Ethereum: Ethereum Improvement Proposals (EIPs)

return to the Ethereum Standards

Source: Ethereum Improvement Proposals (EIPs)

> *Ethereum Improvement Proposal (EIP) describe standards for the Ethereum platform, including core protocol specifications, client APIs, and contract standards.*

EIP status terms

Draft

> *an EIP that is open for consideration and is undergoing rapid iteration and changes.*

Last Call

> *an EIP that is done with its initial iteration and ready for review by a wide audience.*

Accepted

> *a core EIP that has been in Last Call for at least 2 weeks and any technical changes that were requested have been addressed by the author. The process for Core Devs to decide whether to encode an EIP into their clients as part of a hard fork is not part of the EIP process. If such a decision is made, the EIP will move to final.*

Final (non-Core)

> *an EIP that has been in Last Call for at least 2 weeks and any technical changes that were requested have been addressed by the author.*

Final (Core)

> *an EIP that the Core Devs have decided to implement and release in a future hard fork or has already been released in a hard fork.*

Deferred

> *an EIP that is not being considered for immediate adoption. May be reconsidered in the future for a subsequent hard fork.*

EIP Types

EIPs are separated into a number of types, and each has its own list of EIPs.

Standard Track

Describes any change that affects most or all Ethereum implementations, such as a change to the the network protocol, a change in block or transaction validity rules, proposed application standards/conventions, or any change or addition that affects the interoperability of applications using Ethereum. Furthermore Standard EIPs can be broken down into the following categories.

Core

Improvements requiring a consensus fork (e.g. EIP5, EIP101), as well as changes that are not necessarily consensus critical but may be relevant to "core dev" discussions (for example, the miner/node strategy changes 2, 3, and 4 of EIP86).

Networking

Includes improvements around devp2p (EIP8) and Light Ethereum Subprotocol, as well as proposed improvements to network protocol specifications of whisper and swarm.

Interface

Includes improvements around client API/RPC specifications and standards, and also certain language-level standards like method names (EIP6) and contract ABIs. The label "interface" aligns with the interfaces repo and discussion should primarily occur in that repository before an EIP is submitted to the EIPs repository.

Ethereum Request for Comment (ERC)

Application-level standards and conventions, including contract standards such as token standards (ERC20), name registries (ERC137), URI schemes (ERC681), library/package formats (EIP190), and wallet formats (EIP85).

Meta

Describes a process surrounding Ethereum or proposes a change to (or an event in) a process. Process EIPs are like Standards Track EIPs but apply to areas other than the Ethereum protocol itself. They may propose an implementation, but not to Ethereum's codebase; they often require community consensus; unlike Informational EIPs, they are more than recommendations, and users are typically not free to ignore them. Examples include procedures, guidelines, changes to the decision-making process, and changes to the tools or environment used in Ethereum development. Any meta-EIP is also considered a Process EIP.

Informational

Describes a Ethereum design issue, or provides general guidelines or information to the Ethereum community, but does not propose a new feature. Informational EIPs do not necessarily represent Ethereum community consensus or a recommendation, so users and implementers are free to ignore Informational EIPs or follow their advice.

Final ERCs

Please refer to the Ethereum Improvement Proposals (EIP) Final.

- EIP 20: ERC-20 Token Standard
- EIP 55: Mixed-case checksum address encoding
- EIP 137: Ethereum Domain Name Service - Specification
- EIP 141: Designated invalid EVM instruction
- EIP 155: Simple replay attack protection
- EIP 162: Initial ENS Hash Registrar
- EIP 165: ERC-165 Standard Interface Detection
- EIP 181: ENS support for reverse resolution of Ethereum addresses
- EIP 190: Ethereum Smart Contract Packaging Standard
- EIP 191: Signed Data Standard (DRAFT)
- EIP 211: New opcodes: RETURNDATASIZE and RETURNDATACOPY
- EIP 214: New opcode STATICCALL
- EIP 721: ERC-721 Non-Fungible Token Standard
- EIP 777: ERC-777 Token Standard
- EIP 1167: Minimal Proxy Contract
- EIP 1820: Pseudo-introspection Registry Contract

Interface ERCs

Please refer to the Ethereum Improvement Proposals (EIP) Interface .

- EIP 107: safe "eth_sendTransaction" authorization via html popup (DRAFT)
- EIP 234: `blockHash` to JSON-RPC filter options (DRAFT)
- EIP 695: Create `eth_chainId` method for JSON-RPC (DRAFT)
- EIP 712: Ethereum typed structured data hashing and signing (DRAFT)
- EIP 758: ERC-NN Subscriptions and filters for completed transactions (DRAFT)
- EIP 1102: Opt-in account exposure (DRAFT)
- EIP 1186: RPC-Method to get Merkle Proofs - eth_getProof (DRAFT)
- EIP 1193: Ethereum Provider JavaScript API (DRAFT)
- EIP 1474: Remote Procedure Call (RPC) specification (DRAFT)
- EIP 1767: GraphQL interface to Ethereum node data (DRAFT)
- EIP 1803: ERC-NN Rename opcodes for clarity (DRAFT)
- EIP 1898: ERC-NN Add `blockHash` to JSON-RPC methods which accept a default block parameter (DRAFT)