Is Filecoin a $257 million Ponzi scheme?

Marc Juchli
Faculty of Electrical Engineering,
Mathematics and Computer Science
Delft University of Technology
m.b.juchli@student.tudelft.nl

Johan A. Pouwelse
Parallel and Distributed Systems Group
Department of Software and Computer Technology
Delft University of Technology
j.a.pouwelse@tudelft.nl

***Abstract*—**

Keywords*—*filecoin, decentralized storage network, ico, proof-of-spacetime, crypto-currency.

I. Introduction

> ”A MASSIVE AMOUNT OF STORAGE SITS UNUSED IN DATA CENTERS AND HARD DRIVES AROUND THE WORLD.” [34]

With this slogan Protocol Labs is about to disrupt the storage market by using *proof-of-spacetime* as their driving source. The Filecoin project describes a decentralized storage market where anyone, worldwide, is able to participate as a storage provider. The concept is indeed promising and convinced the investors such that a total of $257 million had been raised – the biggest initial coin offering (ICO) as of today (September 2017). However, the idea of a decentralized storage market is not a novel concept. Others [29] [47] have tried in past too, but yet were not able to scale as much as Filecoin advertises to do, and eventually failed. More recent projects [31] [36] are currently working towards building a similar system with conceptual differences which will be uncovered briefly in this paper.

This paper aims to analyze the ICO launch of Filecoin and reasons about the exceptionally large investment using heavily discussed topics in social media channels. Further, the feasibility in terms of technical as well as economical design is studied. We aim to uncover potential weaknesses but also highlight strengths of the proposed white-paper [33]. The novel *proof-of-spacetime* consensus algorithm is being highlighted and compared with consensus proposals from projects such as StorJ and Sia. Eventually, we reason about whether the ICO launch can be considered as to be a ponzi scheme.

The structure of this paper is as follows: Section II lays out a history of decentralized storage projects with a similar ambition as Filecoin, but that have failed over time. Section III mentions recent competitor projects that make use of a blockchain. The following Section IV is an introduction to the Filecoin project by analyzing the ICO including the reasoning about the token sale and allocation thereof. Section V reasons whether the promises of the Filecoin project are technically feasible. By doing so, we analyze the InterPlanetary File System (IPFS) [3] in Section VI and highlight possible ways of its adoption in Filecoin. In Section VII we question the economical feasibility in rudumentary manners and lastly conclude about the entire Filecoin project in Section VIII.

II. 15 years of documented failure

In July 2000, long before the blockchain era, the Mojo Nation software [47] was released, aiming to serve an ”emergent file store” to its users. The project successfully deployed a decentralized storage network in an environment consisting of unmanaged nodes. It used consistent hashing [46] to locate nodes and data blocks. However, the project was shut down in February 2002 due to a number of problems:

- **Data Availability:** the main issue was the inconsistency of data available to its users as it depended upon which server nodes were connected at the given time. According to Maymounkov et al. [48] this could have been avoided by heuristically favoring long-lived nodes and by discriminating newly joined nodes which show a frequent join- and leave behaviour.
- **Firewalls and NAT**: networking hurdles such as firewalls and network address translation (NAT) prevented a substantial amount of nodes to act as servers.
- **Mutual distrust:** in order to have a network of nodes to behave as designed a *motivation to cooperate* needs to be established within the network, combined with sophisticated *attack resistance* mechanism which prevents nodes from using resources of other nodes without offering equal amounts of services in return.

Also academia has been struggling to succeed in building self-organizing systems. Tribler [29] is based on robust reputation and craft collaboration, an environment where one could think of a decentralized marketplace which can handle storage as an asset. The 12 years of development has been a history affected by many hurdles. Security issues such as collusion attacks [30] have slowed down the development progress enormously, preventing the project to scale.

III. Recent Competitors

At the time of writing this paper, an invasion of ICOs has arisen, among some of which are projects that tend to go into a similar direction as Filecoin does. Figure 1 shows that the funds invested in ICOs has overcome the investments made in the venture capital sector. Primary examples of decentralized, incentivized, byzantine fault-tolerant storage networks which shall be competitive with centralized alternatives such as Amazon S3 [16] are: StorJ [31], Sia [36] and MaidSafe [32].

- **Sia**: supports on-blockchain smart contracts which define payments for hosts while providing storage.

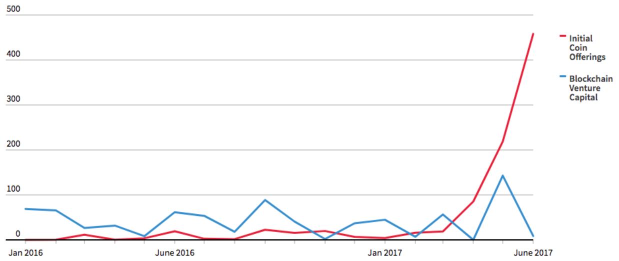

Fig. 1. ICOs vs Venture Capital [17]

Payment is guaranteed once the contract has been created and ensures that the host is being paid in case the the uploader never accesses the file. Additionally, the contract enforces penalties for hosts which go offline or lose data.

- **Storj**: is similar to Sia but does not feature on-blockchain storage contracts but instead offers a pay-as-you-go model for nodes who provide storage. Once the host disappears or goes offline, payments are halted.
- **MaidSAFE** goes beyond decentralized storage with a less sophisticated focus on efficiency. MaidSAFE uses a novel scheme for achieving consensus by not relying on a blockchain but instead on close group consensus and hence is not proof-of-work related.

IV. ICO analysis

The Filecoin ICO started on August 10, 2017 and closed the offering on September 7th. It was the first ICO ever which complied with SEC securities regulations and hence only accredited investors were allowed to contribute (Reg. D, 506(c), see [28]). Further was the ICO conducted using CoinList [27], a platform for token sales, built by Protocol Labs too. CoinList partners with AngelList [26] whose responsibility is on the compliance side regarding the law. In total, approximately $257’000’000 was raised, formed of $52’000’000 from presale and $205’800’000 Reg D investments. For the latter category, $135 million was raised within the first hour.

A. Simple Agreement for Future Tokens

The tokens distributed during the ICO were so called Simple Agreement for Future Tokens (SAFT). This instrument allows Coinlist to distribute investors the *right* to receive units of the actual Filecoin tokens (FIL) in the future. The definiton of SAFT, however, is also equipped with the following statement:

> ”...a significant portion of the amount raised under the SAFTs will be used to fund the Companys development of a decentralized storage network that enables entities to earn Filecoin (the Filecoin Network)”. [23]

Therefore, it is not explicitly mentioned to what extent the development process is being funded, and thus it is left to be decided by Protocol Labs Inc., solely. Additionally, SAFT introduces great flexibility to the token intermediary, CoinList, as the agreement not only allows to verify accredited investors but also to specify *events* transparently. Transparency is certainly appreciated when funds are being transferred, but it is needless to say that the terms have to be fully understood by both parties. One such event, which might be underestimated by the investor, is the *Dissolution Event* and is defined as follows:

> ”Dissolution Event means (i) a voluntary termination of operations of the Company, (ii) a general assignment for the benefit of the Companys creditors or (iii) any other liquidation, dissolution or winding up of the Company, whether voluntary or involuntary.” [23]

Knowing that it is at any given time possible for Protocal Labs to terminate their business, the entire setup of the ICO becomes fragile when reading the *execution plan*:

> ”If immediately prior to the consummation of the Dissolution Event, the assets of the Company that remain legally available for distribution to the Purchaser and all holders of all other SAFTs (the Dissolving Purchasers), as determined in good faith by the Companys board of directors, are insufficient to permit the payment to the Dissolving Purchasers of their respective Returned Purchase Amounts, then the remaining assets of the Company legally available for distribution will be distributed with equal priority and pro rata among the Dissolving Purchasers...” [23]

After all, only the *remaining* assets which are legally available will be distributed in the case of a *Dissolution Event*; including definition (i), the voluntary termination. Unfortunately, if one considers the worst case scenario in which Protocol Labs decides to terminate the operation, while having spent all the funds available, the investors would likely be excluded from the distribution of any reward. Similarly, if the project does not announce a *Network Launch* [23] event by July 18, 2022 (including a 60-days extension), then, by definition (i) or (iii), investors would have to expect a *Dissolution Event* to be announced as a subsequent step.

B. Token allocation

As presented in [22], the allocation of the Filecoin token is distributed to 4 groups of participants:

- 70% to Filecoin Miners as mining block rewards once *Network Launch* is past
- 15% to Protocol labs as genesis allocation with 6-year linear vesting
- 10% to Investors as genesis allocation with 6 months to 3 year linear vesting
- 5% to the Filecoin Foundation as genesis allocation with 6-year linear vesting

The total of $257 million US-dollar raised during the advisor pre-sale and investor sale (see IV-C) therefore only accounts up to 10% of the total coins expecting to be circulating after several years past *Network Launch*. Since the half-life time of Filecoin block rewards is set to 6 years, this will likely not change significantly for several years, considering that the *Network Launch* requires a solid implementation of the Filcoin ecosystem.

C. Token sale

The offering of SAFTs was established in a two-phase process: the first phase allowed Protocol Labs as well as Filecoin advisors to purchase SAFTs prior the broader group of investors. The latter were able to proceed purchases in a second phase. In the first phase, the price was fixed at 0.75 USD/FIL. For the second phase, a pricing function was introduced:

$$price = max(\$1, \frac{amountRaised}{\$40'000'000})$$

The pricing function increases linearly with the amount being raised, as indictated in Figure 2. As a result, the closing price, past $257’000’000 investments, can be estimated to be approximately $6.425. Hence, advisors and Protocol Labs Inc. itself were able to purchase for a price which is a factor of 8.57 lower than the late investor.

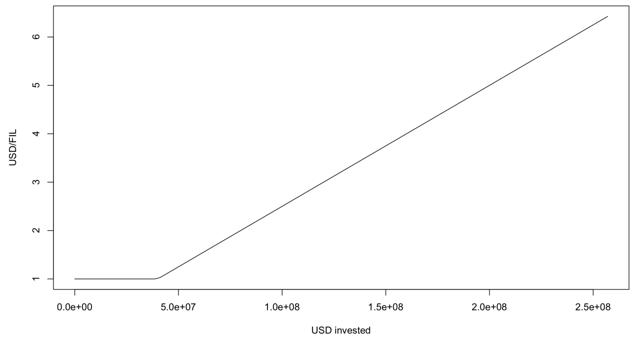

Fig. 2. Filecoin sale price function

V. Is the design technically feasible?

In this section, technical details will be highlighted with reasoning whether the decisions stated in the Whitepaper [33] are chosen appropriately or arise potential weaknesses. We further aim to make comparisons to more recent decentralized storage projects involving a blockchain, such as StorJ [31] and Sia [36].

A. Decentralized Storage Network

The notion *Decentralized Storage Network* (DSN) is a scheme with associated protocols that enables to aggregate and provide data in decentralized coordination manners. The protocols verify that involved parties execute their operations securely and therefore allows coordination without a trusted party. The Filecoin DSN introduces three types of participants: (1) *Clients* who pay FIL tokens to store and retrieve data, (2) *Storage Miners* who pledge FIL tokens as a collateral while providing storage which is worth equal amounts of FIL and are eligible to mine new tokens by doing so, and (3) *Retrieval Miners* who serve data upon client requests, and oftentimes act as Storage Miners too. Further does Filecoin personify all the participating parties which run a *full nodes* as *The Network (N)*. Hence, the protocol scheme is being castellated as a tuple in a very suitable and abstract definition: `(Put, Get, Manage)`. Storing and retrieving data is done by Clients using the Put, or respectively Get, Protocol. The Network is being coordinated using the Manage protocol, which serves to control the available storage, audit offered services by Storage Providers and repair possible faults. Little is known about the details of `Manage.AssignOrders` and `Manage.RepairOrders`, such as, whether the manage protocol follows incentive and rewards nodes for actively maintaining the network. However, from what is known, the Filecoin DSN, compared to the competitors described in Section III, is indeed conceptually splendid. The simplicity of `(Put, Get, Manage)` implicitly covers additional client operations operations such as `PING` and `FIND_NODE`, as defined in StorJ [31].

The flow of storing data, initiated by a client, involves the data to be split into `pieces` in order to be stored within `sectors` as part of the disk space provided by storage miners. Conceptually, the data handling is very similar to StorJ [31] where data, after it underwent AES256-CTR encryption executed by the client, is split into *shards*. While StorJ dynamically adjusts the size of the shards depending on the file to be stored, it is yet not known how Filecoin will approach the splitting into pieces. Filecoin, like StorJ, delegates the encryption task to the client. However, whereas StorJ enforces encryption in its reference client implementation, Filecoin currently has no plan for offering built-in encryption and leaves this responsibility solely to the client before approaching the Filecoin network. In order to introduce redundancy for the to be stored data, Filecoin neatly introduces a *replication factor* to be choosen during the `Put` protocol which allows to increase the tolerance of storage faults. In contrast, StorJ herefore uses a distinctive *mirror* method.

The guarantees and requirements laid out by the Filecoin DSN can be summarized as follows:

- *Integrity:* In order to ensure that clients do not accept altered or falsified data, cryptographic hashes are used as a naming convention and serve as identifier for data retrieval (`Get` protocol) and verification of its content. Filecoin does not rely on any meta data, such as its competitor StorJ [31], but relies on the hash only.
- *Retrievability:* after the data is successfully stored, clients are ensured that the very same data can eventually be retrieved. The $(f, m) - tolerant$ system specifies that given m storage miners, a maximum of f faults are tolerated. Increasing the replication factor hence implicitly increases the chances of recovery.
- *Public Verifiability and Auditability:* as storage miners are obligated to submit proofs of storage (see V-C to the blockchain, any user can verify their validity without having access to the data. Since *proof-of-spacetime:* implicitly guarantees the continuous existence of the data on the storage miner side, no challenge-response communication is required. Compared to StorJ [31], the communication overhead is therefore reduced while providing the same guarantees.
- *Incentive compatibility:* like any of the projects mentioned in III, Filecoin enforces incentive by rewarding

parties which store data and punishing those who loose data.

- *Confidentiality:* as mentioned earlier in this Section, Filecoin is weaker in terms of confidentiality as it fully delegates the encryption task to the client.

B. Ledger

The `Ledger` L being used in Filecoin will be represented by a native blockchain, as announced in [35], and supports various types of data structures. The state of the DSN is stored within an `AllocTable`, which keeps track of `pieces` and their assigned `sectors`. The `Orderbook` is responsible for storing `Orders` which either state a request to store data (`Bid order`), offer a service (`Ask order`) or confirm a match of bid- and ask orders in form of a `Deal order`. Lastly, a `Pledge` engraved in the ledger represents the collateral of the storage miner in order to accept orders from the clients, as described in V-A. In contrast, Sia only stores storage contracts which define the terms of the formed agreement between parties and therefore relies on a variant of the Bitcoin protocol [21]. As Filecoin is clearly more diverse in terms of data structures additional complexity in the clients is to be expected. From another perspective, this decision might allow to extend the protocol more easily as the ledger is already lied out to handle various types of data structures. Having a native blockchain further was a necessary decision in order to employ Proof-of-Spacetime, see V-C. However, by not relying on the Ethereum Network [19] as previously planned and announced at DEVCON2 [20], and its absence of an ERC20 token [18] hence prevents direct compatibility to Ethereum and therefore requires a bridge of some sort.

C. Proof-of-Spacetime

The white-paper [33] describes a novel consensus protocol: *Proof-of-Spacetime* (PoSt). With Proof-of-Spacetime it becomes possible to check if a prover is storing data for a range of time, or formally:

> ”A `PoSt` scheme enables an efficient prover `P` to convince a verifier `V` that `P` is storing data `D` for some time `t`.” [33]

It takes advantage of the capabilities of *Proof-of-Storage* [50], which can confirm if a storage provider is storing data at the time of the challenge, by sequentially generating such proofs and recursively compose the executions thereof. The concrete implementation of Proof-of-Storage is described as *Proof-of-Replication* (PoRep), a novel concept that lets a storage miner to convince a client that its data has been replicated to a uniquely dedicated physical storage. Other Proof-of-Storage schemes such as Provable Data Possession (PDP) [51] and Proof-of-Retrievability (PoR) [52] essentially guarantee possession of some data at the time of the challenge/response. Proof-of-Replication, however, improves those schemes by preventing *Sybil Attacks, Outsourcing Attacks*, and *Generation Attacks*. As Proof-of-Spacetime is based on Proof-of-Replication, these properties are inherited. Thus, PoSt prevents pretentious copies which are not physically stored (*Sybil Attack*). It further denies storage miners to offer more storage than physically available (*Outsourcing Attack*) and also protects from on-demand data generation while this should effectively be stored on a physical disk (*Generation Attack*). The cryptographic building blocks of PoRep and PoSt rely on zero-knowledge Succinct Non-interactive Arguments of Knowledge (zk-SNARKs). Zero knowledge proof, including zk-SNARKs have shown to have great potential allowed to build large scale projects such as ZeroCash [53].

Apart from the extensive capabilities in terms of storing data, Proof-of-Spacetime attempts to reduce resource waste, by considering that storing users data is a form of work. It is therefore a consensus protocol based on *useful work*. The usefulness implies that computational power is not wasted—as this is the case for *Proof-of-Work* [21] and other consensus algorithms. In Filecoin, the voting power (the probability of a miner being elected to create a new block) increases proportionally with the storage offered to the network in relation to the storage resources of the entire network. Due to the fact that Proof-of-Spacetime composes Proof-of-Replication executions sequentially, it is further considered as non-parallelizable and thus greater computational resources will not have any noticeable effect. [33]

However, given that storage miners may offer varying amounts of storage results, their influence on the network is being distributed continuously and unequally. Hence, a naive Byzantine Fault Tolerance approach that uses the number of faulty nodes is in the case of Filecoin not a sufficient measure for determining the outcome of a protocol. Instead, Filecoin proposes *Power Fault Tolerance* (PFT) [54], an abstraction that defines byzantine faults in terms of the *influence* of a participant.

D. Decentralized Markets

Filecoin introduces two types of markets, *Storage Market* and *Retrieval Market*, represented by two independent, decentralized exchanges. The notion *Verifiable Markets* is presented and describes a market where participants are able to verify the exchange between buyers and sellers. Specifically, the protocol describes a two-phase process: *Order matching* allows participants to add buy and sell orders to the orderbook and eventually allows to create deal orders once the two opposite (buy and sell) orders have matched. The second phase, described as *Settlement*, involves the network to ensure the correct execution of the transfer of goods (data) and eventually initializes a payment. The *Verifiable Market Protocol* applies to both of the aforementioned markets. [33]

The main purpose of the Storage Market is for clients to request storage and for storage miners to offer their resources. Bid- and ask orders from those parties will be placed into an orderbook and are therefore publicly available to any participant of the network. Filecoin states that ”every honest user has the same view of the orderbook”. That is, the orderbook is a data-structure incorporated in the blockchain. Hence, orders are added to the blockchain if they are valid. As the Storage Market is a *verifiable market*, orders of type *bid, ask* and *deal* can be validated by every participant.

The only security related parameter required in the order matching phase is a field (`ts`) which describes the duration of how long a *deal order* is valid. This prevents a client from holding back data once the storage miner has committed its resources. The settlement phase further involves the storage miner to seal their sectors and submit the generated proofs of storage to the blockchain. [33]

The retrieval markets sole purpose is for retrieval miners to provide data to the clients upon their requests. One very challenging requirement for this market is the *fast* retrieval of data. Therefore, and unlike the storage market, the retrieval market maintains an off-chain orderbook that allows clients and retrieval miners to find each other. The absence of a blockchain acting as a trusted party introduces the need for other ways of forming trust between client and retrieval miners. Filecoin essentially relies on token exchange as trust instrument. The to be delivered data is being split in multiple pieces and for every successful exchange of a piece the client pays the miner. If one of the involved parties does not come after its duties, the other party is free to stop.

In order to process payments in the first place a network of payment channels is required. Filecoin has not stated more detailed plans of how to integrate payment channels into the retrieval market. Although this being a risk, much research has been done and promising projects have evolved [49].

The order matching phase of the verifiable market protocol differs much, compared to the storage market. Since the orderbook cannot be recorded in a blockchain, clients and retrieval miners have to *gossip* their orders. Filecoin assumes that this point that there is always at least one honest retrieval miner. [33]

VI. IPFS

As described in [33], Filecoin serves as an incentivized seeding layer on top of the InterPlanetary File System (IPFS) [3]. The Filecoin project is therefore beneficial to IPFS as it intends to add more storage to the network. At the same time, is strongly depending on the technical capabilities and robustness of IPFS. This sections aims to highlight the capabilities of IPFS and explains the conceptual decisions made while building a peer-to-peer hypermedia protocol. The latter involves a brief overview of the libp2p project [**?**], a modular peer-to-peer networking stack. Lastly, we reason about possible approaches on how to integrate IFPS in Filecoin.

A. The flaws of HTTP

The Hypertext Transfer Protocol (HTTP) can be seen the global information protocol that standardized how people distribute and present information among each other. Publishing content with HTTP is nowadays almost free and led to great birth of innovation and has been driven innovation, economics and culture ever since. However, the way this valuable content is being distributed a significant flaw: HTTP allows content to *erode* [15]. Covered by error code 404 [14], HTTP fails to maintain links between websites and allows valuable content to vanish completely. The main reason for this problem to occur are centrally managed servers, free to shut down at any given time. While short-term availability of content tends to be sufficient, the ongoing erosion of data lacks to maintain long-term availability and results in a vast amount of broken links [13]. HTTP naturally empathized large organizations to centralize their services which constitutes in an ever increasing potential loss of data. As a result, what used to be a decentralized web is now moving towards a more and more centralized web structure.

B. Distributed file system

According to the readme [12]: ”IPFS is a distributed file system that seeks to connect all computing devices with the same system of files.” In practice, IPFS is a peer-to-peer distributed system that connects all networks using the same system of files. The files are version controlled and represented with their hash and as opposed to HTTP, where content is being searched by location, IPFS searches by content. Hence, IPFS is a content-addressable peer-to-peer hypermedia distribution protocol.

Nodes, which are incentiviszed to remain the same and loose network benefits otherwise, are identified by a cryptographic hash of a public-key. IPFS does not rely on a particular hash function format but instead uses the multihash [11] format such that any well-established cryptographic hash function can be used in order to create a node.

The *Network* then allows nodes to communicate while providing a framework which ensures any client, no matter what network stack looks like, can participate. That is, the client can use any transport protocol and IPFS provides reliability functions if the underlying network does not provide so. Connectivity is enhanced using ICE NAT traversal techniques [10] which for example also includes relaying such that nodes can find nother nodes on their behalf to provide demanded content. Further more, IPFS does not rely on IP only but instead uses the multiaddr [9] format to express addresses and their protocols.

Routing is required for nodes to find other peers which can serve objects. A *distributed sloppy hash table (DSHT)* [8] serves those purposes and is based on S/Kademlia [6] and Coral [7]. Again, IPFS remains true to its flexible and modular structure and thus allows to change the implementation of `IPFSRouting` interface, allowing, for example, in a local environment to use a regular hash table. [3]

Proceeding a *block exchange* for distributing data in IPFS is implemented with BitSwap [4], a protocol inspired by BitTorrent [5]. BitSwap is incentivized as it uses a credit system where peers track their balance with other nodes and sending blocks to debtor peers is implemented with probabilistic approach. Hereby, an additional timeout is being applied in case a sender does not come after its duties, which prevents gaming the probabilities. Whereas BitTorrent uses tit-for-tat strategy, BitSwamp relies on a debt ratio factor:

$$r = \frac{bytes\_sent}{bytes\_received + 1}$$

A probabilistic function further describes the likelihood of a peer sending data as follows:

$$P(send|r) = 1 - \frac{1}{1 + exp(6 - 3r)}$$

The function implies that once the debt ratio of a peer surpasses twice the amount of its credit, the likelihood of sending and hence its trustworthiness drops radically. As a result, this BitSwamp implements a measure of trust using the debt ratio factor and therefore provides resistance against Sybil attacks and values successful relationships among peers yet with the tolerance of temporary unavailability. [3]

In order to effectivly store and distribute blocks in quick and robust manners, IPFS introduces the *Merkle DAG*. The directed acyclic graph is built-up with cryptographic `links` of the underlying objects. The `object` is a data structure consisting of a `name`, `multihash` and `size`, allowing any type of data to be represented with. The Merkle DAG, described as "a generalization of the Git data structure" [3], hence provides properties including: Content Addressing (content is uniquely identified by its multihash checksum), Tamper resistance (content is verified with its checksum) and Deduplication (objects with exact same content are equal and are stored only once). [3] Allowing to traverse the Merkle DAG is being enabled with UNIX-like paths describing the multi-hashes of the object:
`/ipfs/<hash-of-object>/<name-path-to-object>`
That is, IPFS makes an attempt to reuse long established file system properties while handling data with a fundamentally different concept than what was known so far.

At last and in order to make IPFS a fully useful file system, IPFS introduces IPNS [2], the InterPlanetary Naming System. The goal is to maintain the Merkle DAG containing immutable content-addressed objects, and apply *Naming* by using mutable pointers to the Merkle DAG. Thus, enabling user friendly naming. [3]

C. Libp2p

The development of IPFS has been a demanding challenge in understanding the internet network stack. [56] As a result, the `libp2p` library was created in order to bundle protocols used for building large scale peer-to-peer applications. Thus, libp2p can be regarded as an entire network stack, represented by a protocol suite. Conceptually, libp2p is structured according to Figure 3. 2.

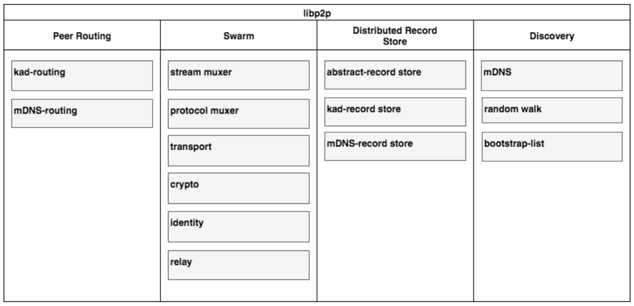

Fig. 3. Libp2p network stack

The interfaces can be described as follows:

- **Peer Routing:** Identifies the peers to which a message should be routed to. The implementations include *kad-routing*, a kademlia routing table where each peer holds a set of k-buckets; and mDNS-routing which identifies local area network peers.
- **Swarm:** handles everything related to the opening of a stream. A *stream muxer* multiplexes connections per peer and streams per connection. The *protocol muxer* enables multiplexing of transport protocols on applications level. This allows several protocols to be muxed in the same socket and therefore only one NAT traversal would be required, if any. Further more, *Relay* provides an end-to-end encrypted connection where a node asks another node to connect on its behalf and thus can overcome NAT traversal.
- **Distributed Record Store:** The aim is to store and distribute records, that is, similar to DNS as it is used for signaling links, announcing peers and content. The record store was further modularized under the name: IPRS, InterPlanetary Record Store [1]. The interface provides record stores for abstract records, kademila routing records as well as mDNS routing records. Having such a record keeping system allows content to be verified by any user of the record store.
- **Discovery:** in order for peers to find and identify each other in the network, libp2p provides several ways to do so. LAN discovery is being implemented with *mDNS*, *Random Walk* using a DHT (distributed hash table) discovery by proceeding random queries enables discovery of peers outside the LAN, and finally a *Bootstrap-list* enables peers to store highly stable (and possibly trusted) peers locally.

D. Filecoin integration

As for now it is not known how Filecoin plans to adapt IPFS as its underlying data store. The previous Section VI-B introduced the components of IPFS briefly and provides the basic knowledge to reason about how Filecoin would be able to take advantage of the IFPS ecosystem. We presume that the most obvious component to hook in is Bitswap (see Section VI-B). Bitswap manages requests from peers in the network and therefore is considered as the "data trading module" of IPFS [4]. Essentially, Bitswap acquires blocks requested by the client and initiates a send to the peers who demand these blocks. We believe that in the case of Filecoin, the native measure of trust of Bitswap, provided by the debt ratio factor, has to change. Instead of relying on the bytes being sent and received, Filecoin provides a measure of trust by relying on FIL token exchange and the guarantees provided by proof-of-spacetime (see Section V-C). Regarding the distribution of the blocks among peers, Bitswap would react on `DEAL` orders (see Section V-B). Depending on whether the order evolved form the storage- or retrieval market, blocks would be sent to the node on either ask- or bid side. As a result, Bitswap serves as the API used by the decentralized markets (see Section V-D) and handles data exchange according to Filecoins incentive.

VII. Is the design economically feasible?

While analyzing the Filecoin ICO, people have also tried to measure the risk they take that comes along with the investment. Since there is no implementation yet, the investor has to realy on what is written in the white-paper and consider hypothetical pros and cons of the project and its team members. Therefore, the autor of [45] introduced a simple measure by adding (or subtracting) on a scale from 1-5 for every hypothetical pro (or con). Table I summarizes this work and extends it with the knowledge accumulated while writing this paper.

TABLE I. ESTIMATION OF ECONOMICAL FEASIBILITY

Aspect	Rating
Image left after ICO	-2
Overestimating teams abilities and/or underestimating the cost	-2
Possible hurdles for users to use Filecoin once ready	-3
Ability to lure miners and customers away from Storj and Siacoin due to technical excellence	+4
Positioning for marketing opportunities and business partnerships	+3
Development team will respond to user demands	+1
Huge pool of essentially free resources available from potential miners	+2

Our estimation shows that the biggest advantage of the project is its technical excellence as well as the sophisticated marketing and its great partnerships. On the negative side we see two major weaknesses. First of all, the ICO left many investors and upcoming users look at the project with mixed feelings. Overestimating the complexity of the implementation is also expressed as an uncertainty and so is the hurdle of whether a user is going to use the project after all, or not.

A. Is it a Ponzi scheme?

Given that Filecoin raised a substantial amount of money with the ICO instrument, it is legitimate raise the question of whether Filecoin itself is a Ponzi scheme?

By definition, a ”Ponzi scheme” is a fraudulent investment operation that pays returns to its investors either from the investors own money or the money paid by subsequent investors, rather than from any actual profit earned by the company. [55]

Indeed, given Section IV there are multiple factors which support this argument:

- Full control over investments by Protocol Labs (IV-A)
- Exponentially higher return for early investors (IV-C)
- Reward for vesting (IV)
- Lack of implementation

On the other hand, there are numerous of logical reasons why an investor would *trust* the people behind the Filecoin project: The fact that Protocol Labs has proven enormous technical capabilities by building IPFS and libp2p leads to believe that mastering the upcoming hurdles by building Filecoin will be handled in equal elegance. Further more, the investments raised are controlled an intermediary (see IV) which holds connections to the SEC.

The authors of this paper conclude that the proven technical capabilities of Protocol Labs should naturally be reason enough to believe that Filecoin is a legitimate project. However, the terms and the way the ICO was proceeded is far from ideal. Essentially, Protocol Labs allowed themselves to remain as the single entity that holds full control over the raised assets, without giving investors the possibility to be able to intervene effectively. Therefore, if good will shall fade away, the entire project could be turned into a Ponzi scheme. *Note: this is a hypothetical scenario and we are not implying that this scenario will occur. Solely, we uncover possibilities.*

B. Back of the envelope calculation

After all, we would like to confront the reader with a speculation drawn from weakly supported statements and some hypothetical numbers. The speculation is meant to be taken with a pinch of salt, however, it shall also serve as an alert to the blindfolded bullish investor.

In a recent interview [37], Juan Benet compares Filecoin with Airbnb [43] where people can rent away storage, instead of their homes. Therefore, the following calculation compares the valuation of both companies against each other by opposing their resources. Apartment space for Airbnb and disk space for Filecoin: As of today (September 2017), Airbnb values at approximately $31 billion while holding around 3 million listings in total [38]. The average apartment in the United States was 934 square feet in 2016 [39]. In a hypothetical scenario, Airbnb is therefore valued $11.06 per square feet. If one would compare this number to the median price per square feet in the United States, which is $123 [40], the Airbnb ecosystem diminishes the median price by a factor of 11.12. The average price for hard drives in 2017 is $0.03 per Gigabyte [41]. Dividing the same factor as Airbnb applies for square feet to the storage price, the average Gigabyte in the Filecoin system results in $0.0027. This in fact means, the $257′000′000 launched at the Filecoin ICO require 95′185′185′185 Gigabyte (95′185 Petabyte) of storage to be offered by storage miners. Considering that Dropbox [44] holds currently around 500 Petabyte of user data [42], one could argue that Filecoin is overvalued.

VIII. Conclusion

References

[1] ”InterPlanetary Record Store”, https://github.com/libp2p/interface-record-store, accessed: October 17, 2017.

[2] ”The Inter-Planetary Naming System”, https://github.com/ipfs/examples/tree/master/examples/ipns, accessed: October 13, 2017.

[3] ”IPFS - Content Addressed, Versioned, P2P File System”, https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/ipfs accessed: October 13, 2017.

[4] ”Bitswap”, https://github.com/ipfs/go-ipfs/tree/master/exchange/bitswap, accessed: October 13, 2017.

[5] Cohen, Bram. ”Incentives build robustness in BitTorrent.” Workshop on Economics of Peer-to-Peer systems. Vol. 6. 2003.

[6] Baumgart, Ingmar, and Sebastian Mies. ”S/kademlia: A practicable approach towards secure key-based routing.” Parallel and Distributed Systems, 2007 International Conference on. IEEE, 2007.

[7] Freedman, Michael J., Eric Freudenthal, and David Mazieres. ”Democratizing Content Publication with Coral.” NSDI. Vol. 4. 2004.

[8] Freedman, Michael J., and David Mazieres. ”Sloppy hashing and self-organizing clusters.” International Workshop on Peer-to-Peer Systems. Springer, Berlin, Heidelberg, 2003.

[9] ”The network address multiformat”, https://github.com/multiformats/multiaddr, accessed: October 11, 2017.

[10] Rosenberg, Jonathan. ”Interactive connectivity establishment (ice): A methodology for network address translator (nat) traversal for offer/answer protocols.” (2010).

[11] ”Self describing hashes - for future proofing ”, https://github.com/multiformats/multihash, accessed: October 11, 2017.

[12] ”IPFS - The Permanent Web”, https://github.com/ipfs/ipfs/, accessed: October 11, 2017.

[13] Markwell, John, and David W. Brooks. ”Broken links: The ephemeral nature of educational WWW hyperlinks.” Journal of Science Education and Technology 11.2 (2002): 105-108.

[14] Fielding, R., et al. ”RFC 2616.” Hypertext Transfer ProtocolHTTP/1.1 2.1 (1999): 2-2.

[15] ”HTTP is obsolete. It’s time for the distributed, permanent web”, https://ipfs.io/ipfs/QmNhFJjGcMPqpuYfxL62VVB9528NXqDNMFXiqN5bgFYiZ1/its-time-for-the-permanent-web.html, accessed: October 11, 2017.

[16] ”Amazon S3”, https://aws.amazon.com/s3/, accessed: September 28, 2017.

[17] ”ICOs vs Venture Capital”, http://fingfx.thomsonreuters.com/gfx/rngs/USA-VENTURECAPITAL-DIGITALCURRENCY/0100502J05M/index.html, accessed: September 28, 2017.

[18] ”ERC-20 token standard”, https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20-token-standard.md, accessed: September 28, 2017.

[19] Wood, Gavin. ”Ethereum: A secure decentralised generalised transaction ledger.” Ethereum Project Yellow Paper 151 (2014).

[20] ”DEVCON2”, https://ethereumfoundation.org/devcon2/, accessed: September 28, 2017.

[21] Nakamoto, Satoshi. ”Bitcoin: A peer-to-peer electronic cash system.” (2008): 28.

[22] ”FilecoinTokenSaleEconomics”,https://coinlist.co/static/media/Filecoin-Sale-Economics.0ae9a53f.pdf, accessed: September 25, 2017.

[23] ”Simple Agreement for Future Tokens”, https://coinlist.co/static/media/Protocol%20Labs%20-%20SAFT%20for%20Filecoin%20Token%20Presale.6ddb6fb6.pdf, accessed September 25, 2017.

[24] Blanchard, Olivier Jean. ”Speculative bubbles, crashes and rational expectations.” Economics letters 3.4 (1979): 387-389.

[25] Gastwirth, Joseph L. ”A probability model of a pyramid scheme.” The American Statistician 31.2 (1977): 79-82.

[26] ”AngelList”, https://angel.co/, accessed: September 22, 2017.

[27] ”CoinList”, https://coinlist.co/, accessed: September 22, 2017.

[28] ”Rule 506 of Regulation D”, https://www.sec.gov/fast-answers/answers-rule506htm.html, accessed: September 22, 2017.

[29] ”Tribler Project”, https://www.tribler.org/, accessed: September 22, 2017.

[30] ”Tribler Issue: blockchain-regulated markets”, https://github.com/Tribler/tribler/issues/2559#issuecomment-307353664, accessed: September 22, 2017.

[31] Wilkinson, Shawn, et al. ”Storj a peer-to-peer cloud storage network.” (2014).

[32] ”MaidSafe.net announces project SAFE to the community”, https://github.com/maidsafe/Whitepapers/blob/master/Project-Safe.md, accessed: September 22, 2017.

[33] FileCoin whitepaper, https://filecoin.io/filecoin.pdf, accessed: September 20, 2017.

[34] ”FileCoin Website”, https://filecoin.io/, accessed: September 20, 2017.

[35] ”Filecoin INvestor FAQ”, https://ipfs.io/ipfs/QmWdXyhqHJWJut5wt4gSCueTjSnFyDHBy3SRfmcqArtz1a/2017-08-08-Filecoin-Investor-FAQ.html, accessed: September 28, 2017.

[36] Vorick, David, and Luke Champine. ”Sia: Simple Decentralized Storage.” (2014).

[37] https://a16z.com/2017/09/14/networks-protocols-labs-tokens/

[38] https://expandedramblings.com/index.php/airbnb-statistics/

[39] https://www.cnbc.com/2017/03/09/airbnb-closes-1-billion-round-31-billion-valuation-profitable.html

[40] http://www.realtytrac.com/news/home-prices-and-sales/march-q1-2016-home-sales-report/

[41] https://www.backblaze.com/blog/hard-drive-cost-per-gigabyte/

[42] https://blogs.dropbox.com/tech/2016/03/magic-pocket-infrastructure/

[43] ”Airbnb”, https://airbnb.com, accessed: September 20, 2017.

[44] ”Dropbox”, https://dropbox.com, accessed: September 20, 2017.

[45] ”Filecoin ICO analysis”, https://hacked.com/ico-analysis-filecoin/, accessed: September 20, 2017.

[46] Karger, David, et al. ”Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web.” Proceedings of the twenty-ninth annual ACM symposium on Theory of computing. ACM, 1997.

[47] Wilcox-OHearn, Bryce. ”Experiences deploying a large-scale emergent network.” Peer-to-Peer Systems (2002): 104-110.

[48] Maymounkov, Petar, and David Mazieres. ”Kademlia: A peer-to-peer information system based on the xor metric.” International Workshop on Peer-to-Peer Systems. Springer, Berlin, Heidelberg, 2002.

[49] Poon, Joseph, and Thaddeus Dryja. ”The bitcoin lightning network: Scalable off-chain instant payments.” Technical Report (draft) (2015).

[50] Kamara, Seny, and Kristin E. Lauter. ”Cryptographic Cloud Storage.” Financial Cryptography Workshops. Vol. 6054. 2010.

[51] Ateniese, Giuseppe, et al. ”Provable data possession at untrusted stores.” Proceedings of the 14th ACM conference on Computer and communications security. Acm, 2007.

[52] Shacham, Hovav, and Brent Waters. ”Compact proofs of retrievability.” Asiacrypt. Vol. 5350. 2008.

[53] Sasson, Eli Ben, et al. ”Zerocash: Decentralized anonymous payments from bitcoin.” Security and Privacy (SP), 2014 IEEE Symposium on. IEEE, 2014.

[54] ”Power Fault Tolerance”, https://filecoin.io/power-fault-tolerance.pdf, accessed: October 31, 2017.

[55] ”Ponzi Scheme”, https://www.sec.gov/fast-answers/answersponzihtm.html, accessed: November 05, 2017.

[56] ”libp2p: Modular peer-to-peer networking stack”, https://github.com/libp2p/specs, accessed: October 31, 2017.