Filecoin Research Roadmap for 2017

Juan Benet
Protocol Labs

July 4, 2017

Abstract

This document describes the Filecoin Research Roadmap for 2017, including background, results achieved so far and to be achieved, known future work, remaining open problems, and more. The works outlined are *works in progress*, and are published as *technical reports* in need of further work. They have received significant amount of peer-review, but more is needed before we are comfortable declaring them published as proper results. Any open problem listed is an invitation to join us and contribute to their solution.

1 Background

The first version of the Filecoin protocol was published on July 15, 2014. Without diving into detail, it was a useful achievement at the time, but left some things as open problems. It wrapped a *Proof-of-Retrievability* for a growing dataset into the mining function, effectively forcing miners to offer a storage service as part of the mining process. It introduced how to store and retrieve data while keeping miners focued on storing as many different pieces of users' data as they could. It proposed the distinction between the price of the currency and the price of the storage itself, allowing the currency to float above. It proposed ways for users to tune the replication and availability guarantees. And more, but it did not fully eliminate the use of *Proof-of-Work* – or reliance on another consensus protocol. This problem would be ideal to solve in such a network. It also needed scalability improvements.

In the subsequent three years, a lot changed.

First, our group spent the majority of that time building, deploying, and growing the *InterPlanetary File System* (IPFS), Filecoin's sister protocol and dependency. The original intention was to build IPFS and turn back to Filecoin in late 2015. However, the quick success and significant adoption of IPFS by a large swath of groups – and the lack of applications that *require* decentralized storage networks like Filecoin – kept us working on IPFS. In building and deploying IPFS, in studying our users' needs, and in understanding how these networks must operate, we were able to refine the requirements for Filecoin and pave a road for adoption into the hands of thousands of developers and organizations. These have real-world practical use cases, which give us a clear picture of desired features, target system performance, required system guarantees, and scalability trajectories. IPFS use-cases and specific user groups have also built a market need for Filecoin.

Second, Ethereum was implemented, shipped, and matured as a platform. Ethereum paved the way for the next generation of blockchain protocols, it proposed and tested dozens of new ideas, it put smart-contracts into large-scale use, it built a vibrant ecosystem where many contracts and apps transact with each other (and need to store or read files), and much more. Ethereum's technology stack has developed and matured to a point where large scale businesses and operations are depending on it. The EVM is developing as a potential standard for smart-contracts. Finally, the blockchain market developed and blockchain applications now *require* a decentralized storage network.

Third, academia and industry have proposed many improvments and whole new blockchain and consensus systems. Notable and relevant improvements include: (a) *Payment Channels*, as in the *Lightning Network*, and their evolutions (*State Channels*, *Sprites*); (b) the first provable *Proof-of-Stake* protocols, and several

variant protocol constructions; the use of *Proofs-of-Space* and *Proofs-of-Storage* for *Proof-of-Work* (Permacoin, Spacemint, and more); the use of zkSNARKs and a trusted setup in a blockchain (Zcash); the invention of zkSTARKs which give SCIP without requiring a trusted setup (*transparency*); and more.

Fourth, Bitcoin and Ethereum mining have created the worlds most powerful computer networks. The magnitude of Bitcoin's hash rate is staggering: Bitcoin is pushing 6 exa-hashes per second! This is an enormous amount of computing power, and an enormous amount of *proof-of-work* energy waste. About 13.72 terawatts are spent in mining Bitcoins, equivalent to the consumption of the entire country of Slovenia. The sheer magnitude of miner revenues is also staggering: Bitcoin miners are earning about $4.4 Billion USD between 2013 and 2017 (priced per year, including EOY 2017 estimate); Ethereum miners are earning $2.7 Billion USD between 2015 and 2017. This is no longer seen as an absurd, eccentric, bound-to-fail scheme, but rather as an odd but highly profitable indusry. An entire industry of special-purpose machines (the ASICs) sprang up, an ecosystem of businesses is meeting the demand for more hash power, and everything else miners need. All of this means that (a) tens of thousands of crypto miners seek to mine as much as they can; (b) large businesses with lots of computing resources are much more willing to entertain mining crypto assets; and (c) the *proof-of-work* waste continues its unabated exponential growth. The original Filecoin aim to completely replace useless *proof-of-work* with a useful storage service is much more needed than ever.

Fourth, the broader industry landscape has developed to the point where both massive and small organziations are trusting blockchain systems for their critical operations. More and more decentralized applications are being built. More and more businesses need a decentralized storage network. The exponential growth of data continues to demand massive expenditures in data storage world-wide, though the prices for storing and transmitting data remain quite high (storing and distributing a PB of data in the cloud can cost half a million dollars per year). Lots of regional storage players are being put out of business by competitors who can provide storage world-wide. Regional ISPs and interconnects now house CDN *points-of-presence* world-wide, and ISP ability to cache is decreasing due to the move to https. The storage market needs to be optimized much more granularly, and algorithmically. Businesses are much more likely to trust blockchain solutions. Vast amounts of latent storage exist around the world, unused and depreciating. The market is ripe for Filecoin, and it needs to scale fast.

2 2017 Q1, Q2: Filecoin v2 and New Results

This changing landscape spelled out new requirements. With these in mind, and the relevant recent results, we set on the path to completely upgrade the Filecoin protocol. In early 2017 we had a very successful research period, we combined solutions achieved over the last few years, and discovered several more. All this work culminated in several important results, and the desigin of a new Filecoin protocol. Some of the results have broad applicability and promise to be fundamental achievements in their sub-field. Here is a listing of some of our results:

- The most important result is that we finally found a way to completely do *useful storage proof-of-work* mining, achieving Filecoin's long-term goal. We do this with *sequential* and frequent *Proofs-of-Replication*. A miner's influence over the consensus (security threshold) and ability to win blocks (profits) are linearly proportional to the amount of storage they provide to users in the network. Reward is not super-linear in terms of the storage, which would be a centralizing problem; the process has little computational overhead; and no advantages are conferred by amassing parallel computational power.

- The above result required several other results. One of them is the discovery of *Proofs-of-Replication*, a new class of *Proof-of-Storage* protocols. *Proof-of-Replication* schemes allow a prover $\mathcal{P}$ to prove to a verifier $\mathcal{V}$ that $\mathcal{P}$ is storing (or has low latency access to) a specific replica $\mathcal{R}^{\mathcal{D}}_{ek}$ of some data $\mathcal{D}$, under some agreed-upon encoding key ek. The proof proves that the replica has been stored in independent physical storage and has not been deduplicated with other replicas. The proof also proves that the replica was actually physically stored and was not computed on demand. This is achieved with an arbitrarily slowable PRP, used to create per-replica encodings.

- Then, we identified the use of *sequential* and *non-parallelizable Proofs-of-Storage* (or even *Proofs-of-Space*), to confirm that a particular amount of storage was used in a particular way (eg storing data $\mathcal{D}$, or a specific replica $\mathcal{R}_{ek}^{\mathcal{D}}$) for some duration of time. We call this sequence of proofs a *Proof-of-Spacetime* because it proves that a particular amount of storage *space* was occupied by a particular data $\mathcal{D}$ for a specific amount of *time*. This could be a growing sequence, auditable randomly and on-demand, or a sequence meant to only last for a well-known amount of time. The intuition here is that a prover $\mathcal{P}$ can produce a *Proof-of-Spacetime* that proves to a verifier $\mathcal{V}$ that $\mathcal{P}$ was storing data $\mathcal{D}$ for a period of time $\mathcal{T}$, between two points in time (when $\mathcal{P}$ and $\mathcal{V}$ interact, or when $\mathcal{P}$ makes some commitment to a timed shared ledger).

- More results came from studying new *Proof-of-Stake* consensus protocols to learn new strategies that manage to commit miners' stake to a single candidate value per protocol epoch, and trying to apply our new *Proofs-of-Spacetime*. The first is a new framing for *Byzantine Fault Tolerance* that we call *Power Fault Tolerance* (PFT). This framing gives a much more accurate model for consensus blockchains using NakamotoConsensus, as the fault-tolerance assumptions are framed in terms of some continuous quantity of *power*, which is scarce in the network and is governed by per-protocol allocation and conservation rules. This result allowed us to understand consensus better, how to design systems that use some other scarce resource as *power* in consensus, and even suggested more systems to design. PFT uses a continuous measure of power, as blockchains tend to do, as opposed to the simple discrete number of participants that most BA protocol use. PFT is useful in thinking about various consensus protocols, and in modeling the properties they must satisfy.

- Another result we use in Filecoin is the *Expected Consensus* (EC), a new blockchain consensus (or byzantine consensus) protocol, that uses a *secret leader election* to pick – on expectation – a single leader per epoch. EC achieves finality based on the amount of *power* committed to a candidate history. EC is modular in that it can work on any notion of *power*.

We are currently writing four papers, three capture individual results usable in other contexts, and one is a new Filecoin paper. We are publishing these as *work in progress technical reports*, so that the ideas can be used by others, so that more peer-review and improvements come our way, and so that people can understand all the pieces that will go into Filecoin. We expect significant more work to go into these papers before they can be deemed finished, but we are well over half-way there.

- ***Proof-of-Replication*** (PoRep) introduces and analyzes *Proof-of-Replication* and *Proof-of-Spacetime*.
- ***Expected Consensus*** (EC) is new PFT consensus protocol one leader per epoch, and finality based on "power committed" to a candidate history.
- ***PFT: Power Fault Tolerance*** is a re-framing of *Byzantine Fault Tolerance* in terms of *power* and *influence* over the outcome of the protocol.
- ***Filecoin: Decentralized Storage Network*** is a full paper describing the new Filecoin protocol, and all its improvements.

Below, we include the abstracts of these papers. The papers themselves will be available on our websites. These papers will be completed during the next few months.

2.1 *Proof-of-Replication*

We introduce *Proof-of-Replication* (PoRep), a new kind of *Proof-of-Storage*, which allows proving data $\mathcal{D}$ has been *replicated* to its own uniquely dedicated physical storage. Enforcing unique physical copies enables a verifier to check that a prover is not deduplicating multiple copies of $\mathcal{D}$ into the same storage space. This construction is particularly useful in *Cloud Computing* and *Decentralized Storage Networks*, which must be transparently verifiable, resistant to sybil attacks, and unfriendly to outsourcing.

This work (a) reviews *Proofs-of-Storage* and motivates use cases; (b) defines the novel *Proofs-of-Replication*,

which can be *publicly verifiable*, *transparent*, *authenticated*, and *time-bounded*; (c) shows how to chain *Proofs-of-Replication* to establish *useful Proofs-of-Spacetime*; (d) gives realizable constructions with zero storage overhead; and (e) provides concrete open problems and future work direction.

2.2 Expected Consensus

In this work we present the Expected Consensus, EC a probabilistic, authenticated, weakly-synchronous Byzantine Consensus protocol with $f < n/2$ fault tolerance. The protocol runs in epochs and it uses a deterministic but unpredictable secret leader election, which elects - on expectation - one leader per epoch, which create a new block and propagate it to the network.

2.3 PFT: Power Fault Tolerance

We introduce the Power Fault Tolerance (PFT) model, which reframes Byzantine Fault Tolerance (BFT) in terms of participants' *influence* over the outcome of the protocol, instead of their number. In PFT, n is the total power, and f is the fraction of power controlled by faulty or adversarial participants.

This work: (a) provides a formal definition and properties for PFT; (b) generalizes Byzantine Consensus (BC) protocols of different classes (*permissioned*, *permissionless*, and *federated*) into a single class of Power Consensus (PC); (c) explores new directions for PC protocols, particularly for *blockchains*, and protocols that can detect and make progress during catastrophic network partitions.

2.4 Filecoin: Decentralized Storage Network

The internet is in the middle of a revolution: centralized proprietary services are being replaced with decentralized open ones; trusted parties replaced with verifiable computation; brittle location addresses replaced with resilient content addresses; inefficient monolithic services replaced with peer-to-peer algorithmic markets. Bitcoin, Ethereum, and other blockchain networks have proved the utility of decentralized transaction ledgers. These public ledgers process sophisticated smart contract applications and transact crypto-assets worth hundreds of billions of dollars. These systems are the first instances of internet-wide Open Services, where participants form a decentralized network providing useful services for pay, with no central management or trusted parties. IPFS has proved the utility of content-addressing and decentralizes the web itself. The IPFS network sees billions of files served and used across a global peer-to-peer network. IPFS liberates data from silos, survives network partitions, works offline, routes around censorship, and gives permanence to digital information.

Filecoin is a *decentralized storage network* that turns cloud storage into an algorithmic market. It is powered by a blockchain and a native protocol token. Miners earn Filecoin by providing storage to clients. And clients spend Filecoin hiring Miners to store or distribute data. As in Bitcoin, Filecoin miners compete to mine blocks with sizable rewards, though Filecoin mining power is measured in storage in use. This creates a powerful incentive for miners to amass as much storage as they can, and rent it out to clients. The protocol weaves these amassed resources into a self-healing storage network that anybody in the world can rely on. The network replicates and disperses content, while automatically detecting and repairing replica failures. Clients can select replication parameters to protect against different threat models. Content is encrypted end-to-end at the client; storage providers do not have access to decryption keys. Filecoin works as an incentive layer on top of IPFS [**?**], which can provide storage infrastructure for any data. It is especially useful for decentralizing data, building and running distributed applications, and implementing smart contracts.

This work:

(a) Introduces the Filecoin network, gives an overview of the protocol, and walks through several components in detail.
(b) Formalizes *decentralized storage network* (DSN) schemes and their properties, then constructs Filecoin as a DSN.

(c) Introduces a novel class of *proof-of-storage* schemes called *proof-of-replication*, which allows proving that any replica of data is stored in physically independent storage.
(d) Introduces a novel useful-work consensus based on sequential *proofs-of-replication*, and storage as a measure of power.
(e) Formalizes verifiable markets and constructs two markets – a Storage Market and a Retrieval Market – which govern how data is written and read from Filecoin.
(f) Discusses the erasure-coding scheme Filecoin uses, how failures are detected, and how the network automatically self-heals.
(g) Discusses use cases, connections to other systems, and how to use the protocol.

3 2017 Q3, Q4: Implementation and more

The remainder of the year will be spent in finishing these papers, implementing Filecoin, making progress on outlined future work, and searching for solutions to a set of open problems. At this point, we consider the core of the new Filecoin protocol design to be stable, though we are open to improvements in some of its pieces. We are particularly keen on finding some constructions that reduce costs of some particularly expensive *publicly-verifiable* and *transparent* proofs.

3.1 Implementation

The majority of our research bandwidth in Q3 and Q4 2017 will be devoted to the implementation of the Filecoin protocol. This will require some amount of research work as we need to produce implementations of several constructions, and to tune several constructions so they work effectively in today's hardware. We expect this work to be hard but straightforward, open to but without requiring any new invention.

3.2 Future Work

There are many things we have left as future work. Some of these have already been specced out, some have been described in detail, and some are ideas only. These are no longer *Open Problems*, and are solved but require writing up or more work to cleanly adapt or use.

- A specification of the Filecoin state tree.
- Detailed performance estimates and benchmarks for Filecoin and its components.
- A full implementable Filecoin protcol specification.
- A *sponsored-retrieval* ticketing model where any client $\mathcal{C}_1$ can sponsor the download of another client $\mathcal{C}_2$ by issuing per-piece bearer-spendable tokens.
- A *Hierarchical Consensus* protocol where Filecoin subnets can partition and continue processing transactions during temporary or permanent partitions.
- Incremental blockchain snapshotting using SNARK/STARK, as described in [**?**].
- Filecoin – in – Ethereum interface contracts and protocols.
- Blockchain archives and inter-blockchain stamping with Braid.

3.3 Open Problems

There are a number of remaining open problems. None of these are show stoppers and none of them have to be solved before launch. That said, finding better solutions is always a good idea.

- A better primitive for the *Proof-of-Replication* Seal function, which ideally is $\mathcal{O}(n)$ on decode (not $\mathcal{O}(nm)$), and *publicly-verifiable* without requiring SNARK/STARK.
- A better primitive for the *Proof-of-Replication* Prove function, which is *publicly-verifiable* and *transparent* without SNARK/STARK.
- A *transparent*, *publicly-verifiable Proof-of-Retrievability* or other *Proof-of-Storage*.

- A better *secret leader election* for the *Expected Consensus*, which gives *exactly one* elected leader per epoch.
- A better trusted setup scheme for SNARKs that allows incremental expansion of public parameters (schemes were a sequence of MPCs can be run, where each additional MPC strictly lowers probability of faults and where the output of each MPC is usable for a system).

3.4 Proofs and Formal Verification

We would like to have formal proofs for as much of our work as we can. Unlike the vast majority of blockchain projects, we strongly value proofs and formal verification. We have a few in progress, and more in mind. But it will be hard, long-term work to prove many properties of Filecoin (such as scaling, offline, or fault-tolerant DSN properties, Filecoin automatic *self-healing* guarantees, etc). Further, we would like to have formally verified protocol specs and formally verified protocol implementations.

- Proofs of correctness for *Expected Consensus* and variants.
- Proof of correctness for *Power Fault Tolerance* asynchronous 1/2 impossibility result side-step.
- Proof of security for *Filecoin DSN*, and prove other guarantees.
- Formal model and proofs for automatic *self-healing* guarantees.
- Formally verify protocol descriptions (eg TLA+ or Verdi).
- Formally verify implementations (eg Verdi).

References