THE KING OF NET

WHITEPAPER

Author: Siberian Cyber, Inc.

Version: Whitepaper v1.5

www.thekingof.net

ABSTRACT

A decentralized peer-to-peer cloud storage platform based on blockchain technology with end-to-end encryption will allow users to engage in transfer and sharing of data securely and privately, without depending on a third-party for data storage. The removal of a traditional centralized control would alleviate most typical data failures and outages, as well as significantly increase security, privacy, and data control, while introducing a self-replicating protocol would eliminate the issue of proof of data redundancy in other decentralised storage providers. We propose a solution in the form of a sensitivity coefficient verification system coupled with direct payments to allow for periodical audits of data integrity, and to offer rewards to peers maintaining data, along with a secondary, Linux-based computer operating independently from the primary computer, making the system more secure, since it would be used uniquely for this purpose. We also propose a model for addressing access and performance concerns with a set of autonomous and independent nodes.

TABLE OF CONTENTS

INTRODUCTION

The entire internet is based on data.

Storing that data is a problem increasing in complexity, which is partially solved by delegating more storage to the cloud. In turn, we have come to rely almost exclusively on large storage providers– "trusted" third parties that store the data. However, this system suffers from the inherent weaknesses of a centralised, trust-based model: the cloud is vulnerable to a variety of security threats. Furthermore, failures are compounded across systems and files because most storage devices rely on the same infrastructure. A decentralised cloud storage network offers multiple advantages over traditional datacentre cloud storage. Data security can be ensured by using multiple security settings and redundancy measures, and data integrity via *Proof of Retrievability* and *Proof of Redundancy,* thus reducing the impact of security breaches and system failures typical of centralized data centres. An open market for data storage will drive down costs for various services and providers due to more competition enabled by more providers and services using new and existing services. Data on the network will be resistant to suppression, interference, unsanctioned access, and data breakdowns. The King of Net offers a further evolution of the decentralised storage, by setting up a storage system for a decentralised internet through creating a Linux-based operating system that will work alongside a primary computer as a secondary computer, with the sole purpose of participating in blockchain networks, eliminating the risk of personal data loss and losing exclusive access to personal data. This paper describes a process of implementing a network, mechanisms for encrypting and storing data, and tools for participating in that network.

What is the King of Net?

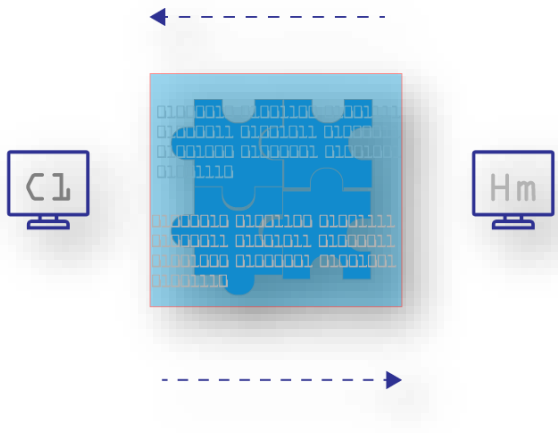
The King of Net is a global decentralised cloud storage solution with the focus on security and stability and deep hardware integration. Based on the Ethereum blockchain technology, TKON

uses a multilevel security system and a motivational model for hardminers who are incentivised to increase the volume and maintain the stability of the network.

In The King of Net System, hardminers provide data for the overall system and users store the data across all nodes in the network. Every hardminer that provides resources to the network is paid according to a formula revolving around the Stability Coefficient.

Hardware integration comes in a form of a specialised tool called the TKON Holder, and is a custom version of an iCTABLE product built for the blockchain.

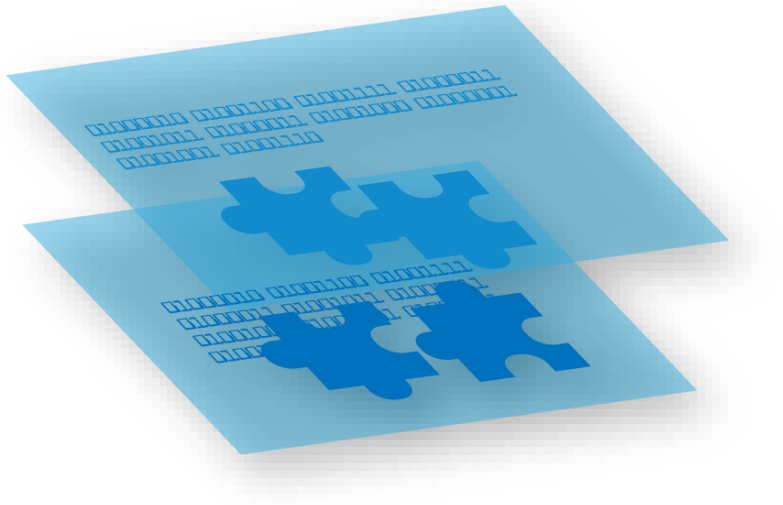
Files as Encrypted Shards



A shard is a portion of an encrypted file that is stored on this network. Sharding has a number of advantages over basic storage of a single file. All Files are encrypted end-to-end, meaning that they are encrypted once client side and once host side. This protects the contents of the data from the storage providers—hardminers, and anyone else but the user. The data owner retains complete control over the encryption key, and subsequently over access to the data.

Cloning

After the fragmentation, shards are cloned for redundancy and are separately encrypted, end-to-end. Depending on the security level chosen, shards can be cloned once or multiple times. Afterward, clones are also distributed across the network, with a protocol in place to ensure that no cloned shard is stored on the same node as the original, ensuring further redundancy.



1. Files are encrypted.
2. Encrypted files are split into shards
3. Audit pre-processing is performed for each shard (see Section 2.3).
4. Shards may be transmitted to the network.
5. Kademlia and Modifications

TKON also uses Kademlia a distributed hash table (DHT) but the shards are not stored in the hash table. Instead, Kademlia creates a distributed network with efficient message routing and other suitable advantages. TKON adds several message types, and

enhancements to core Kademlia functionality. In the future, the hash table may be used as a store for data location information, or other purposes.

Proof of Retrievability

Proof of Retrievability verifies the existence of a certain piece of data on a remote host. The proof should be miniscule in size, be calculated quickly, require minimal processing, and provide proof of the integrity of the data. TKON provides a standard format for issuing and verifying proofs of retrievability via audits—a challenge and response interaction in order to prove data integrity and availability to the data owner.

Our reference implementation uses Merkle trees and Merkle proofs. A sample TKON audit process is detailed below.

The system stores the set of challenges, the Merkle root and the depth of the Merkle tree, for every user using TKON and then transmits the Merkle trees leaves to the hardminer. The hardminer stores the leaves along with the shard. Periodically, the system selects a challenge from the available set, and transmits it to the hardminer. Challenges are randomly selected and are not reused. The hardminer uses the challenge and the data to generate the pre-leaf. The pre-
leaf, along with the set of leaves, is used to generate a Merkle proof, which is sent back to the data owner. The TKON Merkle proof is a small transmission. The system uses the Merkle root and tree depth stored on the user's computer to verify the proof by verifying that its length is equal to the tree depth and the hashes provided recreate the stored root. This scheme does not allow false negatives or false positives, as the hash function requires each bit to remain intact to produce the same output. Thus the data owner can have a known confidence level that a shard is still intact and available. In practice, this is more complex, as farmers may implement intelligent strategies to attempt to defeat partial audits. Fortunately, this is a bounded problem in the case of iterative audits. The probability of several consecutive false positives becomes

very low, even when small portions of the _le have been deleted.

Contracts and Negotiation

A user enters the system and decides to store the data on the network, they are able to choose the amount of data to be stored in the network (Reference Network Capacity) and the security setting for the data storage, ranging from Normal, Secret and Top Secret. The decision on which security setting to apply greatly affects the price of data storage. Because a user does not negotiate with a specific hardminer, since the data is split and randomized across the network equally, the process is automated. The duration of the storage is limitless and the price for storage is secured in the contract. If the stability of the system is consistently above the network minimums, then storing data within the network is a dependable and secure method of storing such data, leading to a higher price. If the stability of the system is at a minimum or slightly above the accepted minimum parameters, then the price is automatically lower to compensate for potential volatility.

Network Capacity

Network capacity is dependent on the following parameters: number of active hardminers, the amount of storage they provide, and the amount of extra storage required for redundancy schemes.

Payment

TKON Coins are used as a form of payment within The King of Net System. Once a user accepts a Smart Contract with the TKON and upload the files to the network, TKON Coins are automatically transferred to the network, based on the current prices, length of a contract and current system capabilities. A 2.5% fee is applied to all transactions. The entire payment

submitted by the user is spread across all hardminers. Hardminers are paid daily, according to their current stability coefficient and storage space provided to the network.

Redundancy Schemes

In typical cloud storage offerings, companies own or lease servers to store files uploaded by their customers. They tend to use RAID schemes or a multiple –datacentres as redundancy schemes to protect from physical or network failure. TKON is a peer to peer decentralized network, where each hardminer contributes various amount of storage volume and have different redundancy schemes, if any. After all, nothing stops a hardminer from suddenly logging off or completely exiting the system. To counter this behaviour, we introduce incentives for the hardminers, such as the stability coefficient (Reference Incentives), to protect the data. Unlike some other blockchain cloud storage solutions, where the data owner must introduce their own redundancy schemes, TKON automates many of the processes including redundancy, largely simplifying the process. The redundancy solutions are described below.

Cloning (Part 2)

Cloning is the basic redundancy scheme that is implanted for all files stored on the network. Depending on a security setting chosen by the user, shards of a file may be cloned once or multiple times. The clones are never stored on the same node as the original so that the integrity of the file is maintained during unforeseen events.

SELF-REPLICATING SYSTEM

Self-Replication Basics

The hallmark differentiating feature of The King of Net system is the inclusion of a self-replicating system to combat potential malfunctions within the system relating to data loss. The decentralized nature of the system is an excellent defensive mechanism against the problems plaguing centralized systems, but with the decentralized nature of the system, other potential problems arise. The most significant such problem is the loss of data due to a host malfunction or host leaving the system, whether temporarily or permanently. Generally, a decentralized, or a blockchain cloud storage separates a single data into many encrypted shards and places them into different hosts. In a true decentralized system, such as the TKON, the file shards are divided placed equally into all available hosts. To protect the file shards, TKON implements a sophisticated self-replicating Salamander protocol dependent on the signal sent by the Cyber Sensilla code, which is in turn is dependent on the system sensitivity coefficient.

Detecting Malfunctions

The Cyber Sensilla code detects the smallest deviations within the system, to a single shard. It constantly monitors the available hosts and is aware of the number of files, files shards, connected hosts, host volume, and rate of activity at any given moment. Once the files are shared, broken, and spread throughout the TKON, the sensor then monitors the additions and subtractions within the system, and sends a signal to Salamander protocol to guide the protocol towards duplicating and protecting data. For once, if a host leaves the system, Salamander protocol intervenes and duplicates the files/file shard across the TKON for security. Once the host deviant is back online, the system restores the files to the original place. If the

host deviant is not back online within twenty-four hours, Cyber Sensilla renders the host errant and places constraints on rewards and further participation on that host, whilst finding a more suitable host for the files places on the host errant.

Restoring Data

After detecting a malfunction, the data must be restored to the network to keep it intact. For that reason, there are two constantly operating company servers connected to the network that temporarily store replicated data until suitable new host is found by the system. Afterwards, the replicated files are uploaded to this new host and system integrity is restored.

DECENTRALISED STORAGE

There is no stopping progress. With a massive increase in popularity, decentralised networks are rapidly ascending as the top option for data storage. In the future, the speed and security of peer-to-peer networks will completely overtake the centralised datacentres.

Security

Decentralised solutions offer many benefits to the security of the data. Since the data is spread across all nodes in the network, it is not vulnerable to typical attacks on massive storage centres. In addition, the system becomes increasingly more secure and more resistant to the potential attacks with each new hardminer joining the network. As the number of nodes in the network increases, so does the security of the stored data.

Opportunity

Decentralisation offers many opportunities for people to earn with unused resources of their computers. This also offers an opportunity to potentially connect all computers creating a massive worldwide storage solutions, where all of the world's data can be stored and be available at a moment's notice.

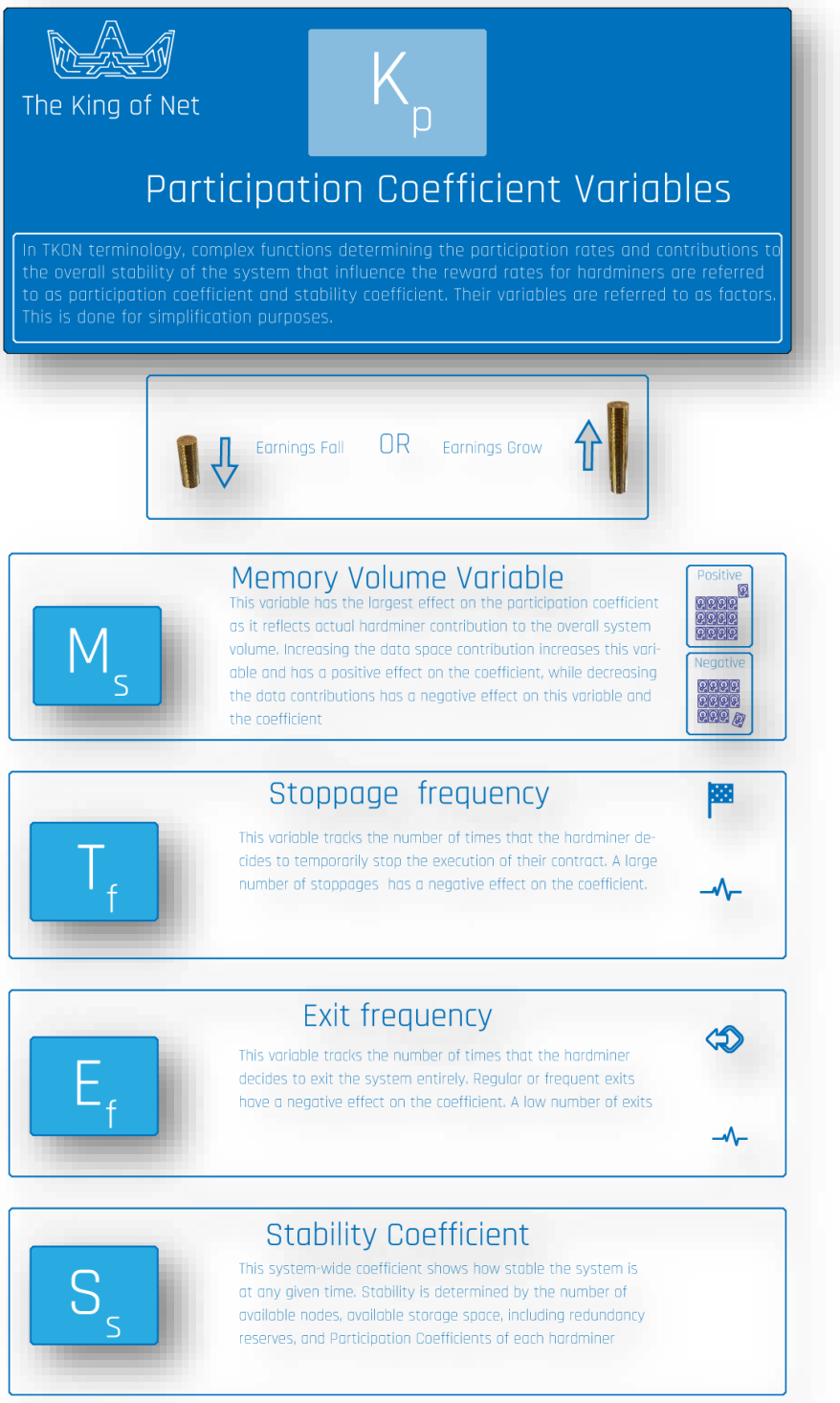
SYSTEM STABILITY

As the demand for fast, reliable, and secure storage solutions increases, the blockchain-based solutions are at the forefront. However, they generally lack stability, due to an inconsistent number of nodes present within the system at any given time and each node is unreliable. To incentivise hardminers to remain in the system, we developed a motivational system which is detailed below.

Coefficients

TKON uses a variety of coefficients, for internal calculations and purposes, with system-wide implications, some of which are detailed in this paper. The primary coefficients that pertain to the prices of storage and hardminers are the participation and stability coefficients: a personal stability coefficient and a system-wide, or network coefficient.

Hardminer Participation Coefficient



Each hardminer is assigned a basic stability coefficient once they join the network. As they continue to participate and contribute their resources to the network, their stability coefficient can increase or decrease based on their activity.

IMPROVING STABILITY

Stability coefficient can be increased through consistently keeping a node online and increasing resources contributed to the system.

DEGRADING STABILITY

Stability coefficient can be decreased through constantly disturbing the system by logging off and decreasing the amount of resources contributed to the system.

System Stability Coefficient

System stability coefficient is calculated by aggregating the stability coefficients of all hardminers registered in the system. The system stability affects the price of storage on the network, which incentivises hardminers to raise their personal stability coefficients, leading to an overall greater stability of the entire network.

Sensitivity Coefficient

Sensitivity coefficient is one of the most important coefficients used in the system. It is linked with the ability of the system to activate the Salamander Self-Replicating protocol. It has many variables that affect it, increasing the difficulty of such calculation, but it also increases the accuracy and effect this coefficient has on the ability of the system to respond to the slightest disturbances within the network.

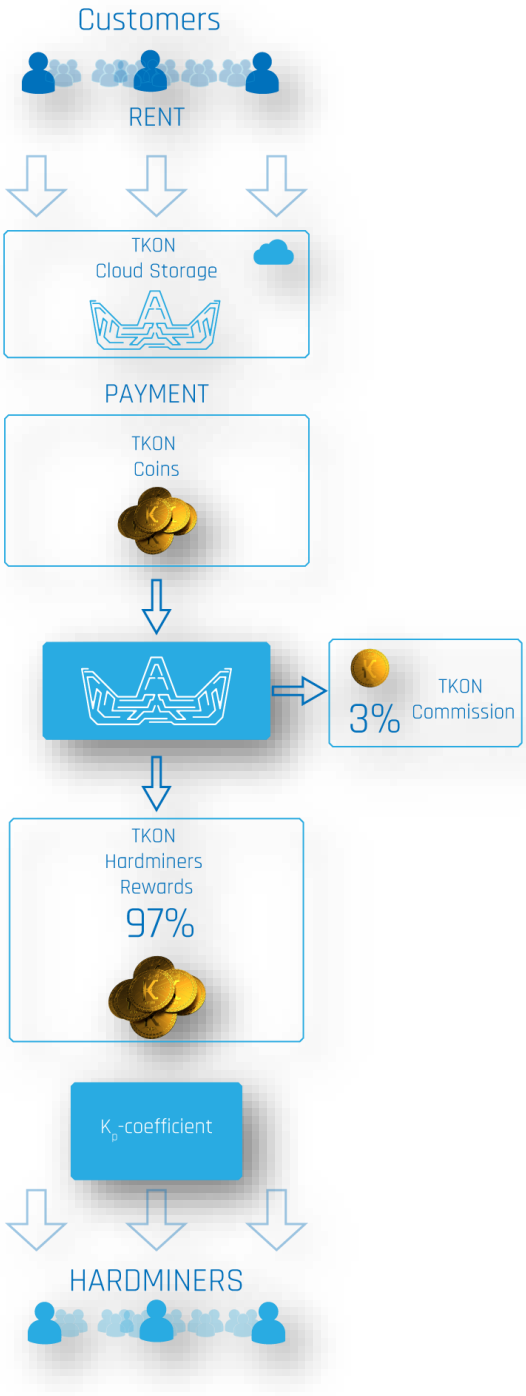
SMART CONTRACTS

What is a Smart Contract?

All transactions taking place within the TKON network are based on smart contracts. Smart contract is a self-executing contract, where certain rules are proscribed, that is transparent, safe, and reliable method for verifying and ensuring transactions occurring within the blockhain.

USING TKON COINS

Reasons for Tokenisation



In recent years, tokens have become an effective outreach of the global cryptocurrency. Made with the technology of the larger cryptocurrency networks, tokens allow for unprecedented freedom of use. In the case of TKON, the coin is Etherium-based and can be purchased, earned, or used for transactions within the network. Because the network is based on blockchain, the transactions are easily verifiable, yet secure. Another reason to use tokens is for stability purposes, as the hardminers are rewarded for more time spent within the system as a host.

Receiving Tokens

The first way to receive a TKON token is to purchase hardware from Siberian Cyber. The number of dollars one pays for a Siberian Cyber product is the number of tokens one is eligible to receive.

Earning Tokens

TKON Coins are earned through continued contributions to the overall network. Each time someone purchases Secret or Top Secret storage on the network, all users with the stability coefficient above 2 will receive a payment and continue to do so as long as the file is stored on the network.

Using Tokens

One can use the TKON Coins for transactions within the network or keep them for further use.

Exchanging Tokens

One can exchange the TKON tokens to any cryptocurrency at any time based on the real-time value.

HARDWARE INTEGRATION

Specialized Tools

There is a tool for every task, but within the blockhain-based storage, there are none available as the users generally are expected to provide the system resources from their primary computers towards the network. With the TKON, we introduce a specialized tool made specifically for blockhain networks—TKON Holder. BASED On a Linux operating system for security, the users will be able to dedicate all of the system's resources to a specific task,

namely storage within the TKON. As a separate computer, the users will never need to share access to their systems and compromise security.

Secure Access

At the heart of any blockchain system are the users, who provide their systems' resources for a larger system. The idea, while excellent and appropriate, presents a unique set of challenges, specifically in the realm of security. By creating a Linux fork that runs as a separate computer and acts as a node in the blockchain, we eliminate potential security risk for personal data stored on a user's primary computer, since the actual participation in the blockchain will be delegated to a secondary computer that. This specialisation of the purpose is already present in the mining community, but thus far has not been explored in the community of system resource sharing. The creation of such system will undoubtedly be beneficial for the users, communities, and networks alike.

Integration

TKON Holder is lightweight, portable, and durable, and can be placed separately as a case or be used in conjunction with iCTABLE. With the Linux fork operating system, TKON Holder is an alt-server for blockchain. It integrates seamlessly with any blockchain and utilises all of the "computer's" power towards participating in the blockchain network. This allows for an unprecedented participation rate within the blockchain, since TKON holder is a dedicated tool built specifically for this task

USING TKON HOLDER FOR OTHER BLOCKCHAIN NETWORKS

Although TKON Holder is recommended for use with the TKON System, user freedom and choice is paramount in any successful decentralisation of the internet. TKON Holder can be repurposed

for use in any blockchain network, whether as a separate case standing beside a user's primary system or integrated into iCTABLE. TKON Holder can offer many benefits to blockchain user or provider notwithstanding its usage as a TKON System node. It can be used as a storage provider in system like TKON, a resource provider in system like Golem, or even work as a mining tool for Bitcoin, Ether, or similar blockchain networks.

REFERENCES

R.C. Merkle, Protocols for public key cryptosystems, In Proc. 1980 Symposium on Security and Privacy, IEEE Computer Society, pages 122-133, April 1980.

B. Cohen. Incentives build robustness in bittorrent, (2003). http://www.bittorrent.org/bittorrentecon.pdf.

P. Maymounkov, D. Mazieres. Kademlia: A peer-to-peer information system. based on the xor metric, (2002). https://pdos.csail.mit.edu/ ~petar/papers/maymounkov-kademlia-lncs.pdf.

H. Shacham, B. Waters. Compact proofs of retrievability, (2008). https://cseweb.ucsd.edu/~hovav/dist/verstore.pdf.

V. Buterin *et al.* A next-generation smart contract and decentralized application platform, (2014). https://github.com/ethereum/wiki/wiki/White-Paper

Ari Juels and Burton S Kaliski Jr. Pors: Proofs of retrievability for large files. In Proceedings of the 14th ACM conference on Computer and communications security, pages 584–597. Acm, 2007.

Hovav Shacham and Brent Waters. Compact proofs of retrievability. In International Conference on the Theory and Application of Cryptology and Information Security, pages 90–107. Springer, 2008.

(Wilkinson, et al., 2016) Storj: a peer-to-peer cloud storage network.

https://storj.io/storj.pdf