ESSH Whitepaper

DRAFT

10.07.2019

Version 1.0.1

TABLE OF CONTENTS

NOTICE OF NON-LIABILITY AND DISCLAIMER

The purpose of the ESSH Architecture document is to outline general information only, its purpose is not to sell any product, asset or item. The information below may not be inclusive and does not imply any elements of a contractual relationship.

ESSH makes no assurances of any kind with respect to the information contained herein and assumes no liability for any direct or indirect damages resulting from the information provided within this document or from the use of any product or service described within. There are no assurances and/or completeness and no warranties of the information provided in this document.

The ESSH Architecture document contains information from third-party sources and the ESSH team has not independently verified the accuracy or completion of this information. Please be informed that circumstances are liable to change and therefore this document may become outdated as a result. The ESSH team has no obligation to update and/or correct this paper. All statements included in this paper, or statements made in press releases, promotional articles, online resources or any other statements made in the public sphere by any ESSH representatives may constitute only subjective, forward-looking statements (including intent, belief, present or future expectations or predictions regarding market conditions, business strategy or risk practices).

All information provided on the ESSH website, architecture document, roadmap or any related documents are to be treated as suggestions and should not be considered as mandatory to be executed unless explicitly stated in the corresponding terms and conditions.

The ESSH Architecture document may be translated into other languages in order to clarify certain terms, conditions or any statements made on the original document. For reference, the English language version of this paper must be considered as the prevailing document.

The use of any trademark, product, platform or company name does not imply direct or indirect affiliation with the third party. Any references in this document to specific names were made for illustrative purposes only.

It is forbidden to copy, distribute, or reproduce the ESSH Architecture document without prior agreement and approval from ESSH.

ABSTRACT

Over the past few decades, technological innovation has disrupted entire industries in significant ways. Just like the proliferation of the Internet at the turn of the century, the advent of blockchain technology opened up a universe of opportunity and a race for innovating solutions. Since 2009, the use of blockchain and digital currencies have gradually but significantly expanded in terms of applicability and use-cases. From financial resources, data storage, identity management to employee contracts, there is no industry that has not been touched by its potential. Since the inception of ESSH, the team has remained focused on solving certain prevalent problems that have inhibited the proliferation of blockchain technology. The most paramount being inaccessibility and inefficiency. Although blockchain technology offers immutability, superior privacy, cheap administration costs and transaction fees, blockchains need to be more simple to use, efficient and less fragmented in order for the world to choose blockchain over its preceding technologies.

The aim of ESSH is to build a global decentralized cryptosystem which offers the best, and latest, blockchain innovations in one convenient environment. It offers fast, stable, secure and private transactions over its network and is designed to remain up to date with incoming technology. Every feature implemented ESSH One blockchain is tailored to create a single ecosystem capable of implementing the latest and most efficient technological practices.

The ESSH engineers have already begun implementing the best features currently available, these include a masternode network, staking technology, token swaps, dApp capability, multi-wallets, and identity management. ESSH remains true to the fundamental principles of decentralization: transparency, open-source code, and interoperability.

An enormous effort has been made to incorporate these features into a single ESSHI unit. This work has enabled the ESSH team to offer a unique product entirely differentiated from any competitors.

These are the primary features of the ESSH blockchain network:

- Security and scalability.
- Decentralized system powered by Masternodes.
- Atomic swaps.
- ESSH superwallet.
- Anonymous Peer-To-Peer currency.
- Growing large community on multiple social resources.
- Higher network security based on PoS (Proof of Stake) consensus instead of PoW (Proof of Work).
- The proactive, experienced and highly skilled dev team.

ESSH'S Features

Security

Scalability

Masternodes decentralized system

Atomic swaps

Essentia superwallet

Anonymous Peer-To-Peer currency

Large and growing community

Higher network security based on PoS consensus

Proactive and highly skilled development team

1 INTRODUCTION

1.1 PREFACE

The big bang of Blockchain technology began with the publication of the Bitcoin Whitepaper in 2008. It proffered decentralization as a system for creating, tracking and storing digital currency. Bitcoin gave the world a fundamental structure, including its groundbreaking consensus model, which enabled the retention of distributed transaction ledgers. The major achievement of Bitcoin was the introduction of a permission-free, open-source computing model with the existence of a Byzantine model of failure. In practice, this implies that nodes could achieve transaction consensus even in the presence of malicious actors. In order to corrupt the blockchain, it would require over 50 percent of the computing power in the network to manipulate past transactions. Although Bitcoin has since been integrated into many applications, its primary purpose remains financial, in other words, a cryptocurrency with a secure means of exchanging and storing value.

Naturally, the idea of immutable records leads to the realization that this technology could be adapted to a vast range of sectors beyond finance. Ethereum was the first to actualize a general-purpose blockchain which allowed programmers to adopt the blockchain to specific user needs. This was achieved through a groundbreaking new innovation; smart contracts. It is worth mentioning that Ethereum-like projects are Turing-complete, meaning they allow problematic forecasting through computational modeling. This allowed blockchain technology to be adapted for different purposes, and not solely for exchanging value. Bitcoin and Ethereum are community-attracted blockchains that permit anyone to engage and interact. The pioneering technology and the masterminds behind them have inspired us to proceed with further innovation and to push the boundaries of the technology to offer new, viable and valuable solutions for consumers, organizations, and businesses alike.

1.2 CHALLENGES FACING BLOCKCHAIN

Three main challenges impeding the growth of blockchain:

Decentralization

Since 2009, more than 900 public blockchain platforms have been created. Unlike centralized software systems, decentralized programs are much cheaper to run. This has meant small companies and individuals could utilize technology and make advancements in the development of the industry. Despite its numerous benefits, decentralized networks are not perfect, and they face certain challenges that impede it from competing with existing technologies. For example, a large number of mining pools were formed to supply PoW (Proof of Work) based blockchain networks, which makes it unfair for small miners to collect a proportional block reward. As well as the problem of differentiation in mining resources, PoW involves the use of a large amount of electricity. Finally, decentralized systems that are based on legacy solutions are vulnerable to hacker attacks.

Security

As its name suggests, blockchains consist of a chain of "blocks" tied together digitally. The design was authored in order to ensure security and achieve immutability for a ledger that not one person or organization owned or operated - a distributed ledger. The realization that blockchain could securely decentralize data was revolutionary, however, just like any burgeoning system, it was not immune to the threat of malicious actors and general vulnerabilities.

Nonetheless, sufficient security was realized through the peer-to-peer network requiring continuous updates and synchronization. In practice, blockchains that utilize PoW as a consensus algorithm would require at least 51% hash power of the entire network to conduct a double-spend attack in order to manipulate any transaction. In general, the success of such attacks depends on the level of decentralization. More decentralization results in a decreased risk as computing power is more evenly distributed.

Scalability

One of the most common practices to achieve scaling (capability to deal with the growth of the network) is to split the chain into smaller pieces, this is called sharding. This practice lowers computational demand and hash power per block, however, smaller chains are less protected due to their size. Although this method achieves scalability to a degree, it compromises security and should therefore not be a long-term solution. Another negative side effect of sharding is that multiple blockchains can limit cross-chain transactions. This has further fragmented the industry and increased costs for trading to account for inefficiency.

Blockchain Challenges

2 ESSH BLOCKCHAIN OVERVIEW

ESSH is an open-source peer-to-peer network geared to provide access to the best and latest features blockchain has to offer. The network is strengthened by Proof-of-Stake (PoS) as its consensus algorithm and masternodes to ensure a maximum level of security for its functionality.

PoS (Proof of Stake) is a type of protocol which achieves the verification of transactions using just a fraction of the electricity required for PoW. PoS provides the framework to create ESSH coins, which is then rewarded to the verificators (usually referred to as “stakers”) as a reward for validating transactions. The first 100 blocks of the ESSH
blockchain are created via PoW, with the next blocks - PoS.

The blockchain keeps track of a set of validators (masternodes), and anyone who holds the blockchain's base cryptocurrency can become one by sending a special type of transaction that locks a minimum amount of ESSH as a deposit. The process of creating and agreeing to new blocks is then done through a consensus algorithm that all current validators can participate in.

ESSH has implemented the following functionality:

- The ESSH blockchain with 2 types of
- masternodes. The ESSH Wallet.
- Atomic swaps.

A schematic example of the blockchain's structure and how to accESSH it:

2.1 CORPORATE ORGANIZATION

ESSH is an independent, community-oriented organization inspired by blockchain technology and the demand to raise utility. It's structure consists of the following entities:

Furthermore, the ESSH project has gathered a team of experienced professionals who have created complex software solutions and work with a desire to share their knowledge through open-source product releases.

2.2 PRIVACY AND CONFIDENTIALITY

We often use the terms confidentiality and privacy interchangeably, but from a legal perspective, they mean noticeably different things. Confidentiality in regards to blockchain relates to the sensitive information shared between network participants by consent. Privacy, on the other hand, refers to the freedom from and avoidance of third-party intrusion. ESSH provides an optimal balance between confidentiality and privacy. In practice,

users can create an ESSH account as well as ESSH wallets without entering personal-related data or verifying their identity. Despite the fact that our **block explorer** is open, so anyone can view block details, there is no personal information. What is more, the ESSH wallet generates anonymous bitcoin-like addresses, but there may be transaction amount restrictions depending on its legality by country or territory.

ESSH cannot access or manage any private data which belongs entirely to the user. With the help of advanced cryptographic methods, ESSH offers seed, mnemonic phrase or Keystore file access for storage and data management.

2.3 TARGET AUDIENCE

Average users

Cryptocurrencies have now become a global phenomenon known to most mainstream audiences. Despite often negative coverage in mass media, there is little doubt that blockchain technology has revolutionary potential. Global corporations and institutions in finance, logistics, customs & immigration and healthcare are already investing millions into research and development. ESSH has its own vision on how the future of the blockchain landscape will look like, and it begins by providing access to anyone with a computer, smartphone or tablet device.

ESSH, as an all-encompassing resource, will be able to serve as a starting point for the average user, business or organization looking to explore and take advantage of the benefits of blockchain.

Masternodes users

In the current blockchain climate, there are certain options available for maintaining blockchain networks. The processes of mining and masternodes have proven the most common methods due to their respective advantages. Despite the fact that mining is a well-known technology, it foresees bigger computational resources to pass the entry barrier. Masternodes can attract new users because of the low-cost maintenance, overall simplicity, and regular profits.

Trend followers

Young and proactive 18 - 35-year-olds are the largest proponents and adopters of cryptocurrencies. ESSH understands the importance of this audience and is ensuring maximum resources are allocated in order to reach, engage and communicate the latest product updates, developments and new opportunities.

Influencers

ESSH has established partnerships with key influencers, thought leaders, blockchain gurus, and various other professionals to inform the industry of our goal and mission.

As the general public lost confidence in the value of crypto and the community as a whole - partly due to genuine concerns regarding the proliferation of fake platforms - the only products to survive in the future will have proven value within their respective fields. ESSH understands that the only way to gain public confidence is to create a demonstrably solid and reliable product.

2.4 ESSH ECONOMICS

The fundamental economic infrastructure of ESSH is simply based on a cryptocurrency with supply restriction. ESSH utilizes this approach to increase the value of the ESSH coin and, at the same time, to bestow anti-inflationary properties for future distribution. The primary purpose is to foster liquidity and ensure the stability of growth. This increases overall coin value and therefore overall liquidity.

2.5 SUPPLYING COINS

As ESSH imposes restrictions on the supply of new coins, there is also a limit on the number of coins generated per year. The conditions of the ESSH coin limit is then met when the limited number of coins is reached. Please see the “Coin Specifications” section for these limits. To make things clear, ESSH's coin supply philosophy is comparable to that of a static currency in which the money supply is auto-adjusted in reaction to financial pressures in the market. This assumes that no one will produce additional coins thereby reducing overall liquidity.

2.6 BITCOIN-LIKE CURRENCIES ROOTS

As the very first cryptocurrency, Bitcoin established the general blockchain philosophy. Because Bitcoin is decentralized, it does not depend on a central source of electricity to maintain, instead, it works on a node network with the network itself verifying transactions.

Bitcoin uses the processing power of the network's mining computers to maintain the integrity of the system. Transactions are registered in pieces of information, each of which is called a block. The overall system, organized as a blockchain, counts on the mining computers' processing energy to solve a cryptographic puzzle by defining an arbitrary amount to hash with. This mining dependence is recognized as a scheme of Proof of Work (PoW). These cryptographic secrets are increasing in complexity as the network expands, becoming more difficult to resolve which requires more energy. In contrast to Bitcoin, ESSH does not rely on the PoW method. A critical problem with Proof of Work has proven to be its extreme sensitivity to mining pools — computer groups working together to fix block hashes and share the reward to stay competitive. This strategy contributes to the strengthening of mining pools and weakening individual miners. Unlike Bitcoin, ESSH does not rely on mining to run the network. The ESSH blockchain utilizes masternodes as network suppliers instead. This was done to reduce the required resources and capabilities to participate in the network, thereby increasing accessibility.

2.7 DEVELOPMENT PRACTICES

ESSH is a decentralized network maintained by a well-organized, proactive and committed team. ESSH's initiative expands beyond the ESSH Core wallet, with such additional software including the ESSH Android wallet, iOS wallet Most of the development and releases of ESSH are maintained via GitHub.

2.8 PROOF OF STAKE CONSENSUS

The ESSH blockchain network operates on a Proof of Stake consensus algorithm. Originally, it was based on “coinage” - a time measure for the Unspent Transaction Output. The way coinage differs from PoW (Proof of Work) is fairly simple; PoS is not based on rewarding miners, but rather rewarding anyone willing to participate in running the network. It is worth noting that since its inception the PoS consensus has undergone several crucial updates to fix security issues and implement new features. Later versions of PoS include the protection against malicious nodes that could make an attempt to manipulate the blockchain. PoS is fundamental to ESSH’s cryptocurrency and it helps to sustain a valid, viable and easy to use blockchain network. Successful staking requires at least 100 confirmations to get rewards, and in turn, they can only be exchanged after confirming 101 blocks. This approach increases protection against malicious staking and control over the network. It was discovered that a successful attack would require more than 70% of the coins on stake for a 51% chance to double-spend or invalidate a single block. Technically, achieving these numbers is nearly impossible. Please see the "staking" section for more details.

ESSH is committed to supporting eco-friendly practices in the cryptocurrency industry. In contradiction to Bitcoin-like coins that consume high amounts of electricity due to their "Proof of Work" consensus, ESSH has built the blockchain on "Proof of Stake" to counterbalance the demand for computational power. In order to understand the scale of the problem, according to BBC research, the amount of overall electricity consumed by mining warehouses is near the same amount used by a medium-sized country. ESSHlly, the PoS mechanism achieves the same task as mining, however, it requires less than 1% of the power. In general, anyone interested in supporting the network can set up a Masternode without the need for unattainable or costly resources.

2.9 A QUESTION OF AVAILABILITY

Since ESSH offers a masternode-based network along with with software solutions a question of high service availability becomes determinative. Since ESSH is based only on approved providers and own capacities, any online operations can be stopped due to any hacking attempts or other unforeseen malignant circumstances. ESSH focuses on the problem of high availability to guarantee that services can continue to function at all times for both customers and companies regardless of any external risk.

2.10 VISION AND STRATEGY

The First level: Masternodes network

As mentioned previously, the masternode network lowers the barrier of entry for anyone interested in becoming a network participant. Due to this, ESSH expects an increase in demand, therefore a series of extensive guidelines have been released to assist in establishing and maintaining masternodes. Please refer to the "Masternode requirements" section for more details. In addition, those who are involved help to promote network growth and also obtain a passive income.

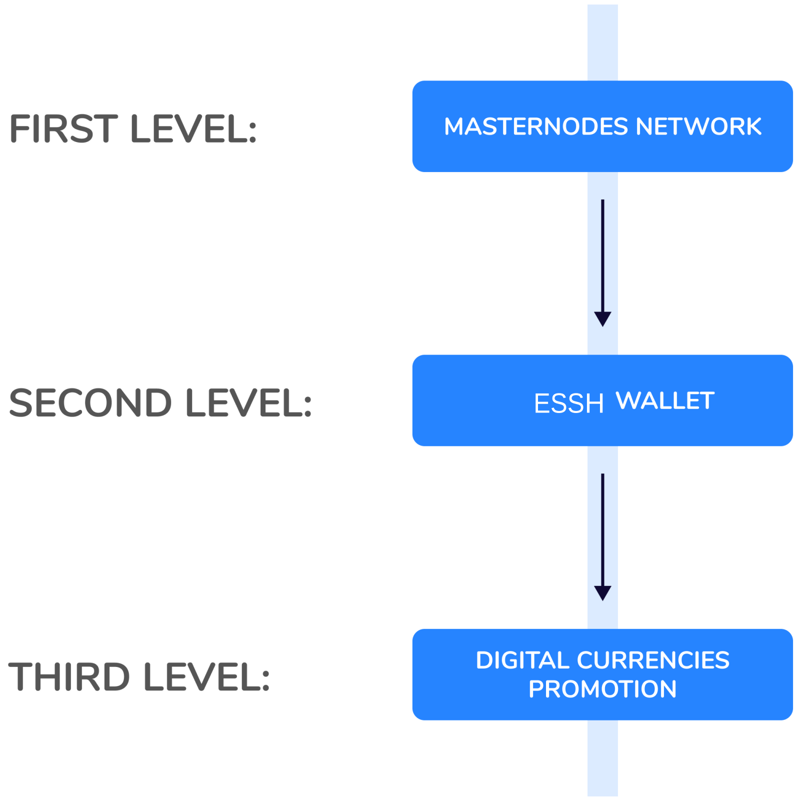
Second level: ESSH Wallet

To build a high-quality software product there are a few fundamental features required for success. Some of these include device-independence, performance efficiency, usability, security, and functional suitability. These principles were identified as determinative during ESSH wallet development. The wallet solution is tailored to simplify user experience and remove the inefficiency and security risk of multiple crypto wallets. It is important to remember that ordinary users do not know-how, time or ability to safely navigate the current blockchain and crypto landscape. The underlying value of the ESSH wallet is based on its user-friendly interface, deep understanding of User Experience (UX) and functionality which can be utilized by businesses, organizations and ordinary users alike. Our team believes that these doctrines can be applied to the ESSH Wallet in full.

Third level: Promoting Digital currencies

As revealed by CoinMarketCap, there are now more than 1500 cryptocurrencies in existence. There can be no doubt that in the near future digital currencies will work alongside contemporary monetary systems on equal terms. The blockchain community must, therefore, spend all efforts constructively to distinguish the best practices and to learn from previous lessons. ESSH is allocating maximum resources to determine and implement the most valuable solutions for future products. To engage new users in the crypto market through opportunities and blockchain potential remains an important strategic goal.

Vision and Strategy



2.11 FUTURE PLANS AND INTENTIONS

Masternode One-Click install

An automated deploy of masternodes will significantly improve user experience by offering a single-click masternode installation feature from the ESSH desktop app. The team is currently working on developing this feature which will be included in an update in the near future.

MultiSig

Multisig stands for “Multisignature” and means that multiple keys are necessary to conduct any transaction. This was primarily designed to guarantee robust security for users, companies, and all ESSH holders.

Distributed Access

The move to Distributed Access holds a variety of advantages. This enables network progression through the decentralization and virtualization of headend functions. In the case of ESSH, distributed access provides additional security by creating a special authorization layer among ESSH software solutions.

ESSH Storage

ESSH Storage is a form of cloud storage driven by the ESS cryptocurrency. The aim is to provide absolute control and sole access for users through the decentralized capabilities of blockchains. Using ESSH storage allows users to:

- The ESSH blockchain with 2 types of masternodes.
- The ESSH Wallet.
- Atomic swaps.

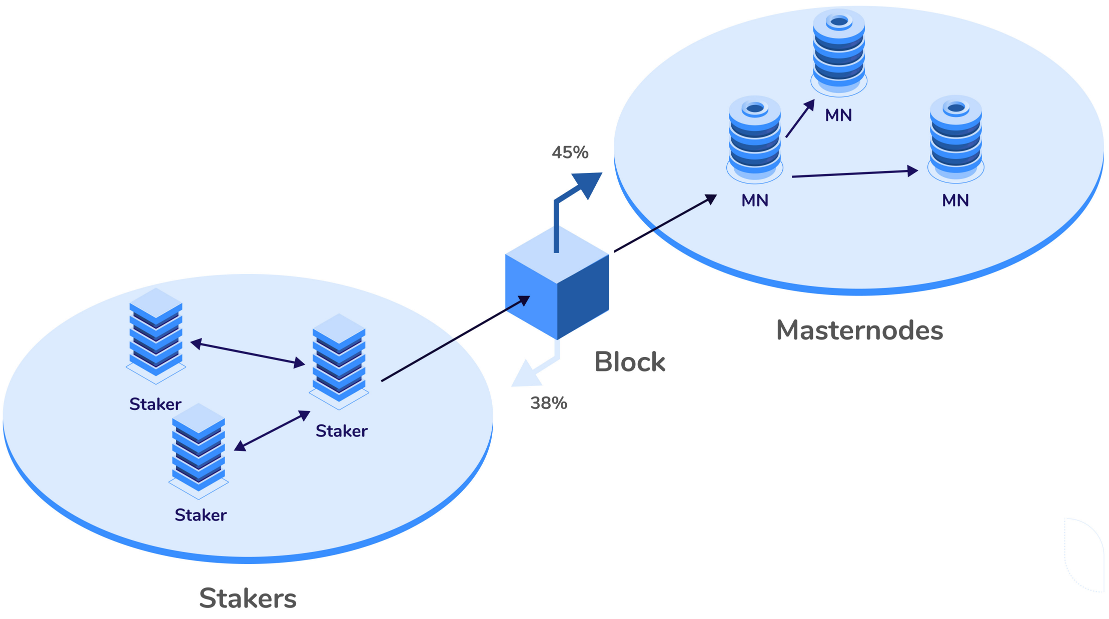
3 STAKING

"Staking" refers to the act of providing the network with the required computer resources for node "selection". This is based on delimited competition and done in order to generate the next block on the chain. When it comes to ESSH, these limits are defined by the balance (UTXOs - Unspent Transaction Output) staked in an ESSH wallet. In practice, this means that each staking node competes to generate a valid block; similar to PoW. It is worth mentioning that nodes are limited in their trial attempts and that the problem with achieving a valid block is inversely proportional to the amount staked. In simple terms, a higher balance implies a greater chance of meeting the difficult requirements to validate the block and obtain the reward.

In general, the origin of staking comes from the idea of "coinage", meaning it shows how long a UTXO has not been spent. Consequently, instead of rewarding only miners, PoS rewards anyone willing to participate in running the network.

An important facet to consider in PoS is the vulnerability of a long-range or history attack where early blocks are overwritten, which in turn compromises the blockchain. In order to prevent this from happening, the ESSH blockchain offers ‘checkpoints’, which are blockchain markers set over certain intervals to restrict any alteration/forking prior to the checkpoints set. They have proven to be crucial in saving valid chains and the protection against long-range attacks.

Reward distribution



Any PoS attack would greatly depreciate the attacker's assets when discovered, and at the same time the PoW attack would incur costs in electricity. In addition, ESSH staking can be easily decentralized among its consumers and therefore cannot be traced by the use of electricity, while mining is generally centralized by mining cartels, focused on cheap electricity areas and traceable by high steady demand for energy.

ESSH stakers are divided into 2 groups - 'verificators' and 'block producers'. In the network, a block is produced by block producer and then broadcast to every node (1). Then, the new block is validated by one of the verificators (2). If a block is valid, the reward from it goes to the Verificator (45% of block reward) and to the block producer that produced this block (38%). The rest of the block reward goes for network support.

4 TRANSACTIONS

Verificators validate transactions and are awarded coins. During the staking process, the wallet checks transactions to ensure that the coins were actually sent by their owner. The block is accepted by the network if most of the online wallets accept the transaction as valid. Every minute one of these wallets are rewarded based on the coins stored within. The verificator must be online and wallet must be synced to peers.

Please take into consideration that staking requires a node funded with **10k ESSH**, have at least 3 connected peers, the wallet must be unlocked and the "age" of coins must be more than 1 hour (mintable coins).

4.1 TRANSACTION FEE

The transaction fee is the amount charged to users when processing crypto transactions. The fee is paid to ensure cryptocurrency transactions appear in a timely manner. In general, the fee should be considered as a tool which directly influences the speed of transaction. The lower the fee the lower the transaction's priority and as a result the longer time it takes to be processed.

Calculating fee

The ESSH transaction fee has its own calculation rules and it depends on the transaction amount, size and global network modifiers. Please refer to the formula below to calculate the fee for your transaction:

$$Fee = (TVM * TV) * TS / TSM$$

Where:

- *TVM* - Transaction value multiplier (in accordance with global value)
- *TSM* - Transaction size multiplier (in accordance with global value)
- *TS* - Transaction size (in bytes)
- *TV* - Transaction value (in ESSH)

Global fee modifiers

These are the default global fee modifiers. By changing these values we can set a global fee for the entire network. By implementing global modifiers we can optimize fee calculation.

- *TSM* - 300
- *TVM* - 0.0001

Minimum & Maximum fees

Currently, the maximum fee value for the network is set as *MaxFee = 100 ESSH*. The default minimum is *MinFee = 10000 satoshis per KB*.

Deduction of fee

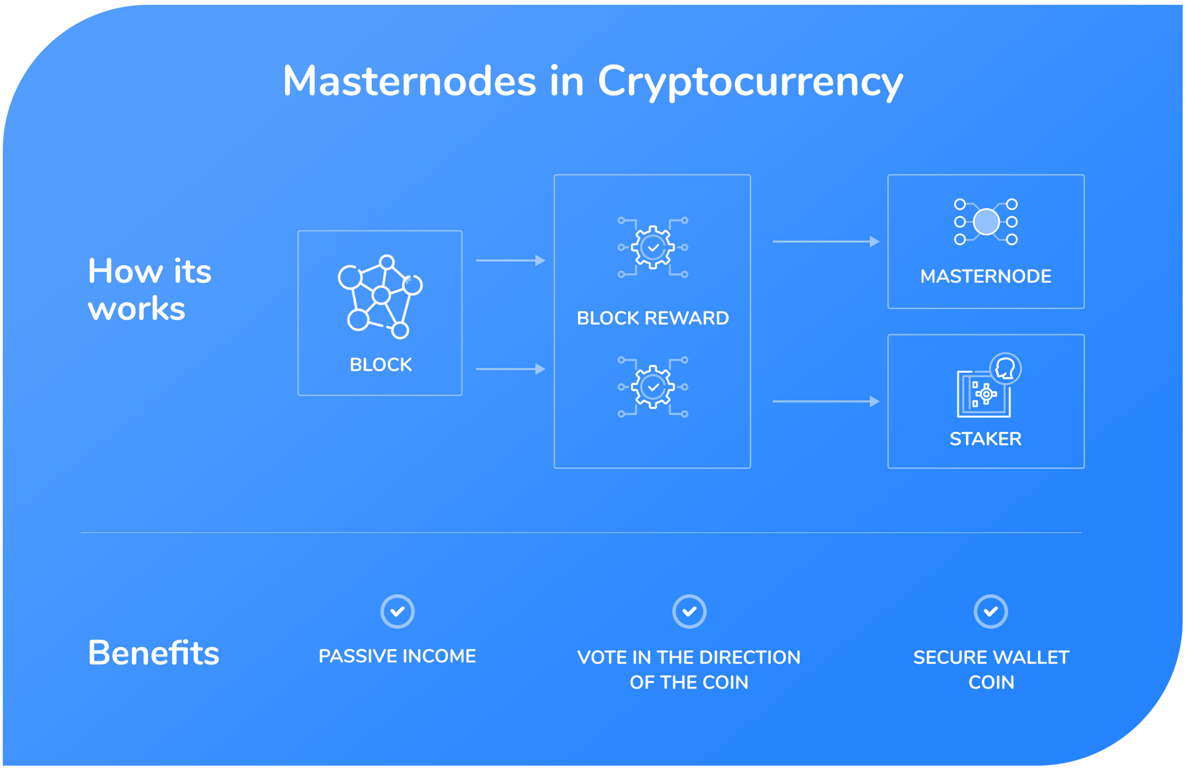
The fee is withdrawn along with transaction and calculated as *Fee = TxIn - TxOut*. Eventually, the fee is paid in the spendable outputs of the original transaction.

5 NODE HIERARCHY

5.1 MASTERNODES OVERVIEW

A Masternode is a full node that processes transactions within a blockchain and in return receives a reward from the blocks created. They operate under a system that is collateral-based to guarantee the provision of services that are genuine. Masternodes act as the ESSH blockchain network backbone or as a bonded validation system. A masternode is considered a crucial part of the decentralized blockchain network that is responsible for the overall system operating. Technically speaking, masternodes are a series of virtual private and secured servers that are running 24 hours a day to ensure blockchain network workability by processing transactions and signing blocks.

The main goal of the masternodes is to support the blockchain network through block creation and signing, reducing transaction time, and providing a truly decentralized environment. It is worth mentioning that masternode owners are paid a reward as motivation and gratitude for their investment, server configuration and support of the ongoing blockchain operations.



These are several reasons why the masternode configuration is useful:

- Higher trust factor and reputation value within the network community.
- Less time to wait for the transaction to be processed.
- Better transaction privacy.
- Masternodes directly contribute to network growth, security, and overall appeal.

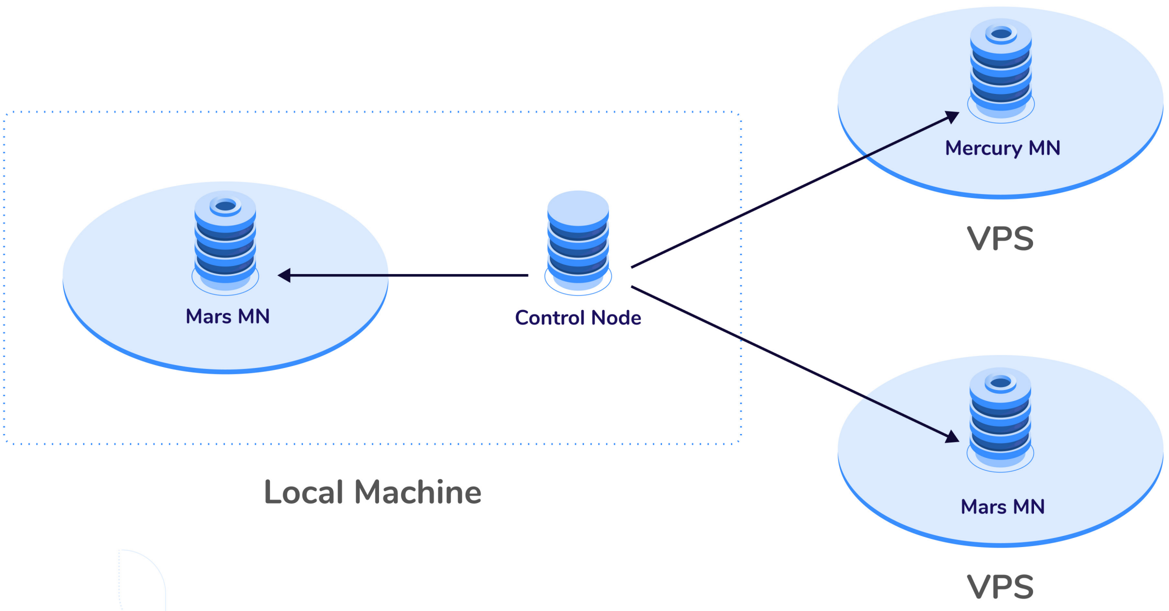
There are two types of Masternodes: **Mercury** and **Mars**. Each node requires a minimum amount of coins as collateral. Mercury masternodes require 100,000 ESSHand Mars masternodes require 300,000 ESSH. This collateral can be spent at any time, doing so removes the associated masternode from the network and prevents Sybil attacks.

The staker receives a reward depending on the tier of Masternode: the more coins allocated in masternode - the larger the reward.

All Masternodes are included in the global list in the blockchain. Masternode payments are specified by a decentralized selected algorithm based on the Masternode type.

All Masternodes in the global list vote for the new winner, who will be paid after the new block has been added to the network.

Masternode connection to the control node



5.2 REQUIREMENTS

The primary requirements for running a Masternode:

Control node -> Masternode organizational structure

To set up a control node (local node) it is necessary to allocate the minimum amount of coins required and store the base info of the masternode in the config. The control node can distribute several amounts of masternodes (not only one). Stakers can allocate only one Masternode at a time. Also, the Control node initiates the Masternode.

A masternode (remote node) can be run locally on a virtual machine or on a VPS (for best results). It should be online 24/7. **VPS** stands for Virtual Private Server and can be treated as a virtual machine provided by an Internet hosting service. It uses virtualization technology to grant the user with the dedicated resources of a server. VPS technology offers advanced management and control over the cloud-based machines and server configurations.

Due to the technical nature of any VPS, they offer the most advanced solution for running a masternode - particularly because of their static IP addresses and uninterrupted uptime. Please note that a dynamic IP will not suffice as a consistent connection with a verified masternode is mandatory. In addition, each masternode can be set to a unique IP only. There is not a possibility to host two or more nodes on a secondary IP. These are mandatory for a stable and efficient node.

Presently, the cloud-service market offers various VPS providers to choose from. We do not endorse or recommend any particular VPS provider and none are affiliated to ESSH in any way. **Following is a list of virtual service providers which each have individual pros and cons:**

- AWS (Amazon)
- DigitalOcean
- GCE (Google)
- OVH
- Linode

NOTE: It is better to use multiple hosting providers at once to build a more decentralized network.

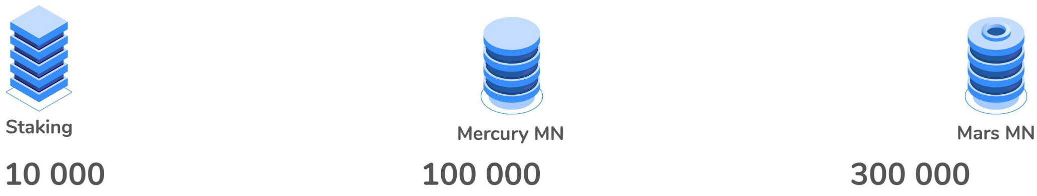
5.3 MASTERNODE REQUIREMENTS

Here are the required items and server specifications to setup the ESSH **masternode:**

Requirements

Device
Masternode server
UNIX
Unix OS
2GB+ of memory
80 GB HDD

Allocation



5.3.1 DEVICES

- A primary computer that can be turned on and off without affecting the masternode.
- A masternode Server (VPS with a unique IP address) to be powered 24/7.

5.3.2 TECHNICAL REQUIREMENTS

- Unix OS (Ubuntu 18.04 is recommended)
- at least 2GB of memory
- at least 80GB hard drive

5.3.3 ALLOCATION REQUIREMENTS

- For Staking: 10,000 ESSH
- For a Mercury masternode: 100,000 ESSH
- For a Mars masternode: 300,000 ESSH

NOTE: To deploy both a masternode and enable staking on one wallet, this requires 10,000 + 300,000 (for a Mars masternode) = 310,000 ESSH in total.

5.4 VPS SETUP

NOTE: Below you will find an example of VPS install instructions based on the Digital Ocean service. Digital Ocean should not be considered the most advantageous as this guide is for illustrative purposes only. Users are free to choose any VPS service provider and ESSH highly recommends an analysis of the cloud-hosting market for the most suitable options for individual needs.

5.4.1 SSH KEY

ESSH (Secure Socket Shell) designed to provide secure access to a remote virtual machine. It is a common practice to use ESSH for accessing a VPS. It is necessary to create an ESSH key first since it is required to proceed with further instructions.

MacOS and Linux

1. Open the terminal app and type the following:
   ```
   ssh-keygen
   ```
2. Choose a file path if needed. Otherwise, simply hit "Enter" or "Return" (↵).
3. Next, enter your passphrase. No text will be displayed while typing as a security measure. If you wish to skip this step and leave the passphrase blank press enter once again.
4. Repeat the passphrase if required.

5 Hit enter to confirm. The following messages should appear:

```
Your identification has been saved in /Users/$USER$/.ssh/tut_rsa.

Your public key has been saved in /Users/$USER$/.ssh/tut_rsa.pub.

The key fingerprint is:

SHA256: $KEY$

Username@$COMPUTERNAME$.local

The key's randomart image is:

+---[RSA 2048]-----+

| Eo              |

|O              |

| **    o        |

|=.+   . + +       |

|*b.  o S        |

|@A+o   . .       |

|=o.      .      |

|+o.y . .   +     |

|oo+-.. . ..+      |

+----[SHA256]-----+
```

6 In order to view the ESSH key, please type:

```
cat ~/.ssh/id_rsa.pub
```

5.4.3 SELECT YOUR DROPLET

1. Click on the "Create" dropdown menu in the top right corner of the Digital Ocean interface.
2. Hit on “Droplets” from the menu.
3. Choose Ubuntu 18.04 x64 in "Distributions" section.
4. Choose the most suitable Droplet plan. You can select the data center region as well as additional options like IPv6, Monitoring and Private networking.
5. Add the SSH key which was created earlier.
6. Finally, add a hostname for the Droplet. Once this step is done, an IP address will be provided.

5.4.4 SECURE YOUR DROPLET

1. Copy Droplet IP which was previously given.
2. Open the terminal app and type the following:

   ```
   ssh root@dropletIP
   ```
3. Type “yes” to proceed
4. Enter ESSH key password which was created previously.
5. During the following steps, a new user account will be created to avoid using root as the main account.
6. Type the following where &USERNAME& is the username you wish to use:

   ```
   adduser $USERNAME$
   ```
7. Answer asked questions and/or press enter to skip.
8. Execute the command below to grant administrative privileges for the new user:

   ```
   usermod -aG sudo $USERNAME$
   ```
9. Next, choose Open ESSH to enable the firewall:

   ```
   ufw allow OpenSSH
   ```
10. Confirm the enabling the firewall. Do not forget to type "y" and hit enter:

    ```
    ufw enable
    ```

Grant access for new user

1 Open the terminal app and type the following:

```
ssh $USERNAME$@serverIP
```

2 Enter server password

3 Enter any command which requires admin permissions:

```
sudo service ufw status
```

4 If no errors are returned then the account can be changed from root to the newly created account.

5.5 BUILDING AND RUNNING

5.5.1 INSTALLING THE NODE

All of the actions described below should be performed in the Terminal console. Please ensure all root privileges are granted as some steps may require additional permissions.

1 Clone:

```
git clone https://github.com/ESSHone/essx
```

2 Install required dependencies:

```
 apt-get install git checkinstall build-ESSHl libtool autotools-dev
automake pkg-config libssl-dev libevent-dev bsdmainutils python3
 libboost-system-dev libboost-filesystem-dev libboost-chrono-dev
 libboost-program-options-dev libboost-test-dev libboost-thread-dev
 libminiupnpc-dev libzmq3-dev libqt5gui5 libqt5core5a libqt5dbus5
 qttools5-dev qttools5-dev-tools libprotobuf-dev protobuf-compiler
 libqrencode-dev libdb-dev libdb++-dev
```

3 Set proper path configuration for further actions:

```
export BDB_PREFIX="$(pwd)/contrib/db4"
```

4 Go to 'contrib' folder:

```
cd contrib/
```

5 Install Berkeley DB by executing the following:

```
sh ./install_db4.sh .
```

6 Go back to the main folder:

```
cd ..
```

7 Execute 'autogen.sh' to generate configuration script:

```
./autogen.sh
```

8 Proceed to the configuration by performing this command:

```
./configure --disable-tests --with-unsupported-ssl --without-gui
LDFLAGS="-L${BDB_PREFIX}/lib/" CPPFLAGS="-I${BDB_PREFIX}/include/"
```

9 Build a generated package:

```
make
```

10 Finally, install the software:

```
make install
```

11 Once you have everything installed, it is necessary to create ~/.ess/ess.conf file with the following preferences:

```
rpcuser=%username%
rpcpassword=%password%
daemon=1
```

12 If all went well, you should execute one last command to start node:

```
essd
```

> **NOTE:** If any trouble is encountered during this process, try to execute "./essd --help" for more details about the available commands.

5.5.2 SETTING UP THE MASTERNODE

> **NOTE:** Please ensure sufficient coins are available to execute the masternode. All numbers are specified in the requirements section of this guide.

1. Make a collateral transaction to allocate funds using the ess-cli tool. To do so, execute the following:

   ```
   ess-cli allocatefunds masternode mn23 mars
   ```

   Where:

   - *allocatefunds* — command
   - *masternode* — purpose
   - *mn23* — alias of masternode
   - *mars* — tier of masternode (Mercury or Mars)

   EXAMPLE: txhash output example (hash of the collateral transaction):

   ```
   {
        “txhash” :
   “27acac9161f54b9ee343cad2289e4f834758926fad03c55796a685e7171df0b”
   }
   ```

2. Wait until the transaction has 15 confirmations. To check the number of confirmations, run the “gettransaction” command. Below is an example of this command in input and output.

   EXAMPLE: gettransaction input:

   ```
   gettransaction
   “27acac9161f54b9ee343cad2289e4f834758926fad03c55796a685e7171df0b”
   ```

EXAMPLE: gettransaction output:

```
{
     “amount” : 0.00000000,
     “fee” : -100.00000000,
     “confirmations” : 15,
     “bcconfirmations” : 15,
     “blockhash” :
“89a37cc47ef0d7f314b95ccf8f9ab4e0855cbdd959e8053e7098d3054bb999”,
     “blockindex”: 2,
     “blocktime”: 1556193465,
     “txid”:
“27acac9161f54b9ee343cad2289e4f834758926fad03c55796a685e7171df0b”
     “walletconflicts”[
],
     “time” : 1556193466,
     “timereceived”:1556193466,
     “details” : [
     {
          “category”: “move”,
          “amount”: 300000.00000000,
          “fee”:-100.00000000,
         “addressed”: {
               {
               “Address”: “DQpdt7LcWg5s3xfk69qBFLwikkdaSSH”,
               “Account”: “alloc->mn23”
               },
               {
               “Address”: “DBmwZnE4vC1wmdDb6R39cha5cMdCqea51p”,
               “Account”: “”
               }
}
```

3. Next, execute the command below to tie the masternode with the VPS:

```
ess-cli fundmasternode mn23 mars <hash_of_collateral_tx> <masternode_ip>
```

Where:

- *fundmasternode* — command
- *mn23* — alias of masternode
- *mars* — tier of masternode
- *<hash_of_collateral_tx>* — txhash value
- *<masternode_ip>* — ip address of the remote server, where masternode will be running.

As a result, a configuration line should be shown as below:

```
mn23 <masternode_ip>:<port> <private_key> <hash_of_collateral_tx>
<output_index>
```

Where:

- *mn23* — alias of masternode
- *<masternode_ip>:<port>* — ip address of the remote server, where masternode will be running
- *<private_key>* — masternode private key
- *<hash_of_collateral_tx>* — txhash value

4. As soon as the configuration is received, the node should be stopped:

```
ess-cli stop
```

5. The configuration must then be copied from the 3rd step and pasted inside the masternode.conf file. In order to do so nano command can be used:

```
nano ~/.<user_name>/masternode.conf
```

6. Configure the remote node by adding the below lines to the ess.conf file:

```
masternode=1
externalip=<masternode_ip>:<port> from paragraph
masternodeprivkey=private_key from paragraph 3.
```

7 Initiate the Masternode. As a last step, execute the ‘start masternode’ command on a local (control) node:

```
ess-cli startmasternode mn23
```

Where:

- *startmasternode* — command
- *mn23* — masternode alias

EXAMPLE: startmasternode success output:

```
{
“status” : “success”
}
```

In order to ensure that everything is up and running, please run the command below. Masternode mn23 must be in the list of existing masternodes.

```
ess-cli listmasternodes
```

EXAMPLE: listmasternodes success output:

```
{
     “network” :  “ “,
     “txhash” :
“27acac9161f54b9ee343cad2289e4f834758926fad03c55796a685e7171df0b”,
     “outidx” : 0,
     “status” : “ENABLED”,
     “addr” : “DQpdt7LcWg5s3xfk629QbLdFLwikkdaSSH”,
     “version” : 70916,
     “lastseen” : 15556194862,
     “activetime” : 0,
     “lastpaid” : 0,
     “tier” : “MARS”
}
```

5.6 STOPPING A MASTERNODE/NODE

There is no lock-up period for running an ESSH masternode, they can therefore be turned off and removed from the network at any time. Despite that, please take into account that dedicated nodes are ESSH for receiving coin rewards. To end the masternode, enter the following command:

```
ess-cli stop
```

To stop running the control node follow these steps:

1. Enter stop command first:

   ```
   ess-cli stop
   ```

2. Then delete the configuration line in **masternode.conf** file.
3. Restart control node.

6 COIN SPECIFICATIONS

Coins mined before release:

- 1.317.488.573 ESSH

Total coins mined after 7 years:

- Total 1.755.313.373

Coins mined:

- PoW blocks: 100 blocks
- PoS blocks: after 100 blocks

Coins per block during the year:

190	1st year
167	2nd year
142	3rd year
117	4th year
92	5th year
75	6th year
50	7th year

Reward distribution:

- 38% - block producers
- 45% - masternodes
- 17% - network support

Block size:

- 2MB

Boot nodes IPs:

- 4 IPs

7 THE ESSH WALLET

One of the core missions of the ESSH blockchain is to create an easy-to-use blockchain-powered platform with the most up-to-date crypto and tech solutions integrated for efficient access and management. By simplifying the user experience with the help of intuitive and clear UI (User Interface) design, the ESSH Wallet can act as a key to the decentralized world for those who could not utilize the technology before. The ESSH wallet is based on a user-friendly architecture that covers all major software platforms. The Smart Wallet can be treated as an ESSHl utility for any crypto operation (to meet any business, organization or consumer needs).

In general, a cryptocurrency wallet is nothing more than an application for storing public and private keys, cooperatively interacting with various blockchains to ensure digital currency transactions among its accounts.

A cryptocurrency wallet can be used for a variety of different tasks. For example, users can monitor their digital balance, send or receive money and view transaction history. Digital wallet transactions can be simply described as senders ESSHlly signing off ownership of a certain balance to another wallet, which is broadcast to the blockchain. At the same time, in order to spend the received coins, a private key in the personal wallet must match with the public address to which the currency is assigned. For cases when keys are the same, the wallet balance will increase and the senders will decrease for the sent amount.

One significant advantage of the ESSH wallet is its multi-coin functionality, meaning the likes of Bitcoin, Ethereum, Litecoin, ESSH, Dash, Bitcoin Cash and more can be stored, accessed and managed in a single application. Additionally, these cryptocurrencies are ecosystem ready, meaning users can benefit from built-in features such as instant fiat currency exchanges, advanced transaction history, auto-sync API and much more.

As mentioned before, the ESSH wallet can be used in a multitude of ways. In terms of storing (in the case of software wallets) and accessing digital currencies, there are currently three means of managing the ESSH wallet: the desktop, Android/iOS, and web applications. Below are the benefits and drawbacks of each:

- **Online.** This wallet is located on cloud storage and can, therefore, be accessed from various devices and places. Such a wallet offers convenience, however, the private key cannot be directly controlled..
- **Desktop.** As this wallet is installed on a personal PC or laptop, this wallet can only be accessed from a single computer. Despite that, it offers higher-security as the key can only be reached locally.
- **Mobile.** In some cases, a mobile wallet combines the pros from both desktop and online wallets. The mobile offers portable access, and the key is saved only on a personal device.

Wallet security is a question of high priority. The overall security level varies and depends on the wallet type. For obvious reasons, an online wallet works in riskier environments compared to offline wallets. This is due to possible vulnerabilities of the platform and online specifics. On the other hand, offline wallets cannot be hacked because they are not connected to an online network. Please take into account that (unless backup options have been saved) no matter the wallet type, a loss of the private key will result in permanent loss of access to the wallet and contained balance. The same goes if the wallet is hacked or when money is sent to an unwanted address, it is not plausible for the balance to be retrieved. Due to the immutable nature of blockchain, it is not possible to reverse transactions, so an enormous amount of care must be taken, just as with any fiat money, credit card details or online banking.

Five important methods of protecting assets:

- **Wallet Backup:** Backing up should be considered an imperative and the very first thing that should be done after creating the wallet. Naturally, there can be no assurances that any mobile, computer, hardware waller or another device won't fail. Therefore, ESSH recommends a backup of Keystore files to be saved on several devices. The ESSH Wallet offers multiple ways to do so, including a mnemonic phrase, seed or keystore file.

- **Keep passwords, mnemonic phrases, and keystore files offline.** It is strongly recommended to keep passwords offline. In fact, highly sensitive or private information should always be stored offline if and when possible in order to minimize any risk of an attack. ESSH recommends passwords are stored in safe places, such as a safe or bank box.
- **Never share wallet details with anyone.** Please be aware of any wallet information requests. The ESSH team will never, under any circumstances, request any credentials or access to wallets or keys. If any type of request is encountered, please contact the support team immediately and do not make any decisions without being 100% certain.
- **Keep software up to date.** Software updates are important and should not be ignored as they include not only new functionality and features but also potential security enhancements. Do not forget about updates especially if the device is rarely connected to the network.
- **Think about additional security.** ESSH recommends users set a long and complex password that cannot be guessed or hacked. Next, store only a small amount of coins for everyday use per wallet. Do not hesitate to double-check addresses before transactions as they are not reversible.

Multi-currency or single-use?

Despite the fact that Bitcoin is the most well-known and prevalent digital currency, hundreds of new cryptocurrencies have since followed suit with their own infrastructure, architecture, and ecosystems. If interests go beyond using only one currency, the good news is, the ESSH Smart Wallet application supports Bitcoin, Litecoin, Ethereum, Dash and over 700 tokens. There is no need to choose an isolated wallet as ESSH incorporates a multiverse of functionality.

Are there any transaction fees?

Generally, transaction fees are analogous to traditional banking fees paid for transactions. In the case of blockchain, fees are required to remunerate network miners for their work in ensuring the security of the blockchain. On the other hand, some transactions are free from any fees. It is important to understand that the transaction fee depends on the choice of priority, as higher fees ensure a faster transaction. In terms of opening an ESSH account or wallet, there are no fees and the process is free.

Are cryptocurrency wallets anonymous?

Crypto wallets can be considered completely anonymous, considering they are not linked to any personal information. But they are often regarded as pseudonymous because, although it is not possible to determine user identity via the wallet, the history of any transaction stored on the blockchain can always be viewed and cannot be removed, thus, such details as wallet addresses and transactions are scrutable.

8 ESSH ATOMIC SWAPS

8.1 WHAT ARE ATOMIC SWAPS?

Atomic Swaps are a direct peer-to-peer exchange of cryptocurrencies from one user to another without the use of a centralized crypto exchange services. The exchange process foresees that both users have full control over their private keys.

In addition, the atomic swap can be either directly processed between independent blockchains with different coins or executed through off-chain channels.

What began in July 2012, the idea of a peer-to-peer cryptocurrency exchange was introduced by Sergio Demian Lerner. The main concept of which however, never left its foundational stage and was not released.

The next stage in the development of atomic swaps came about in May 2013 after Tier Nolan introduced the first account of a plausible procedure for the swap.

Below is how atomic swaps work and what benefits they bring to the crypto community.

8.2 PROBLEMS WITH CENTRALIZED EXCHANGES

One significant problem atomic swaps solved is, for example, when a person wishes to exchange Bitcoin to another cryptocurrency or altcoin,, normally this person would need to use a third party centralized exchange to facilitate the transaction. Certainly, this method works, however, it has its pitfalls.

There is always a risk of hacking with third-party exchanges. One of the most notorious instances being Coincheck, which had $550 million worth of NEM stolen by malicious actors. The worst of the damage was inflicted on crypto holders in Japan, a country that was traditionally known for being crypto-friendly. This event, and many others like it, have made an enormous impact by decreasing the levels of trust around the world.

8.3 VOLUME DEMANDS

Third-party exchanges cannot cope with changes in demand, particularly when demand suddenly increases. A pertinent example of this occurrence can be seen in the case of Bitcoin Cash where the value decreased by over a half on the 12th November. This was caused by a considerable increase in demand in which most centralized exchanges could not cope with. In specific, Bithumb experienced 90 minutes of downtime and lost a quantity of 60,000 BTC.

8.4 GOVERNMENTAL REGULATION

Since centralized exchanges are located in specific countries, they are subject to local laws and regulations. Due to this, third party exchanges can not be trusted to provide long term solutions as frequent changes in the industry result in fluctuating regulations.

8.5 HOW DO ATOMIC SWAPS WORK?

In general, the working concept behind the atomic swap is fairly simple. It can be described in the following way; two sides involved in an atomic swaps decide on a shared secret code. These two users will exchange crypto if and only if their secrets match. Therefore, the exchange process can be controlled only by the owners of these secret codes. Technically, the swap process is realized via Hashed Timelock Contracts or HTLCs. Below, you will find a brief description on what hashed timelock contracts are in particular, otherwise please refer to Hash Time Locked Contracts for more details.

8.6 WHAT ARE HASHED TIME CONTRACTS?

Hashed timelock contracts are a particular form of payment channels. In essence, payment channels are off-chain state channels that manage payments. State channels are a two-way communication between participants used to conduct interactions similar to ones in the blockchain. This dramatically decreases the time to confirm a transaction as there is no need to wait for validation from third parties such as miners.

Below you will find the basic requirements to run an off-chain state channel:

- A locked segment of the blockchain state via multi-signature or a form of smart contract which is agreed among exchange participants.
- Participants' interaction with each other via signed transactions without submitting anything to the third party.

Please notice, it is possible to close the state channels at a stage predetermined by the users. Following are reasons why closures can occur:

- Expired TTL (time to live). Users can agree to set the TTL on the required time, for example, 1 hour.
- It could be based, for example, on the total amount of transactions made. The chain can be closed after $100 worth of transactions were made.

Hashed timelock contracts are one of the most convenient methods of processing payment channels.

It is worth mentioning that earlier versions of payment channels were built on timelocks. An HTLC follows this practice along with the use of "Hashlocks".

The HTLC allows payment channels to be opened where funds can be transferred between participants before an agreed deadline. These payments are recognized by submitting cryptographic proofs. Another valuable characteristic of HTLCs is that they enable a party to forfeit payment and return it to the payer. The concept uses a multi-signature transaction scheme in which both traders are held responsible for a successful swap.

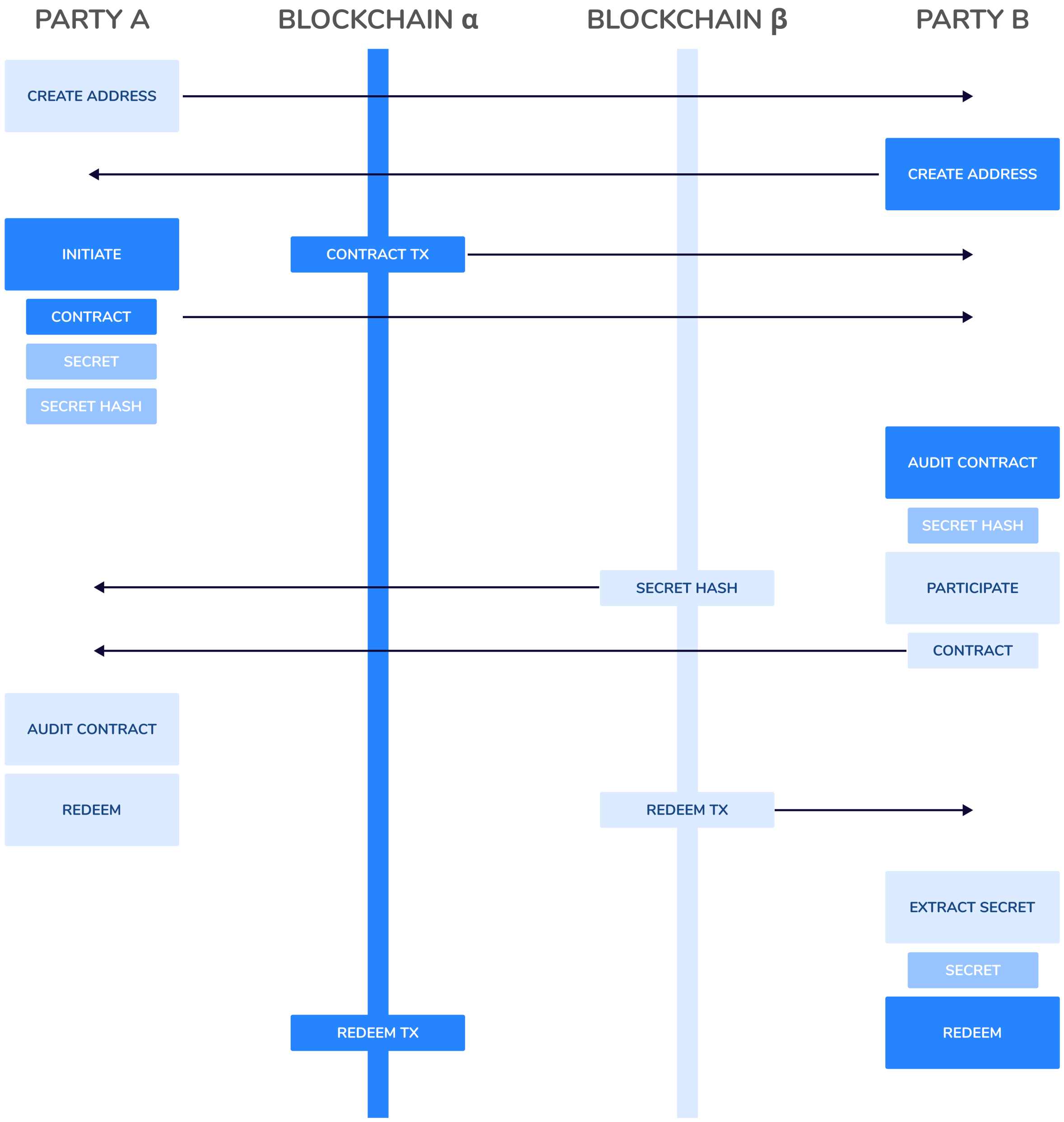
To make things clear, one party (also called an initiator) produces a secret key and initiates the trade into a transaction contract. The second party (also called the participant) can redeem the contract output only in case the secret becomes known. It is worth mentioning if a time period (usually 48 hours) expires after a transaction contract has been mined - but at the same time not redeemed - then the contract can be reimbursed back into the wallet of the initiator.

For example, if we suppose that the initiator wishes to trade with the participant in ESSH ERC20 for ESS H coins. In parallel, the initiator can also trade ESS H coins for ESSH ERC20 tokens in the same way, however, on the other blockchain.
At this point, the participant cannot spend from the Bitcoin contract of the initiator because they do not know the secret code.

The participant generates a contract transaction comparable to that of the initiator but on the ESSH blockchain and pays the ESSH amount into the contract. However, their own secret codes must be disclosed to the initiator to redeem the output. The initiator must reveal not the secret, but a cryptographic hash of the secret key to the participant in order for the participant to create their contract. The contract of the participant may also be reimbursed by the participant, but only after half the time that the initiator is required to wait for the reimbursement of their contract (typically 24 hours).

As long as each side paying into a contract on the blockchain, and both parties can not perform their reimbursement until the time allotted expires, the initiator redeems the participant's ESSH contract by revealing the code to the participant. The secret code is then extracted from the redeeming ESSH transaction of the initiator, which gives the participant the ability to redeem the contract of the initiator.

This method is atomic (with timeout) as it allows each party to redeem their coins on the other blockchain for at least 24 hours before a refund can be made. The following picture offers a visual of each party's steps and the information transferred between each party:


PARTY A
BLOCKCHAIN α
BLOCKCHAIN β
PARTY B
CREATE ADDRESS
CREATE ADDRESS
INITIATE
CONTRACT TX
CONTRACT
SECRET
SECRET HASH
AUDIT CONTRACT
SECRET HASH
SECRET HASH
PARTICIPATE
CONTRACT
AUDIT CONTRACT
REDEEM
REDEEM TX
EXTRACT SECRET
SECRET
REDEEM TX
REDEEM