

# FAQ

## Roadmap

Q: What's the roadmap of IPC, and will IPC have a beta version go live?

The IPC team is working on delivering IPC features incrementally based on the roadmap planning in milestones. We expect to deliver a market-ready GA version by Milestone 3. The rough roadmap of IPC is:

- ✓
- Milestone 1:
  - The testnet-ready version to deploy a Lotus-based IPC subnet.
- ✓
- Milestone 2.5:
  - IPC preview.
- - This milestone migrates away from the Eudico/Lotus stack to the Fendermint stack.
- - Developers can spin up new IPC subnets anchored on Filecoin's Calibration network for general purposes with the fast block time and finality, not customization at this stage.
- \*
- [WIP] Milestone 3:
  - Production-grade manually-created, customizable L2+ networks
  - Audited and tested a version of the Fendermint stack that is safe to deploy production apps and move customers' funds into (well-designed) subnets.
- Milestone 4
  - : Support for user-deployed Wasm actors + multi-subnet apps + QoL improvements.
- - Developments from the FVM team make it safe to support user-deployed Wasm actors and runtimes.
- - Streamlines the operation of multi-subnet/federated apps by introducing multi-hop networks and subnet-aware wallets.
- - Network-wide and subnet explorers, plus better infra recipes are introduced at this stage.
- \*
- 

Q: IPC is currently compatible with Filecoin and Ethereum. Will more L1 chains be supported in the future?

IPC is implemented to be compatible with Filecoin and EVM-compatible chains. The current focus is on improving the features, like general message passing, subnet customization, atomic upgrades, native multi-coin support, etc.

Once the IPC stack achieves stability and maturity to meet the requirements of general web3 scalability, detailed roadmaps will be established to support additional Layer 1 (L1) chains such as Bitcoin, Solana, etc.

## Design & Architecture

Q: What are the best use cases of IPC?

As a multi-chain scalability framework, IPC offers hyper customization, lower fees, and faster transactions for decentralized technology. It offers enhanced performance for decentralized ecosystems and dApps e.g. decentralized computation, AI/ML, and metaverse/gaming use cases, etc. Read more about different use cases of IPC [here](#).

Q: What is the difference between IPC and other Ethereum L2 solutions (e.g. Polygon, Arbitrum, and Optimism) How does it compare to rollups ?

IPC is a flexible scalability framework that can be configured to resemble rollups, side chains, and more. Using IPC, developers can build customized scaling L2 or L3 solutions (operating as independent side chains) on-demand while anchoring their trust to the root chains, like Filecoin, Ethereum, or even BTC in the future.

IPC achieves horizontal scalability by using a hierarchical subsystem of subnets. This allows for flexible scaling throughout the network and is not limited to the rootnets on Layer 2 solutions.

Q: What is the advantage of using IPC for scalability?

- Highly customizable on consensus, runtime, gas fee, throughput, and membership.
- Inherit security and other unique features from the rootnet.
- Multi-chain interoperability via FVM WASM-based polyglot execution environment.
- Efficient chain storage with [IPLD](#)

•  
Read more in [How IPC Compares](#) .

Q: Is deploying an IPC subnet more convenient for accessing the Filecoin storage network?

Supporting data storage and retrieval on Filecoin, natively in an IPC subnet, requires specific system calls enabled from Filecoin to IPC and is in the roadmap.

Q: How does the checkpoint work?

To learn more about checkpointing [here](#) .

Subnet customization

Q: Is it possible to launch an IPC subnet without a pledge required and block rewards?

Yes, it is possible since pledging and block rewards are configurable for an IPC subnet.

When you launch a subnet, the default configuration is to use FIL as the native token and not include block rewards. But you can configure the subnet to use any ERC20 token as the native token for the utility, fee, and block rewards in that subnet.

For the pledge, you have a couple of different subnet modes to choose from when you set up a subnet.

- Collateral
    - : Validator power is determined by the collateral staked, and then you need an arbitrary positive pledge.
  - Federate
    - : Validator power is assigned by the owner of the subnet, with no need for a pledge.
  - Static
    - : Validators need to provide a fixed initial pledge to join the subnet when the subnet is set up. After that, no validators can join the subnet.
- 

Q: Can we modify chain parameters like the message size limits and gas price within the subnet?

Currently, IPC utilizes the default values from Filecoin Mainnet, including the gas model and block limits.

However, a core feature of IPC is its high customizability. As part of the future roadmap, there are plans to introduce mechanisms that allow users to configure these chain parameters for a subnet.

Q: Can we configure the logic for selecting subnet validators based on the geography or physical location of nodes?

The concept behind the subnet actor in IPC's parent architecture is to allow for the inclusion of specific logic and requirements for individual subnets. While anyone can become a user of a subnet, the subnet administrators can restrict the pool of validators and define the capabilities of validators that can participate in their subnet.

The development plan is to incrementally add interfaces and "levels" to help subnet admins configure their subnet capabilities. This allows subnet admins to have a greater degree of control over the validators and capabilities of their subnets, enabling them to tailor the geography and physical node selection to align with their intended use cases and objectives.

Deploying & joining a subnet

Q: Can a node join the IPC subnet as a validator? Does it require staking or depositing enough funds?

When a subnet is created by its maintainer, the subnet permission mode will be configured for this subnet. The available options are collateral, federal, and static. So whether or not a node can join an IPC subnet depends on its permission mode configuration.

When creating a subnet in IPC, the subnet permission mode can be configured by its maintainer. There are three available options for the permission mode: collateral, federal, and static. The permission mode determines how a node can join the subnet based on its configuration.

- Collateral
    - : validators are required to provide collateral or meet the criteria to join the subnet.
  - Federate
    - : validators are assigned by the owner of the subnet. Not any node can join it as a validator.
  - Static
    - : Once a static subnet is created, a node cannot join that IPC subnet anymore.
- 

Q: How to leave a subnet and release the funds properly?

We can use the following command to leave a subnet as a validator.

...

Copy `ipc-clisubnetleave--subnet`

...

Leaving a subnet will release the collateral for the validator and remove all the validation rights from its account. This means that if you have a validator running in that subnet, its validation process will immediately terminate.

If an IPC account holds enough funds inside this subnet, it can release funds to the parent subnet from its subnet with `ipc-cli cross-msg release` command. Refer to [Release funds from a subnet](#) for more details.

Q: Who is responsible for generating and signing a checkpoint transaction for the parent subnet?

Subnets are periodically committing checkpoints to their parent every `bottomup-check-period` (parameter defined when creating the subnet) by the subnet validators. A bottom-up checkpoint will be submitted after it is populated, signed, and agreed on their validity by a majority of validators in the child subnet.

Once subnet validators have agreed on the bottom-up checkpoint to be submitted in the parent for a specific epoch, relayers need to pick up the checkpoint and submit it in the parent. Then relayers are rewarded through cross-net message fees for the timely submission of bottom-up checkpoints to the parent.

Q: What is the difference between ISA and IGA?

ISA (a.k.a subnet actor) is subnet-specific and user-defined, which means that subnet operators are free to implement their own. They define the specific logic for the subnet such as the collateral policy, minimum number of validators, or any other subnet-specific logic that needs to be implemented. This contract lives in the parent of the subnet.

The IGA (a.k.a gateway actor) is the contract that implements the logic for the IPC protocol. There is one per subnet and depending on if the network is behaving as a root network, a parent, or a child, it triggers different logic to propagate messages, enforce collateral requirements, etc.

Read more at [IPC Actors](#).

Developing on Subnet

Q: Can we deploy smart contracts within a subnet?

IPC used FVM (Filecoin virtual machine) as the runtime environment for the IPC subnet. Since FVM is built as a polyglot VM to enable on-chain programmability, IPC is initially compatible with Filecoin and Ethereum.

FVM is the transaction execution layer of the IPC subnet, allowing builders to deploy their smart contracts on top of subnets to build any use cases. Since FVM is EVM-compatible, builders can use EVM-compatible toolings to develop smart contracts for a subnet, such as solidity libraries, Hardhat, foundry, MetaMask, etc.

Q: How to send a transaction between a subnet and its parent subnet?

At the moment, the `ipc-cli` only expose commands to perform the basic IPC interoperability primitives for cross-net communication, which is the exchange of FIL (the native token for IPC) between the same address of a subnet. Mainly:

- `fund`
- `, which` sends native tokens from one public key address to the address in the child subnet.
- `release`
- that moves native tokens from one account in a child subnet to its counterpart in the parent.
- 

To learn how to use those `ipc-cli` commands for [cross-subnet messages](#).

The smart contract interaction between subnets is achieved by using GMP (General Message Passing), you can learn more about GMP [here](#).

Cost & Performance:

Q: Are the gas usage costs the same as the Filecoin mainnet (independent from gas price)?

Currently, IPC adopts the default values from the Filecoin mainnet, including the gas model and block limits. However, as part of the roadmap, there are plans to introduce mechanisms that allow for the configuration of gas-related parameters within IPC.

Q: What is the current maximum TPS the subnet can achieve? And how far is it from the target?

IPC client (Fendermint) uses Tendermint under the hood which potentially can reach [10k tps](#) in single-node lab conditions as a theoretical limit.

In practice, the actual TPS achieved with Tendermint can vary depending on various factors, including the number of validators in the network. As the number of validators increases, the TPS tends to decrease. Real-world deployments of Tendermint have been known to achieve TPS numbers around 1,000 to 4,000.

Those numbers are theoretical and might not be possible if we benchmark an IPC subnet since we've been focusing on adding features instead of performance optimizations.

[Previous Troubleshooting](#) Last updated 18 days ago On this page \* [Roadmap](#) \* [Design & Architecture](#) \* [Subnet customization](#) \* [Deploying & joining a subnet](#) \* [Developing on Subnet](#) \* [Cost & Performance](#):

Was this helpful? [Edit on GitHub](#)