The Filecoin Virtual Machine

The Filecoin Virtual Machine (FVM) is a runtime environment enabling users to deploy their own smart contracts on the Filecoin blockchain. This page covers the basics of the FVM.

Introduction

Filecoin's storage and retrieval capabilities can be thought of as the base layer of the Filecoin blockchain, andFVM can be thought of as a layer on top of Filecoin that unlocks programmability on the network (e.g. programmable storage primitives).

Whereas other blockchains do have smart contract capabilities, FVM's smart contracts can use Filecoin storage and retrieval primitives with computational logic conditions. FVM will also enable Layer 2 capabilities, such as "compute over data" andcontent delivery networks .

Some additional notes about FVM's technical specifications:

- WASM-based: The FVM is a WASM-based polyglot execution environment for IPLD data, meaning that FVM gives developers access to IPFS / IPLD data primitives and can accommodate smart contracts (actors) written in any programming language that compiles to WASM.
- FEVM Compatibility: Are you an Ethereum / Solidity developer? You can build the next killer app on FVM and make use of theFilecoin Solidity library
- . Learn more about how FVM is Ethereum runtime and solidity compatible in the next section.
- VM Agnostic: The FVM is built to be VM-agnostic, meaning support for other foreign VMs can be added in the near future. Future versions of FVM can serve as a useful hypervisor enabling cross run-time invocations.
-

FVM brings user programmability to Filecoin, unleashing the enormous potential of an open data economy through various applications.

Use Cases

FVM Actors enable a huge range of use cases to be built on Filecoin. Here are just a few potential examples:

- Data Access Control: FVM Actors can enable a client to grant retrieval permission for certain files to a limited set of third-party Filecoin wallet addresses.
- DataDAO: FVM Actors can enable the creation of decentralized autonomous organizations where members govern and manage the storage, accessibility, and monetization of certain data sets and pool returns into a shared treasury.
- Perpetual Storage: Because all Filecoin storage deals are time-limited, when a client makes a deal with a storage provider to store a data set with them, the client has to begin to consider whether they will want to renew this deal for the next time-period with the same storage provider or seek out other storage providers that may be cheaper. However, FVM enables a client to automatically renew deals or find a cheaper storage provider when the time limit of a given deal has reached maturity. This automated renewal of deals can persist, even in perpetuity, for as many cycles as can be financed by an associated endowment of FIL. FVM Actors enable the creation and management of this endowment.
- Replication: In addition to allowing a client to store one data set with one storage provider in perpetuity, FVM Actors enable data resiliency by allowing a client to store one data set once manually and then have the Actor replicate that data with multiple other storage providers automatically. Additional conditions that can be set in an automated replication Actor include choices about the geographic region of the storage providers, latency, and deal price limits.
- Leasing: FVM Actors enable a FIL token holder to provide collateral to clients looking to do a storage deal, and be repaid the principal and interest over time. FVM Actors can also trace the borrowing and repayment history of a given client, generating a community-developed reputation score.
-

Additional use cases enabled by FVM include, but are not limited to, tokenized data sets, trustless reputation systems, NFTs, storage bounties and auctions, Layer 2 bridges, futures and derivatives, or conditional leasing. Visit the Protocol LabsFVM Request for Startups page to see in-depth descriptions of the use cases you may want to build and how Protocol Labs prioritizes them.

Start building on the FVM

If you're ready to start building on the FVM, here are some resources you should explore:

- FVM Reference Implementation: TheGithub repo
- containing the reference implementation for FVM.
- FVM Quickstart Guide: The Quickstart guide will walk you through deploying your first ERC-20 contract on FVM. In addition to being provided this code, we also walk you through the developer environment set-up.
- Developing Contracts: If you are ready to build your dApp on FVM, you can skip ahead and review ourbest practices
- section for developing contracts. Here, you can find a guide for the Filecoin solidity libraries, details on tools such as Foundry, Remix, and Hardhat, and tutorials for calling built-in actors and building client contracts.

-

The next page will walk you through the process of deciding whether you need to use FVM’s programmatic storage when building a dApp with storage on Filecoin.

Previous Fundamentals Next Filecoin EVM runtime

Last updated4 months ago