Filecoin Project Roadmap

Version History

- **2018-08-27.** Initial publication
- **2018-12-20.** Adjusted timelines
- **2019-02-13.** Adjusted timelines, added milestones, and updated go-filecoin completion
- **2019-09-24.** Adjusted timelines, added milestones, and updated go-filecoin completion
- **2019-12-11.** Adjusted milestones, added lotus, added link for Gantt chart

Table of Contents

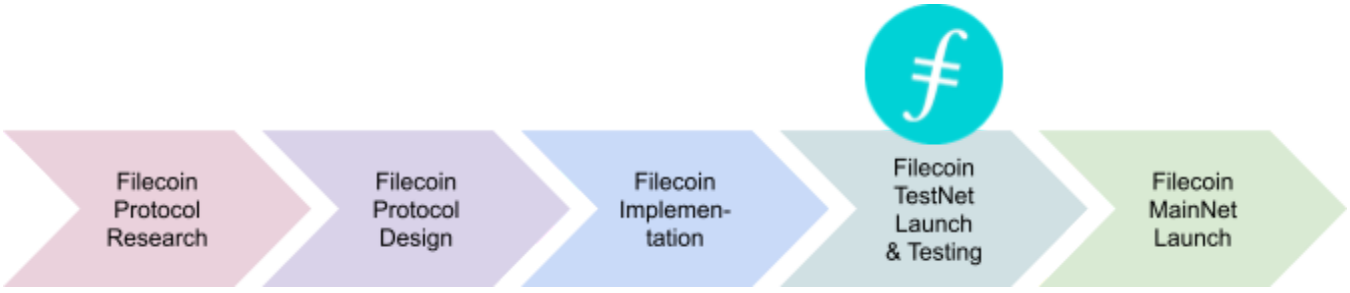
Optimistic Timeline

While we cannot commit to any exact dates on this timeline, we want to give at least an indication of where we are, what's next, and when our next milestones will be reached. We hate giving dates that may turn out to be too optimistic or slip, but we hate the silence and keeping our community in the dark even more. So, against typical mainstream software & product development wisdom, we are publishing this optimistic roadmap with expected dates of targeted milestones, and we plan to revise it as we need. The pros: a much clearer and transparent planning approach, easier coordination across the community, and excitement as milestones get closer. The cons: timelines will certainly have to change -- some things may come sooner or later than initially anticipated.

For more detailed timing information, check out our public Gantt chart.

Where we are

In rough terms, this is a high level view of where we are. (These phases are not equally long)



Upcoming Milestones

Our coming milestones are as follows:

- ✅**go-filecoin demos published (2018 Q3)**

    - As part of this update we are publishing a set of demos demonstrating features of **go-filecoin**.
- ✅**go-filecoin collaborator & contributor preview (2018 Q4)**
    - We will be inviting a number of collaborators & contributors to our codebases before opening them up. You can register your interest in this form.
    - The main goals of this are to improve documentation, take preliminary questions, and make general preparations for a public release.
    - This will also help us transfer knowledge to a broader set of contributors who can help us with the upcoming onslaught of questions, issues, and PRs.
- ✅**Opening up the go-filecoin codebase (ETA: 2019 Q1)**
    - We will be opening the github repositories hosting the go-filecoin implementation and related tools.
    - This will be a major point of involvement for the community at large, and we strongly encourage the participation of developers interested in developing Filecoin or in building applications on it.
    - In the short term, this will likely slow our implementation pace, as our team will face open source maintainer duties and spend time responding to questions from our broader developer community learning the codebase.
    - In the long run, this will speed up our implementation pace, as the amount of people who can contribute will increase.
- ✅**Launching the first public Filecoin devnets (ETA: 2019 Q1)**
    - We are testing and developing **go-filecoin** in test environments (devnets).
    - Use-case specific devnets are there for protocol developers, miners, and other users to work with
    - **devnet-nightly:** a public devnet reset nightly, for use by go-filecoin developers & contributors
    - **devnet-user:** a long-running public devnet, reset as needed, for miners, tool developers, and users
- ✅**Multiple implementations in development (ETA: 2019 Q3)**
    - We are working with a few teams to build additional implementations of the Filecoin protocol
    - Multiple implementations are extremely important for network security and community development purposes -- we aim to launch with more than one Filecoin implementation
    - If you are interested in building an additional Filecoin implementation, get in touch
- ✅**Launching the first long-running Filecoin alphanet (ETA: 2019 Q3)**
    - We are launching a go-filecoin alphanet (long-running network)
    - As additional functionality is implemented in go-filecoin, we will be able to upgrade the network without requiring network resets
    - This will allow us to test chain and network behavior over longer periods of time, important preparation before we launch our testnet next quarter
- ✅**Launching the first public Filecoin testnet (ETA: 2019 Q4)**
    - This will be a testnet of lotus nodes, and will gradually scale up -- starting with 100s of nodes, and ramping up orders of magnitude to 1,000s, 10,000s, and so on.
    - We will run our testnet in conditions as close to real as we can, so we will be pushing lots of data to testnet miners.
    - We will help onboard miners & clients signed up on our forms. If you have already completed the form, stay tuned!
- **Testing with sets of large scale miners (10 PB+) (ETA: 2020 Q1)**
    - We are getting close to running our first tests with very large scale miners.
    - We will be selecting from miners signed up on our form.
- **Security Review & Audit (ETA: 2020 Q1)**
    - Before we can launch, we need to conduct full security reviews and external audits of our codebases.
    - This is a critical step for secure software crypto-systems.
    - If you are interested in reviewing & auditing our code, get in touch.

- **Testnet phase 2 & Filecoin v1.0.0 feature freeze (ETA: 2020 Q1)**
    - Before launch we will reach a feature freeze for the Filecoin protocol and implementations, after which only bug fixes and polish before launch.
    - Testnet phase 2 will have lotus, go-filecoin, and other implementations interoperating on the network.
    - This enables reviews, audits, and preparations for launch.
- **Launching the Filecoin mainnet (ETA: 2020 Q1)**
    - Once we are certain we are safe to launch, we will set a target launch date.
- **Scaling the network (2020+)**
    - Once launched, we will work closely with miners seeking to onboard.
    - We will work on technology improvements to enable scaling, including solutions to general blockchain scaling problems (off chain, sharding, partition tolerance).
- **Growing the utility of the network (2020+)**
    - Once launched, we will work closely with clients seeking to use Filecoin.
    - We will work on integrations and supporting applications to enable broader uses of Filecoin.
- **Improving the Network (2020+)**
    - We are already researching significant improvements to Filecoin and blockchains in general.
    - We will continue improving the network, through a standard open source improvement proposal process.
- **Forming the Filecoin Foundation (2020+)**
    - Form and boot up the Filecoin Foundation, to safeguard the network, drive governance, and more.
    - Foundations are critical institutions in blockchain networks, and we have big hopes for ours. If you are interested in working with us on governance questions, driving future research and usage of Filecoin, and public storage utilities, please reach out.
- **And more.**
    - We have many more plans, but we are focused on the milestones listed above now.

go-filecoin Completion Overview

We are currently implementing core Filecoin protocol features and functionality. As one of our engineers puts it, “we are waist-deep in protocol.” The following feature set should give an indication of completion.

Legend: Green means “mostly done”, yellow means “work in progress”, and white (no highlight) means “still todo”.

- **Blockchain Basics**
    - Blockchain Data Structures - Blocks, State Tree, Messages, Actors
    - Nodes can craft and sign transactions (messages)
    - Nodes can propagate and receive transactions (msg pool)
    - Miners can assemble transactions into a block (block creation)
    - Nodes can propagate blocks via pubsub
    - Nodes can choose the heaviest (or highest quality) chain (consensus)
- **Repo (on-disk)**
    - Nodes can store and load the blockchain in local storage (on-disk)
    - Nodes can store, load, and mutate a configuration file
- **Networking (libp2p powered)**
    - Nodes can listen for incoming connections
    - Nodes can dial to and connect to other nodes

    - Nodes can accept incoming connections from behind a NAT (autorelay)
    - Nodes can establish secure (encrypted and authenticated) connections to each other
    - Nodes can connect to specific other nodes by network address
    - Nodes can connect to random sets of nodes discovered (gossip)
    - Nodes can propagate messages to each other (pubsub)
    - Nodes can propagate blocks to each other (pubsub)
    - Nodes can look up other nodes (lookup/kademlia)
    - Nodes can look up other nodes (lookup/v2)
- **Storage Mining**
    - Miners can pledge storage capacity
    - Miners can assemble pieces into sectors
    - Miners can seal sectors (PoRep.Encode) and generate a proof
    - Miners can submit Proofs-of-Spacetime to prove correct storage
    - Miners can re-pledge storage capacity
- **Proofs & Crypto**
    - Production implementation of Proofs-of-Replication (Setup, Prove, Verify)
    - Concept implementation of Proofs-of-Replication (Setup, Prove, Verify)
    - Production implementation of Proofs-of-Spacetime (Setup, Prove, Verify)
    - Concept implementation of Proofs-of-Spacetime (Setup, Prove, Verify)
    - Production implementation of Seal Commitment
    - BLS Signature aggregation
    - Blocks aggregate all signatures
- **Storage Market**
    - Miners can submit Storage Market Asks (on chain)
    - Clients can submit Storage Market Bids (on chain)
    - Clients and Miners can find matching parties (off chain)
    - Clients can propose Storage Market Deals to Miners (off chain)
    - Miners can approve Storage Market Deals (off chain)
    - Clients can send data pieces to Miners (off chain)
    - Miners can check deal data, and finalize Deals (off chain)
    - Miners or Clients can post resulting Deals (on chain)
- **Retrieval Market**
    - Clients can set up payment channels to Miners (off chain)
    - Miners can fulfill Client requests (off chain)
- **Repair**
    - Clients can chunk files into small pieces
    - Clients can disperse pieces to many different miners manually
    - The Network can automatically detect sectors that have failed
- **Consensus**
    - Nodes can choose the heaviest (or highest quality) chain (consensus)
    - Nodes can choose the chain based on Expected Consensus rules
    - EC parameters and rules are finalized.
    - Nodes can handle null blocks
    - Nodes can handle blocks with more than 1 parent
- **VM and Actors**
    - Function invocations can be expressed in serialized, authenticated messages
    - Messages can be executed sequentially from mined blocks

    - Messages can be assembled into transactions with transactional semantics
    - Successful function invocations mutate the state tree
    - Failed function invocations do not mutate the state tree
    - VM code is metered and will halt if gas runs out (failed invocation)
    - Actors mutate their local storage according to the messages their receive
    - System Actors are all implemented
- **System Actors**
    - Null Actor
    - Account Actor
    - Storage Market Actor
    - Storage Miner Actor
    - Payment Channel Broker Actor
    - Multisig Actor
    - Storage Power Actor
    - Cron Actor
- **Wallets**
    - go-filecoin supports **secp256k1** keys
    - go-filecoin supports **BLS** keys
- **Filecoin Token**
    - Users can send payments (tokens) to each other
    - Tokens can be sent to smart contracts as part of an invocation
    - Tokens buy gas according to gas price metering (Ethereum model)
    - Tokens can be transferred between actors (inc system actors)
    - Tokens can have vesting
    - Tokens can have locking
- **Formats**
    - Address Format
    - On-disk Chain Data Structures Representation
    - On-wire Chain Data Structures Representation
- **Tool & API**
    - Core filecoin node API is well defined
    - Core filecoin node API is fully implemented
    - Core filecoin node API is exposed via network api (cmdkit JSON endpoint, etc)
    - Core filecoin node API is callable via cli
    - go-filecoin long-lived node can run as a daemonized process
    - go-filecoin can interact with a long-lived node
    - go-filecoin node can be instrumented and orchestrated according to production standards
    - go-filecoin has versioning information

lotus Completion Overview

The following feature set should give an indication of completion for the **lotus** codebase.

Legend: Green means “mostly done”, yellow means “work in progress”, and white (no highlight) means “still todo”.

- **Blockchain Basics**
  - Blockchain Data Structures - Blocks, State Tree, Messages, Actors
  - Nodes can craft and sign transactions (messages)
  - Nodes can propagate and receive transactions (msg pool)
  - Miners can assemble transactions into a block (block creation)
  - Nodes can propagate blocks via pubsub
  - Nodes can choose the heaviest (or highest quality) chain (consensus)
- **Repo (on-disk)**
  - Nodes can store and load the blockchain in local storage (on-disk)
  - Nodes can store, load, and mutate a configuration file
- **Networking (libp2p powered)**
  - Nodes can listen for incoming connections
  - Nodes can dial to and connect to other nodes
  - Nodes can accept incoming connections from behind a NAT (autorelay)
  - Nodes can establish secure (encrypted and authenticated) connections to each other
  - Nodes can connect to specific other nodes by network address
  - Nodes can connect to random sets of nodes discovered (gossip)
  - Nodes can propagate messages to each other (pubsub)
  - Nodes can propagate blocks to each other (pubsub)
  - Nodes can look up other nodes (lookup/kademlia)
  - Nodes can look up other nodes (lookup/v2)
- **Storage Mining**
  - Miners can pledge storage capacity
  - Miners can assemble pieces into sectors
  - Miners can seal sectors (PoRep.Encode) and generate a proof
  - Miners can submit Proofs-of-Spacetime to prove correct storage
  - Miners can re-pledge storage capacity
- **Proofs & Crypto**
  - Production implementation of Proofs-of-Replication (Setup, Prove, Verify)
  - Concept implementation of Proofs-of-Replication (Setup, Prove, Verify)
  - Production implementation of Proofs-of-Spacetime (Setup, Prove, Verify)
  - Concept implementation of Proofs-of-Spacetime (Setup, Prove, Verify)
  - Production implementation of Seal Commitment
  - BLS Signature aggregation
  - Blocks aggregate all signatures
- **Storage Market**
  - Miners can submit Storage Market Asks (on chain)
  - ~~Clients can submit Storage Market Bids (on chain)~~
  - Clients and Miners can find matching parties (off chain)
  - Clients can propose Storage Market Deals to Miners (off chain)
  - Miners can approve Storage Market Deals (on chain)
  - Clients can send data pieces to Miners (off chain)
  - Miners can check deal data, and finalize Deals (on chain)
  - Miners or Clients can post resulting Deals (on chain)
- **Retrieval Market**
  - Clients can set up payment channels to Miners (off chain)
  - Miners can fulfill Client requests (off chain)

- **Repair**
  - Clients can chunk files into small pieces
  - Clients can disperse pieces to many different miners manually
  - The Network can automatically detect sectors that have failed
- **Consensus**
  - Nodes can choose the heaviest (or highest quality) chain (consensus)
  - Nodes can choose the chain based on Expected Consensus rules
  - EC parameters and rules are finalized.
  - Nodes can handle null blocks
  - Nodes can handle blocks with more than 1 parent
- **VM and Actors**
  - Function invocations can be expressed in serialized, authenticated messages
  - Messages can be executed sequentially from mined blocks
  - Messages can be assembled into transactions with transactional semantics
  - Successful function invocations mutate the state tree
  - Failed function invocations do not mutate the state tree
  - VM code is metered and will halt if gas runs out (failed invocation)
  - Actors mutate their local storage according to the messages their receive
  - System Actors are all implemented
- **System Actors**
  - ~~Null Actor~~
  - Account Actor
  - Storage Market Actor
  - Storage Miner Actor
  - Payment Channel Broker Actor
  - Multisig Actor
  - Storage Power Actor
  - Cron Actor
- **Wallets**
  - lotus supports **secp256k1** keys
  - lotus supports **BLS** keys
- **Filecoin Token**
  - Users can send payments (tokens) to each other
  - Tokens can be sent to smart contracts as part of an invocation
  - Tokens buy gas according to gas price metering (Ethereum model)
  - Tokens can be transferred between actors (inc system actors)
  - Tokens can have vesting
  - Tokens can have locking
- **Formats**
  - Address Format
  - On-disk Chain Data Structures Representation
  - On-wire Chain Data Structures Representation
- **Tool & API**
  - Core filecoin node API is well defined
  - Core filecoin node API is fully implemented
  - Core filecoin node API is exposed via network api (cmdkit JSON endpoint, etc)
  - Core filecoin node API is callable via cli

- lotus long-lived node can run as a daemonized process
- lotus can interact with a long-lived node
- lotus node can be instrumented and orchestrated according to production standards
- lotus has versioning information