Programming on Filecoin

Once data has been stored, it is possible to run computations and calculations on that data, without needing to retrieve the data from a storage provider. This page covers the basics of how programmin

Compute-over-data

When it comes to data, a common need beyond storage and retrieval is data transformation. The goal with the compute-over-data protocols is generally to perform computation over IPLD, which is the data layer used by content-addressed systems like Filecoin. There are working groups working on different types of computing on Filecoin data, such as large-scale parallel compute (e.g., Bacalhau) and cryptographically verifiable compute (e.g.Lurk), etc.

For example,Bacalhau is a platform for public, transparent, and optionally verifiable distributed computation. It enables users to run arbitrary Docker containers and WebAssembly (wasm) images as tasks against data stored in the InterPlanetary File System (IPFS).

It is worth noting that Filecoin is uniquely positioned to support large-scale off-chain computation since the storage providers have to compute resources such as GPUs and CPUs colocated with their data. By supporting compute-over-data on the Filecoin network, we enable a new paradigm of computing on the data where the data exists rather than moving the data to external compute nodes.

Filecoin virtual machine

The Filecoin virtual machine (FVM) is a runtime environment for smart contracts on the Filecoin network. Smart contracts enable users to run any bounded computation, including those that create and enforce rules for storing and accessing data on the network. The FVM is responsible for executing these smart contracts and ensuring they are executed correctly and securely.

FVM is designed to support native Filecoin actors written in languages that compile to WASM, as well as smart contracts written for foreign runtimes, including Solidity contracts for Ethereum Virtual Machine (EVM), Secure EcmaScript (SES), and eBPF. Thereference FVM and SDK are written in Rust.

According to the FVM roadmap, we initially support smart contracts written in Solidity and eventually support any language that compiles to WASM.

The FVM enables compute-over-states on the Filecoin network and allows developers to build endless new use cases on top of Filecoin. Some example use cases are:

Data organizations

FVM can create a new kind of organization – one built around datasets of various kinds.

Data DAO and tokenized datasets

The FVM enables the creation and management of data-based decentralized and autonomous organizations – data DAOs. The FVM allows a group of individuals, or organizations, to curate and preserve data collection. Data DAOs can govern and monetize data access and pool the returns into a shared treasury to fund the collections preservation and long-term growth. One could even exchange those data tokens between peers and request computation services on that data, such as validation, joins, analysis, feature detection, and extraction, moving into machine learning.

Perpetual storage

FVM allows users to store once and have repair and replication bots manage the repetitive storage deal creation tasks so that data can be stored perpetually. Using a smart contract, users can provision a wallet with FIL, and storage providers can use that to fund data storage permanently. Repair bots can monitor the storage deals and replicate the data with other storage providers when necessary. This process gives users long-term storage permanence.

Financial services for miners

FVM can provide a variety of financial services for storage providers. The needs of these SPs are unique to the Filecoin ecosystem.

Lending and staking protocols

Users can lend out Filecoin to storage providers to use it as storage collateral and receive interest in return. These loans can be undercollateralized based on the on-chain storage history of past storage provider performance. Community members can use this history to generate reputation scores, enabling everyone to identify good borrowers. On top of that, loans can be automatically paid back to investors by using a multisig as the storage provider’s owner address, including lenders and a third party, to help negotiate payback. New FVM-enabled smart contracts give every FIL token holder access

to new yield opportunities on their holdings while also benefiting the whole Filecoin economy by allowing entry ramps for providing storage on the network.

Insurance

SPs need to have financial products that help protect them from the risk they are undertaking in creating more storage solutions. Certain characteristics such as payment history, length of operation, and availability can be used to craft insurance policies just as they can be used to underwrite loans to SPs. This can protect them from the financial consequences of active faulting or token price drops.

Core chain infrastructure

We expect that FVM will gain feature parity with other chains that persist. This is required for any EVM chain to operate but is not necessarily tied to storage primitives.

Decentralized exchanges

Users on FVM need to be able to exchange FIL for other tokens issued on-chain. This may be a decentralized exchange such as a fork of Uniswap or Sushi or involve building a decentralized order book similar to Serum on Solana.

Token bridges

While not immediately on the roadmap, bridges are needed from EVM chains, Move chains, and Cosmos chains to bring wrapped tokens from other ecosystems into the fold. With the current launch, we are more focused internally since the value proposition of Filecoin is unique enough that it does not need to bootstrap TVL from other chains. However, in the long run, we expect FVM to be part of a broader family of blockchains.

Besides these, there are a lot more use cases that the FVM could enable, such as data access control (Medusa), retrieval and trustless reputation systems, replication workers, storage bounties, and L2 networks. To learn more about what you can build on top of FVM, check out our Request for Startup post.

If you are interested in building these use cases, there is a list of solution blueprints that might help as a reference point regarding how some of these could work on a high level:

- DataDAO Solution Blueprint
- Perpetual Storage Solution Blueprint
- Lending pool cookbook
-

Filecoin EVM

The Filecoin EVM runtime (FEVM) is the Ethereum Virtual Machine (EVM) virtualized as a runtime on top of the Filecoin Virtual Machine (FVM). It will allow developers to port any existing EVM-based smart contracts straight onto the FVM and make them work out of the box. FEVM emulates EVM bytecode at the low level, supporting contracts written in Solidity, Vyper, and Yul. The EVM foreign runtime is based on preexisting OSS libraries, including SputnikVM and Revm. You can find out more details in the EVM <> FVM mapping specification .

Because Filecoin nodes offer the Ethereum JSON-RPC API support, FEVM is also completely compatible with any EVM development tools, such as Hardhat, Brownie, and MetaMask. Most smart contracts ported to Filecoin shouldn't require changes or auditing. For example, new ERC-20 tokens can be launched on the Filecoin network or bridged directly to token pools on other chains.

Developers can deploy actors on either the FEVM or native FVM; which one should you choose? The decision can be summed up as such: if you want better performance, write actors that are compiled to WASM and deployed to native FVM. If you are familiar with Solidity and want access to the EVM ecosystem of tools, but don't mind slightly less performance, deploy to the FEVM.

To sum it up, the FEVM allows current Web3 developers to quickly start writing actors on the Filecoin blockchain while using all of the tools, software packages, and languages they are used to while having access to Filecoin storage deals as a native.

The difference between FEVM and EVM contracts is that contracts deployed on FEVM can interact with built-in actors to interact with Filecoin-specific actors, such as miner actors, as mentioned in the built-in actor section. This allows developers to build Filecoin-native decentralized applications for the new use cases mentioned above. Smart contracts deployed to the Ethereum blockchain have no direct access to the Filecoin network or Filecoin-specific actors.

To allow Solidity smart contracts on FEVM to seamlessly call methods on Filecoin built-in actors and access Filecoin-specific syscalls idiomatically, a Filecoin-Solidity API library has been developed, you can use it for building your use cases, such as interacting with storage deals.

If you build on FEVM, you might find some of the example contracts here helpful.

Previous Retrieval market Next Networks

Last updated7 months ago