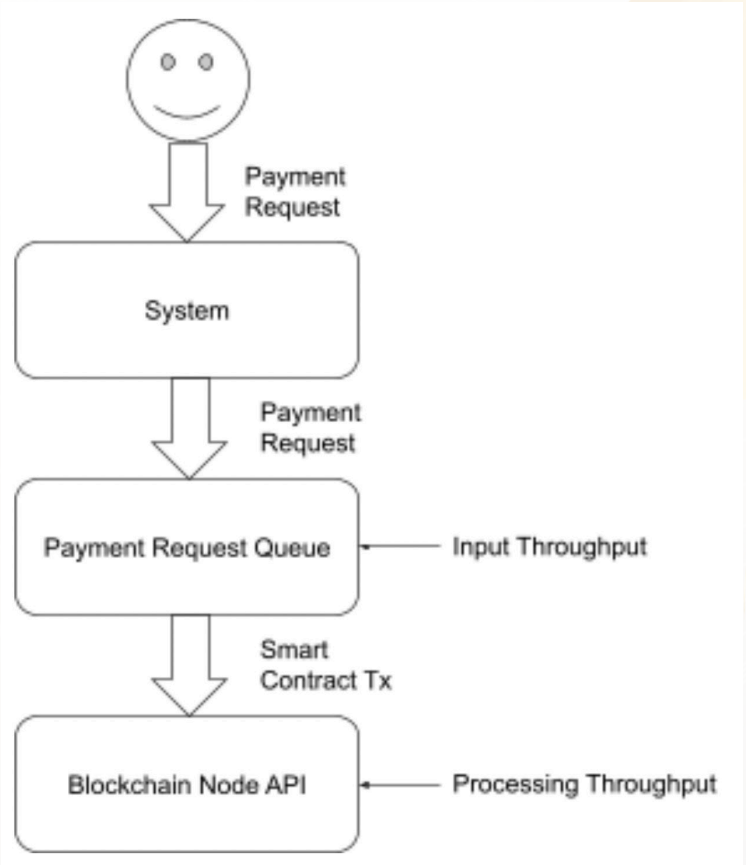



*Figure 5. Payment Request Flow*

processing is 1000 requests/s. The API should handle this number by either horizontal or vertical scaling means.

### 3.2. Processing throughput (transactions/sec)

When a payment request arrives at the level of the payment request queue it has to be prioritized and broadcasted to the blockchain network. This process practically should initiate execution of a smart contract for one of the supported actions according to the requestee's role. This happens by means of broadcasting - a mechanism used in blockchains to advertise a pending transaction which can be picked by any of the entitled block validators and placed as a sealed transaction in the new mined block. This process of "transaction processing" is a sole duty of the blockchain nodes. The speed with which a transaction is processed is determined by a number of factors, such as number of validator nodes, maximal number of gas consumed per block and block mining time. To cover the functional properties and to satisfy user experience, the processing throughput should be achievable with 100 transactions/sec.

### 3.3 Security Measures

The system will use the solution as an internally connected module. This requires the solution to be able to be deployed on premises and within the security domain of the system as a first necessary security requirement. Every API call must be actionable ONLY over a securely established channel, such as SSL enabled tunnel or HTTPS. Additionally, the exchange of user keys between the system and the solution must be secured by encrypting the keys with a shared secret credential. The solution should be accepting requests only from a limited predefined list of internet addresses and rejecting any other request that comes from a non-specified or known source address.

Figure 6 illustrates visually the deployment context of the security measures and requirements

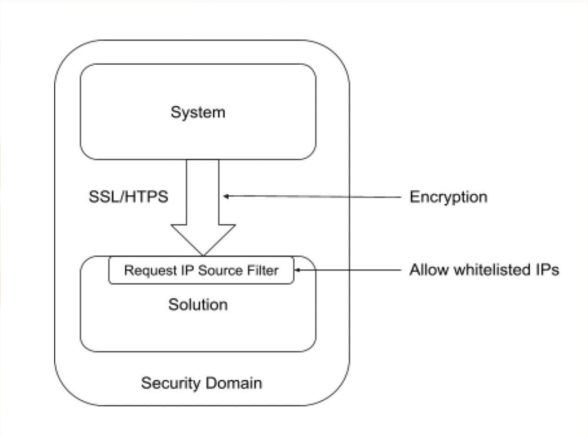



*Figure 6. Levels of Security*

Whenever a private key is being passed to the solution for the purpose of initiating blockchain transaction, the key must remain accessible only till the moment of transaction broadcasting after which the key must be wiped from the memory. The key must only be stored in the operative memory and not be stored in a file or any data-store related services such as databases.

### 3.4. Scalability

When a server which hosts the solution has not enough resources for handling the input throughput it should be possible to spread the load by using traditional load-balancing techniques to achieve horizontal scalability. Since the solution should be / is designed in a stateless way this should be straight forward to achieve.

Figure 7 gives an overview of the possible layout that has to be supported.

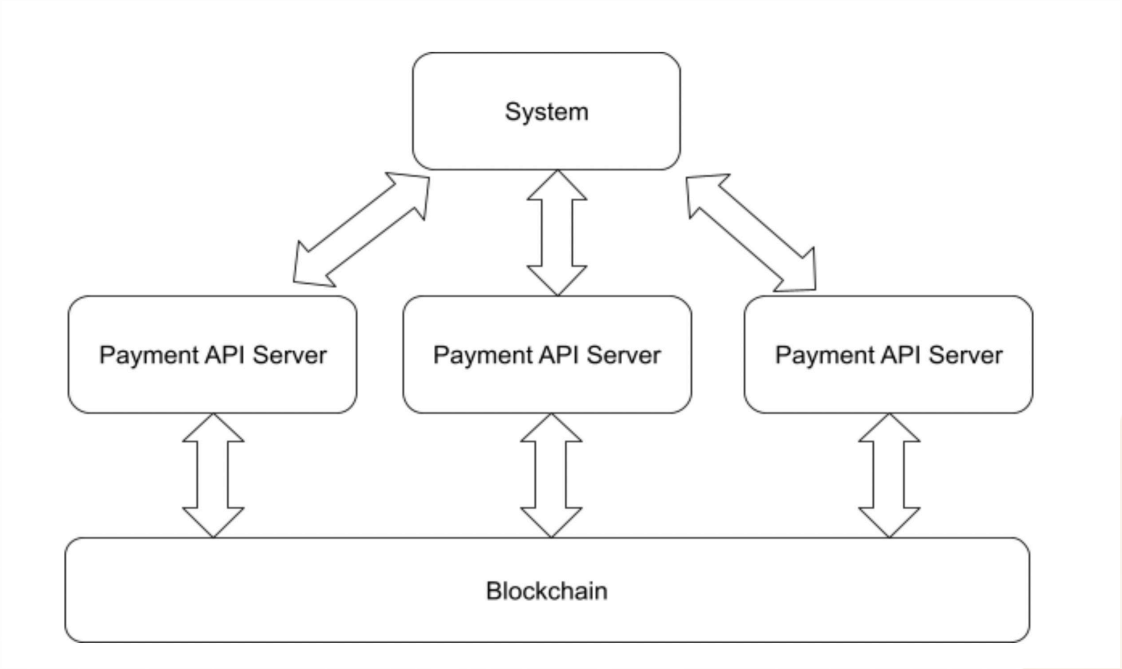



*Figure 7. Horizontal Scaling of the solution*

Each of the payment servers will handle its own set of transaction requests and will submit and execute smart contract transactions to the blockchain network independently and in parallel to achieve greater throughput.

# General Architecture and Context of the Realization

The architecture of the solution is driven by all requirements but its core components are driven by the need for input throughput and the requirements for migration. Figure 7 illustrates the building blocks and the system context of the solution.

## *4.1. Input Throughput Support*

To satisfy this requirement the solution includes a transaction queue, a manager and a scheduler.

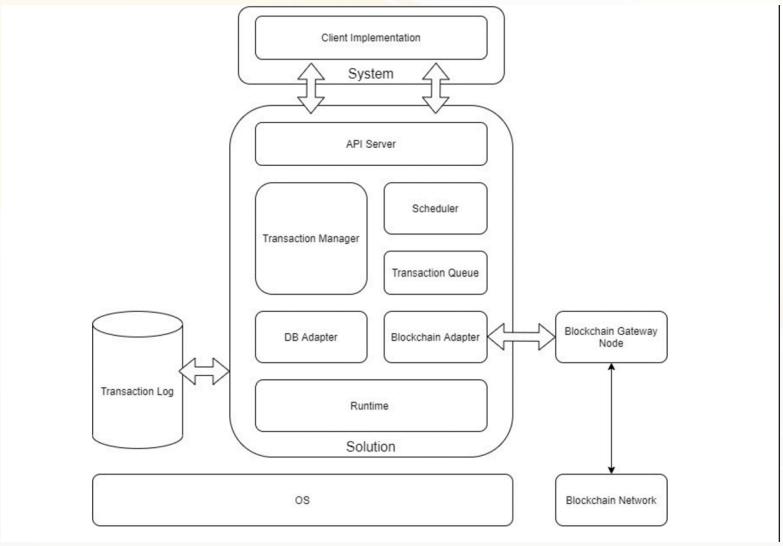



*Figure 8. Solution Architecture*

Whenever a payment request is coming in through the API server, the BPO object inside its payload is converted into a generic structure that represents a concrete smart contract call that needs to be broadcasted to the blockchain network as a transaction. This prepared structure is recorded into a transaction log after which the payment request is complete. This simple process assures a very high insertion rate and practically utilizes efficiently the capabilities of the underlying web server realization without introducing a blockchain latency factor or anything related to long asynchronous interactions with it (Fig. 9).

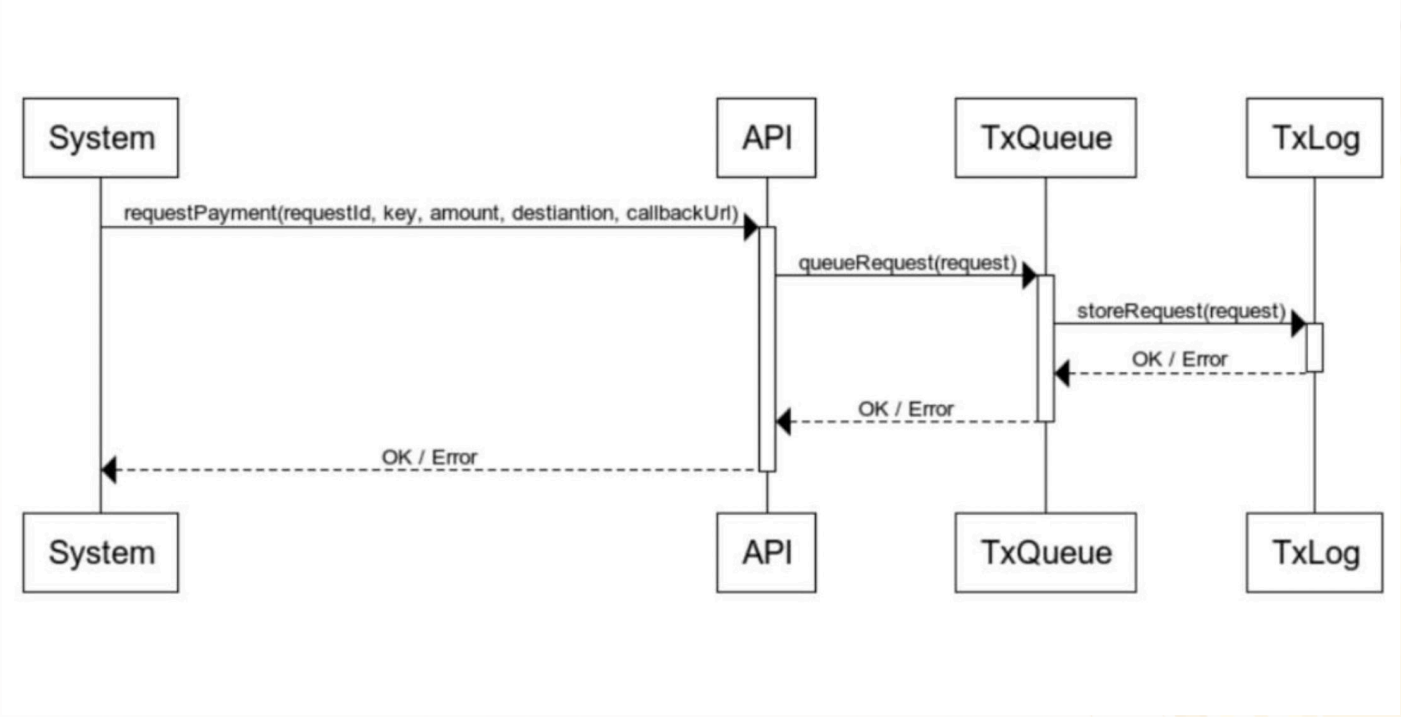



*Figure 9. Payment Request Input*

## 4.2. Transaction Log

All requests are converted into transaction objects and stored in the transaction log. This way, if necessary, a selected transaction or set of transactions can be "replayed" on any blockchain network in case a migration has to be performed and requires a complete history of the transactions to be present. Even without the need for migration, any transaction may end in a state that is not resolvable on the blockchain side and has to be resolved on the system side and replayed again to fulfil the specific payment order.

## 4.3. Blockchain Adapter

It is important to note that the migration requirement imposes a possible dependency on more than one blockchain network. This means that all the transaction details and history should be applied the same way on different networks. To solve this challenge the architecture foresees an adapter-based mechanism that allows interaction with any blockchain via a unified interface. Specific blockchain interaction is backed by a concrete implementation of that interface for the specific APIs and protocols offered by the respective blockchain gateway.

# General Framework

There are a number of  principles that drive the design and the usage of the solution, outlined here as a “framework”. Most of them are related to how a blockchain operates and how it handles user identity. A significant part of the framework is dependent on basic cryptographic concepts such as private and public keys (PKI), cryptographic signatures and hashing. The main concepts that need to be understood are User Identity, Cryptographic Keys, and the Transactional Processes into which the user identity and cryptography meet to achieve security and reliability.

## *5.1. User Identity*

User identity is a core concept that almost any multi-user environment has to deal with. The system provides identity to its users via credentials stored in a central database, a traditional way adopted by many web-based systems. Users are given these credentials and once provided they are matched, and the result of matching confirms the identity of the authorizing user.

In the case of blockchain, user identity is not stored in a central database, rather the users themselves are holding the keys with which they are in a position to authenticate themselves. This reverse way of authentication forces the management of a private key on the side of the user. If this key is lost, his identity is lost too. Going this way for a full-blown web-based platform is not optimal because users are often not in a position to hold their keys in a consistently secure environment and or keep them or remember them forever. That is why the solution foresees a hybrid approach that solves the issue of lost keys and introduces flexibility in key management.

The well-known approach of user identity handling on the side of the system is known as “custodian wallet”. In such a configuration, it is important for the system to assure well-defined and secure access to the private keys of the users which will be used to execute smart contract transactions for the purpose of the payments.

## *5.2. Use of keys and cryptography*

The fact that the payments are going to be managed by a blockchain radically changes the options of the users regarding the interaction with their digital assets (currency). While in the traditional systems they 100% rely on the central database for integrity and truthfulness, the blockchain introduces the independence and higher level of integrity by simply turning over the way identity works - private keys bound to the identity of the user.

### *5.2.1. Key Pair Binding*

For each user that is managed by central credentials there will be a corresponding private-public keypair that will be used to manage the blockchain wallet of the user. This key will be handled on the system side and assures that all payments are reflecting a single balance of a single account that corresponds to the user identity.

### *5.2.2. User Taking Account Control*

The concept of custodian wallet is that it provides user comfort in keeping their digital assets secure and at the same time assures that the assets are still in the reach of the user if she wants to take full control of them. This can only happen if the user is in full possession of the private key that corresponds to the account into which the assets are residing. Looking at this requirement functionally, If a user wants to get full control on the wallet, he can do so by requesting the private-public keypair from the system.

This means that from that moment he gets his private key, the user is responsible for the safety of his digital assets (coins) because the private key is not within the security domain of the system. If the user wants to move the assets to an account of his choice on the same blockchain network he can do so by opening a standard wallet implementation and importing his private key, for example in “Metamask”.

There is an additional measure to be taken if such a process takes place. Once the user gets his private key, the system should not be able to access that account anymore. In practice, it is impossible to provide the actual private key used by the system previously because the system will be still holding a copy of the private key somewhere. That is why the user will be downloading a new private key for a new account, which will be with a balance of 0 before the completion of the process of taking control. Upon completion, the system will initiate coin transfer action to the new account for which it has no private key, respectively no control. This way the user will be sure that he has access to the assets that the system has kept previously and can move them to another self-managed account.

## 5.3. Transactional Processes

Given the nature of the blockchain technology each function call that is requested from the system must be thought of as an asynchronous one. Depending on the block sealing speed the result from a call may be delayed which inevitably reflects in the way the user experience and user interface must be designed so that the users are informed about an ongoing process in the background. After a successful input of a request, there are two strategies of approaching the issue with resolving the status:

- poll - the system polls the solution via an API about a status of a request by providing request identifier
- push - the system is waiting for a call-back call from the solution which executes a specified ReST call on the system with provided status changes.

Figure 10 and Figure 11 are visualizing the sequence for both strategies.

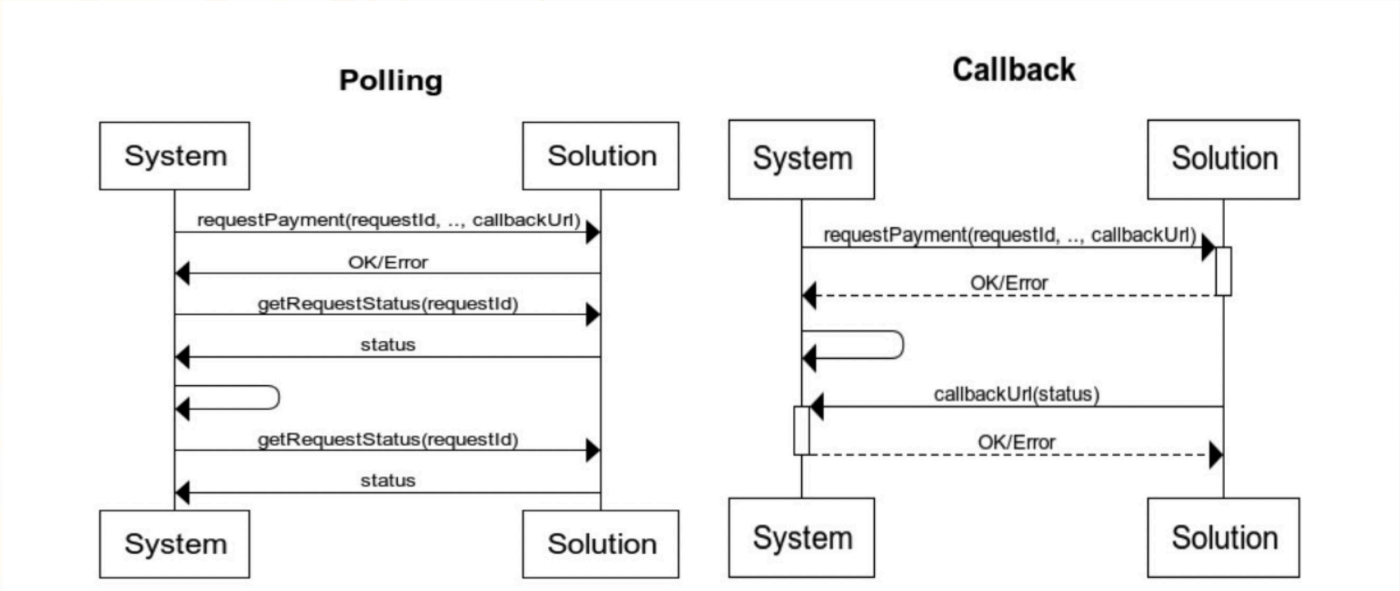



*Figure 10. Payment Request Status Polling*




*Figure 11. Payment Request Status Callback*

It is important to note that the callback Url must be executed more than once if the system returns “Error”. The number of executions must be limited to a maximal value of 4.

The requirement for the implementation of the solution includes both strategies because a system may need to get a status on user or on system demand but also the system should get the status at the first possible moment without polling.

# Interface Specification

There are two categories of interfaces according to the architecture - application level interfaces, namely the ReST services layer and the smart contract interfaces.

## 6.1. Smart Contracts

The most essential smart contract is the coin contract. As already mentioned, it must follow the ERC-20 standard for interoperability and integrity reasons. To achieve the desired functionality for minting and burning, the ERC-20 interface will be extended with options for minting and burning but with a multisig option, so that only when all necessary signatures are placed the action will be fired.

Another important aspect of an Ethereum software-based blockchain is the notion of Gas. It represents a security constraint that increases the economical burden on an attacker who aims at a Denial of Service (DoS) attack towards accessible nodes. Limiting the available gas to regular users decreases the chance of attack. That is why it is recommended to have a gas supply strategy for an account to be able to execute a payment transaction. The gas should be supplied in the form equivalent to the native "ether token" at the moment after creating the account. Additionally, after each transaction executed by an account, its gas balance is reduced with the required gas for the executed transaction. That is why the solution may implement a top-up smart contract that will handle automated fill up of the account with required gas for at least one transaction. This way the contract may decide to fill up or not to fill up a specific account and/or to ban it if there are indications for malicious behaviour on the network. The manual gas supply function should be also accessible to the owner or admins of the smart contract that handles the coin.

### 6.1.1 ERC-20 Contract

***Standard Interface Methods***

function name () public view returns (string) function symbol() public

view returns (string) function decimals() public view returns (uint8)

function total Supply( ) public view returns (uint256)

function balance of (address _owner) public view returns (uint256 balance)

function transfer (address _to, uint256 _value) public returns (bool success)

function transfer from (address _from, address _to, uint256 _value) public returns (bool success)

function approve (address _spender, uint256 _value) public returns (bool success)

function allowance (address _owner, address _spender) public view returns (uint256 remaining)

***Events***

event Transfer (address indexed _from, address indexed _to, uint256 _value)

event Approval (address indexed _owner, address indexed _spender, uint256 _value)

### 6.1.2 Extended Interface Methods

The methods for minting and burning (mint and burn) are executed only when the number of required signees is reached. Each signee must execute the mint or burn method in order to confirm where the _amount must match every time. The number required is set by setRequiredSignees. For an address to

become a signee the owner of the contract must execute the setSignee method. Signees can be excluded from the list of entitled signees when the owner of the contract executes setSignee with value false of the _allowed parameter. Every time action of mint or burn is executed an event gets fired - Mint, respectively Burn. Respectively, whenever a change in the status of a signee occurs the event Signee is fired.

***Methods***

function setSignee(address signee, bool _active) public returns (bool success) function

setRequiredSignees(uint8 _signatures) public returns (bool success) function

mint(uint256 _amount) public returns (bool success) function burn(uint256 _amount)

public returns (bool success)

***Events***

event Mint(uint256 _value) event

Burn(uint256 _value)

Event Signee(address indexed _signee, bool _active)

### *6.1.3 Faucet Contract*

The faucet contract is taking care of filling up the accounts after they execute a transaction through the coin smart contract. The loadGas method is payable, thus it accepts posted value in the transaction object. The value will get accumulated in the contract and will be available for disposition to claiming accounts. The method claimGas will execute an ether transfer transaction to the calling account. Respectively the ClaimGas is fired when an account successfully claims its gas. The loadGas method is evoked by the coin contract right after its method transfer or transferFrom are called.

***Smart Contract Interface Methods***

function loadGas() public payable returns (bool _success) function claimGas() public returns(bool _

success)

***Events***

ClaimGas(address account, uint256 amount)

### *6.1.4 Sales Contract*

The sales contract will handle the conditional payments as described in the flow in paragraph 2.2.2. It should keep the blocked balances for each sale transaction between buyer and seller with the possibility to include multiple transactions by a single buyer to a single seller and multiple buyers to multiple sellers. The system will provide an identifier for each such conditional transaction initiated with method startSale as a way to address concrete transactions for the actions confirmSale or cancelSale respectively for finalizing the payment and for cancelling the payment and returning back the balance to the buyer.

***Smart Contract Interface Methods***

function startSale(uint256 amount, address buyer, address seller, bytes32 identifier) public payable

returns (bool _success) function confirmSale(bytes32 identifier) public returns(bool

_success)

function cancelSale(bytes32 identifier) public returns(bool _success)

***Events***

NewSale(uint256 amount, address buyer, address seller, bytes32 identifier)

SaleConfirmed(bytes32 identifier) SaleCancelled(bytes32 identifier)

### 6.1.5 Trader Contract

The trader contract will handle the token exchange process as described in the flow in paragraph 2.2.3. It should keep its own balance on each pair of exchanged tokens and transfer tokens or system currency on demand. The exchange rates will be set as a parameter at the moment of the execution of the method exchangeToken. The balances for each supported token will be topped up using standard ERC20 transfer to the token account. Available tokens can be withdrawn at will by the administrator by using the method withdrawTokens. The case for voucher exchange can be covered by two pairs - CURRENCY-VOUCHER and VOUCHER-CURRENCY. The method exchangeTokens is expecting as parameter pairId. For these two pairs, the respective IDs are 1 and 2. The method withdrawTokens is expecting token identifier which for CURRENCYis 1 and for VOUCHERis 2.

***Smart Contract Interface Methods***

function exchangeTokens(uint8 pairId, uint256 amount, address buyer) public payable returns (bool _success)

function withdrawTokens(bytes32 identifier, address beneficiary) public returns(bool _success)

***Events***

TokenExchange(uint8 pairId, uint256 amount, address buyer) Withdraw(bytes32 identifier, address beneficiary)

## 6.2. Application Interfaces

The API provides ReST endpoints for writing and reading information from the blockchain.

The requests that are related to reading are invoked as GET HTTP requests, while requests that are writing to the blockchain are submitting a POST payload.

### 6.3.1 Account creation

The action of creating an account is not resulting in any reading or writing on the blockchain. The server is creating a keypair from entropy and encrypts the private key with a shared password. The result of the action is a read keypair that the user can supply when blockchain operations are invoked on the server.

An example for creating a user is a GET request with the following format:

| Endpoint | Method | Parameters |
|---|---|---|
| /new | GET | --- |

*http://localhost:3000/new*

*The invocation results into the following response:*


{
"status": "OK",
"code": 200,
"data": {
"address": "0x1CD6bfD36b9EF425849CA25EEB248f5cE439Cabd",
"privateKey":
"qSjsOp7bT2um+D+muNpd7bgLm2qIaNqGMuqon4jpOzVe4qH9dXsZOdubBY9Bej2E6Btb9pFpE3c1
t7EfhpLiCz24mjwCBakbPIbuKCldIpduzg=="
}
}


### 6.3.2 Read Operations

Each read operation is requested to an API through a GET request which has the structure:


{
"status": "OK",
"code": 200,
"data": {
}
}


#### 6.3.2.1 Reading the coin balance of a user

| Endpoint | Method | Parameters |
|---|---|---|
| /coinbalance | GET | Param 1: privateKey |

It is important to note that the privateKey parameter expects URI-encoded string.

http://localhost:3000/coinbalance?privateKey=JxD37oc%2FWwq
di9kifDiyZ9ClBNb5tL8lZeF3tSQkauWH6l8IufIR6t9Rwvf2%2FGojR3O
faZBf4OzBM95pX5Wan1p43Pi1FxwEwk%2BP5U3%2FfKeNTA%3D%3D

***This request results into:***


{

“status”: “OK”,

“code”: 200,

“data”: {

“address”: “0x5ecE9372c0BaC1d408b446C0a9a0E713106a3fD8”,

“balance”: “999999000000000000000000”

}

}


### 6.3.2.2 Reading the gas balance of a user

| Endpoint | Method | Parameters |
|---|---|---|
| /ethbalance | GET | Param 1: privateKey |

It is important to note that the privateKey parameter expects URI-encoded string.

http://localhost:3000/ethbalance?privateKey=JxD37oc%2FWwqd
i9kifDiyZ9C1BNb5tL8lZeF3tSQkauWH6l8IufIR6t9Rwvf2%2FGojR3Of
aZBf4OzBM95pX5Wan1p43PilFxwEwk%2BP5U3%2FfKeNTA%3D%3D

***The request results into:***


{

“status”: “OK”,

“code”: 200,

“data”: {

“address”: “0x5ecE9372c0BaC1d408b446C0a9a0E713106a3fD8”,

“balance”: “18191570690000000000”

}

}


### 6.3.2.3 Retrieving transaction info

At any time information about ongoing requests can be retrieved through the API by just providing the requestId initially set in the request.

| Endpoint | Method | Parameters |
|---|---|---|
| /ethbalance | GET | requestId |

**requestId** is a reference that the client specifies for tracking purposes at the moment of request POST.

http://localhost:3000/txstatus?requestId=paymenttx-1

This request results in a detailed object that contains the current status of the request, as well as information related to the execution of the transaction on the blockchain (the transaction receipt object).

If a transaction request was submitted with a repeating requestId the result of txstatus will be a list of transactions.


*{*

*“status”: “OK”,*

*“code”: 200,*

*“data”: [*

*{*

*“tx_id”: 167,*

*“request_id”: “paymenttx-1”,*

*“request_body”:*

*“{\”action\”:\”payment\”,\”params\”:{\”recipient\”:\”0x1231231231 23\”,\”amoun t\”:10000000}, “currency” : “TOKEN”}”,*

*“request_sig”: “”,*

*“request_timestamp”: “2021-09-29T13:56:16.000Z”,*

*“keypair”:*

*“{\”publicKey\”:\”0x5ecE9372c0BaC1d408b446C0a9a0E713106a3fD8\”,\”secretKey\”: \”CII28K0IJBfkWt4RKiYLoG/S66p/nGzzexWdzsKRkmIKActfikbFj0OBGDckgq6+U/zom5zPGB+ rowE/ ayf4Zx4isiosa48Ja+GGQmEN/Q1PNQ==\”}”,*

*“contract_id”: “0x9cdA24945a0fAA87E92a159E0aB3B827fb84e06E”,*

*“tx_hash”:*

*“0x09951088ccb1d3d557954e817778d10f0131c4b73051f121e28d7398f39e13e9”,*

*“tx_receipt”: “sample receipt”,*

*“tx_receipt_timestamp”: “2021-09-29T13:56:30.000Z”,*

*“tx_attempts”: 1,*

*"tx_last_attempt_timestamp": "2021-09-29T13:56:30.000Z",*

*"tx_status": "complete",*

*"callback_url": "http://localhost:3000/callback",*

*"callback_response": "{\"status\":\"OK\",\"code\":200,\"data\":{}}",*

*"callback_status": "complete",*

*"callback_attempts": 1,*

*"callback_last_attempt_timestamp": "2021-09-29T13:56:32.000Z",*

*"last_update_timestamp": "2021-09-29T13:56:30.000Z"*

*}*

### 6.3.3 Write operations

Every request that results in writing information on the blockchain needs a Blockchain Payment Object (BPO) which is submitted as a payload to the **/action** ReST endpoint (see below).

#### 6.3.3.1 BPO Format

The Blockchain Payment Object (BPO) is a JSON structure that describes the intent of the user to execute an operation provided by a smart contract and the location of the callback that has to be invoked upon execution.

The information which BPO wraps is sufficient to

- Identify the user who is executing a transaction
- The specific action that has to be executed
- The parameters of the action
- The callback URL that has to be invoked upon completion of the action, specifically upon the validation of the blockchain transaction that invoked the smart contract behind the requested action

***An example BPO is:***

*{*

*"id" : "paytx-1",*

*"payload" : {*

*"action" : "payment",*

*"params" : {*

*"currency" : "CHR",*

*"amount" : "1",*

*"recipient" :*

*"0xC843dAEca73F236386D9e3AF69E0a551b3bFF9db"*

}

},

“identity” : {

“keyPair” : {

“publicKey”:

“0x5ecE9372c0BaC1d408b446C0a9a0E7131106a3fD8”,

“secretKey”:

“CII28K0IJBfkWt4RKiYLoG/S66p/nGzzexWdzsKRkmIKActfikbFj0OBG

Dckgq6+U/zom5zPGB+rowE/ayf4Zx4isiosa48Ja+GGQmEN/Q1PNQ==”

}

},

“callback” : {

“type”: “url”,

“action”: “http://localhost:3000/callback”

}

}


As it can be seen there are three major sub-objects - payload, identity and callback.

In this specific example the **payload** object describes the requested action “payment” which writes data to the blockchain. The parameters that the action expects are “currency”, “amount” and “recipient”.

In the **identity** object the public and private key in encrypted form are provided as a means for carrying out the execution of the transaction on behalf of the user. The secret key is encrypted with a password that the backend used to encrypt it in the first place when the creation of the account was requested.

The **callback** object is providing the concrete location of an HTTP callback endpoint that will be invoked upon execution of the transaction on the blockchain, or upon hitting an error state. The information is sent as payload of a POST request to the supplied URL.

### *6.3.3.2 The /action endpoint*

All POST operations (writing to the blockchain) are performed through the **/action** endpoint. It has no parameters but expects as request body the JSON form of the BPO object.

| Endpoint | Method | Parameters |
|---|---|---|
| /action | POST | --- |

Each successful call of **/action** endpoint should return immediately with the following expected body with JSON structure.

{
“status”: “OK”,
“code”: 200,
“data”: {
“id”: “send-IIIII”,
“internalId”: 173
}
}

The result contains the initial request id and the internal identifier for that transaction.

#### 6.3.3.3 Sending ether to account

When a new user account is created it contains 0 ether. Without ether the account is not capable of executing transactions on the blockchain. To enable an account to start executing transactions one needs to send ether via the /action api:

| Payload Action | Parameters |
|---|---|
| reload | Amount : 18 decimals number,<br>recipient: 0x leading address |

“payload” : {
“action” : “reload”,
“params” : {
“currency” : “ETH”,
“amount” : “1000000000000000”,
“recipient” :
“0x24428B1DeB024b25F940281980Cd823b19440BC9”
}
}

#### 6.3.3.4 Sending coins to account

Coins can be moved (sent and received) between accounts using the **payment** action. It is very similar to the **reload** action, but instead of ether, coins are exchanged.

“payload” : {
“action” : “payment”,
“params” : {
“currency” : “CHR”,

*"amount" : "1",*

*"recipient" :*

*"0xC843dAEca73F236386D9e3AF69E0a551b3bFF9db"*

*}*

*}*

### 6.3.3.5 Paying to a seller with a conditional payment

When a buyer needs to pay for his delivery to a seller, the payment object should specify this with a special property in the params object conditional. If this property is not specified the conditional property should be considered to be with value false and the payment type atomic. The conditional payment requires an identifier too, which is provided through the property depositId. It must be unique for each conditional payment transaction.

*"payload" : {*

*"action" : "payment",*

*"params" : {*

*"currency" : "CHR",*

*"amount" : "1",*

*"recipient" :*

*"0xC843dAEca73F236386D9e3AF69E0a551b3bFF9db",*

*"conditional" : "true",*

*"depositId" : "0x1231231231231231231"*

*}*

*}*

### 6.3.3.6 Confirming or cancelling conditional payments

A conditional payment must be confirmed or cancelled for it to be complete. This action is only available for the system role. The action in the BPO is "confirm" or "cancel" for confirming or cancelling a specific payment transaction respectively. The important parameter is the payment ID that was initially supplied when the transaction was initiated with "payment" action.

*"payload" : {*

*"action" : "confirm",*

*"params" : {*

*"depositId" : "0x1231231231231231231"*

*}*

*}*

### 6.3.3.7 Exchanging Tokens

Coins can be exchanged for other tokens or vouchers which are presented in the network as ERC20

tokens using the **exchange** action. Specific for this action are the parameters pair and buyer which specify the trade pair and the beneficent of the received token. The parameter rate is optional. It specifies the exchange rate specific for this trade pair and if not specified the default exchange rate set in the system configuration will be used.

*"payload" : {*

*"action" : "exchange",*

*"params" : {*

*"pair" : "1",*

*"amount" : "100",*

*"rate" : "1.33",*

*"buyer" :*

*"0xC843dAEca73F236386D9e3AF69E0a551b3bFF9db"*

*}*

*}*

## 6.4 Errors

Each request may be completed with either a success or error. There are three types of errors that each call (read or write) can produce - validation error, network error, blockchain error. This section describes the minimal set of errors that must be covered by the solution.

### 6.4.1 Error Result Structure

When a request fails with an error, the server will return its HTTP response with an error code of 5XX and a body payload. In the body payload, the error code will match the HTTP error code and will indicate the status in the response structure as "ERROR". If the error is on the level of blockchain, such as a smart contract error, then the concrete blockchain receipt will be returned in the data property of the body payload which follows the defined response structure.

***Example:***

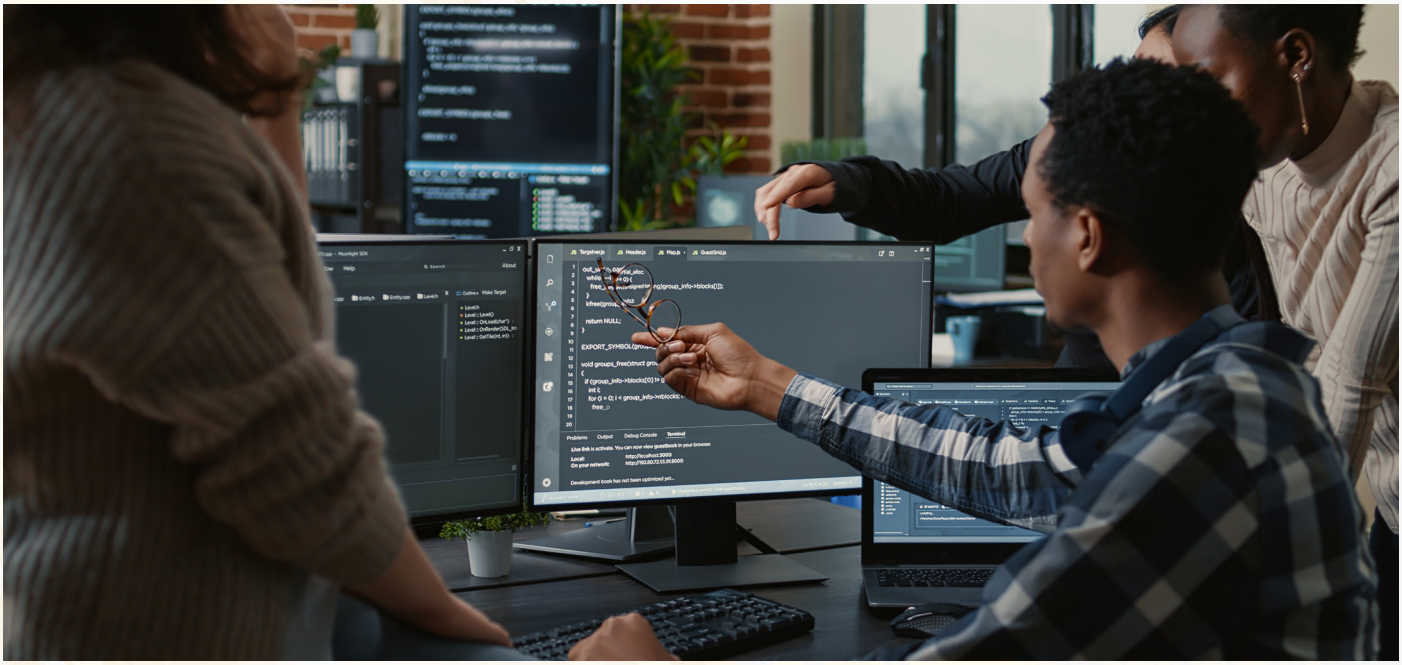
***{***
***"status": "ERROR",***
***"code": 511,***
***"data": {***
***"receipt": {***
***}}***
***}***

### 6.4.2 Validation Errors

A validation error is returned to the requesting client if some of the parameters values are not meeting the validation rules of the service layer. The validation may check the payload structure and return error if there is a missing parameter. Another check that can raise validation error is the value check. An example for such error is the failure to match private keys with the specified account addresses in

the payload. Another example is the error raised when trying to transfer a bigger number of tokens than the current balance of the user.

| Error Code | Description |
|---|---|
| 510 | Wrong action. The specified action is not recognized. |
| 511 | Missing parameter for the specified action. |
| 512 | Private key missing or invalid |
| 513 | Mismatch of private key and account number or account number invalid |
| 514 | Insufficient token balance |
| 516 | Unknown account address |
| 517 | Unknown request ID |
| 518 | Invalid number specified as amount |
| 519 | Unrecognized token symbol |



# Validation Errors

### 6.4.3 Network Errors

The service layer is communicating with blockchain nodes in a network environment. It is not excluded that the communication between the service layer and the blockchain gateway node fails. In this case the client can receive an error that indicates that something with the blockchain access is failing.

| Error Code | Description |
|---|---|
| 520 | Blockchain gateway unreachable. The service layer cannot access the blockchain node. |
| 521 | Blockchain response timeout. After accessing the node the service layer is not receiving a response. |

*Network Errors*

### 6.4.4 Blockchain Errors

| Error Code | Description |
|---|---|
| 530 | Smart contract returns an error. The transaction receipt is returned as a data field in the body. |
| 531 | Transaction inclusion is delayed. It takes too long for the transaction to be mined. This does not necessarily indicate complete transaction failure, rather only its delay. |
| 532 | Transaction inclusion failed. There were repetitive trials to push the transaction but every time it failed. |
| 515 | Insufficient gas balance. The executing account does not have enough gas balance to cover the transaction costs. |

# Blockchain Errors

When a transaction that includes smart contract execution is submitted to the blockchain, the smart contract can raise an error or its execution can be completed with success. Another case in which a blockchain error can be raised is if a transaction is not included within a threshold of newly mined blocks. This specific error may indicate a blockchain restriction limitation that hinders transaction inclusion. Examples of factors that may bring this type of error are insufficient gas amount on the balance of the executing account or specification of the incorrect nonce.

## *7. Migration Strategy*

An important concept that the solution should be able to support is the concept of migration to another blockchain network, different from the one it started and was used for a period in time during which the users had used the solution and created a record of transactions.

### *7.1. ReplayingTransactions*

The process of migration may or may not need to transfer the complete list of transactions on another network, but in order to ensure replayability, the solution must keep an archive of transactions and their final status. It should reflect the reality recorded on the blockchain. This way one can replay all transactions on the new network and have identical state at the end of the process, with different timestamps of course.

### *7.2. Migration of Balances*

A different approach to that of replaying transactions is the ability of the system to directly set the balances on the new network and disable the transactions on the old network in an atomic way per user account.

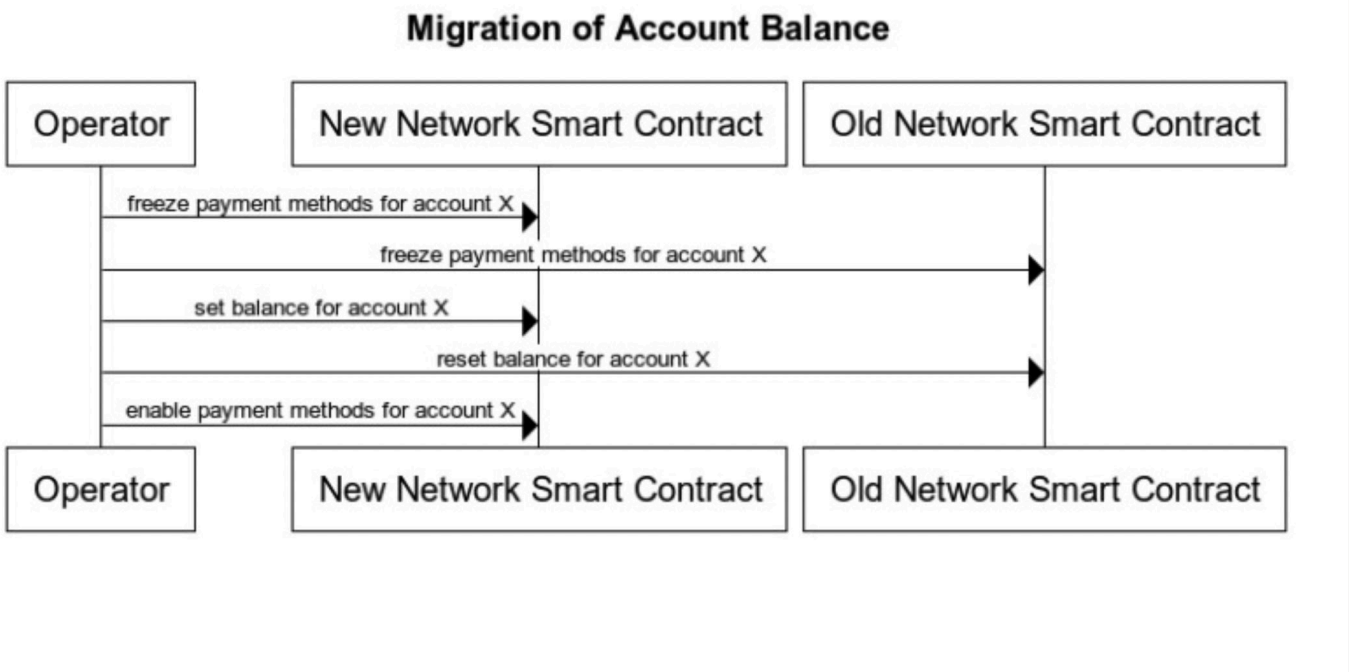



*Figure 12 Balance Migration Sequence*

Figure 12 shows the sequence of actions necessary to achieve atomic balance migration per account. This sequence must be performed for each account from the old network. The sequence ensures the fact that while the balance is migrated it will not be changed accidentally or on purpose, nor can this happen on the new network before the balance has been migrated finally. After the migration the new smart contract methods are enabled for that user while the old smart contract becomes unusable for the user.

### *7.3 Account Identifier Migration*

The way cryptography works in respect to user account identifiers and private keys relation does not allow for having a different private key for the same identifier. This means that the target new network must have the same cryptography mechanism and account abstraction as the initial network for a full compatibility between the account identifiers, which in practice must remain the same on both networks.

This limitation, however, does not make the migration impossible if the two networks are different in their consensus algorithms. For example, accounts can be safely migrated between two Ethereum based networks one running PoW and the other PoA or any other supported by the official Ethereum software consensus.

# Guidelines for Implementation

The implementation can be realized using any technology stack capable of:

- Serving ReST endpoints
- Communicating to and with Ethereum Node API
- Achieving input throughput as required

Such available technology stacks can be built on top of .NET, Java, Python or NodeJS.

### *8.1. Work Environment*

There are no limitations to how the implementation will be developed and what development environment will be used, however the project should keep a record of development history in code repository with a distributed version control, preferably Git.

Git is supported by most development toolchains including Visual Studio Code, IntelliJ Webstorm, Netbeans and others. The advised work environment is Linux for its greater flexibility in terms of file management, directory management and process control compared to Windows and Mac.

### *8.2. Technological Stack and Dependencies*

#### *8.2.1. Web Services Middleware*

Having in mind the fairly stable implementation of a Ethereum client, web3js is a notable choice when it comes to interaction with the blockchain, thus for similar applications the preferred platform is NodeJS. The advised version of NodeJS is 14.x for which there is an adapted web3js package that reflects recent package dependency chains.

### 8.2.2. Database

Database solutions can make a big difference in how an application performs. The specific of the solution does not imply features that are exclusive for a specific database engine. MySQL has a consistent development track and on-time security updates, performs well in the Linux environment and has already a large base of developer support and accumulated best practices as well as paid support.

### 8.2.3. Blockchain Connector

While there are different libraries available for connecting to a blockchain node and executing transactions, the official web3js library enjoys the full features support and is tailored to specific smart contract APIs. Another usable library is ethers - it enjoys a healthy community and is used in a significant number of projects.

### 8.2.4. Operating System

The OS-level environment a matter of choice but having in mind that many of the web-based technology stacks (NodeJS, PHP, Python, etc) being developed mostly on Linux and a lot of the available packages are taking advantage of the native implementations of cryptography libraries Linux distributions are likely a better choice for an operating environment. Both Ubuntu 20.04 and CentOS Linux 8 have proven themselves as very capable operating environments with good support for security updates.

## 8.3. Configuration Parameters

The solution components are implying configuration parameters that vary depending on what the environment is and what is the nature of the network into which the solution is deployed.
The known configuration parameters are outlined in Table 2. Specific configuration keys are not advised and should be assigned by the developers.

| Component | Parameter | Comment |
|---|---|---|
| Blockchain Adapter | Smart Contract Address | The address which a payment transaction will use to transfer tokens between user accounts |
| | Smart Contract ABI | The description in machine and human readable form that defines strictly the payment and the management interfaces of the contract |
| | Master Account Address | Account address which is entitled to distribute gas for new accounts. |
| | Master Account Key | The credentials used to sign a blockchain transaction on behalf of the master account. |

| Component | Parameter | Comment |
|---|---|---|
| Transaction Scheduler | Trigger Timeout | The time period between triggers of the transaction management routine and transaction broadcasting. |
| | Tx Error Retries | Number of retries for transaction broadcasting in case of error. |
| | Callback Error Retries | Number of retries for callback execution in case of callback error. |
| ReST Service Server | Server Port | Port on which the server will start listening for HTTP/S requests |
| | Server Interface | The IP interface on which the server will listen |
| | HTTPS Support | Indicates if the server is capable of serving HTTPS |
| Database | Database host | The remote host which will serve the data |
| | Database port | The port on which the database server is listening for requests from the ReST server. |
| | Database Username | The username part of the access credentials to the database server |
| | Database password | The password part of the access credentials to the database server |
| | Database name | The name of the database schema that will be used by the ReST server. |
| Logging | Log level | The level which will be used to start the server with. |
| | Log location | Directory into which logs will be stored. |

*Table 2. Configuration Parameters*

# e. THE ROADMAP

The world of cryptocurrency consists almost exclusively of international clients, mainly companies and wealthy private individuals, there is need for global financial solutions.
Corporate clients especially have in addition to a desire for comprehensive advice and solutions from a single source, considerable needs on specifically tailored solutions. A significant number of the customers are coming from Asia.

**The One Ecosystem's aim** is to build a best of class cryptocurrency operation, capitalizing on the solid existing platform and people, ensuring continuity of the cryptocurrency activities, services, and clients, strengthening its balance sheet through new capital and corresponding relations, while bringing new retail clients to the business from a diversified range of Emerging Markets ("EM") and developed jurisdictions. The objective is to continue to capitalize on the excellent business contacts and reputation.

During the next 12 month period the company is planning to go on  a main net to make the blockchain public and we will join a public network, in that matter our transactions will be confirmed by the miners  of the network, this will lead to a more clear process and will be a complete proof of concept visible outside our centralized blockchain, keeping with our very transparent One Ecosystem processes.

The existing business model will remain unchanged, ensuring continuity of activities, while expanding the retail client base. The targeted increase in profitability will be achieved through expansion of the retail client base introduced by the new stakeholder, which in turn will boost the income, as well as synergies created with the corporate business introduced by the existing stakeholder.
The capital base of One Ecosystem will be significantly strengthened, to ensure solid financial prospects of the institution during difficult economic conditions in the market. After the blockchain was improved in 2021, the company focused its efforts on completing the One Forex and One Exchange projects. Through significant client base enlargement and wise operating cost policies the Company expects to recover its profitability, increase the variety of the services and products offered.

Due to the continued interest of more and more consumers around the world, in order to make the products more accessible to the end user, the company's business plan includes defining areas and providing franchises for these areas.

You may have heard of the study which says that only one in ten start-ups survives after the 5th year. And only one in ten of them continues to exist after the 10th year, ie. only 10% of all new companies have a chance to grow and survive for more than 10 years. It must be true. According to another study by the International Franchise Association (IFA), about 90% of franchised businesses were still on the market at the end of the fifth year.

But what is a franchise? According to one of the accepted definitions, franchising is a vertically cooperative organizational system of legally independent companies that sell goods and services on a contract basis. This system manifests itself as a single structure and is characterized by the distribution of functions between partners. One company (called a franchisor or parent company) gives a branch company (called or franchisee) a license to carry out a certain business. This is done under specific contractual terms.
In more understandable language, the franchisee buys the right to do business under the name and approved structure of the franchisor, for which he/she pays, etc. franchise price.
It consists of: Initial franchise fee (entry fee). Current royalties, which are usually monthly fees for national advertising fund

***The most important advantages of this form of business are:***

- The entrepreneur uses the successful and well-known trademark of the franchisor, as well as his reputation.

- Franchising allows you to start a business with significantly less capital than in the usual case of self-employment. This is due to the saved costs for research and building a successful system of work provided by the parent company.

- Another advantage for start-up entrepreneurs is the help they receive from the parent company. This enables businesspeople, without any economic experience or knowledge of the industry, to create a very successful business.

This concept requires those who will receive a franchise for specific areas to follow the rules imposed by the operational manuals, as well as to consult about any change.

***Organizational Structure & Management Process***

Business continuity is of utmost importance for the stakeholders, therefore current management and structure will not be significantly changed in the following 12 to 24 months. It is planned to strengthen the Company with non-executive experienced professionals, mainly to capitalize on their knowledge of industry in the regions the Company plans to expand in the forthcoming years. The introduction of significant client base and expansion to new geographical markets will require additional human resources in all the departments involved and these requirements are already properly budgeted.

Moreover, the Company will re-evaluate all the business flows and processes and will proceed with any optimizations required to further increase efficiency as well as ensure provision of high-quality services to a larger client base. That exercise should enable the Company to achieve even higher automation and integration of new technologies as well as the introduction of new locations in Asia and LATAM.

***Business Model***

To provide flexible, cost-efficient, and intelligent solutions to clients in need of international banking, building on the existing IT and human infrastructure. Clients are expected to come from a wide range of jurisdictions in Asia, Middle East, Europe, and Latin America.

***Corresponding relationships***

A prominent focus is to ensure all compliance policies and controls are there, the systems are robust, and all the operational processes are in place and work properly. The Company aims to improve its corresponding bank relationships by enforcing strict regulatory compliance, by strengthening its capital base, and of course by bringing additional transactional business in the corresponding currencies.

The Company aims to expand and develop not only the existing corresponding relationships but also add new to provide its clients with better services. Highest priority is to be devoted to establishment of corresponding relations in Asia and Latin America, in order to facilitate the needs of the markets the Company is planning to expand in the near future.

# f . COMPLIANCE PROGRAM

The Due Diligence (DD) Compliance Procedure

The DD procedures are performed to assess the risks to which an entity may be exposed, particularly the risk of money laundering and terrorist financing. The main point of the DD analysis is to identify and verify the customer based on account information provided to the company. When conducting DD, it is important to consider factors such as:

- the type of customer – individual or business
- the type of products/services/transactions the customer is using or conducting
- the geographical areas of the customers' operations
- the number of the customers' operations

Identification and verification of the customer is being completed before establishing a business relationship. For higher risk customers it is important to perform further Enhanced Due Diligence (EDD).

***The EDD Compliance Procedure***

Enhanced Due Diligence (EDD) compliance measures include obtaining further information about higher risk customers, including additional evidence of identity as documents provision, checks in an available software, politically exposed persons, individuals in trusts and fiduciary relationships or if a customer comes from a jurisdiction considered to be a high risk.

EDD compliance is also performed in cases when there is a suspicion of money laundering or terrorist financing. The company conducts retrospective due diligence periodically on all existing customers.

***KYC Procedures***

Since global business opportunities demand a sophisticated international customer identification and verification solution, the KYC policy adopted by One Ecosystem includes identifying the user and verifying the identity by examining reliable and independent documents.

Know Your Customer (KYC), Know your Business (KYB) and Know Your Customers' Customer (KYCC) analyses are made to assess the extent to which the customer exposes the organization to a range of risks. KYCC analysis is important, because a company should know who their third-party customers' business dealings are with, their sources of funds and legitimacy, and whether the risks for these third parties may be related to areas such as money laundering, fraud and terrorist financing.

All submitted documents of identity/identification remain confidential. Each user must go through a verification procedure every time his/her identity information is changed, thus confirming that each payment is not anonymous.

***AML/CFT and CDD and KYC/KYB Compliance***

Standard due diligence procedures require customers' identification as well as verification of the customer's identity. In addition, the company ensures gathering of information for the purpose of understanding the customers' intention of the business relationship with the company. All collected data give evidence of the customers' aim and the objectives of the products and services application, to ensure they are not being used for money laundering or for any kind of other criminal activity.

Documentation of identity must be supplemented with additional identification such as a recent utility bill or a bank statement, which is less than 3 months' old and showing the customer's name and address.

***KYC Compliance procedure***

List of documentation the company requests:

1. Proof of Identity documents:

2. Proof of Address documents:

- National ID card - all sides must be
- Utility bills (electricity, gas, water, waste, etc.) scanned in ONE file
- less than 3 months old
- National Passport

- Document issued by a Bank less than 3 months old
- International Passport
- Document issued by Municipality/Government Agency/Tax Authorities - most recent All sides of a valid National ID Card/Passport (identity page and detailed address page uploaded in ONE file)
- Other documents issued by the Government, where the names and the detailed address are shown, for example: NOT expired: Residence permits, Driving licenses, Voter ID cards; Nation-al/ Municipal
- Residential Agencies or Registers
- The following details must be clearly visible: visible:
- Issue date
- Personal photo
- Issuing authority's official letter
- All personal names • Head/logo
- Date of birth • Personal names of the customer
- Document serial number
- Detailed address (Country, City, Zip code)
- Issuing date of the document
- (Postal) Code, street name, street number, etc.)
- Expiry date of the document
- Official sign and stamp of the issuing authority (if available for the respective document)
- Issuing authority

***KYB Compliance Procedure***

KYB procedure includes the assessment of the high-risk activities/ businesses, taking into consideration factors such as:

- The customers' occupation.
- The type of transaction the customer is using or conducting.
- The geographical areas of the customer's operations

Furthermore, collection of some entity documents will be required in order for the company to fulfil the requirements of its KYB (CDD) procedure.

The following documentation may be requested and provided by the business entity on demand (certified copy of the documents with official English translation):

Certified true copy of Business Registration Certificate or corresponding commercial register extract (Certificate of Incorporation), including one for any shareholding companies for the customers' company.

If there are shareholding companies - Business Ownership Chart

(STAKEHOLDERS' ORGANISATION CHART)

***Board resolution***

Identity documents and proof of residency of the stakeholders holding more than 10 per cent of the paid-up share capital and/or key functionalities are required

- Copy of Tax registration certification (if any)
- Photocopy of Business Registration License (if any)
- Fiduciary agreement (if any)
- Annual financial statement at least for the previous year

All the information and documents collected are recorded in a secured database. Any change of the information and/or documents is recorded and kept for 5 years dating from the end of the relationship of the company with the customer.

The Company, under the protection of the safe harbour from liability, may voluntarily receive or otherwise share information with any of the other financial or governmental institution regarding individuals, entities, or other organizations for purposes of identifying and when needed reporting activities, that may involve possible terrorist or money laundering activities.

# g. LEGAL CONSIDERATIONS AND RISKS

**Industry/competition risk** – companies engaged in the high-technology business sphere face significant competition, due to the high-speed development of the technologies and amortization of the currently exploited systems, software etc. Risks can be related to larger client base, name recognition, bad commercial practices, and implementation of practices in violation of the common competition rules.

**Media risk** – as usual, development of technologies is related to increasing media interest, where the sector is tide by high level of confidentiality, commercial and trade secrets. In this case excessive media interest may seriously harm the business, as for example activating other risks as the above described "competition risk" or via negative publications to affect client base etc.

**Regulatory & compliance risk** – because of the "cutting edge" status of the sector, regulations are following the practical implementation of cryptocurrency and blockchain technologies. The rapidly developing sector may suddenly become a subject to statutory and regulatory requirements, which may potentially alter the business.

**Taxes** - Holders of the ONE may be required to pay taxes associated with the transactions contemplated herein. It will be a sole responsibility of holders of ONE to comply with the tax laws of the jurisdictions applicable to them and pay all relevant taxes.

**Force Majeure** - The performances under the One Ecosystem may be interrupted, suspended or delayed due to force majeure circumstances. For the purposes of this Whitepaper, force majeure shall mean extraordinary events and circumstances which could not be prevented and shall include acts of nature, wars, armed conflicts, mass civil disorders, industrial actions, epidemics, pandemics, lockouts, slowdowns, prolonged shortage or other failures of energy supplies or communication service, acts of municipal, state or federal governmental agencies, other circumstances beyond control, which were not in existence at the time of Whitepaper release.

**Disclosure of Information** - Personal information received from holders of the ONE, information about the number of coins owned, the wallet addresses used, and any other relevant information may be disclosed to law enforcement, government officials, and other third parties when required to disclose such information by law, subpoena, or court order. OneEcosystem shall at no time be held responsible for such information disclosure.

**Bitcoin and Ethereum Volatility** - The ONE may be significantly influenced by digital currency market trends and the ONE value may be severely depreciated due to non- ONE related events in the digital currency markets. Cryptocurrencies exchange rate volatility may impact the Company's ability to provide services at the indicated prices. Assumptions with respect to the foregoing involve, among other things, judgments about the future economic, competitive and market conditions and business decisions, most of which are beyond the control of the ONE team and are therefore difficult or impossible to accurately predict.

**Delayed Projects & Competition** - Although the ONE team believes that its assumptions underlying its forward-looking statements are reasonable, any of these may prove to be inaccurate. As a result, assurances that the forward-looking statements contained in this Whitepaper will prove to be accurate cannot be provided.

Considering the significant uncertainties inherent in the forward-looking statements contained herein, the inclusion of such information may not be interpreted as a warranty that the objectives and plans of the OneEcosystem project will be successfully achieved.

Competition may introduce the same or better prediction market solutions and cause loss of market share and eventually failure to deliver the declared business goals.

# h. DISCLAIMER

PLEASE READ THIS SECTION AND THE FOLLOWING SECTIONS ENTITLED "DISCLAIMER OF LIABILITY", "NO REPRESENTATIONS AND WARRANTIES", "REPRESENTATIONS AND WARRANTIES BY YOU", "CAUTIONARY NOTE ON FORWARD-LOOKING STATEMENTS", "MARKET AND INDUSTRY INFORMATION AND NO CONSENT OF OTHER PERSONS", "NO ADVICE", "NO FURTHER INFORMATION OR UPDATE", "RESTRICTIONS ON DISTRIBUTION AND DISSEMINATION", "NO OFFER OF SECURITIES OR REGISTRATION" AND "RISKS AND UNCERTAINTIES" CAREFULLY.

IF YOU ARE IN ANY DOUBT AS TO THE ACTION YOU SHOULD TAKE, YOU SHOULD CONSULT YOUR LEGAL, FINANCIAL, TAX OR OTHER PROFESSIONAL ADVISOR(S).

The ONE is not intended to constitute securities or financial instruments in any jurisdiction.
This Whitepaper does not constitute a prospectus or offer document of any sort and is not intended to constitute an offer of securities or a solicitation for investment in securities in any jurisdiction. This Whitepaper does not constitute or form part of any opinion on any advice to sell, or any solicitation of any offer by the Distributor (the free "Token Provider") to purchase any ONE because ONE is not for sale nor shall it or any part of it nor the fact of its presentation form the basis of, or be relied upon in connection with, any contract or investment decision.

No regulatory authority has examined or approved of any of the information set out in this White Paper. No such action has been or will be taken under the laws, regulatory requirements, or rules of any jurisdiction. The publication, distribution or dissemination of this Whitepaper does not imply that the applicable laws, regulatory requirements, or rules have been complied with.

## DISCLAIMER OF LIABILITY

To the maximum extent permitted by the applicable laws, regulations and rules, or any entity or person being a part of the ONE Ecosystem shall not be liable for:
any indirect, special, incidental, consequential or other losses of any kind, in tort, contract or otherwise (including but not limited to loss of revenue, income or profits, and loss of use or data), arising out of or in connection with any acceptance of or reliance on this Whitepaper or any part thereof by the participants.

## NO REPRESENTATIONS AND WARRANTIES

The Token Provider and any entity or person being a part of the One Ecosystem does not make or purport to make, and hereby disclaims, any representation, warranty or undertaking in any form whatsoever to any entity or person, including any representation, warranty or undertaking in relation to the truth, accuracy and completeness of any of the information set out in this Whitepaper.

## REPRESENTATIONS AND WARRANTIES BY YOU

By accessing and/or accepting possession of any information in this Whitepaper or such part thereof (as the case may be), participants represent and warrant to the Token Provider or any entity or person being a part of the One Ecosystem that they agree on following:

(a) participants agree and acknowledge that the ONE does not constitute securities in any form in any jurisdiction.

(b) participants agree and acknowledge that this Whitepaper does not constitute a prospectus or offer document of any sort and is not intended to constitute an offer of securities in any jurisdiction or a solicitation for investment in securities and participants are not bound to enter into any contract or binding legal commitment and no cryptocurrency or other form of payment have to be accepted on the basis of this Whitepaper.

(c) participants agree and acknowledge that no regulatory authority has examined or approved of the information set out in this Whitepaper, no action has been or will be taken under the laws, regulatory requirements or rules of any jurisdiction and the publication, distribution or dissemination of this Whitepaper to participants does not imply that the applicable laws, regulatory requirements or rules have been complied with.

(d) the distribution or dissemination of this Whitepaper, any part thereof or any copy thereof, or acceptance of the same is not prohibited or restricted by the applicable laws, regulations or rules in participants jurisdiction, and where any restrictions in relation to possession are applicable, participants have observed and complied with all such restrictions at their own expense and without liability to any entity or person being a part of the ONE Ecosystem.

(e) participants agree and acknowledge that in the case where participants wish to mine ONEs, the ONE is not to be construed, interpreted, classified, or treated as:

- any kind of currency other than cryptocurrency
- debentures, stocks, or shares issued by any person or entity
- rights, options, or derivatives in respect of such debentures, stocks or shares.

- units in a collective investment system.
- units in a business trust.
- derivatives of units in a business trust; or
- any other security or class of securities.

participants have a basic degree of understanding of the operation, functionality, usage, storage, transmission mechanisms and other material characteristics of cryptocurrencies,

block-chain-based software systems, cryptocurrency wallets or other related token storage mechanisms, blockchain technology and smart contract technology.

participants are fully aware and understand that in the case where you wish to mine ONEs, there are risks associated with the One Ecosystem and their respective services, business and operations.

participants agree and acknowledge that neither One Ecosystem nor any entity or person being a part of the One Ecosystem is liable for any indirect, special, incidental, consequential or other losses of any kind, in tort, contract or otherwise (including but not limited to loss of revenue, income or profits, and loss of use or data), arising out of or in connection with any acceptance of or reliance on this Whitepaper or any part thereof by you; and rights under a contract for differences or under any other contract the purpose or pretended purpose of which is to secure a profit or avoid a loss;

All the above representations and warranties are true, complete, accurate and non- misleading from the time of participants access to and/or acceptance of possession this Whitepaper or such part thereof (as the case may be).

**CAUTIONARY NOTE ON FORWARD-LOOKING STATEMENTS**

All statements contained in this Whitepaper, statements made in press releases or in any place accessible by the public and oral statements that may be made by an entity or person being a part of the One Ecosystem, including their respective directors, executive officers or employees acting on behalf of them that are not statements of historical fact, constitute "forward-looking statements". Some of these statements can be identified by forward-looking terms such as "aim", "target", "anticipate", "believe", "could", "estimate", "expect", "if", "intend", "may", "plan", "possible", "probable", "project", "should", "would", "will" or other similar terms. However, these terms are not the exclusive means of identifying forward-looking statements. All statements regarding financial position, business strategies, plans and prospects and the future prospects of the industry which One Ecosystem is in are forward-looking statements. These forward-looking statements, including but not limited to statements about revenue and profitability, prospects, future plans, other expected industry trends and other matters discussed in this Whitepaper are matters that are not historical facts, but only predictions. These forward-looking statements involve known and unknown risks, uncertainties and other factors that may cause the actual future results, performance, or achievements to be materially different from any future results, performance or achievements expected, expressed or implied by such forward-looking statements.

These factors include, amongst others:

(a) changes in political, social, economic, and stock or cryptocurrency market conditions, and the regulatory environment in the countries in which a part of the ONE Ecosystem conducts its respective businesses and operations.

(b) the risk that being a part of the One Ecosystem may be unable or execute or implement their

respective business strategies and future plans.

(c) changes in interest rates and exchange rates of fiat currencies and cryptocurrencies.

(d) changes in the anticipated growth strategies and expected internal growth of being a part of the One Ecosystem.

(e) changes in the availability and fees of being a part of the One Ecosystem in connection with t heir respective businesses and operations.

(f) changes in the availability and salaries of employees who are required being a part of the One Ecosystem to operate their respective businesses and operations.

(g) changes in preferences of customers being a part of the One Ecosystem.

(h) changes in competitive conditions under which being a part of the One Ecosystem operate, and the ability being a part of the One Ecosystem to compete under such conditions.

(i) changes in the future capital needs of being a part of the One Ecosystem and the availability of financing and capital to fund such needed

(j) war or acts of international or domestic terrorism.

(k) occurrences of catastrophic events, natural disasters and acts of God that affect the businesses and/or operations being a part of the One Ecosystem.

(l) other factors beyond the control being a part of the One Ecosystem; and
All forward-looking statements are expressly qualified in their entirety by such factors.

Given that risks and uncertainties that may cause the actual future results, performance, or achievements to be materially different from that expected, expressed or implied by the forward-looking statements in this Whitepaper, undue reliance must not be placed on these statements. These forward-looking statements are applicable only as of the date of this Whitepaper.

The actual results, performance or achievements may differ materially from those anticipated in these forward- looking statements. Nothing contained in this Whitepaper is or may be relied upon as a promise, representation or undertaking as to the future performance or policies. Further, being a part of the One Ecosystem disclaim any responsibility to update any of those forward- looking statements or publicly announce any revisions to those for-ward-looking statements to reflect future developments, events or circumstances, even if new in-formation becomes available or other events occur in the future.

**MARKET AND INDUSTRY INFORMATION AND NO CONSENT OF OTHER PERSONS**

This Whitepaper includes market and industry information and forecasts that have been obtained from internal surveys, reports, and studies, where appropriate, as well as market research, publicly available information and industry publications. Such surveys, reports, studies, market research, publicly available information and publications generally state that the information that they contain has been obtained from sources believed to be reliable, but there can be no assurance as to the accuracy or completeness of such included information.

Save for being a part of the One Ecosystem and their respective directors, executive officers and employees, no person has provided his or her consent to the inclusion of his or her name and/or other

information attributed or perceived to be attributed to such person in connection therewith in this Whitepaper and no representation, warranty or undertaking is or purported to be provided as to the accuracy or completeness of such information by such person and such persons shall not be obliged to provide any updates on the same.

While being a part of the One Ecosystem have taken reasonable actions to ensure that the information is extracted accurately and in its proper context, being a part of the One Ecosystem have not conducted any independent review of the information extracted from third party sources, verified the accuracy or completeness of such information, or ascertained the underlying economic assumptions relied upon therein.

An IMA and/or any other entity being a part of the ONE Ecosystem, nor their respective directors, executive officers and employees acting on their behalf makes any representation or warranty as to the accuracy or completeness of such information and shall not be obliged to provide any updates on the same.

**RESTRICTIONS ON DISTRIBUTION AND DISSEMINATION**

The distribution or dissemination of this Whitepaper or any part thereof may be prohibited or restricted by the laws, regulatory requirements, and rules of any jurisdiction. In the case where any restriction applies, you are to inform yourself about, and to observe, any restrictions which are applicable to your possession of this Whitepaper or such part thereof (as the case may be) at your own expense and without liability to being a part of the One Ecosystem. Persons to whom a copy of this Whitepaper has been distributed or disseminated, provided access to or who otherwise have the Whitepaper in their possession shall not circulate it to any other persons, reproduce or otherwise distribute this Whitepaper or any information contained herein for any purpose whatsoever nor permit or cause the same to occur.

**NO OFFER OF SECURITIES OR REGISTRATION**

This Whitepaper does not constitute a prospectus or offer document of any sort and is not intended to constitute an offer of securities or a solicitation for investment in securities in any jurisdiction.

No person is bound to enter into any contract or binding legal commitment and no cryptocurrency or other form of payment is to be accepted on the basis of this Whitepaper. Any agreement in relation to mining of ONE (as referred to in this Whitepaper) is to be governed by only the T&Cs of such agreement and no other document. In the event of any inconsistencies between the T&Cs and this Whitepaper, the former shall prevail.

No regulatory authority has examined or approved of any of the information set out in this Whitepaper. No such action has been or will be taken under the laws, regulatory requirements, or rules of any jurisdiction. The publication, distribution or dissemination of this Whitepaper does not imply that the applicable laws, regulatory requirements, or rules have been complied with.

ONE ECOSYSTEM
registered office address:
Oberneuhofstrasse 8, 6340 Baar
Switzerland