NEWFANG

A Decentralized Cloud Storage Platform

V1.0

01 Aug 2019

Mayur Relekar

mayur@newfang.io

newfang.io

Introduction 2

Design 3

- Secure and Private 3
- Distributed 3
- Decentralized 6
- Fast 7
- Transparent 7
- Intuitive 8

Implementation 9

- Architecture 9
- Tahoe LAFS 10
- Client SDKs 11
- Storage Node App 12
- Smart Contracts 12
- Governance(WiP) 13

Business and Token Model 14

- Structure 14
- Business Model 14
- Token Model 14

References 16

1. Introduction

The Internet is the biggest working model of a decentralized and distributed network but everything built on it or for it is largely centralized. The data on the internet is especially centralized with large entities constantly vying to gain a larger piece of the data pie. Bitcoin has shown us how it is indeed possible to perhaps “break the wheel” by giving the world a base framework for building decentralized systems that put the power firmly back into the hands of users. All this while being secure, trustless and economically viable.

We at Newfang believe the future is decentralized and large parts of the internet as we know it today will be redesigned to be so. We’ve chosen to attack the data centralization problem because storage is not only ubiquitous and presents a significant market opportunity(~$90B by 2022[1]) but also because decentralization can significantly improve the quality of service.

In the centralized approach:

- Data breaches are fairly commonplace[2]. Our approach allows users to secure files even before they hit the network.
- Network outages are not as common but when they occur have a pretty damning effect[3]. Newfang steers clear of the standard data center approach in favour of a more reliable distributed architecture with configurable redundancies.
- Speed is another concern. A standard storage service with a CDN is the norm even for “cold” data downloads, increasing cost and effort for users. The inherent distributed nature of our storage clubbed with erasure coding techniques mean files can not only be stored reliably but can be retrieved faster, from multiple “seeds” asynchronously, a la BitTorrent.

Storage/Bandwidth costs are not dropping linearly with drop in hardware costs. We believe there is a significant opportunity to provide an even more economical service by leveraging the high availability of largely unorganised and smaller service providers at the edge of the network. Our non-reliance on maintaining hardware means we have the ability to scale massively to parts of the globe ignored by centralized players.

Our biggest insight, however, from years of having built several Apps and DApps is that centralized solutions are all too familiar and indeed very easy to use. Most decentralized solutions thus far are unwieldy or have poor user experience. Adoption of decentralized solutions has to be evolutionary and it is our mission to provide our users with the obvious benefits of decentralization and distributed architecture but above all, be very easy to use.

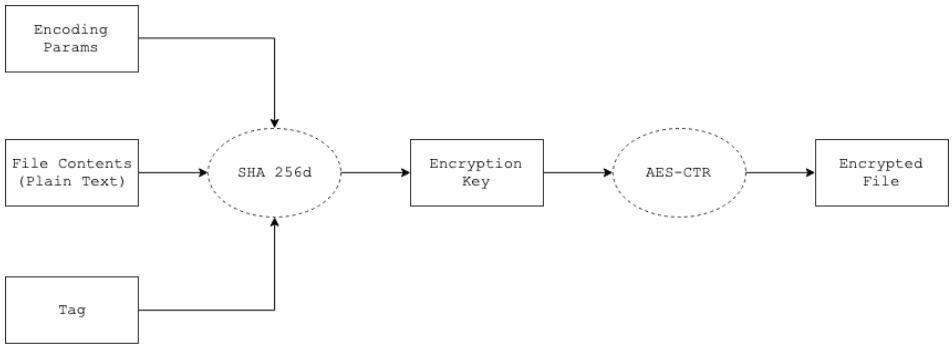
2. Design

2.1. Secure and Private

The Newfang Network is secure by default. All files are AES[4] encrypted with a secret-key on the client side. This secret-key is included in the file handler that is returned when any file is uploaded. This file-handler is a return value at the end of an upload call in the Newfang SDK which can be saved by the developer or passed on to the end user depending on the use case.

There is very little performance overhead as Encryption and Decryption of files takes up only a sliver of the total upload/download time of a file which is something that Newfang takes advantage of and encrypts all files. Also, since the secret-key is included in the file handler, which is used to retrieve the file, it is also very easy to share files that perhaps do not necessarily warrant security.

Encryption of files on the client side means that the independent storage nodes hosting the files(actually just pieces of an encrypted file) have no way of knowing the contents which ensures provider independent security and complete privacy of content.



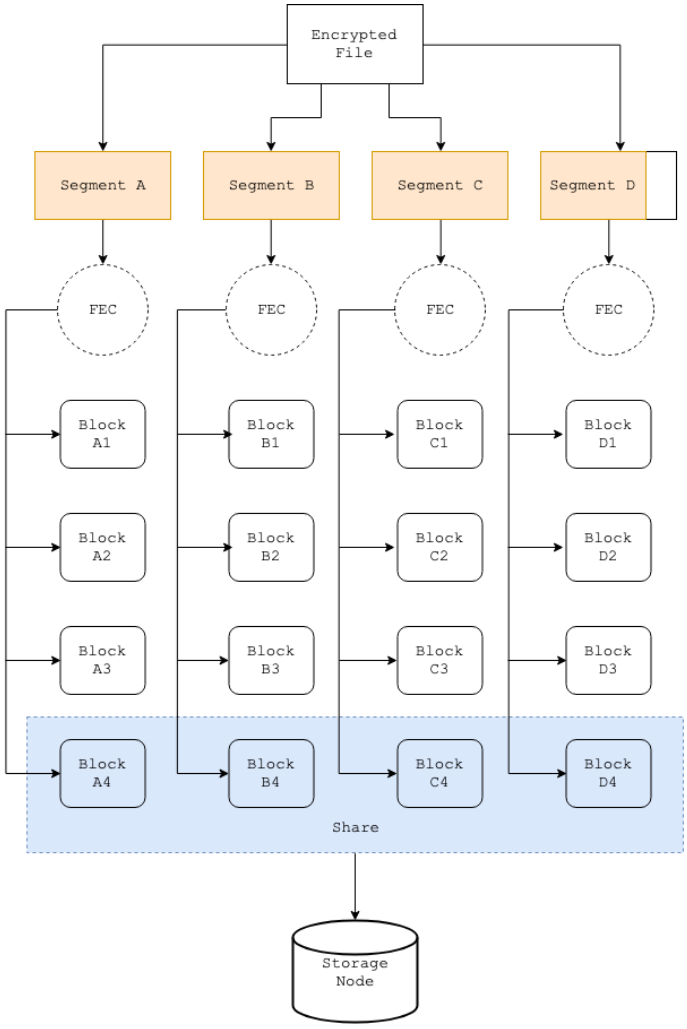
2.2. Distributed

Every file uploaded is erasure coded or undergoes “Forward Error Correction” (FEC). Essentially, each file, post encryption, is uploaded to the network and broken up into smaller pieces or “shares” using a method of data protection in which data is broken into fragments, expanded and codified with redundancies and stored across a selected set of unique storage nodes.

Encoding is per a specified configuration which describes a “K of N” scheme where N is the total number of shares created for a file and K is the minimum number of shares

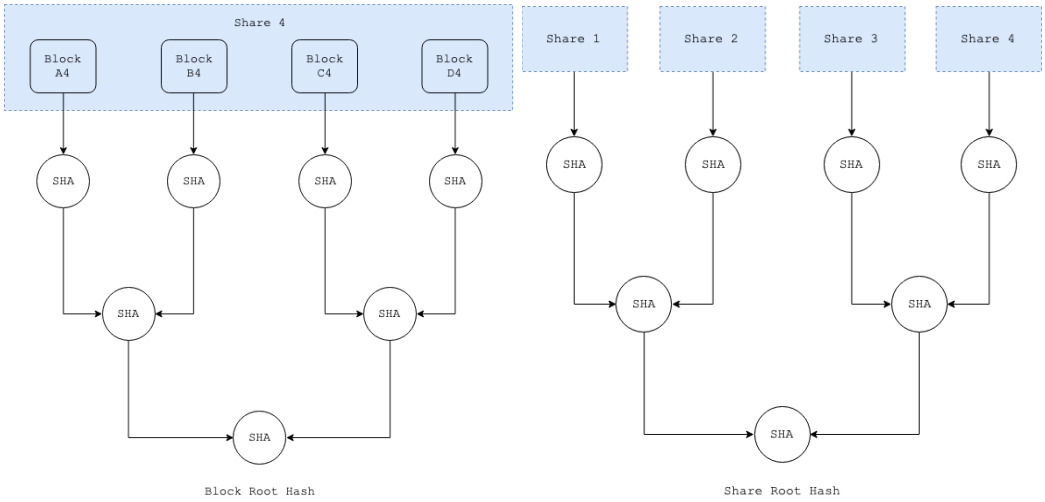
required to decode and retrieve the file. Newfang allows its users to set up this configuration for each of the apps they intend to use Newfang for, depending on the use-case of the said app. A higher N value could mean faster downloads but slower uploads, a K value closer to N could mean lower expansion costs but lower redundancy. Users are free to decide on the configuration their app would need by carefully evaluating their K and N selections.

To generate “shares”, the Encrypted File which is the input to the Encode step is broken into a number of segments where each segment is of a certain size barring the last segment which is in all likelihood smaller. Each segment is then put through an encoding step to produce blocks that are then grouped to produce “shares”.

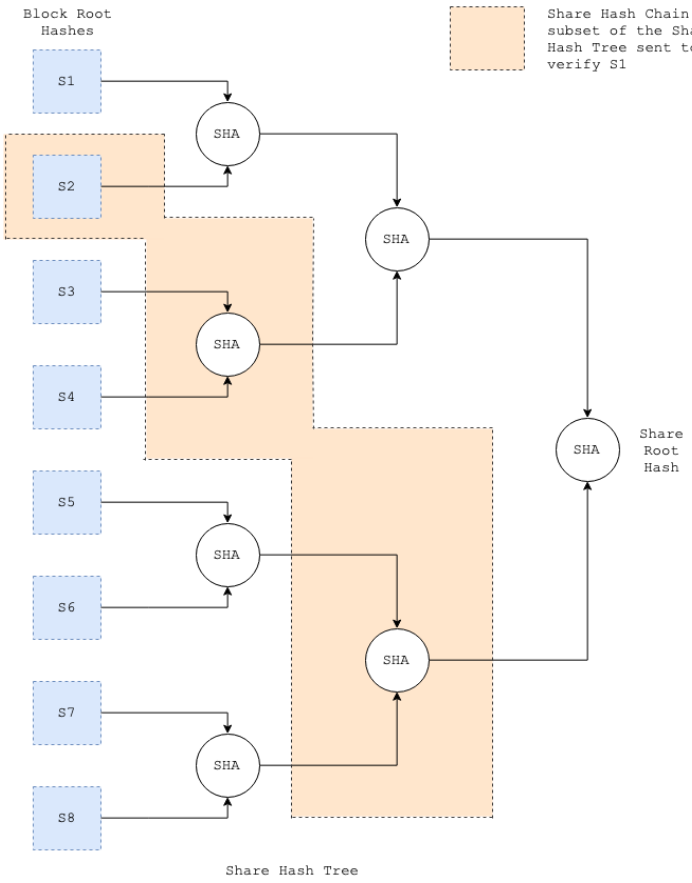


Additionally, every block, when generated, is also hashed and put into a merkle tree. The tip of this merkle tree is the Block Root Hash. Also, every Root Block Hash generated is

also further hashed and placed into another merkle tree whose tip is the Share Root Hash.



During upload, all blocks are sent first, followed by the block hash tree, followed by the share hash chain. During download, the share hash chain is delivered first, followed by the block root hash. The client SDK then uses the hash chain to validate the block root hash. Then the storage node delivers enough of the block hash tree to validate the first block, followed by the first block itself. The block hash chain is used to validate the block, then it is passed (along with the first block from several other peers) into decoding, to produce the first segment of the encrypted file, which is then decrypted to produce the first segment of plaintext, which is finally delivered to the user. This cycle repeats multiple times over till the entire file is decrypted and returned to the user.



Although this process seems very involved, it is very performant and again takes only a sliver of the total upload/download time. On the contrary, this system results in some hefty performance gains in terms of download times which we will discuss in 2.4.

2.3. Decentralized

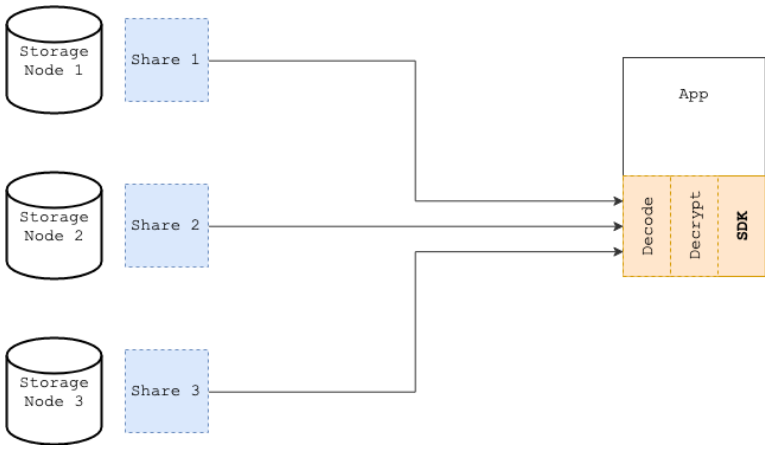
Decentralization is essentially the absence of a single service provider or central point of failure. Absence of an entity that can solely control costs and disbursement of services on a Network for the purposes of cornering the highest benefits.

Storage Nodes on Newfang are independent resource(storage + bandwidth) providers whose numbers are massive. There is a huge long-tail of such providers all over the world but aren't organised. One of the key motivations for us is to help organise this community and pass on the cost benefits these untapped markets could provide, to our customers. Additionally, we are able to further raise the cost benefit bar by not having to provision and maintain hardware. This also results in an ability to massively scale, very fast.

The trade-off for such a system is the obvious dependence on potentially untrustworthy service providers, as is the case in any unorganised segment, but we believe the governance we intend to build into Newfang will help alleviate any potential risks. A governance protocol where each node on the network is tasked with running proofs on its peers, reporting findings to a public ledger and a self correcting system that takes action in the event of malicious activity will eventually allow for a vibrant marketplace comprised of users trustlessly paying to consume resources and providers incentivised to provide resources and disincentivised to act maliciously.

2.4. Fast

Thanks to the distributed architecture of Newfang, users stand to enjoy a significant performance gain with file downloads. Since each storage node house an individual share of a file, it is possible for each of the required shares to be downloaded asynchronously and be decoded and decrypted on the client to provide a BitTorrent like effect with downloads. The more shares that exist, the more unique storage nodes that serve as “seeders”.

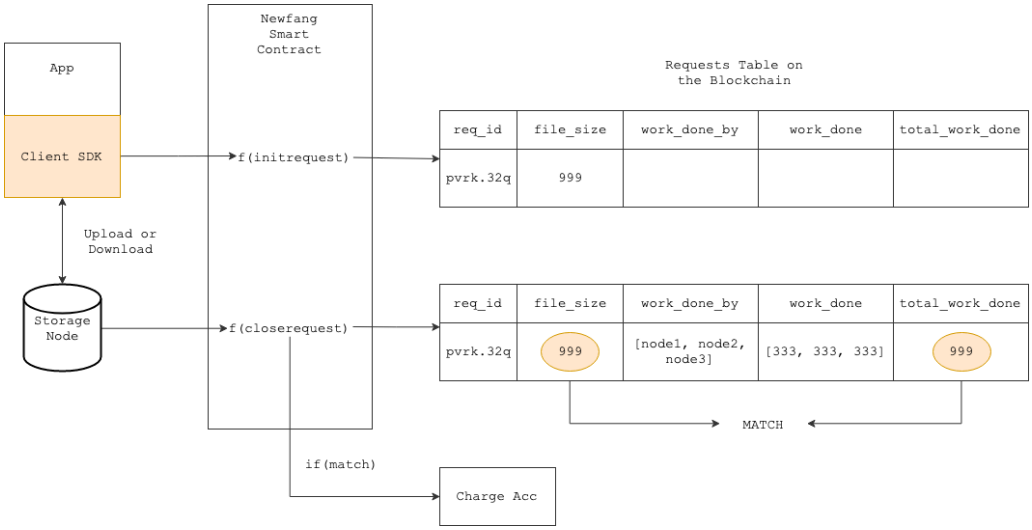


There are logical limitations to how many shares can be created per file and also limitations on how many can be asynchronously handled on a client depending on the resources available on the underlying system. Too many shares would not only be too heavy on the network resulting in potential failures but also lower the overall performance of an upload/download operation at the encode/decode step.

2.5. Transparent

All upload/download actions begin as requests made to the network which are logged as transactions on the blockchain. The size of the file is one of the details logged by the client SDK as part of an *initrequest* transaction. A node on the network is tasked with overseeing the operation to its logical conclusion and to report back the actual work done by the underlying storage nodes to the blockchain as part of a *closerequest* transaction. Once it is ascertained that the work done i.e. bytes stored(upload) or bytes

transmitted(download) matches the initial file size, the user's account is charged the appropriate amount..



The process of the client stating a certain file size i.e. work needed to be done and an adversarial storage node reporting the bytes stored or transmitted i.e. the work actually done helps bring the necessary transparency and fairness to the system.

Newfang uses the high throughput EOS blockchain which is a decentralized public blockchain where all transactions are open to public viewing bringing the necessary aspect of public verifiability of the Newfang system.

2.6. Intuitive

This is the cornerstone of Newfang. Anything else we do is rendered useless if we do not ensure that the platform is intuitive to our users. To this effect, we have taken the route of abstracting away not only complexity but also the poor user experience that is the norm with blockchain based solutions in general or other decentralized storage platforms. Our users do not have to:

A. **Wrangle with private keys to use their accounts:**
   This is abstracted away and managed by Newfang for the user. The way the EOS blockchain and its auth layer is structured means we can perform transactions on behalf of the user and pay for them too.

B. **Purchase or hold any cryptocurrency:**
   Users pay bills in USD. Users are charged a fixed price for upload/download and make payments against recurring charges accumulated into a bill, using their credit card or any of the several other payment mechanisms making their costs predictable. This also means that our users can be from any geography, part of

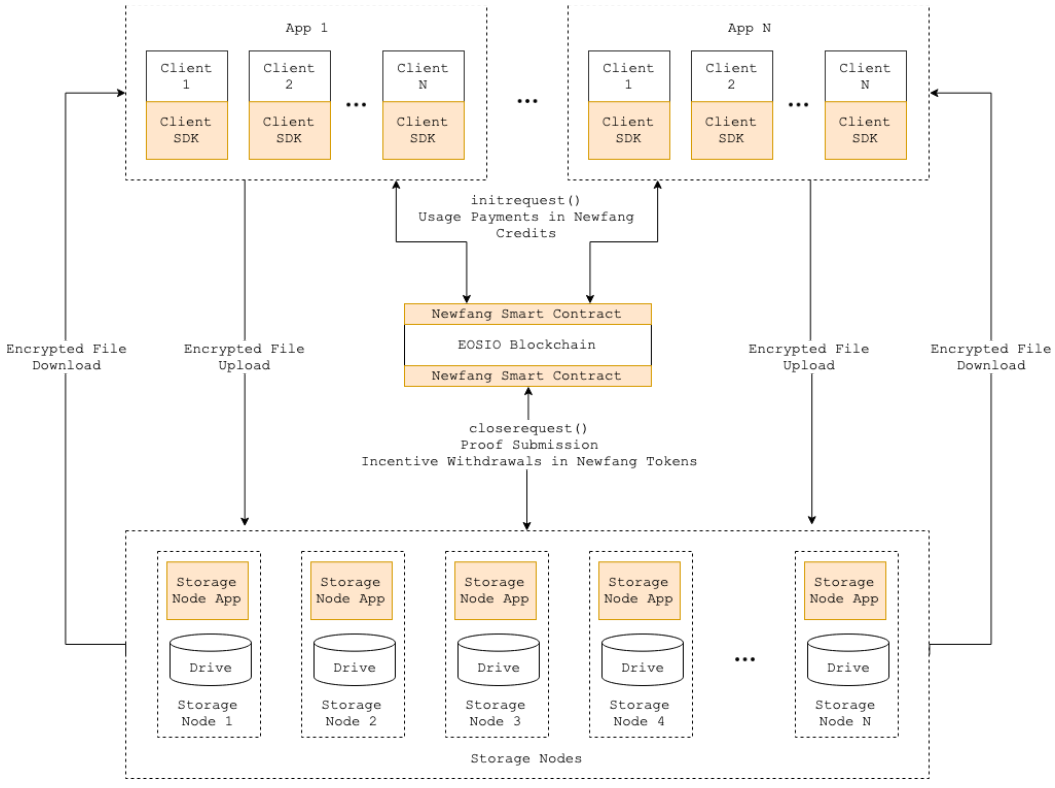
any enterprise large or small and they do not have to worry about the legalities or regulatory responsibilities that come with purchasing/holding cryptocurrencies.

C. **Worry about any volatility in the price of the underlying Newfang token:** Our novel dual token system, described in detail in section 4, means that any volatility in the Newfang Tokens price is encapsulated and does not affect how much it would cost our users to upload or download a file

D. **Perform complex procedures to join the network and use it:** No running full nodes, no complex setup, nothing… Easy to integrate SDKs for all platforms that expose a handful of simple methods to perform the core tasks of upload, download, delete. Users simply need to ensure prompt bill payments so that users of their apps have a seamless file management experience.

3. Implementation

3.1. Architecture



A. Post signup, Newfang users get to download customised SDKs for platform(s) of their choice. These are light-weight, easy to integrate SDKs that expose basic methods to upload, download and manage files which the user's apps can consume. Additionally, the SDKs internally interact with the Newfang Smart

Contract on the EOSIO Blockchain to know and update the current state of variables that drive the payments, payouts and transparency on the network.

B. Independent storage nodes on Newfang have to install and run simple software on their servers to be able to participate in the network. These apps are configured for a specific node and help it not only accept and handle requests to upload and download files from clients but also interact with the Newfang Smart Contract on the EOSIO Blockchain to know and update the current state of variables that drive the payments, payouts and transparency on the network. Additionally, these apps perform various checks and balances on peer nodes in order to ensure the integrity of the network is maintained at all times and also report findings back to the Newfang Smart Contract which in turn drives the governance of each of the nodes on the network.

C. The Newfang Smart Contract on the EOSIO blockchain serves almost as a decentralized middleware within the network and is responsible for informing and maintaining the state of various actions on the network. It is also what drives the token economics on the platform and ensures fair payments and payouts to all stakeholders.

3.2. Tahoe LAFS

Tahoe LAFS[5] forms the core of the Newfang network. It is a secure, decentralized, fault-tolerant and distributed cloud storage system. It has a great many virtues which have compelled Newfang to use it over other existing systems or building one from scratch. Among it:

A. Provider Independent Security:
This basically refers to all files being encrypted before being sent to potentially untrustworthy storage nodes because of which storage nodes are incapable of ever knowing the contents of the data being stored. The key to decrypt the files is only available with the client.

B. Decentralized by design:
All storage nodes produce the same homogeneous response to any request they receive. No single points of authority or failure.

C. Distributed architecture:
All files uploaded undergo a custom erasure coding step via an efficient erasure coding tool called zfec[6]. The encoded shares generated are pushed to unique storage nodes and only a subset of these shares, and consequently storage nodes, are required to retrieve the entire file.

D. Self Correcting:

Tahoe LAFS has a repair mechanism driven by timely checks on file and share integrity on the network. On finding any missing or damaged shares, Tahoe is able to initiate a repair step that restores the lost/incorrect share and restore the integrity of the file's encoded shares and K of N configuration of the files themselves.

There are obvious limitations to using Tahoe LAFS as is. Newfang has had to and will continue to customise it to suit its needs such as building light clients, on-demand file deletion, automated repairs…

3.3. Client SDKs

Newfang will build and provide SDKs for all major platforms for our users to be able to download and integrate into their apps. These SDKs perform the following tasks in order to successfully complete core network operations or gracefully fail:

A. SDKs expose 3 main methods for developers to implement in their apps i.e. Upload, Download and Delete. There will be more functionality that will be added in the future.

B. The SDKs also perform encryption/decryption on the client before any Upload operation or post a Download.

C. They are responsible for learning the identity of the appropriate storage node which will oversee the current operation the user intends to perform.

D. They also place the initial request transaction with the EOS blockchain by calling the *initrequest* method of the Newfang smart contract. In this step it also logs the size of the file that it expects to upload/download which is verified in the reconciliation step that occurs post the action completing and before any payment.

E. Finally, they are responsible to decode a file being downloaded before it is decrypted. It receives all the pieces of the file generated in the encode step on upload and puts them back together on the client.

These are light-weight SDKs that are extremely easy to integrate. Once signed up, the developer simply needs to create an application on the developer portal with a specific configuration, download application specific SDKs for the platform(s) of their choice and be up and running in minutes.

3.4. Storage Node App

Each Storage Node that is onboarded to provide resources to the network and receive an incentive in return, post sign up, simply has to download and install this Storage Node App to be able to join the network. The primary tasks of the Storage Node App are:

A. Accept and execute Upload/Download/Delete requests.
B. Encode the encrypted files received on accepting an Upload request.
C. Keep track of where each encoded piece of every file is located.
D. Perform the reconciliation step post completion of an Upload/Download action, reporting back the work done by the nodes in the process i.e. the number of bytes stored or transferred depending on the type of action performed. This closes the loop and helps the network ascertain that the expected amount of work was indeed done and that payment for completion of the action can be withdrawn from the developer's account.
E. Execute and gather proofs for Uptime, Retrievability and Storage on peer Storage Nodes. Report the results back to a high throughput DLT that is responsible for the Governance on the platform(details in 3.6).

3.5. Smart Contracts

Newfang uses the EOSIO[7] blockchain as its smart contract platform. The reasons why Newfang chose EOSIO are:

A. EOSIO is a decentralized public blockchain which runs a dPOS[8] consensus among 21 validators, voted in from a larger set of candidates by token holders. The fact that it is a public blockchain is crucial to ensure the transparency and fairness with which the Newfang Network operates.
B. EOSIO has a mainnet which offers very high throughput compared to similar platforms. Blocks are mined every 0.5s and when combined with its Byzantine Fault Tolerant dPOS consensus, transactions get confirmed within 1s. EOS, in its current state, offers up around 3000 transactions per second.
C. Transactions on EOSIO are feeless. One does need to pay for using up any storage, bandwidth or cpu time but these are rental charges that are returned when resources are freed up.
D. It also provides a very neat abstraction layer over user authentication that basically allows for transactions to be placed and paid for by a permissioned representative. This means that Newfang can foot the bill and execute all transactions allowing the developers and their users to be free from all the complexity and poor UX of wrangling with private keys. It also allows the developers of applications to be more flexible with the business model they intend to have in their apps and not be tied into the platform's business model directly.

The Newfang smart contracts are primarily used to:

A. Store and update the public state of several variables that describe the requests made on the platform and their status.
B. Compute the payments needed to be collected from developer accounts when a request is closed successfully.
C. Compute payouts needed to be made to storage node accounts at regular epochs based on the work done by them among other parameters.
D. Help create and keep track of the Newfang Token via the standard eosio.token contract.

3.6. Governance(WiP)

This is pending concrete implementation and we are currently working on:

A. Mitigating attacks. Namely Sybil Attack, Outsourcing Attack, Generation Attack and DDoS. Tahoe LAFS gives us some great in-built constructs to implement a Proof of Retrievability[9] and Proof of Replication[10] and also defend against a Sybil attack. Additionally we are looking to implement a VDF enabled PoST[11] that would help us mitigate against a Generation Attack.

B. Incentive distribution. What are the various data points that would help us arrive at a sound 'rating' system for Storage Nodes? What is the interval at which tokens get distributed? What is the amount of tokens minted? Will we need a periodic inflation of the fixed mint amount at all?

C. Staking and Slashing. What amount does a Storage Node need to stake to join the network? What are scenarios that warrant stake slashing? How much should be slashed?

Largely, we are inclined towards the idea of not introducing any new stakeholders on the network and have governance be performed on-chain by peers in a truly decentralized fashion. The actual implementation could perhaps involve a high throughput, DPoS consensus DLT like Tendermint/Cosmos or Polkadot/Substrate.

We have several ideas at various stages of development and implementation, coming together before our Beta Launch later this year.

4. Business and Token Model

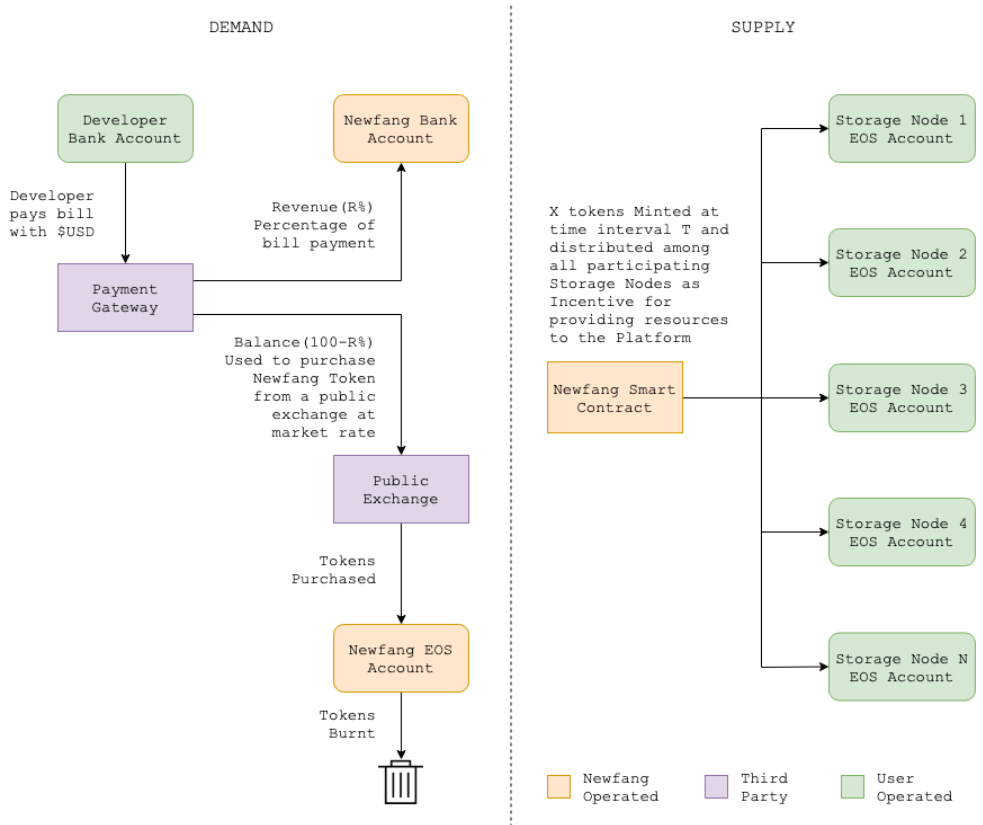
4.1. Structure

Newfang Tokens are a token created on the EOSIO Blockchain using the standard EOSIO token smart contract. They are a cryptocurrency that will be listed on public exchanges, have a trading value and be subject to market speculation. The Newfang Token is what will be distributed as a payout to Storage Nodes on the network in lieu of the resources they provide to the platform. A fixed number of tokens are minted every epoch to be distributed to the Storage Nodes as per the distribution algorithm. There may or may not be an annual inflation of the minted amount. This will be decided prior to the Beta Launch of the platform.

4.2. Business Model

Once signed up, developers pay bills that accumulate transparent charges against uploads/downloads on their apps, using a fiat currency(USD). A percentage(say R%) of this amount received(after setting aside payment gateway fees) is Newfang's revenue. The remainder of the amount(100-R%) feeds into the token model as described below in 4.3.

4.3. Token Model



Newfang employs the Burn-Mint Equilibrium[12] token model where Newfang tokens are Minted on the Supply side and Burnt on the Demand side. In the absence of market speculation, constant burning and minting of tokens causes the price of the token to adjust itself owing to the relative inflation/deflation that comes with a currency increasing/reducing in supply, striving to reach an equilibrium.

Newfang mints a fixed amount of tokens on the Supply side and distributes among the participating storage nodes. This causes an increase in supply and inherent inflation which in turn causes the price of the token to drop.
On the Demand side, each time a user pays a bill, after keeping a portion of the payment as revenue, Newfang automatically moves the balance of the purchase value to an account on a public exchange where the Newfang token is being actively traded and uses the entire amount to purchase tokens at the prevailing market rate. These tokens are then moved out of the exchange account and into one of Newfangs EOSIO accounts and are then burnt or destroyed. This causes a decrease in supply and inherent deflation which in turn causes the price of the token to rise.

With inflation and a drop in token price, we are able to purchase and burn more Newfang tokens which cause a relatively larger deflation and price rise. As a fixed number of tokens are minted at every epoch and with usage on the platform being low in the initial stages, the burn rate will keep trying to catch up with the mint rate and eventually reach equilibrium at which point, barring any potential annual inflation introduced into the system, token supply would reach a stable value.

This token model helps Newfang:

A. Abstract away the need for users to hold/manage a cryptocurrency.
B. Allow developers to pay a fixed $ amount against uploads/downloads, when paying bills and not contend with any token price volatility. Leading to predictable expenses.
C. Ensure that increase in usage on the platform is the main driving force behind the increase in the price of the Newfang Token.
D. Provide stable unit economics to storage nodes.

5. References

[1]https://www.marketsandmarkets.com/PressReleases/cloud-storage.asp

[2]https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed/

[3]https://www.medianama.com/2019/06/223-google-cloud-outage-shows-how-a-highly-centralised-internet-is-bad-for-everyone/

[4]https://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[5]https://tahoe-lafs.org/trac/tahoe-lafs

[6]https://tahoe-lafs.org/trac/zfec/

[7]https://eos.io/

[8]https://en.bitcoinwiki.org/wiki/DPoS

[9]http://www.arijuels.com/wp-content/uploads/2013/09/BJO09b.pdf

[10]https://eprint.iacr.org/2018/678.pdf

[11]https://eprint.iacr.org/2016/035.pdf

[12]https://multicoin.capital/2018/02/13/new-models-utility-tokens/