```
BIP: 123
Title: BIP Classification
Author: Eric Lombrozo <elombrozo@gmail.com>
Comments-Summary: No comments yet.
Comments-URI: https://github.com/bitcoin/bips/wiki/Comments:BIP-0123
Status: Active
Type: Process
Created: 2015-08-26
License: CC0-1.0
         GNU-All-Permissive
```

Abstract

This document describes a classification scheme for BIPs.

BIPs are classified by system layers with lower numbered layers involving more intricate interoperability requirements.

The specification defines the layers and sets forth specific criteria for deciding to which layer a particular standards BIP belongs.

Copyright

Motivation

Bitcoin is a system involving a number of different standards. Some standards are absolute requirements for interoperability while others can be considered optional, giving implementors a choice of whether to support them.

In order to have a BIP process which more closely reflects the interoperability requirements, it is necessary to categorize BIPs accordingly. Lower layers present considerably greater challenges in getting standards accepted and deployed.

Specification

Standards BIPs are placed in one of four layers:

1. Consensus
2. Peer Services
3. API/RPC
4. Applications

Non-standards BIPs may be placed in these layers, or none at all.

1. Consensus Layer

The consensus layer defines cryptographic commitment structures. Its purpose is ensuring that anyone can locally evaluate whether a particular state and history is valid, providing settlement guarantees, and assuring eventual convergence.

The consensus layer is not concerned with how messages are propagated on a network.

Disagreements over the consensus layer can result in network partitioning, or forks, where different nodes might end up accepting different incompatible histories. We further subdivide consensus layer changes into soft forks and hard forks.

Soft Forks In a soft fork, some structures that were valid under the old rules are no longer valid under the new rules. Structures that were invalid under the old rules continue to be invalid under the new rules.

Hard Forks In a hard fork, structures that were invalid under the old rules become valid under the new rules.

2. Peer Services Layer

The peer services layer specifies how nodes find each other and propagate messages.

Only a subset of all specified peer services are required for basic node interoperability. Nodes can support further optional extensions.

It is always possible to add new services without breaking compatibility with existing services, then gradually deprecate older services. In this manner, the entire network can be upgraded without serious risks of service disruption.

3. API/RPC Layer

The API/RPC layer specifies higher level calls accessible to applications. Support for these BIPs is not required for basic network interoperability but might be expected by some client applications.

There's room at this layer to allow for competing standards without breaking basic network interoperability.

4. Applications Layer

The applications layer specifies high level structures, abstractions, and conventions that allow different applications to support similar features and share data.

Classification of existing BIPs

Number	Layer	Title
1		BIP Purpose and Guidelines
2		BIP process, revised
9		Version bits with timeout and delay
10	Applications	Multi-Sig Transaction Distribution
11	Applications	M-of-N Standard Transactions
12	Consensus (soft fork)	OP_EVAL
13	Applications	Address Format for pay-to-script-hash
14	Peer Services	Protocol Version and User Agent
15	Applications	Aliases
16	Consensus (soft fork)	Pay to Script Hash
17	Consensus (soft fork)	OP_CHECKHASHVERIFY (CHV)
18	Consensus (soft fork)	hashScriptCheck
19	Applications	M-of-N Standard Transactions (Low SigOp)
20	Applications	URI Scheme
21	Applications	URI Scheme
22	API/RPC	getblocktemplate - Fundamentals
23	API/RPC	getblocktemplate - Pooled Mining
30	Consensus (soft fork)	Duplicate transactions
31	Peer Services	Pong message
32	Applications	Hierarchical Deterministic Wallets
33	Peer Services	Stratized Nodes
34	Consensus (soft fork)	Block v2, Height in Coinbase
35	Peer Services	mempool message
36	Peer Services	Custom Services
37	Peer Services	Connection Bloom filtering
38	Applications	Passphrase-protected private key
39	Applications	Mnemonic code for generating deterministic keys
42	Consensus (soft fork)	A finite monetary supply for Bitcoin
43	Applications	Purpose Field for Deterministic Wallets
44	Applications	Multi-Account Hierarchy for Deterministic Wallets
45	Applications	Structure for Deterministic P2SH Multisignature Wallets
47	Applications	Reusable Payment Codes for Hierarchical Deterministic Wallets
49	Applications	Derivation scheme for P2WPKH-nested-in-P2SH based accounts
50		March 2013 Chain Fork Post-Mortem
60	Peer Services	Fixed Length "version" Message (Relay-Transactions Field)
61	Peer Services	Reject P2P message
62	Consensus (soft fork)	Dealing with malleability
64	Peer Services	getutxo message
65	Consensus (soft fork)	OP_CHECKLOCKTIMEVERIFY
66	Consensus (soft fork)	Strict DER signatures
67	Applications	Deterministic Pay-to-script-hash multi-signature addresses through public
68	Consensus (soft fork)	Relative lock-time using consensus-enforced sequence numbers

Number	Layer	Title
69	Applications	Lexicographical Indexing of Transaction Inputs and Outputs
70	Applications	Payment Protocol
71	Applications	Payment Protocol MIME types
72	Applications	bitcoin: uri extensions for Payment Protocol
73	Applications	Use "Accept" header for response type negotiation with Payment Request
74	Applications	Allow zero value OP_RETURN in Payment Protocol
75	Applications	Out of Band Address Exchange using Payment Protocol Encryption
80		Hierarchy for Non-Colored Voting Pool Deterministic Multisig Wallets
81		Hierarchy for Colored Voting Pool Deterministic Multisig Wallets
83	Applications	Dynamic Hierarchical Deterministic Key Trees
99		Motivation and deployment of consensus rule changes ([soft/hard]forks)
101	Consensus (hard fork)	Increase maximum block size
102	Consensus (hard fork)	Block size increase to 2MB
103	Consensus (hard fork)	Block size following technological growth
105	Consensus (hard fork)	Consensus based block size retargeting algorithm
106	Consensus (hard fork)	Dynamically Controlled Bitcoin Block Size Max Cap
107	Consensus (hard fork)	Dynamic limit on the block size
109	Consensus (hard fork)	Two million byte size limit with sigop and sighash limits
111	Peer Services	NODE_BLOOM service bit
112	Consensus (soft fork)	CHECKSEQUENCEVERIFY
113	Consensus (soft fork)	Median time-past as endpoint for lock-time calculations
114	Consensus (soft fork)	Merkelized Abstract Syntax Tree
120	Applications	Proof of Payment
121	Applications	Proof of Payment URI scheme
122	Applications	URI scheme for Blockchain references / exploration
123		BIP Classification
124	Applications	Hierarchical Deterministic Script Templates
125	Applications	Opt-in Full Replace-by-Fee Signaling
126		Best Practices for Heterogeneous Input Script Transactions
130	Peer Services	sendheaders message
131	Consensus (hard fork)	"Coalescing Transaction" Specification (wildcard inputs)
132		Committee-based BIP Acceptance Process
133	Peer Services	feefilter message
134	Consensus (hard fork)	Flexible Transactions
140	Consensus (soft fork)	Normalized TXID
141	Consensus (soft fork)	Segregated Witness (Consensus layer)
142	Applications	Address Format for Segregated Witness
143	Consensus (soft fork)	Transaction Signature Verification for Version 0 Witness Program
144	Peer Services	Segregated Witness (Peer Services)
145	API/RPC	getblocktemplate Updates for Segregated Witness
146	Consensus (soft fork)	Dealing with signature encoding malleability
147	Consensus (soft fork)	Dealing with dummy stack element malleability
150	Peer Services	Peer Authentication
151	Peer Services	Peer-to-Peer Communication Encryption

Number	Layer	Title
152	Peer Services	Compact Block Relay