BIPs, SLIPs, EIPs and BOLTs: Four Proposal Series

October 28, 2026 · Document #411 · bip39-phrase.com



Photo by Jerry Wei on Unsplash

Compare BIPs, SLIPs, EIPs and ERCs, and Lightning BOLTs: their maintainers, scope and roles in wallet compatibility.

Contents

BIPs: Bitcoin proposals beyond consensus 2

SLIPs: shared conventions from SatoshiLabs 2

EIPs and ERCs: Ethereum changes and interfaces 3

BOLTs: Lightning protocol interoperability 3

FAQ 3

Sources 4

Example BIP39 word sequences 4

Reading a BIP39 phrase 5

See also 5

A wallet can use BIP39 recovery words, a SLIP-0044 coin type and an ERC-20 token interface without following contradictory standards. These documents govern different layers of the same product. Their prefixes identify proposal collections and communities; they do not establish a universal approval authority or guarantee that every wallet implements a document.

BIPs: Bitcoin proposals beyond consensus

Bitcoin Improvement Proposals describe changes, conventions and processes relevant to Bitcoin. Contributors submit proposals to the `bitcoin/bips` repository, where BIP editors administer the publication process. Editors check matters such as scope, formatting and readiness for inclusion. Assigning a number does not mean that Bitcoin users, developers or miners have agreed to implement the proposal.

BIPs cover more than Bitcoin's consensus rules. Some specify transaction or script changes; others describe wallet interoperability, application conventions or community processes. A consensus proposal requires adoption through the relevant software and network mechanisms. A wallet convention can spread through wallet implementations without changing which blocks Bitcoin nodes accept.

For self-custody, several BIPs commonly work together:

- BIP39 defines the conversion between entropy and mnemonic words, and derives a 512-bit seed using PBKDF2-HMAC-SHA512 with `2048` iterations and a passphrase that may be empty.
- BIP32 defines hierarchical deterministic key derivation, allowing a tree of keys to descend from a seed.
- BIP44 defines a derivation hierarchy: `m / purpose' / coin_type' / account' / change / address_index`.

These are distinct specifications. Supporting BIP39 alone does not establish which derivation paths, address types or assets a wallet supports.

SLIPs: shared conventions from SatoshiLabs

SatoshiLabs Improvement Proposals are maintained in the `satoshilabs/slips` repository under the SatoshiLabs project. Contributors can propose documents, while repository maintainers administer review and publication. SLIPs provide specifications and registries used by wallet developers, including conventions that extend across multiple blockchains.

A particularly important example is **SLIP-0044**, the registry of coin types used at BIP44's `coin_type'` level. Bitcoin's registered value is `0`, making that hardened path component `0'`. The registry coordinates assignments so applications can distinguish asset namespaces within the shared derivation structure.

Other SLIPs address different problems. SLIP-0010 specifies hierarchical key derivation for several elliptic curves. SLIP-0039 specifies a mnemonic secret-sharing scheme with groups and thresholds. SLIP-0039 shares and BIP39 mnemonic sentences have different formats and recovery procedures; a wallet needs explicit support for the relevant scheme.

The SLIP prefix therefore does not mean “a Bitcoin proposal awaiting promotion.” It identifies a separate publication series. A SLIP can complement a BIP, define an alternative mechanism or support ecosystems beyond Bitcoin.

EIPs and ERCs: Ethereum changes and interfaces

Ethereum Improvement Proposals cover Ethereum protocol changes, application standards and processes. EIP editors administer the proposal process and repository. Authors develop proposals through technical discussion and review; client developers and the wider Ethereum community determine implementation and adoption through the applicable coordination process.

ERC means Ethereum Request for Comments. ERCs are application-level standards and a category within the broader Ethereum Improvement Proposal framework. They describe standards such as token contracts and interfaces. ERC proposals are published through the Ethereum Improvement Proposal process and repository.

ERC-20 specifies a fungible-token interface, including functions such as `transfer` and `approve`. ERC-721 specifies an interface for non-fungible tokens. A wallet uses these interfaces to interpret assets and construct interactions with contracts. Neither standard specifies how the wallet must generate its recovery words.

This explains why an Ethereum wallet can combine Bitcoin-origin key standards with Ethereum application standards. BIP39 and BIP32 can govern seed and key derivation, SLIP-0044 can identify the coin type, and an ERC can govern interaction with a token contract. Compatibility must be checked at each layer.

BOLTs: Lightning protocol interoperability

BOLT stands for Basis of Lightning Technology. The specifications live in the `lightning/bolts` repository and are maintained collaboratively by Lightning protocol contributors, including developers of different implementations. Repository maintainers manage contributions; interoperability depends on implementations agreeing on protocol behavior.

BOLTs specify how Lightning participants communicate and operate payment channels. Their subjects include message encoding, channel establishment, commitment transactions, routing, payment forwarding and invoices. BOLT 11 specifies an invoice format; BOLT 12 specifies offers, invoice requests and related messages.

For wallet recovery, this distinction matters: a seed backup may restore on-chain keys without restoring usable Lightning channel state. Channel recovery depends on the wallet's backup design and implementation. Following BOLTs does not imply that every Lightning wallet accepts another wallet's channel backup.

When checking compatibility, identify the exact document and feature, then check the wallet's implementation. A proposal number, repository merge or published status alone cannot establish recovery compatibility, network adoption or complete feature support.

FAQ

Does publication make a proposal mandatory?

No. Publication records a specification or proposal. Its practical effect depends on adoption by the relevant software, users and network participants.

Can one wallet follow all four series?

Yes. A wallet offering on-chain Bitcoin, Ethereum assets and Lightning can use documents from all four collections for different functions.

What should I record alongside recovery words?

Record the wallet, assets, backup standard, derivation paths and whether a passphrase is required. Follow the wallet's separate instructions for Lightning backups.

Sources

- Bitcoin Improvement Proposals repository (github.com)
- BIP-0039: Mnemonic code for generating deterministic keys (github.com)
- BIP-0032: Hierarchical deterministic wallets (github.com)

> Continue reading on bip39-phrase.com: Bitcoin Improvement Proposal (BIP): Process and Deployed BIP List
> More on this topic: BIP Standards

Example BIP39 word sequences

Randomly generated sequences of words from the BIP39 English wordlist. None of them passes the BIP39 checksum, so no wallet will accept them: they show the format only. Never store funds with a phrase you found in a document.

Example 1											
1	armor	2	claim	3	slab	4	early	5	divert	6	join
7	bright	8	custom	9	rice	10	twelve	11	awesome	12	mistake

Example 2											
1	evoke	2	mistake	3	filter	4	torch	5	output	6	love
7	noble	8	tape	9	parent	10	then	11	clarify	12	outdoor

Example 3											
1	leisure	2	slender	3	jar	4	always	5	broken	6	safe
7	knock	8	solve	9	ozone	10	parent	11	metal	12	heavy

Example 4											
1	seed	2	nerve	3	hurt	4	gift	5	kitchen	6	paper
7	similar	8	wheel	9	aerobic	10	short	11	camp	12	winter

Example 5											
1	wolf	2	strong	3	recall	4	much	5	review	6	peace
7	seminar	8	polar	9	mean	10	ozone	11	destroy	12	member

Example 6											
1	jelly	2	identify	3	ride	4	balcony	5	weekend	6	fiscal
7	cost	8	hammer	9	slab	10	grocery	11	easy	12	risk

Example 7												
1	stairs	2	fee	3	around	4	clock	5	coin	6	jacket
7	aspect	8	situate	9	pond	10	success	11	wink	12	abstract

Example 8												
1	fish	2	equal	3	sun	4	flower	5	blue	6	wasp
7	vendor	8	glance	9	good	10	april	11	brass	12	warrior

Example 9												
1	plastic	2	warrior	3	fringe	4	seminar	5	tortoise	6	frame
7	february	8	meadow	9	raw	10	never	11	flee	12	rabbit

Reading a BIP39 phrase

Every word stands for an 11-bit number: its position in the 2048-word list, counted from 0. Here is Example 1 decoded word by word. Each word links to its page on bip39-phrase.com, where the list is numbered from 1.

#	Word	Index (0-2047)	Word no.	11 bits
1	armor	95	96	00001011111
2	claim	333	334	00101001101
3	slab	1623	1624	11001010111
4	early	553	554	01000101001
5	divert	510	511	00111111110
6	join	961	962	01111000001
7	bright	224	225	00011100000
8	custom	436	437	00110110100
9	rice	1481	1482	10111001001
10	twelve	1880	1881	11101011000
11	awesome	132	133	00010000100
12	mistake	1135	1136	10001101111

The 12 words carry 132 bits: 128 bits of entropy followed by a 4-bit checksum. The checksum must equal the first 4 bits of the SHA-256 hash of the entropy.

```
Entropy (hex): 0be5372ba293fcf04701b4b93d604246
```

This sequence ends in the checksum bits 1111, but SHA-256 of its entropy starts with 1000, so wallets reject it as an invalid phrase. That is the point of the checksum: about 15 of every 16 random or mistyped sequences fail it.

See also

- Why There Is No BIP39 Version 2 · October 16, 2026

- ERC-20, ERC-721, and ERC-1155 Token Standards Compared · September 10, 2026

Educational document. The example sequences are random and fail the BIP39 checksum: they are not wallets and must never hold funds. Security notes: bip39-phrase.com/security/