ZentaChain

E-mail:Team@zentachain.io

WHITEPAPER CONTRIBUTORS

Harun Kacemer

Alfred Schlosser

Safi Razag

Ali Muhammed

SPECIAL THANKS TO

ZentaChain and friends!

ZentaChain

A decentralized blockchain based ecosystem, designed for security and anonymity.

ZentaChain Labs
ZentaChain.io

(Dated: December 2018, Version 0.1)

ZentaChain is a high-throughput decentralized blockchain based ecosystem, designed with a strong focus on security and anonymity. The ecosystem is capable of storing encrypted data and hosting scalable decentralized applications and distributed services. The ZentaChain MVP will be build on the Ethereum blockchain and will utilize the ERC20 standard also called Zenta. To meet the desired performance and sustainability requirements, ZentaChain interacts with state-of-the art lightning networks such as Raiden or Casper and innovates on several layers. Ethereum Smart contracts enable ZentaChain developers to build applications and deploy custom services on the platform. After alpha launch the ZentaChain team will launch their own blockchain. Currently ZentaChain Labs is considering several options to build it's core on, among others: Nano (block lattice), EOS and Ontology.The ecosystem holds all necessary building blocks for creating and hosting scalable decentralized applications and distributed services and P2P decentralized cloud storage. It's 5-tier architecture is composed out of five logical layers, each with a dedicated function.

CONTENTS

1. The problem worth solving

Need for secure interaction and digital data storage - owned by the initiators.

A. Resolve the thread of unauthorized data access and manipulation of information streams.
B. Protect against hackers - getting unauthorized access and stealing information, logging, harvesting and selling your data.
C. Leverage net neutrality
D. Ensure ownership of data and data security
E. Provide a secure framework for the hosting and creation of dApps
F. Ownership and security of cloud storage
G. Unencrypted, unsafe data storage Every transaction made on your device of issued by your action on some digital medium is potentially logged and stored somewhere in a database or on a file server. It is very likely this at this data isn't properly encrypted or anonymized.

2. Vision

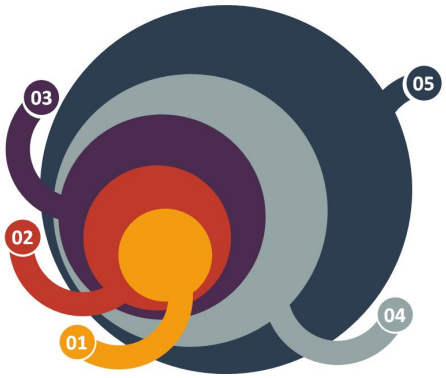
ZentaChain envisions a decentral ecosystem build for net neutral data- and transactions interchange and data storage. The ecosystem is maintained by its users and immune against several forms of cyber attacks and hacking. Next to that, viable solutions for security and data ownership problems are present. The ZentaChain is open source. ZentaChain aims to become the missing gap between decentralized meshnet cloud services such as IPFS (https://ipfs.io) and dynamic routing and addressing protocols such as DNS and HTTPS. This comes down to the fact that Zenta Labs will upgrade the IPFS peer-to-peer hypermedia protocol with state-of-the art blockchain technology.

The web will not only become ultra secure and decentralized and made permanent, it will also become faster and more open. ZentaChain will enable the ability to address large amounts of data with IPFS, and place the immutable, permanent IPFS links onto the Zenta ledger by using a blockchain transaction. This timestamps and secures content, without having to put the data on the ZentaChain itself. Next to that, ZentaChain will add additional encryption to files stored on the meshnet. ZentaChain brings the freedom and independent spirit of the web at full force and at low cost. The ecosystem will help deliver content in a way which can save you considerable money.

High latency networks are a real barrier of entry to developing world, the ZentaChain provides resilient access to data, independent of low latency or connectivity to the backbone. ZentaChain wants to be truly decentral, without any exception and will design the ecosystem in such a way that it never keep track on its users and will never save IP-address or any personal information. By design the ecosystem simply doesn't have this information nor can link transactions to a certain user or identity.

3. Architecture of the ZentaChain ecosystem

The ZentaChain 5-tier architecture is an enabler for a secure and at the same time has a modularity optimized for development of dApps and secure transport layers.



01 **zentacore**: security & business core

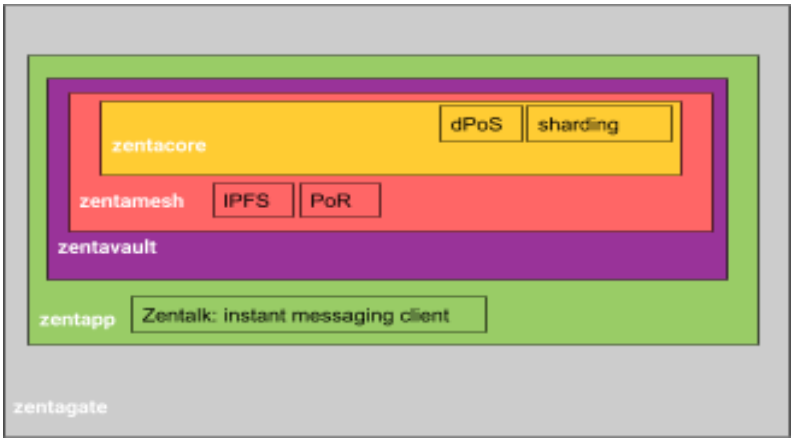
02 **zentamesh**: decentral meshnet

03 **zentavault**: decentral data storage

04 **zentapp**: dApp building and hosting

05 **zentagate**: gateway to internet

This makes ZentaChain awesome:



Security & Business

Zentacore holds all business logics of the ecosystem. It houses smart contracts - governing and consensus models - meshnet governing and advance blockchain technology such as sharding ZentaChain will drastically improve the Ethereum scalability by utilizing these kind of technologies. Sharding enables horizontal partitioning of data on the network. By scaling horizontally, thus dividing the systems transactions and spreading the load, a high-throughput of data is achieved resulting in a high Transactions Per Second (TPS) ratio.

3.1 Zentacore

3.2 Zentamesh decentral meshnnet Merkle Tree

3.3 Zentavault decentral data storage

3.4 Zentapp dApp building and hosting

3.5 Zentagate gateway to internet Components

4.What makes ZentaChain awesome?

ZentDapp: Designed for decentralized privacy services

Zenta enables users to build and host dApps within the ecosystem. All the running dApps will be anonymous and secure by design, no record of users and linked transactions will be stored within the platform. To prove and demonstrate the capabilities of ZentaChain, the team introduces a decentralized ultra secure messenger

Zentalk

Zentalk is a secured, decentralized peer-to-peer messenger dApp. Next to great usability, under the hood you'll find state-of-the-art encryption, security and since it's decental, ZentaChain guaranties full anonymity. All communication sent and received through Zentalk will be tunneled, pushed and dragged through a Zentamesh, a high performing MeshNet.This gives Zentalk the unique advantage that it's virtually immune against any forms of hack- or modification events like censorship and eavesdropping attempts.

Zentavault

Zentavault is a high-throughput encrypted and distributed file vault (encrypted storage) and transfer service. Unlike regular data storage systemsk, Zentavault will store nothing on the users device. Zentavault acts as an encryption delivery vehicle, with the ability to encrypt and dynamically distribute content securely onto the InterPlanetary File System (IPFS). IPFS is a peer-to-peer distributed file system that seeks to connect all computing devices with the same system of files. In some ways, IPFS is similar to the World Wide Web, but IPFS could be seen as a single BitTorrent swarm, exchanging objects within one Git repository.

Zentagate

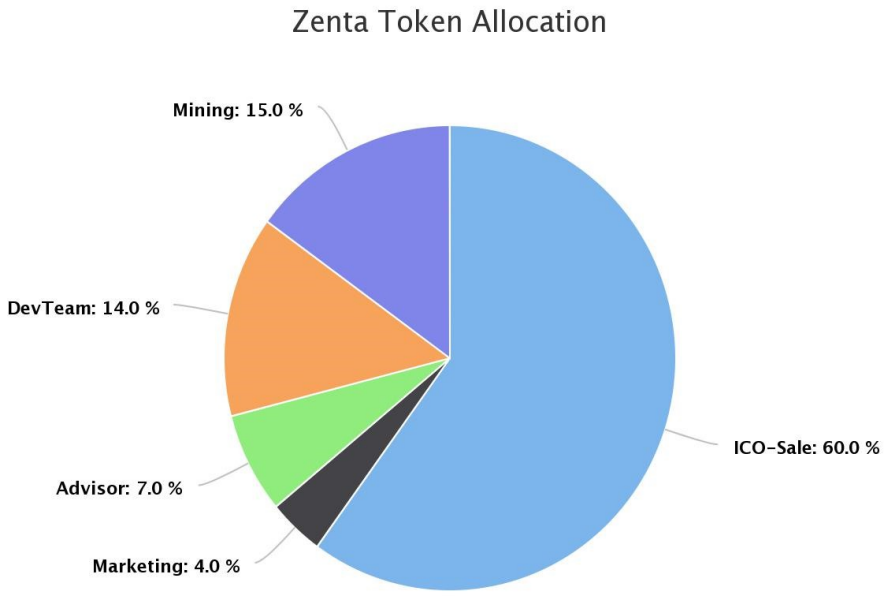
„IPFS-DNS and HTTP-Gateway will be 2 tasks for ZentaGate“

At ZentaChain we take security and anonymity very serious. We designed ZentaChain in such a way that it is running on it's own private Mesh network. Zentagate connects the ecosystem to unsafe networks such as the Internet. Zentagate provides an additional encryption and anti-hack layer, to ensure userdata and transactions are guaranteed - safe. Next to gatekeeping and safe-unsafe network relaying we plan on implementing a decentralized name service as well. Zentagate will run this service - enabling routing and rerouting data and transactions in and out the ZentaChain.

5.Economics & ICO

Token distribution

Official token name: **ZENTA**
Token Symbol: **ZENTA**
Algorithm: **DPOS**
Blockchain: **Ethereum - ERC20**
Number of tokens (total supply): **260.514.201 ZENTA**
Softcap initial token sale: **$3.000.000**
Hardcap initial token sale: **$10.000.000**
Number of tokens for sale: **60% - 156,308,520**



Token price will be calculated with this formula:

AverageETHprice / (Hardcap/TokensForSale/Rate/ProCenture)(PubSale/TokenForSale/Rate/ProCenture-0%)

6.Revenue streams

DPoS - staking Zenta and governing the ecosystem

Delegated Proof of Stake (otherwise known as DPoS) is a consensus algorithm maintaining irrefutable agreement on the truth across the network, validating transactions and acting as a form of digital democracy. Delegated proof of stake uses real-time voting combined with a social system of reputation to achieve consensus. It can be seen to be the least centralized consensus protocol compared to all others as it is the most inclusive. Every token holder can exercise a degree of influence about what happens on the network. In other words: holding Zenta tokens will reward you with voting power and extra tokens.

1) All the nodes in the network are connected to each other.

2) Existing nodes will not leave the network and no new nodes will join the network.

3) All nodes have the same weight or voting power and thus have the same stake.

4) There is no latency in the network.

PoR - relaying data on the meshnet

Relaying transactions will get rewarded with tokens. Zentamesh will Being server independent moves transaction management to clients that in this case will call peers belonging to the same mesh network. Each mesh is well-identified by all participants: the information life is then tied to the number of active peers in the mesh. The information exchanged is the blockchain itself. It is therefore necessary to have a peer synchronization mechanism.

In the literature there are several algorithms that also refer to cryptovalute, our blockchain has a proprietary implementation based on proof of relay:a dynamic evolution of proof of time. We have thus devised a relay that moves between the peers that make up the mesh network and groups the validation of the ledger. The advantages of using the proof of relay are as follows:

1. **It fits according to the mesh configuration**
2. **It responds to network congestion**
3. **Later it acts as a proof of time**
4. **It can be paused**

All of the positive factors listed have been implemented to have a blockchain that over time can increase the response rate and at the same time consumes little electricity.
Chain services will require a small network transaction fee. The ecosystem will be designed as such that inflow fees will go back to the governors of the ZentaChain.

Whitepaper

zentachain.io

"the future illuminates the potential of the present"

8.Prephase

The digital revolution rewrote the world we work in. We already live in a new version of our future. As Barbrook said in 2006: 'The importance of new technology is not for what it can do in the here and now, but for what more advanced models might be able to do one day. The present is understood as the future in embryo – and the future illuminates the potential of the present.' Digital transformation pays a larger contribution to our future. These days, humans depend on technology more than ever. Since the personal computer and more recently the smartphone got invented, it seems that almost the entire world has become interconnected. We cannot imagine a world without digital communications and interactions, social media, user-generated websites, and free online encyclopedias. The importance of the internet, the fibers that connect us all, keeps on growing. Now you have access to the world, in turn, the world has access to you. Further digitalization can be considered as a positive because it transforms the whole way of working, making it more usable and efficient. It increased mobility and productivity and reduced the need for a dedicated workspace. But the digital revolution has a dark side as well, and that is exactly what ZentaChain is going to protect against the company who is saving your data files and document on a central server. Zentalk protects also your communication history from a big company which they have your Message history and selling your Message history.

9.The problem worth solving

Need for secure interaction and digital data storage - owned by the initiators Unauthorized data access and manipulation of information streams has always been a threat, only this threat is expanding ever since the first network was invented.

Hackers can now attack you from numerous countries, different regions and multiple locations simultaneously. If that's not bad enough, there are also multi billion dollar corporations that have made a lucrative business out of logging, harvesting and selling your data.

They sell your location history, messages, photos, documents, and even sell the profiles they created on you. Everything about you is up for auction to any corrupt organization throughout the world who wants this data, no matter how malicious their intentions are. Some telcos may harvest on user data and actions. Regardless if they are regulated, and shouldn't do that, you can never assume they don't do that, perhaps they'll need it one day for 'training' purposes. Unencrypted, unsafe data storage Every transaction made on your device of issued by your action on some digital medium is potentially logged and stored somewhere in a database or on a file server.

It is very likely this at this data isn't properly encrypted or anonymized. Services offered by social media companies and messaging applications advertise themselves as a "free service" that does not cost the consumer any money. Instead, these companies are enticing regular people to put all of their data and content online to be stored and sold to sleazy organizations that are known to be manipulative. It is what it is, you pay for the service in data, not in terms of monthly fees.

10.Data Hacks & Breaches

The number of internet users rapidly grows. (from 1.36 billion in 2007 to 3.57 billion in 2017, an increase of over 2 billion users in the last 10 years), it is observable how our daily lives are becoming more and more virtualized and our feelings, thoughts and personal information - basically, our identities, are making their way into the online world, making them vulnerable to thieves who are looking to extract that data for their personal gain.

We have come to a point in time when everything and everyone can become a target. From personal attacks executed via phishing, ClickJacking, social engineering and other similar techniques to large scale infiltrations in centralized corporate databases which potentially give hackers access to large amounts of personal information. Although the former can, in most cases, be avoided by conscientious use of safe browsing practices, what worries us the most is the latter, because of the sheer scope of affected data and the fact that, in this case, the security of our personal information is completely out of our own hands.

11.Privacy

In the past two decades, mostly due to the impact internet has had on our everyday life, the question of preserving an individual's right to privacy has broadened from our physical to our digital environment. The famous quote "Knowledge is Power", translates more to "Data is Power" inmtoday's world, as we continue pouring our personal information into proverbial "gold-mines" that attract, not justhackers and ill-meaningindividuals, but also companies and conglomerates seemingly operating completely by the law. On one hand we are being targeted, for our identities, our credit card numbers,digital assets,etc.,while on the other,we ourselves are handing over our information in exchange for a more comfortable user experience. Although measures,like Europe's General Data Protection Regulation (GDPR), are being implemented to protect the users, their influence and level of execution still raise a lot of questions.

Weak and inefficient country administrations have allowed for dubious practices by companies who gather their users' data, effectively giving them free reigns to handle our information at their complete discretion. We have already mentioned some of the most recent privacy affairs in the Data Breaches chapter of this Whitepaper. These and other similar incidents seem to either sadly fade into oblivion without proper repercussions and reaction from the wider public, or they sometimes create an atmosphere of paranoia and distrust which is detrimental to our interpersonal communication. This is why it is imperative that we become aware of our privacy rights and we look to new ideas, like the blockchain, as solutions to the burning privacy issues. Blockchain provides the tools users need to be fully in control of the means of their personal data distribution. It allows for higher degrees of anonymity (some even enable complete anonymity) and deployment of different encryption methods to safeguard that data. It gets rid of the vulnerable centralized databases and is immune to data tampering and manipulation. Ultimately, it presents itself as a platform with tremendous potential for re-establishing trust lost by unscrupulous cash-grabbing behaviour exhibited by world's leading businesses.

12.Centralized messenger Apps

How they work and what problems they bring

As the presence of the Internet in our daily lives grew, so did the need for more efficient, more widespread means of online communication. Subsequent decades brought forth progress and the emergence of popular messaging apps like AIM, ICQ and PowW which started incorporating more advanced features including private messages, multi-user groups and file sharing.

In recent times the increase in popularity of messenger apps has correlated directly with the increase in smartphone usage. Desktop apps have been replaced by their mobile counterparts, enabling for more instant, on-the-go means of communication. The user base has grown exponentially, with eading service providers like WhatsApp,Facebook Messenger and WeChat individually amassing over 1 billion monthly active users.

However, today's messenger systems are far from being flawless. Recent events have raised serious questions about the issues of privacy. Facebook (owner of the two most popular messaging platforms) has been hit with indictments over selling user data for use in political purposes and the most popular messaging app in blockchain space, Telegram, has agreed to disclose its users' data to "the relevant authorities" if it receives a court order to do so. This is an issue related directly to the centralized architecture of existing messenger apps. Such an architecture enables the app owners to use, manipulate and restrict content, leaving its users vulnerable to potentially serious privacy breaches.

Today's messengers seemingly attempt to solve this by implementing end-to-end encryption solutions, but again, these close sourced implementations raise questions about the quality and actual levels of encryption. Another weakness of today's messenger apps is it's vulnerability to SIM swap attacks. As most apps require a phone number to register, hackers can potentially gain access to the user's messenger content by convincing the carrier to switch the victim's number to a SIM card they own.

13.Centralized Clouds

How they work and what problems they bring

Centralized clouds are services which enable storing of data on remote servers and access to that data via the internet. They are either free to use or require a monthly fee which is usually based on the length of the contract and storage capacity. Centralized clouds work by utilizing virtualized data centers which can be linked to the user via a web interface. The user uploads his files over the internet which then get stored on data servers.

They can be accessed only by providing a unique ID which then triggers metadata assembly allowing the user to view, edit, transfer or synchronize the files. To ensure uninterrupted data retrievability and integrity the files should be stored on more than one server. There are different types of centralized cloud storages:

Public Cloud Storage - is a shared resource environment in which the client is charged only for the resources being used and the service provider is responsible for the cloud infrastructure maintenance. Private Cloud Storage - is usually an on-premise service used by a single client/organization and maintained by the service provider. Hybrid Cloud Storage - is a combination of public and private cloud storage which enables the flexibility of storing sensitive and publicly accessible information on different cloud types. It is obvious why centralized clouds have gained popularity in recent times.

The increased demand for data mobility and accessibility is causing users to switch over from their physical counterparts like HDDs and USB flash drives. However, although a step in the right direction, centralized could services still have their problems. Firsty, the client is not the data owner.

The centralized server architecture puts the client's data into the hands of the service providers and also makes it vulnerable to data breach hacks as well as DDoS attacks which interrupt the continuity of data access. Public Cloud services are especially exposed because of their resource sharing component which allows for malicious intrusions via one of the cloud's clients.

Laws and regulations are another major concern, with data security and privacy being susceptible to the ever changing rules set by different governments of the world. For smaller organisations looking to employ complex data onto the cloud, cost can be a serious issue, as large bandwidth requirements might prove to be financially unviable.

14.The Vision of ZentaChain

ZentaChain presents viable solutions to these problems. Open source, content that is not stored on centralized servers and integration of technologies like MeshNet for an increased level of security, clearly show the benefits of using apps developed on this platform.

We look forward to safeguarding customers data with a fully secure data protection-ecosystem, against breaches and all kind of cybercrime!

The ZentaChain team breaks with the way mainstream tech companies treat their customers. It seems as if every service that people use today is a secret ploy to retrieve information, for retailers and organizations, to use for targeted consumer advertisement. ZentaChains goal is to provide people and companies a way to remain secure, without fear of eavesdrop, spying, or retrieving their data. ZentaChain pledges to never keep tabs on our customers and will not save your IP address, your email, or phone number. Nothing will be required for service use, except our installed app wallet and ZentaChain (Zenta) tokens. As with all our services ZentaChain will have no record of your identity, what region you live in, or any personal information about you. ZentaChain does not care about these personal effects because our only goal is to provide world class services that will give the consumer absolute anonymity, security and privacy.

15.ZentaChain Ecosystem and the Zentacore

Zentacore will employ sharding to drastically improve on scalability and transaction speeds of the Ethereum blockchain. The easiest way to understand how sharding functions is to draw parallels with horizontal database partitioning. Imagine having a large database containing millions of records and trying to search for ones containing only a certain attribute. From a technical perspective this is a task that takes a significant amount of time to execute. However if we were to partition the database into tables according to its attributes it would decrease the search time, as only a single table would be queried in that case. The sharding solution is interconnected with the entire ZentaChain ecosystem and enables partitioning and storing of the entire blockchain within nodes (shards) which are then responsible for validating solely the transactions contained within them as opposed to each node having to process every single transaction on the blockchain. Each shard contains nodes called collators, tasked with creating collations which hold information about the shard and its state before and after a transaction is processed. Collations are gathered by super-nodes and, if found valid (all transactions and states of collations must be valid and signed by ⅔ of all collators), they are placed into individual blocks and added to the blockchain. By utilizing sharding, Zentacore becomes a key ecosystem component for providing users with a seamless , delay free experience of all ZentaChain products.

Smart Contracts /HTTP

Zentacore use another important feature of the Ethereum blockchain- Smart Contracts. These digital contracts secure and automate processes which are needed to run ZentaChain´s dApps and maintain the network; tasks and commands are being carried out seamlessly with speed, precision and consistency. They can interact with other smart contracts and are defined by the developer. If these contracts are built up they can never be changed or altered, it becomes resolute law and makes it impossible to ever be tampered with. This leads to a secure and consistent ZentaChain ecosystem.

There are many disadvantages of HTTP such as inefficiency, no historic versioning, and centralization. So IPFS overcomes the disadvantages of HTTP. Zentachain presents a framework where the scientific research record keeping can be done in a secure, tamper proof environment using blockchain technology, IPFS and Zentachain smart contracts. For the storage of documents such as project reports, memorandum of understanding, funding projects documents, attendance records, and minutes of meeting, IPFS and Zentacore is utilized,along with certain access control methods, since all participating nodes in the network need not necessarily be able to access all the important information. Methods like secret sharing and asymmetric key cryptosystem can be implemented as additional functionality in the system for limiting the access structure only to certain users of the system. The provenance metadata information of the documents stored in Zentavalut is further uploaded to the blockchain in order to ensure the integrity of the information. The Principal Investigator(PI) can ensure that these documents are only accessed and modified only by intended users, who are allowed access, using this audit information on the blockchain.

16.ZentaChain Smart Contract and IPFS

The documents such as project reports, project funding details, memorandum of understanding, attendance records, and minutes of meeting are encrypted and stored in the IPFS and Zenta. IPFS is a distributed file system that creates a unique hash of each document, and the other nodes on the network can access and view the files only if the unique hash of the file is known to them. In order to restrict access to particular nodes on the network, certain access control methods can be applied.

Smart contracts are written in a high-level coding language called Solidity which is influenced by coding languages such as C++, javascript and Python. To develop ethereum smart contracts, Remix IDE can be used, which is a browser based IDE. Another oneis the Truffle framework which supports built-in smart contract compilation, linking, deployment and binary management.Two ways of access control can be applied: One way of restricting access is by implementing asymmetric encryption scheme through Sha256/AES/GnuPG.In this scheme, the users who have the key only can decrypt the document. Others can not decrypt the document.or data.

Even if the link of the document is provided to the users, the users can only decrypt the document when they have the key. So in the proposed system, this particular secret key is called as the master key. This master key is provided to the users who are allowed to access the system when they are registering with the system. Using this master key both the PI and JRF can access the IPFS file system and access the files. So, all the files are accessible by the users who are registered into the system. The user can login into the Zentavalut and access the file, download it, decrypt it and then view it. But, an important restriction is applied here. Only the PI can upload and create documents on the IPFS network. Uploading and creating new documents in IPFS network is restricted for the JRF. Consider the first scenario, where the JRF wants to access a particular document from IPFS. In this case, there is no problem since the JRF can obtain the link of the document, download it, and decrypt the document using the master key, and then view the document.JRF can do anything with the downloaded document such as modifying it or deleting it. But all these operations will be affected only in the local copy of the JRF's downloaded document. The original document in the IPFS remains unchanged.

The derived key is usually derived from a password by using Password based key derivation function. A key derivation function (KDF) is a function which derives one or more secret keys from another secret value such as a password or a master key using a Pseudo Random Function (PRF). There are many modern based key derivation functions. This provenance data is very much necessary for further auditing purposes. The logs of the IPFS file system can be used to collect the provenance data of the documents stored. This provenance data information can be further embedded in Ethereum blockchain as transactions using smart contracts which are built using Solidity Programming Language , Truffle and Remix IDE Algorithm provides the glimpse of the smart contract to retrieve a file from IPFS, whose hash has been stored in blockchain using smart contracts.

1)Automatically AES Key Encryption 256-bit on Zentavalut over APIs

Encrypt: 1) FileEncrypt(F) → (CTF ,K, kw): DO selects a keyword set kw from the file F, randomly selects AES key K from the key space, computes CTF = EncK (F), where EncK (F) denotes using AES algorithm to encrypt F, the encryption key is K. DO uploads the ciphertext CTF to IPFS, records the file location hlocation returned by IPFS. 2) KeyEncrypt(PK,K, hlocation,P) → CTmd : DO computes CTI = EncK (hlocation). In order to encrypt the AES key K under the access policy P, DO computes where <bXi,bYi> = <Xi.ki , Yi,ki>, IP is a subscript set of P. Then, DO randomly selects sRZ p , computes CTK = <P,C0,C1,C2>, where C0 = K · Y s P , C1 = g s , C2 = X s P . DO randomly selects AES key K1, computes CTmd = EncK1 (CTK ,CTI), embeds CTmd into transaction TXct , and broadcasts TXct to the Zenta blockchain. when transaction has been approved, record the transaction id txid and corresponding key K1. The length of txid and K1.

Decryption by AES Privat Key 256-bit on Zentavalut over APIs

(CTI,CTk , SKd , PK) → F: DU computes d = F(kwf||1,Ks), computes txidj = d ⊕ txid gj , K1j = d ⊕ Kf1j for txid gj Stxid , Kf1j Sk1. DU reads transaction txidj data from the Ethereum blockchain, and computes (CTK ,CTI) = DecK1j (CTmd), where DecK1j (CTmd) denotes using AES algorithm to decrypt CTmd , the decryption key is K1j . DU first checks whether his attributes set S PP, and if not, returns ⊥ and reads the next transaction data; otherwise, computes σ P = Q i I P b σ i , then AES key K is recovered as computes

$$\text{where } b\sigma i = \sigma i,ki = g\, H0(y||i||ki)H1(sk)\, H0(x||i||ki)$$

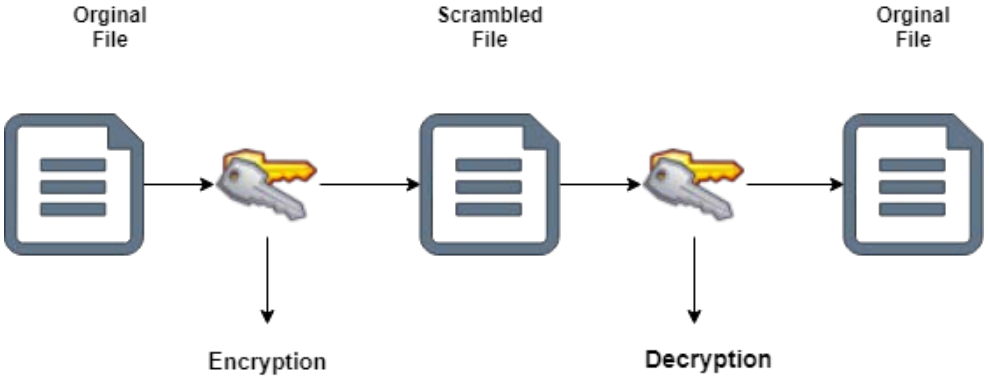
DU computes hlocation = DecK (CTI), then downloads CTF from IPFS based on hlocation, computes F = DecK (CTF) to recovery original file F.

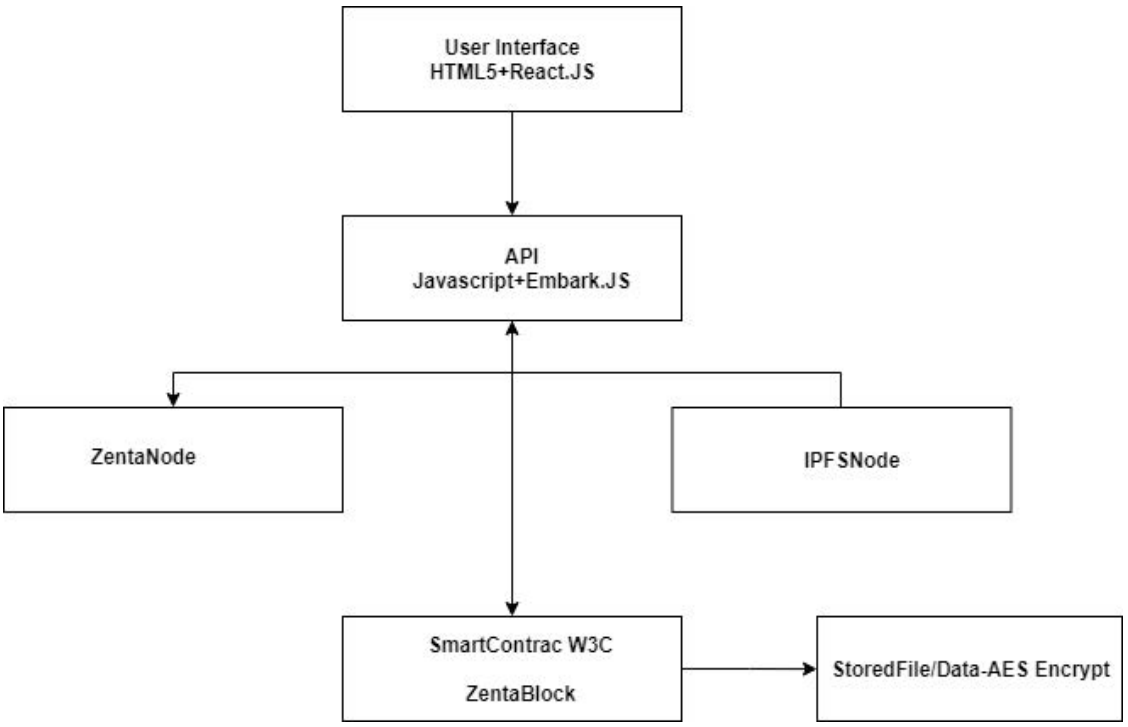
Implement Symmetric And Asymmetric Cryptography Algorithms

The difference between symmetric and asymmetric algorithms is the performance and size. Symmetric encryption is faster and used to encrypt a large data sets. Asymmetric is well suited for encrypting a small messages. But using these two strategies lead you to implement a robust security system in the Zentavalut dapplication.

Symmetric-Key Encryption

With symmetric-key encryption, the encryption key can be calculated from the decryption key and vice versa. With most symmetric algorithms, the same key is used for both encryption and decryption.





2.)Zentachain Web Cryptography API

ZentaChain JavaScript API for performing basic cryptographic operations in web applications, such as hashing, signature generation and verification, and encryption and decryption. API for applications to generate and/or manage the keying material necessary to perform these operations.

Uses for this API range from user or service authentication, document or code signing, and the confidentiality and integrity of communications. The Zentachain Web Cryptography API defines a low-level interface to interacting with cryptographic key material that is managed or exposed by user agents. Cryptographic transformations are exposed via the SubtleCrypto interface, which defines a set of methods for performing common cryptographic operations.

In addition to operations such as signature generation and verification, hashing and verification, and encryption and decryption, the API provides interfaces for key generation, key derivation and key import and export.

A web application may wish to permit users to protect the confidentiality of data and documents stored with remote service providers prior to uploading. Using the Web Cryptography Zentachain API, the application may have a user select a private or secret key, optionally derive an encryption key from the selected key, encrypt the document, and then upload the encrypted data to the service provider using existing APIs. This use case is similar to the Protected Document Exchange use case, with viewership of the document limited to the user themself.

Instead, by allowing the CryptoKey to be used with the structured clone algorithm, any existing or future web storage mechanisms that support storing structured clonable objects can be used to store CryptoKey objects.

In practice, it is expected that most authors will make use of the Indexed Database API, which allows associative storage of key/value pairs, where the key is some string identifier meaningful to the application, and the value is a CryptoKey object. This allows the storage and retrieval of key material, without ever exposing that key material to the application or the JavaScript environment. Additionally, this allows authors the full flexibility to store any additional metadata with the CryptoKey itself.

CryptoKey interface Zentachain (Exsample)

```
enum KeyType { "public", "private", "secret" };

enum KeyUsage { "encrypt", "decrypt", "sign", "verify", "deriveKey", "deriveBits",
"wrapKey", "unwrapKey" };

[SecureContext,Exposed=(Window,Worker)]
interface CryptoKey {

  readonly attribute KeyType type;   readonly
attribute boolean extractable;   readonly
attribute object algorithm;   readonly attribute
object usages;

};
```

KeyAlgorithm dictionary

The KeyAlgorithm dictionary is provided to aid in documenting how fixed, public properties of a CryptoKey are reflected back to an application. The actual dictionary type is never exposed to applications.

```
dictionary KeyAlgorithm {
  required DOMString name;
};
```

KeyType The type of a key.

The recognized key type values are "public", "private" and "secret". Opaque keying material, including that used for symmetric algorithms, is represented by "secret", while keys used as part of asymmetric algorithms composed of public/private keypairs will be either "public" or "private". KeyUsage A type of operation that may be performed using a key. The recognized key usage values are **"encrypt", "decrypt", "sign", "verify", "deriveKey", "deriveBits", "wrapKey" and "unwrapKey".**

Symmetric Encryption (JavaScript Example)

```
var encoder = new TextEncoder('utf-8');
var clearDataArrayBufferView = encoder.encode("Plain Text Data");

var aesAlgorithmKeyGen = {

  name: "AES-CBC",
  // AesKeyGenParams
  length: 128
};

var aesAlgorithmEncrypt = {
  name: "AES-CBC",
  // AesCbcParams
  iv: window.crypto.getRandomValues(new Uint8Array(16))
};

// Create a key generator to produce a one-time-use AES key to encrypt some data
window.crypto.subtle.generateKey(aesAlgorithmKeyGen, false, ["encrypt"]).then(
  function(aesKey) {
    return window.crypto.subtle.encrypt(aesAlgorithmEncrypt, aesKey,
[ clearDataArrayBufferView ]);
  }
).then(console.log.bind(console, "The ciphertext is: "),
       console.error.bind(console, "Unable to encrypt"));
```

Generate a signing key pair (JavaScript Example)

```
var encoder = new TextEncoder('utf-8');

// Algorithm Object
var algorithmKeyGen = {
  name: "RSASSA-PKCS1-v1_5",
  // RsaHashedKeyGenParams
  modulusLength: 2048,
  publicExponent: new Uint8Array([0x01, 0x00, 0x01]),  // Equivalent to 65537
  hash: {
    name: "SHA-256"
  }
};

var algorithmSign = {
  name: "RSASSA-PKCS1-v1_5"
};

window.crypto.subtle.generateKey(algorithmKeyGen, false, ["sign"]).then(
  function(key) {
    var dataPart1 = encoder.encode("hello,");
    var dataPart2 = encoder.encode(" world!");
  return window.crypto.subtle.sign(algorithmSign, key.privateKey, [dataPart1, dataPart2]);
},
  console.error.bind(console, "Unable to generate a key")
).then(
  console.log.bind(console, "The signature is: "),
  console.error.bind(console, "Unable to sign")
);
```

17.Sharding (Zentashard)

The demand for scalability on the Ethereum is becoming increasingly urgent. The Cryptokitties incident demonstrated how quickly the Ethereum network can clog-up. ZentaChain will use Sharding to establish a high TPS, future proof infrastructure. Understanding why ZentaChain uses Sharding: One of the major problems of a blockchain is that an increase in the number of nodes reduces it's scalability. This may seem counterintuitive because more nodes doesn't result in more power or more speed, the opposite is true: One of the reasons a blockchain has its level of security is because every single node must process every single transaction. This is like having your report checked by every single accountant in the company. While this may ensure that your report is faultless, it will also take a really long time before you get it back and finally approved. Ethereum faces a similar problem. The nodes are your accountants, each transaction is your report.

How to solve this problem?
ZentaChain could reduce the number of nodes (accountants) until we are satisfied with the speed. But as the assignment (transaction) backlog increases, we will need to further decrease the number of accountants. This will eventually lead us to rely on a few "trusted" accountants. A centralized group. This is against the ideology of blockchain decentralization. It's much easier to compromise/corrupt a smaller group of nodes than the entire company (the entire network). As a result, we sacrifice security in an effort to scale.

What is "Sharding"?

Can we have a system that has sufficient number of nodes to still maintain the security – while being small enough to increase the speed at which your reports (transactions) are returned (throughput of the network)? Essentially, we are conceding that we can't "max-out" on all three of the attributes: Scalability, Security, Decentralization.

Can we have just "enough" decentralization & security so as to achieve more scalability?

Our global financial system is highly centralized making it resistant to change, vulnerable to failures and attacks, and inaccessible to billions of people in need of basic nancial tools. On the other hand, decentralization poses new challenges of ensuring a consistent view among a group of mutually-distrusting participants. The permissionless mode of operation, which allows open membership and entails constant churn (i.e., join/leave) of the participants of the decentralized system, further complicates this task. Furthermore, any agile financial system, including a decentralized one, should be able to adequately serve realistic market loads. This implies that it should scale easily to a large number of participants, and it should handle a high throughput of transactions with relatively low delays in making their outputs available. Achieving these properties together should also not require sign cant resources from each of the participants since otherwise, it runs contrary to the idea of constructing a tool easily accessible to anyone.
Sharding is a fancy way of saying, "let's break down the network into smaller groups & pieces". Each group is a shard. A shard consists of nodes and transactions. So in our accountant analogy, a shard would consist of a group of accountants and assignments. Now, instead of an accountant having to check the reports across the entire network, he would be only responsible for the assignments within his shard(group). This greatly reduces the number of transactions (assignments) each node (accountant) has to validate.

Bitcoin Scalability Today: A Reality Check We analyze some of the key metrics of the Bitcoin system as it exists today. Maximum throughput. The maximum throughput is the maximum rate at which the blockchain can confirm transactions. Today, Bitcoin's maximum throughput is 3.3–7 transactions/sec. This number is constrained by the maximum block size and the inter-block time. Latency. Time for a transaction to confirm. A transaction is considered confirmed when it is included in a block, roughly 10 minutes in expectation.

Some blockchains, like the Bitcoin blockchain, are public; this means that anyone can purchase equipment, connect to the network, and start "mining". Others, however, require participants in the consensus process to fulfill some predetermined requirements set forth by the founding core developers.

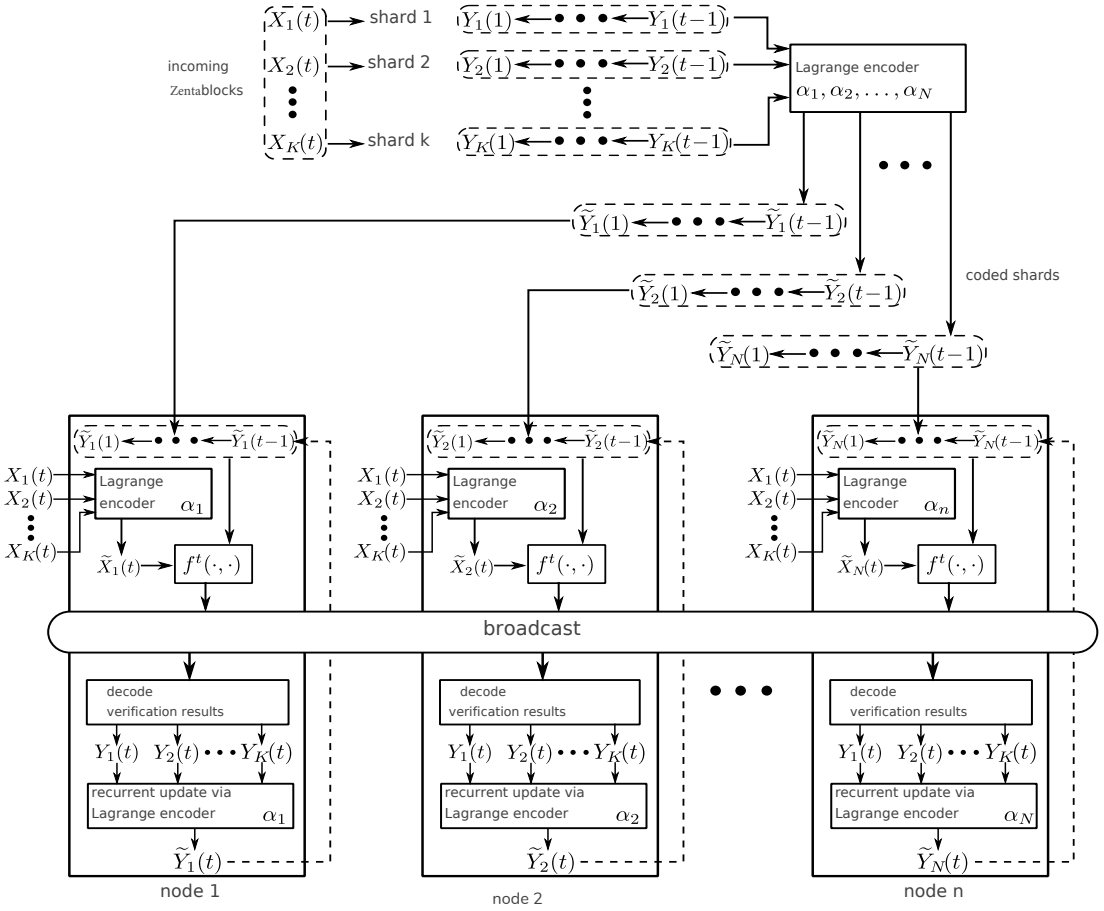
• Proof-of-Work: The original blockchain, the Bitcoin blockchain, uses a procedure called Proof-of-Work (PoW) to validate transactions and create new blocks. PoW was originally intended as an economic measure to deter distributed denial-of-service attacks (DDoS) and other service abuses, such as spam, on a network. It achieves this by requiring that some "work" from the service requester to be performed, which usually requires processing time by a computer. For example, on the Bitcoin blockchain the "work" is to calculate a valid double SHA256 hash. The "work" that the service requester is obligated to perform requires expenditure of capital-intensive resources, such as processing power, which raises the opportunity cost of engaging in fraudulent or unethical behavior. However, this presents a trade-off between data security and wasting capital resources. Therefore, the work should be neither be too hard nor too easy. If the work required is too hard, it creates inefficiency since the work could be easier and still yield the same results. However, if the work required is too easy, the system fails to serve its purpose, resulting in inefficient use of resources. Currently, all public blockchains rely on some form of Proof-of-Work validation and some type of consensus process. However, the main problem with PoW is that it is an incredibly energy intensive process. Many potential solutions have been set forth that are aimed at reducing the energy requirements in maintaining the blockchain. The most promising replacement for PoW is the concept of Proof-of-Stake

• Proof-of-Stake: Proof-of-Stake (PoS) is an algorithm to achieve distributed consensus within blockchain networks. Proof-of-Stake has been proposed as a potential replacement for Proof-of-Work, and is intended to solve the problem of inefficient use of capital resources, such as computing power and energy. The basic idea of Proof-ofStake is to allocate mining privileges based on how much "stake" a member has in the network. Many different versions of Proof-of-Stake have been proposed, and although opinions may differ on how to achieve optimal implementation, the core principle remains the same. The simplest version of Proof-of-Stake is delegating mining privileges based on ownership of the blockchain's native digital currency. Many different versions of the Proof-of-Stake system have been proposed, where mining privileges not only depend on ownership of currency but other factors as well. Such factors include frequency of transactions and how long a member has been part of the network.

We propose ZentaChain, a Byzantine-resilient public blockchain protocol that improves upon the scalability and security limitations of previous work in several ways. At a high level, ZentaChain partitions the set of nodes into multiple smaller groups of nodes that operate in parallel on disjoint blocks of transactions. Such a partitioning of operations and/or data among multiple groups of nodes are often referred to as sharding and has been recently studied in the context of blockchain protocols. By enabling parallelization of the consensus work and storage, sharding-based consensus can scale the throughput of the system proportional to the number of committees, unlike the basic Satoshi Nakamoto consensus. Let n denote the number of participants in the protocol at any given time, and m = n denote the size of each ZentaChain creates c = n/m committees each of size m = c logn nodes, where c is a constant depending only on the security parameter . Sharding-Based Consensus Unlike Bitcoin, a sharding-based blockchain protocol can increase its transaction processing power with the number of participants joining the network by allowing multiple committees of nodes process incoming transactions in parallel. Thus, the total number of transaction processed in each consensus round by the entire protocol is multiplied by the number of committees. While there are multiple exciting, parallel work on sharding-based blockchain protocols such as we only study results that focus on handling sharding in the Bitcoin transaction model.

Zentashard Structure

In each shard/group, we have nodes that are assigned as “Collators”. Collators are tasked with gathering mini-descriptions of transactions & the current state of the shard. In our analogy, you can think of Collators as Accountant Assistants. All the AA’s in shard/group do the first run through of all the assignments within the shard. Finally, we have super-nodes. Each super-node receives the collations created by the collators of each shard. They then processes the transactions within those collations. Furthermore, they maintain the full-description/state data of all the shards – which they get from the collators as well.



of decoding a Reed-Solomon code with dimension (K – 1)d+1 and length N (see, e.g., [38]). In order for this decoding to be robust to μN malicious nodes (i.e., achieving the security $\beta_{\text{Zentashard}}$ = μN), we must have μN ≤ (N –(K –1)d)/2. In
other words, a node can successfully decode $f^t(u_t(z), u^{t-1}(z))$ only if the number of shards K is upper bounded as $K \leq \frac{(1-2\mu)N}{d} + 1$.

$$\tilde{Y}_i(t) = \sum^{K} \ell_{ik} z_k^t X_k(t) = \sum^{K} \ell_{ik} Y_k(t),$$

Updating the sub-chains has the same computational complexity with the block encoding step, which is O(NK).

Since the set of coefficients ℓ_{ik}s in are identical to those in appending a coded block to a coded sub-chain is equivalent to appending uncoded blocks to the uncoded sub-chains, and then encoding from the updated sub-chains. This commutativity between sub-chain growth and storage encoding allows each node to update its local sub-chain incrementally by accessing only the newly verified blocks instead of the entire block history.

The total number of operations during the verification and the storage update processes is O(NK) + N c(f^t) + O($N^2 \log^2 N \log\log N$), where the term O(NK) + O($N^2 \log^2 N \log\log N$) is the additional coding overhead compared with the uncoded sharding scheme. Since $K_{\text{zentashard}} \leq N$, the coding overhead reduces to O($N^2 \log^2 N \log\log N$).

We can see that since the complexities of the encoding and decoding operations of Zentashard do not scale with the coding overhead becomes irrelevant as the chain grows. The Zentashard scheme simultaneously achieves optimal scaling on security, storage efficiency, and throughput.

RESULTS

Zentashard detailed simulations to assess the performance of Zentashards in the payment blockchain system described. This system keeps records of all the balance transfers between clients, and verifies new blocks by comparing them with the sum of the previously verified blocks computing the verification function. More specifically, the system contains K shards, each managing M clients. At each time epoch t, one block of transactions is submitted to every shard k. We simulate this system over N nodes using the full replication, uncoded sharding, and Zentashards schemes respectively. We measure the throughput of each scheme under different values of N and t to understand its scalability. Throughput is defined as the number of blocks verified per time unit, and is measured by dividing K (the number of blocks generated per epoch) by the average verification time (to be measured) of the N nodes. For Zentashard, the verification time also includes the time each node spent on encoding the blocks. However, since the encoding time is a constant, whilst the balance summation time increases with t as the chain becomes longer, it is it is expected that the encoding time is becoming negligible. We note that the storage efficiency and security level of each scheme are decided by system parameters and, thus, do not need measurements. We simulate this system for t = 1000 epochs, using different number of shards K. Each shard manages M = 2000 clients. We fix the ratio N/K= 3. Thus, the number of nodes is N. We plot the complete relation between N, t, and throughput of the three schemes in Fig.. For a closer look, we plot the relation between t and throughput when N = 150 in Fig. 5, and the relation between N and throughput when t = 1000 in Fig. 2 in Section.

$$\text{Zentashard} = \liminf_{t\to\infty} \frac{K_{\text{Zentashard}} N c(f^t)}{N c(f^t) + O(N^2 \log^2 N \log\log N)}.$$

Storage and security of the three schemes under different network size N.

Storage efficiency

N	15	30	60	90	120	200
γ_{full}	1	1	1	1	1	1
γ_{sharding}	5	10	20	30	40	60
zentashard	5	10	20	30	40	60

Security

N	15	30	60	90	120	200
β_{full}	7	15	30	45	60	100
β_{sharding}	1	1	1	1	1	1
zentashard	5	10	20	30	40	100

18.Zentalk

Zentalk is a distributed and peer to peer messaging service, designed to completely alter the way consumers currently think about messaging services.

Zentalk will be hosted on the ZentaChain platform where the service is aiming to be the most secure and private messengers available.

Zentalk achieves its privacy and security through the integration of Mesh Networking technologies. Mesh Networking (MeshNet) is known as one of the safest and most reliable variations of networking available. MeshNet technology is powerful, has excellent load distribution and contains zero central administration.

A mesh network is a network topology in which each node relays data for the network.

ZentaChain implements .cjdns- nodes an encrypted IPv6 network using public-key cryptography for address allocation and a distributed hash table for routing. This provides near-zero-configuration networking, and prevents many of the security and scalability issues that plague existing networks. All mesh nodes cooperate in the distribution of data in the network. Devices that are using Zentalk act as nodes on the MeshNet. These nodes have the unique ability to interconnect with one another in a distributed fashion.

Mesh networks can relay messages using either a flooding technique or a routing technique. With routing, the message is propagated along a path by hopping from node to node until it reaches its destination. To ensure all its paths' availability, the network must allow for continuous connections and must reconfigure itself around broken paths, using self-healing algorithms such as Shortest Path Bridging. Self-healing allows a routing-based network to operate when a node breaks down or when a connection becomes unreliable. As a result, the network is typically quite reliable, as there is often more than one path between a source and a destination in the network. Although mostly used in wireless situations, this concept can also apply to wired networks and to software interaction.

An example of common mesh network technology used in everyday life would be wireless domotica (like the Z-wave protocol). When you register a new node in the house, let’s say a new light bulb, the device pairs with the control center through a self-configured mesh network. Each new device is a new node in the mesh, relaying the data communication. Mesh networks are typically wireless - however ZentaChain Meshnet is about the blockchain based network topology and less aibout infrastructure.

Zentalk achieves technological darkness, this means no metadata can be discovered about its users or their messages. This is achieved through the integration of the MeshNet and its architecture. Zentalk pushes, drags, and tunnels all messages and data through the MeshNet. This insures that any messages shared between the sender and recipient have the highest levels of privacy.

In a meshed network, each network node is connected to one or more nodes. When multiple nodes are interconnected, this is known as a fully meshed network. When a message is sent from Zentalk, the the data is sent through the MeshNet and is passed from one node to the next, until the message has reached the desired recipient. By design nodes in the MeshNet don’t know which node sent which message or exactly which node receives said message. This leads to total anonymity, for or sender and recipient. The sender uses 1 node to get to it’s receiver. This one node will be rewarded in ZENTA to deliver the encrypted message .

19.Zentamesh

The ZentameshNet has self healing properties that contribute to the ability to achieve censorship resistance. Self-healing means if a node connection is blocked or disabled, the network mesh can patch and rerouted around the lost node. The data is redirected and the network is still functional. Meshed networks can be applied to both wired and wireless networks, as well as the Zentalk will establish a meshed WLAN (Wireless Local Access Network). This MWLAN is achieved through the use of a nodes Meshed WiFi. This will be required for offline communications.

This means that the ZentaChain token holder has the ability to establish and share a network uplink (online to offline), as well as establish communication with devices with no internet access (offline to offline). This can be accomplished with little to no direct infrastructure. Relaying data will be rewarded with Zenta tokens.

Nodes are active network components like cell phones, routers, switches, bridges and gateways.

A node in a network is a connection point. This can either be a redistribution point or an endpoint in the data transfer. It has the feature to discover, process, and forward transmissions to other network nodes. A network node has at least two, but usually more connections to other network elements.

On Zentalk, the role of a node will be the connected device with the running peer-to-peer messenger dApp on a mobile phone, tablet or a computer.

3 Reasons for using Mesh technology:

1. **Network stability**

The data in the (wireless) mesh network can be forwarded through various nodes If one or some of the nodes are faulty or disturbed, the data is routed through other nodes on the ecosystem.

2. **High bandwidth**

Mesh networks are designed to follow most optimal (dynamic) routes, allowing a higher bandwidth. With the increase in the number of nodes and the Number of possible paths, the overall bandwidth is greatly increased.

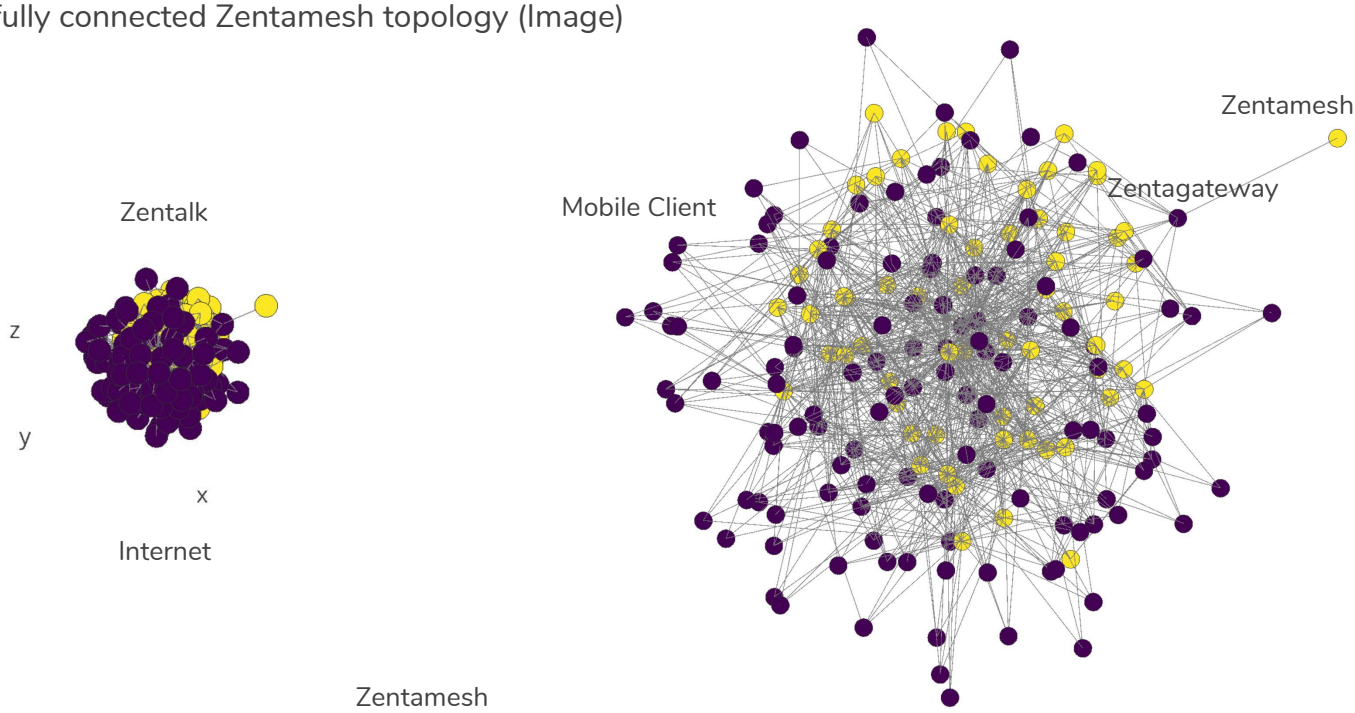
3. **Safety and Security**

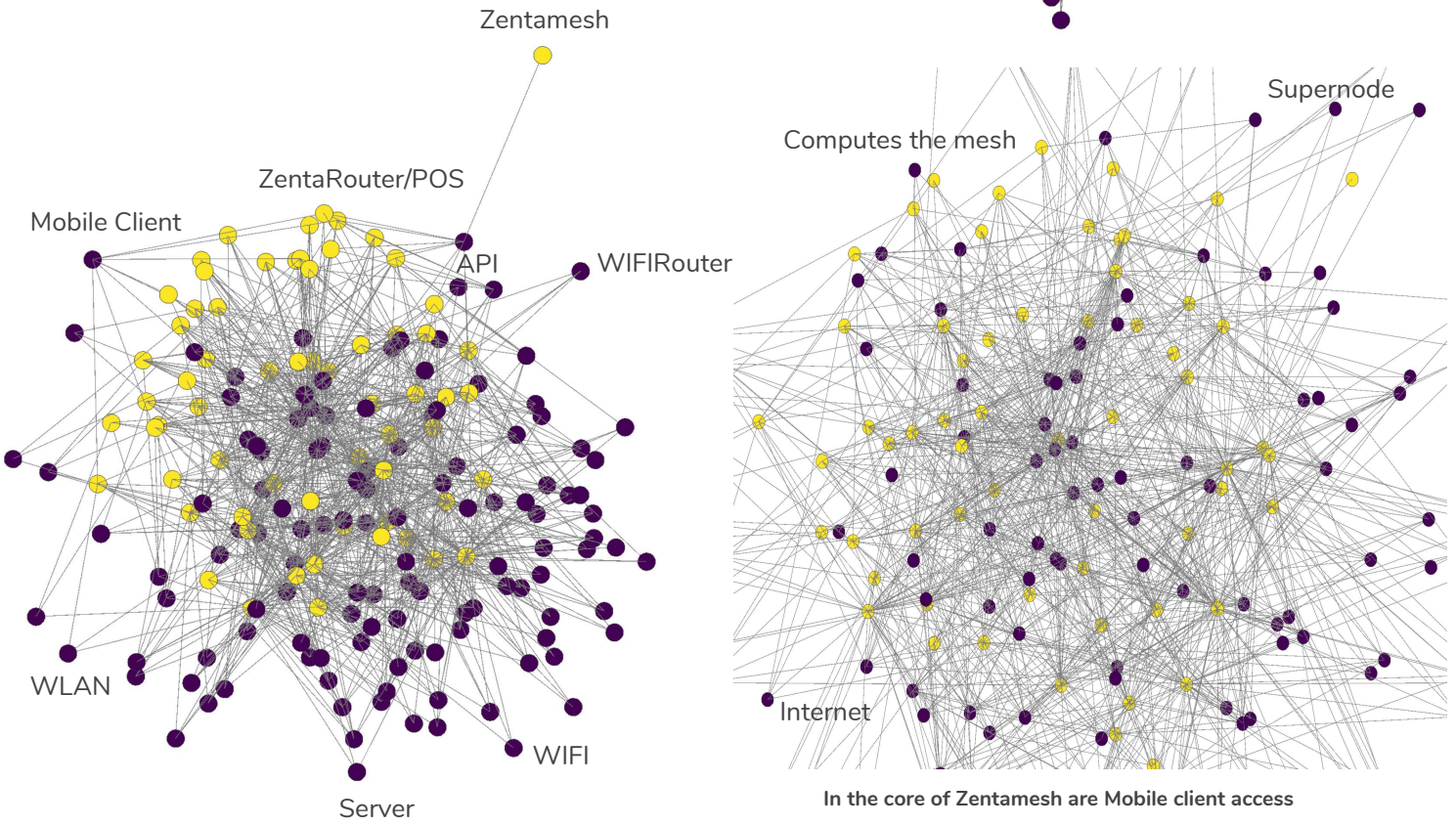
Compared with the single-hop mechanism of WLAN, the multi-hop mechanism of a Mesh network determines that the user communication needs to go through several nodes.

4. **Zentalk & Hyperboria**

ZentaChain looks to address issues of privacy and online security through a decentralized alternative to the Internet called Hyperboria. Hyperboria is a Mesh Network which utilizes the cjdns protocol to ensure complete end-to-end encryption. This is achieved by using public-key cryptography for address allocation and a distributed hash table for routing. Communication between two nodes can only be established after the connection has been verified, eliminating the possibility of third party intrusion or eavesdropping. Routing is executed using a system similar to Kademlia DHT enabling the network to maintain an optimal load across all nodes. ZentaChain believes in the Open Internet. By employing Hyperboria we give our users the opportunity to transmit their data in a fast, secure and private way, without having to fear unwanted intrusion from malicious entities or centralized service providers.

A fully connected Zentamesh topology (Image)





In the core of Zentamesh are Mobile client access

Even in the partially connected mesh, each device can communicate with each other.

Q:Why should you be passionate about mesh networks?

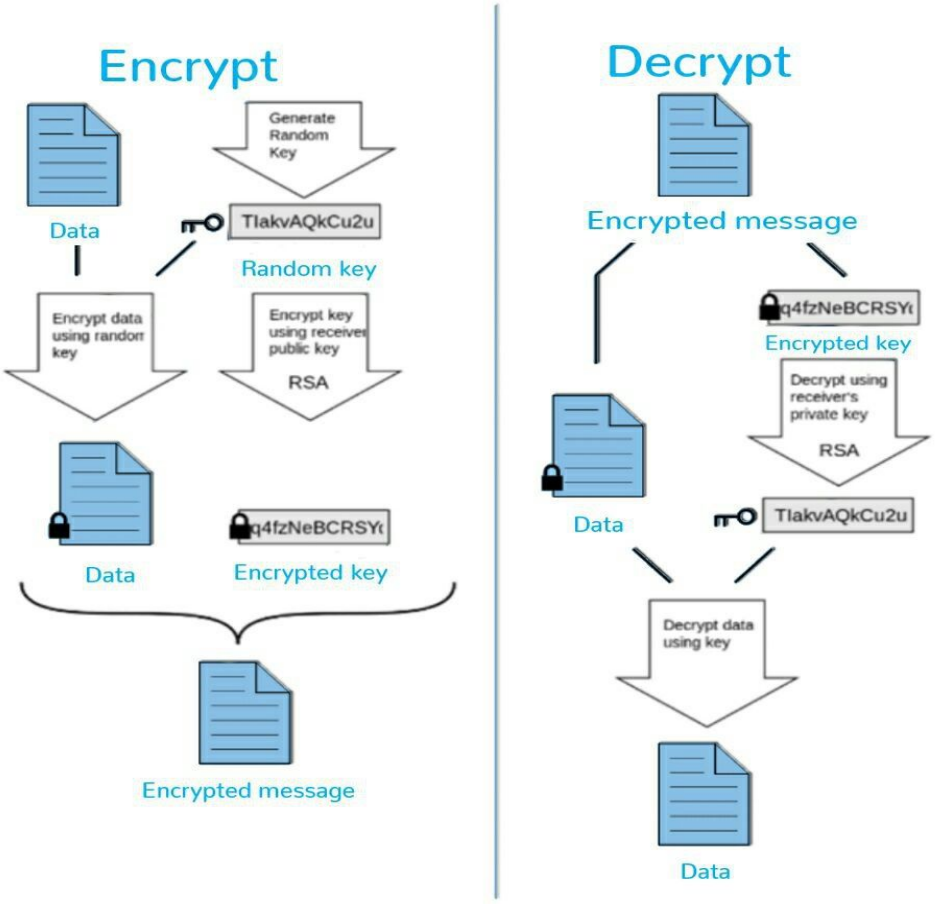
A:The main reason for exciting mesh networks is that no central infrastructure is required.

The privacy on a meshnet is guaranteed by „End-to-end encryption“

With “**End-to-End encryption**”(E2EE),the data is encrypted on the sender's system or device and only he recipient can decrypt it. No one in between is able to ead or manipulate the data. It doesn't matter if it is an ISP(Internet Service Provider),Application Service Provider or a malicious hacker.

The encryption and decryption thus takes place only at the end points of the transmission.The cryptographic keys used for encryption and decryption are stored exclusively on the terminals.

Therefore, the security of this type of encryption is very high, because without the secret key, no ciphertext can be decrypted.



Another advantages of the “End-to-end encryption”: **More security and authenticity**. E2EE can be combined with digital signature. A digitally signed and encrypted message proves, that the sender is indeed the 'real' sender of the message. It also ensures that the message has not been tampered with during transmission. NO mass surveillance. End-to-end encryption protects your messages from mass surveillance.

20.ZentaVault

Zentavault is the second dApp to be released from the ZentaChain application pipeline. Zentavault is a high through-put dApp, which is designed to be a highly encrypted and distributed storage service.This service will be hosted on the ZentaChain platform. ZentaChains dApps will never rely on centralized systems. And will absolutely never have any backup databases for users metadata. By doing this, Zentavault will exceed in confidentiality, anonymity and commercial performance.

Zentavault requires no monthly usage fees, instead there is only a small transaction fee to upload data. Zentavault is a file encrypting and distribution vehicle. Putting the user behind the wheel and in control, to encrypt, store and share content the way they choose.

With Zentavault you can ensure your data is encrypted and embedded permanently onto the InterPlanetary File System, also referred to as IPFS. It is a tailored network that allows for content to be embedded and shared using an associative memory strategy.

When content is on IPFS it is assigned a unique identifier, known as a cryptographic hash or hash Id. This can be used to locate the users data or share it between two parties. Once your content is embedded onto IPFS, a hash Id is assigned to the file allowing a way to find or share content with others.

By employing a peer-to-peer hypermedia protocol technologies such as IPFS, Zentavault is able to achieve a more expeditious, protected and accessible file storage and transfer service.

IPFS (InterPlanetary File System)

The modern internet, although one of the breakthrough technologies of our age, has shown to have limitations since its inception in the 1990s. As technological advances progress, more and more tech is shown to be in need of either upgrading or even complete conceptual reimagining. Such is the case with the HTTP protocol.

In recent times we have seen an increased demand for solutions that would address the issues of privacy, security and speed, the areas in which the HTTP protocol hasn't shown to be "up to the task", so to speak. Fortunately, there has been an influx of ideas giving answers to said problems, one of the most viable being the IPFS.

But how does it work?

We can look at it in the terms of a BitTorrent swarm but with the ability to store and track file versions over time. Unlike the HTTP which works by mapping the resources via location-based IP addresses, IPFS uses a content-addressed system. This decentralized system stores files across peers and enables access to them via a cryptographic hash on a file which is used as the address. This means that the user becomes the client and the host at the same time.

It is made possible by the Merkle DAG (Directed Acyclic Graphs) data architecture and ensures oimmutability and content versioning on IPFS. Because of their similar structures, IPFS is a perfect fit for blockchain integration. It goes a bit further than that, though, solving blockchain's nagging issue of data storage and together with blockchain creates a solution for storing, encrypting, and sharing large data and files. Although still in its infancy, IPFS has all the tools to become the successor to HTTP and usher in a new era of the World Wide Web.

IPFS Identities

Nodes are identified by a NodeId, the cryptographic hash3 of a public-key, created with S/ Kademlia's static crypto puzzle [1]. Nodes store their public and private keys (encrypted with a passphrase). Users are free to instatiate a "new" node identity on every launch, though that loses accrued network benefits. Nodes are incentivized to remain the same.

IPFS Network

IPFS nodes communicate regualarly with hundreds of other nodes in the network, potentially across the wide internet. The IPFS network stack features:

- Transport: IPFS can use any transport protocol, and is best suited for WebRTC DataChannels (for browser connectivity) or uTP(LEDBAT).
- Reliability: IPFS can provide reliability if underlying networks do not provide it, using uTP (LEDBAT) or SCTP.
- Connectivity: IPFS also uses the ICE NAT traversal techniques .
- Integrity: optionally checks integrity of messages using a hash checksum.
- Authenticity: optionally checks authenticity of messages using HMAC with sender's public key.

IPFS Routing

For routing IPFS uses Distributed Sloppy Hash Tables based on S/Kademlia and Coral.
It's purpose is to:

1. Announce data being added to the nodes
2. Locate data requested by specific nodes

Data equal or lesser than 1KB in size is stored directly on the DHT. For data larger than 1KB, DHT stores references, which are the NodeIds of peers who can serve the block

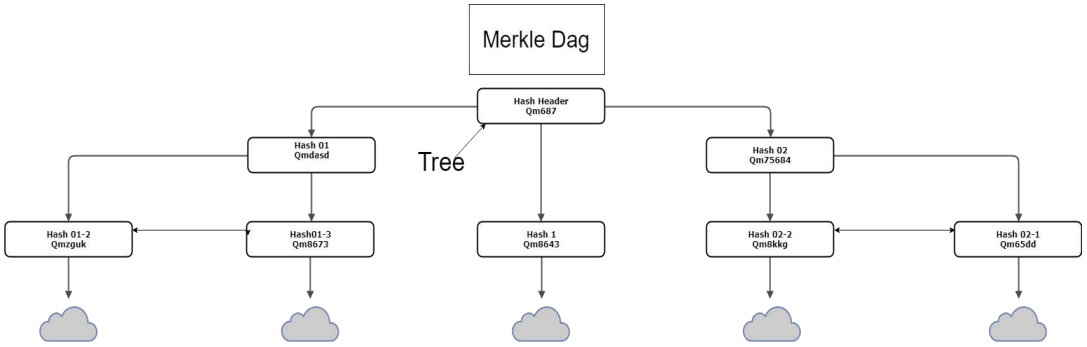
Objects Merkle DAG

Merkle DAG (Merkle Directed Acyclic Graph) is used to keep order of object files in the InterPlanetary File System. Merkle DAG allows Files to become linked to each other by their unique cryptographic hash, which is known as a hashid. The hashid (object) includes all hashid object links. Merkle DAG gives IPFS the useful properties such as:

• Content safeguard: Merkel DAG insurers the security, safety and integrity of all content on ipfs. If any object that stored or hosted on ipfs has been tampered with or otherwise corrupted, Merkle DAG changes the root hash automatically. This shows the changes made to the File.

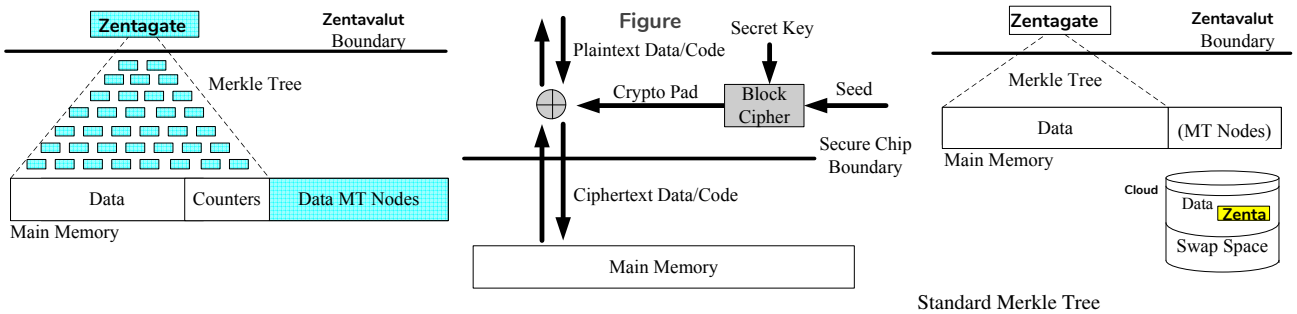
• Anti-duplication for efficiency: All objects holding content on IPFS is sorted through, duplicated objects are recognized then deleted. This insures that content isn't stored multiple times on IPFS.

• Content addressing: All content is able to be located by identifying the unique hashid or multi hash including links.



Using Address Independent Seed Encryption

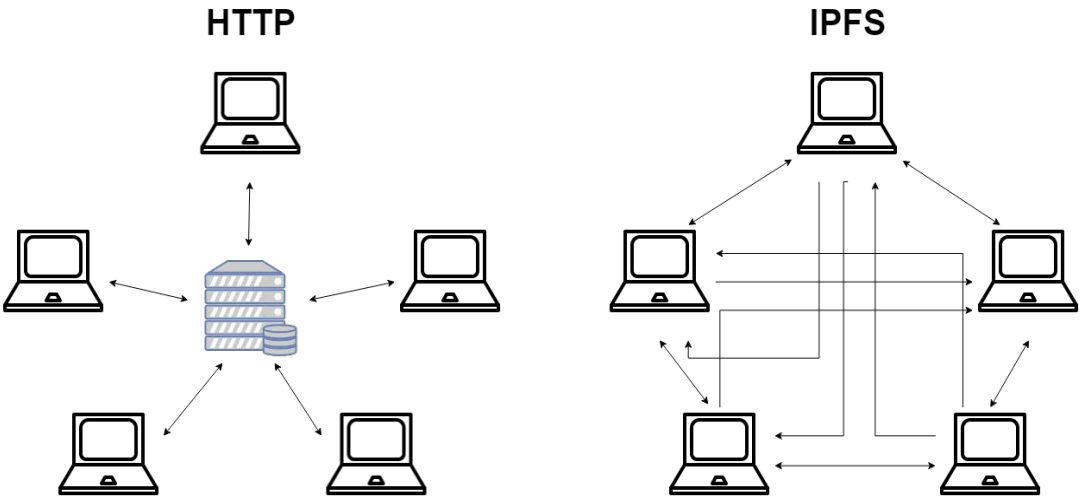
The goal of memory encryption is to ensure that all data and code stored outside the secure processor boundary is in an unintel-ligible form, not revealing anything about the actual values stored. Figure illustrates how this is achieved in counter-mode encryp-tion. When a block is being written back to memory, a seed is en-crypted using a block cipher (e.g. AES) and a secret key, known only to the processor. The encrypted seed is called a cryptographic pad, and this pad is combined with the plaintext block via a bit-wise XOR operation to generate the ciphertext of the block before the block can be written to memory. Likewise, when a ciphertext block is fetched from memory, the same seed is encrypted to gen-erate the same pad that was used to encrypt the block. When the block arrives on-chip, another bitwise XOR with the pad restores the block to its original plaintext form. Mathematically, if P is the plaintext, C is the ciphertext, E is the block cipher function, and K is the secret key, the encryption performs C = P EK(Seed). By XORing both sides with EK(Seed), the decryption yields the plaintext P = C EK(Seed).



IPFS Files

IPFS also defines a set of objects for modeling a versioned filesystem on top of the Merkle DAG. This object model is similar to Git's:

1. block: a variable-size block of data.
2. list: a collection of blocks or other lists.
3. tree: a collection of blocks, lists, or other trees.
4. commit: a snapshot in the version history of a tree.



21.Why do we need IPFS ?

Bandwidth

Only being able to access data from one central location can have disadvantages. Imagine you want to share a file with a room full of people. You would then upload that file to a central server that is probably located somewhere far away from you (backbone of the internet) and be routed over several servers on the way. The other people would then access this file by again connecting to this far away server and retrieving the file. Especially with big files like pictures and videos this can lead to a lot of bandwidth being used for sharing that file with a room full of people that are probably all connected to the same local area network as you are.

With IPFS that file can be served directly by all the computers in the room that have the file through the local area network and thus use a lot less bandwidth because the detour to the central server is no longer needed. This becomes particularly important when looking at the cost of connection speed that is decreasing much slower than the cost of storage. If this trend continues, users will be able to store a lot more data and thus will increase their use of the network. But with the bandwidth not improving at the same pace, the connection speed will appear to get slower and slower.Security also increases — DDos attacks, for example, wouldn't work, since they rely on attacking a central distribution system, which IPFS doesn't have. Speed is another factor that increases. In a distributed web, every node that requests something, requests it to the node closest to him, instead of to a single, central location.

Latency

A related problem is latency. As the speed of light is a constant and cannot be changed, the only way of reducing latency is serving the data from a point that is closer to the user. That is why the big cloud service providers now started offering storage locations by region. IPFS aims to reduce the distance to the computer that serves the requested data if possible.

Being Offline

A lot of the services we use every day and rely on heavily only work when we are online. If you want to work collaboratively on a document with other people in the same room or want to transfer data from your phone to your laptop you are mostly only able to do that if you are connected to the central servers of the backbone of the internet. If there are bandwidth or infrastructure problems like congestion, ISP outage or datacenter problems you are not able to use these services anymore. IPFS hopes to change that by letting you connect to other peers directly without needing to connect to these central servers.

Censoring

Access to data or services can be censored or restricted a lot easier if everything is stored and run on central servers compared to a P2P network. One example of this is when the government of Egypt cut all access to the Internet during the Arab spring in 2011 to prevent the organization of and restrict communication between the protesters. Through being connected P2P IPFS hopes to make this form of censorship impossible.

Permanence

Everyone has encountered an Error 404 before. This means that the required content could not be found because it was deleted or moved. This can be a huge problem if you want to link to this content for example because it is essential for the content you are providing. In general it would be beneficial for society if most of the knowledge that is accumulated in the web would stay accessible and not be deleted because someone purposely or accidently shut down some website. With IPFS you are able to save and host some version of the linked content yourself and that way ensure that this content will always be available to users even though the original hosts no longer host it. The ultimate idea is to create a permanent web where no content is ever lost because all content will be hosted by a number of people that find it valuable.

Security

As the numerous hacks in the recent years have shown, only thinking about the security in the communication between servers and clients is not sufficient. IPFS aims to protect and armor the data itself through enhanced methods of authentication and encryption.

Self-Certified Filesystems - SFS

No secure network file system has ever grown to span the Internet. Existing systems all lack adequate key management for security at a global scale. Given the diversity of the Internet, any particular mechanism a file system employs to manage keys will fail to support many types of use. We propose separating key management from file system security, letting the world share a single global file system no matter how individuals manage keys. We present SFS, a secure file system that avoids internal key management. While other file systems need key management to map file names to encryption keys, SFS file names effectively contain public keys, making them self-certifying pathnames. Key management in SFS occurs outside of the file system, in whatever procedure users choose to generate file names. The Self-certifying File System (SFS)[8] addresses the issue of key management in cryptographic filesystems and proposes separating key management from file system security.

Servers have a public key and clients use the server public key to authenticate the server and establish a secure communication channel. To allow clients to authenticate servers on the spot without even having heard of them before, SFS introduces the concept of a `` self-certifying pathname.''

A self-certifying pathname contains the hash of the public-key of the server, so that the client can verify that he is actually talking to the legitimate server. Once the client has verified the server a secure channel is established and the actual file access takes place. Remote SFS file systems are accessed through the /sfs mount point. An SFS pathname obeys the following syntax: /sfs/ location:hostid/real/pathname, where `` location'' is the name (IP address or DNS Name) of the server exporting the file system and `` hostid'' is the hash of a string containing the server's public key and some other information. SFS does not care on how the pathname has been obtained by the user; a user can eventually obtain hostid's using an existing PKI (Public Key Infrastructure). On the other hand, once a self-certifying pathname for the files he is interested in has been obtained, users do not need to remember any key.

This is used to implement the IPNS name system for IPFS. It allows us to generate an address for a remote filesystem, where the user can verify the validity of the address. SFS introduced a technique for building Self-Certified Filesystems: addressing remote filesystems using the following scheme:

```
/sfs/<Location>:<HostID>

where Location is the server network address, and:

HostID = hash(public_key || Location)

Thus the name of an SFS file system certifies its server.
```

It is designed to function like the web already functions, as you can read above. Instead of a special link that only certain programs understand, or a file to download other files, IPFS is designed to work with common links that work in the browser, and you don't need to install special software.
IPFS can use any network architecture and any kind of files can be used as a DAG.
This feature is called IPLD (InterPlanetary Linked Data).

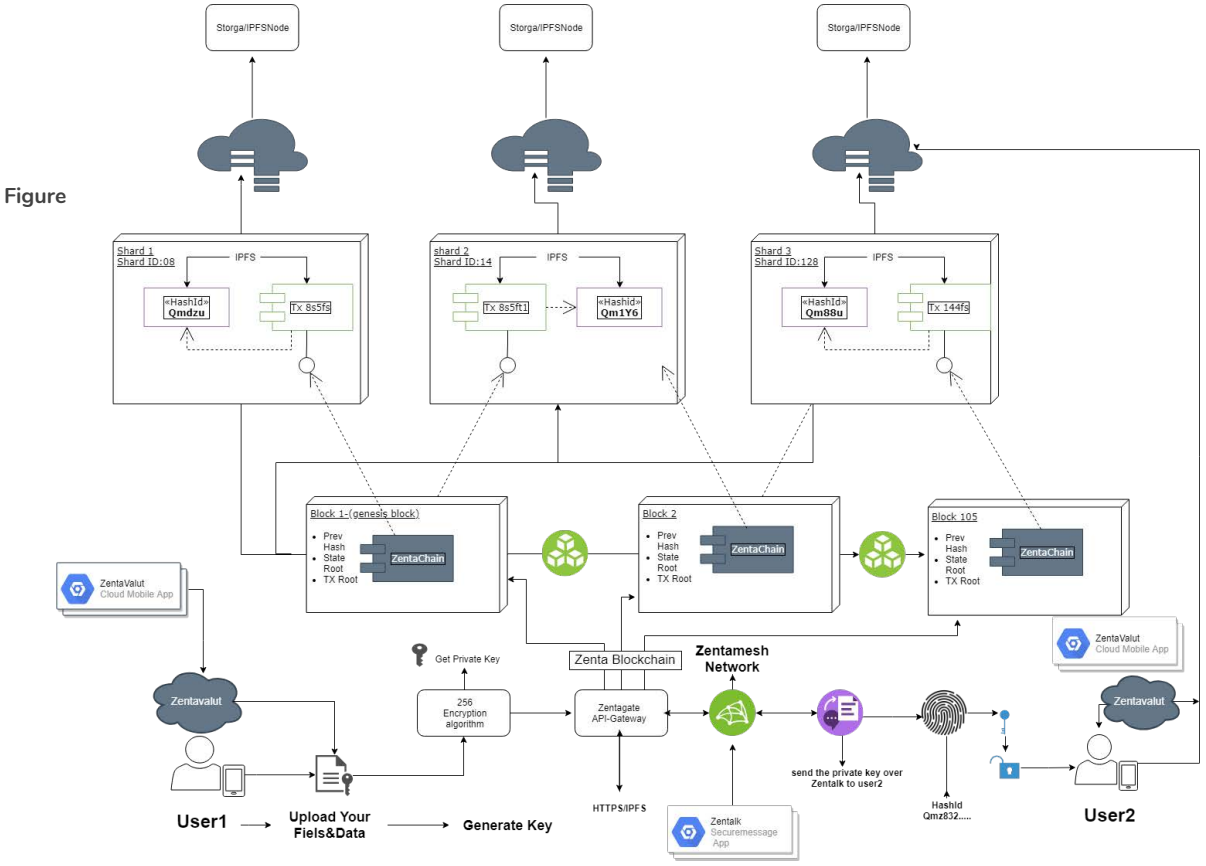
22.Zentagate

Zentagate is a gateway that allows you to easily upload and access content stored on IPFS, through our dapp zentavault. Zentagate itself is a fixed gateway, in which machines are connected to IPFS. These Machines which are linked to Zentagate are used as part of a distributed swarm, which helps with facilitating the publishing and fetching of files in the swarm. Zentagate also has a Javascript API allowing front end Dapps to interact with IPFS directly, without a server. This will make it possible for Dapps like Zentavault to add content directly to ipfs. Zentavault uses Zentagate to create static web sites used to host encrypted content that is served entirely over IPFS. This allows users to create secure content storage containing information that cannot be censored by governments,companies, or other organizations. For each file that ZentaChain's gateway requests on your behalf, it check the hash of the file to ensure that the content has not been modified in transit. Every file will added automaticliy over the zentagate encrypt to the IPFS is given an address derived from a hash of the file's content. That address is like a fingerprint. It belongs uniquely to that file and will be the same, no matter where the file is stored in IPFS.

Q:Do I need to keep my node up and running after I upload my files over ZentaValut?

A:No, all is done when you get the hash.(Qm.....)

Figure



23.References

IPFS

Sharding JDBC

Ethereum (Smart)

Hyperboria IpV6

Gateway-Blockchain

Meshnet Blockchain

Network Landscape

Sha256/516

API Application Cyber

Protect/P2P Cjdns

Protocol Whitepaper

Sharding Ethereum

Hyperboria/IPV6 W3C/

AES JavaScript APIs

rsaEncryption

RSAPrivateKey

ECMAScript

Blockchain shards

Polynomially code (PS)

Scaling Efficiency

Efficiency of Storage

Cjdns

Cjdns supernode

Hyperboria API

Seed Encryption

1. Mathis T. Ethereum: Your Guide To Understanding Ethereum, Blockchain,and Cryptocurrency [Internet]. Level Up Lifestyle Limited; 2018. Available: https://market.android.com/details?id=book-NzVODwAAQBAJ

2. Adams M. Ethereum: The Beginners Guide to Understanding Ethereum, Ether, Smart Contracts, Ethereum Mining, Ico, Cryptocurrency, Cryptocurrency Investing [Internet]. Createspace Independent Publishing Platform; 2018. Available: https://books.google.com/books/about/Ethereum.html?hl=&id=SH4atwEACAAJ

3. Blokdyk G. Erc20 Second Edition [Internet]. Createspace Independent Publishing Platform; 2018. Available: https://books.google.com/books/about/Erc20_Second_Edition.html?hl=&id=qEl HtAEACAAJ

4. Skvorc B, Kendel M, Attard D, Javor M, Jankov T, Ward C. A Developer's Guide to Ethereum [Internet]. SitePoint; 2018. Available: https://market.android.com/details?id=book-5RFqDwAAQBAJ

5. Dannen C. Advanced Concepts. Introducing Ethereum and Solidity. 2017. pp. 173–179. doi:10.1007/978-1-4842-2535-6_11

6. Li J, Wu J, Chen L. Block-secure: Blockchain based scheme for secure P2P cloud storage. Inf Sci . 2018;465: 219–231. doi:10.1016/j.ins.2018.06.071

7. Patil S. Preventing unauthorized Data Access in Fully Obscure Attribute Based Encryption with Security Solutions. International Journal for Research in Applied Science and Engineering Technology. 2018;6: 2950–2957. doi:10.22214/ijraset.2018.5481

8. Asarudeen SS, Sheik Asarudeen S, Mba FY, Srinivasan College of Arts &Science, Perambalur, Priya RPR. Recruitment of Ethical Hackers. J Appl Sci Res. 2012;2: 145–146. doi:10.15373/22778179/jan2013/50

9. Guo B. Why Hackers Become Crackers – An Analysis of Conflicts Faced by Hackers. Public Administration Research. 2016;5: 29. doi:10.5539/par.v5n1p29

10. Fischetti M. Data Theft: Hackers Attack. Sci Am. 2011;305: 100–100. doi:10.1038/scientificamerican1011-100

11. Cornwall H. Data Theft: Computer Fraud, Industrial Espionage and Information Crime [Internet]. 1990. Available: https://books.google.com/books/about/Data_Theft.html?hl=&id=Db0lnQEACA AJ

12. Gilroy AA. Access to Broadband Networks: The Net Neutrality Debate [Internet]. DIANE Publishing; 2011. Available: https://books.google.com/books/about/Access_to_Broadband_Networks.html?hl=&id=IVGN5J28tuEC

13. Nelson D. The Problems of Data Ownership and Data Security. Science Trends. 2017; doi:10.31988/scitrends.3265

14. Tiemensma J, Biermasz NR, van der Mast RC, Wassenaar MJE, Middelkoop HAM, Pereira AM, et al. Increased psychopathology and maladaptive personality traits, but normal cognitive functioning, in patients after long-term cure of acromegaly. J Clin Endocrinol Metab. 2010;95: E392–402. doi: 10.1210/jc.2010-1253

15. Loukas G. Physical-Cyber Attacks. Cyber-Physical Attacks. 2015. pp. 221–253. doi:10.1016/b978-0-12-801290-1.00007-2

16. Rampling B, Dalan D. DNS For Dummies [Internet]. For Dummies; 2003. Available: https://books.google.com/books/about/DNS_For_Dummies.html?hl=&id=8EBjIp 8w5pwC

17. Reschke J. Initial Hypertext Transfer Protocol (HTTP) Method Registrations [Internet]. 2014. doi:10.17487/rfc7237

18. The New York Times Editorial Staff. Net Neutrality: Seeking a Free and Fair Internet [Internet]. The Rosen Publishing Group, Inc; 2018. Available: https://books.google.com/books/about/Net_Neutrality.html?hl=&id=g9xoDwAA QBAJ

19. Drescher D. Reinventing the Blockchain. Blockchain Basics. 2017. pp. 213–220. doi:10.1007/978-1-4842-2604-9_23

20. Solidity — Solidity 0.4.23 documentation. (2018). Solidity.readthedocs.io

21. Remix - Solidity IDE. (2018). Remix.ethereum.org. [Online]

22. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system

23. The GNU Privacy Guard. (2018). Gnupg.org

24. Blockchain Proof Engine | API. (2018)

25. Blockchain and Distributed Ledgers as Trusted

26. Recordkeeping Systems

27. Practical scheme for non-interactive verifiable secret sharing

28. Truffle Suite - Your Ethereum Swiss Army Knife. (2018). Truffle Suite. [Online]. Available:http://truffleframework.com

29. K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer et al., “On scaling decentralized blockchains,” in International Conference on Financial Cryptography and Data Security. Springer, 2016, pp. 106–125.

30. Y. Sompolinsky and A. Zohar, “Secure high-rate transaction processing in bitcoin,” in International Conference on Financial Cryptography and Data Security. Springer, 2015, pp. 507–527.

Thank you

Zentachain.io