

u^b

b

**UNIVERSITY
OF BERN**

Community-Driven Variability: Characterizing a new Software Variability Paradigm

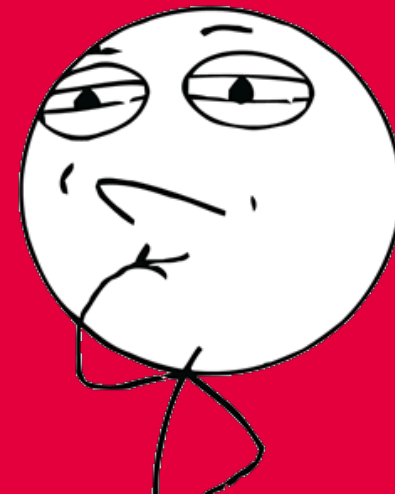
Published in Autom Softw Eng (2026)

Roman Bögli, Alexander Boll, Alexander Schultheiß, and Timo Kehrer

Software Engineering Group (SEG), University of Bern, Switzerland

VARIABILITY, 2026-09-30, Limassol, Cyprus

Who rules bitcoin?





Operating System

Mobile



Desktop



Hardware



User type

☐ New



☐ Experienced



Criteria

☐ Control



☐ Validation



☐ Transparency



☐ Environment

Not available

☐ Privacy



☐ Fees



Features

☐ 2FA



☐ Bech32



☐ Full Node



☐ Hardware Wallet



☐ Legacy Addresses



Below is a list of wallets available for your operating system

Linux Wallets

Control

Validation

Transparency

Environment

Privacy

Fees



Armory



Bitcoin Core



Bitcoin Knots



Bither



BitPay



Electrum



Green



Sparrow



Specter





☐ Privacy i

☐ Fees i

Features

☐ 2FA i

☐ Bech32 i

☐ Full Node i

☐ Hardware Wallet i

☐ Legacy Addresses i

☐ Lightning i

☐ Multisig i

☐ SegWit i

☒ Go



bitcoin / bips

What is a BIP?

BIPs are improvement proposals for Bitcoin. The main topic is information and technologies that support and expand the utility of the Bitcoin currency.

Most BIPs provide a concise, self-contained, technical description of one new concept, feature, or standard.

bips / bip-0173.mediawiki

Preview

Code

Blame

415 lines (340 loc) · 20.3 KB

Raw

BIP: 173

Layer: Applications

Title: Base32 address format for native v0-16 witness outputs

Author: Pieter Wuille <pieter.wuille@gmail.com>
Greg Maxwell <greg@xiph.org>

Comments-Summary: No comments yet.

Comments-URI: <https://github.com/bitcoin/bips/wiki/Comments:BIP-0173>

Status: Final

Type: Informational

Created: 2017-03-20

License: BSD-2-Clause

Replaces: 142

Superseded-By: 350

Table of Contents

- [Introduction](#)
- [Abstract](#)

github.com/bitcoin/bips/...bip-0173

- 1 BIP Purpose and Guidelines
- 2 BIP process, revised
- 3 Updated BIP Process
- 8 Version bits with lock-in by height
- 9 Version bits with timeout and delay
- 10 Multi-Sig Transaction Distribution
- 11 M-of-N Standard Transactions
- 12 OP_EVAL
- 13 Address Format for pay-to-script-hash
- 14 Protocol Version and User Agent
- 15 Aliases
- 16 Pay to Script Hash
- 17 OP_CHECKHASHVERIFY (CHV)
- 18 hashScriptCheck
- 19 M-of-N Standard Transactions (Low SigOp)
- 20 URI Scheme
- 21 URI Scheme
- 22 getblocktemplate – Fundamentals
- 23 getblocktemplate – Pooled Mining
- 30 Duplicate transactions
- 31 Pong message



 IPFS Standards

InterPlanetary Improvement Proposals

An InterPlanetary Improvement Proposals (IPIP) provides an orderly mechanism for considering proposed changes to IPFS specifications. An IPIP proposal is not to be the spec itself; the approval of an IPIP leads to an update to a specification.

 [ethereum / EIPs](#) Public

The Ethereum Improvement Proposal

 eips.ethereum.org/

 CC0-1.0 license

☆ 13.2k stars  5.5k forks

 [nostr-protocol / nips](#) Public

Nostr Implementation Possibilities

☆ 2.5k stars  634 forks  Branches

Nostr Implementation Possibilities

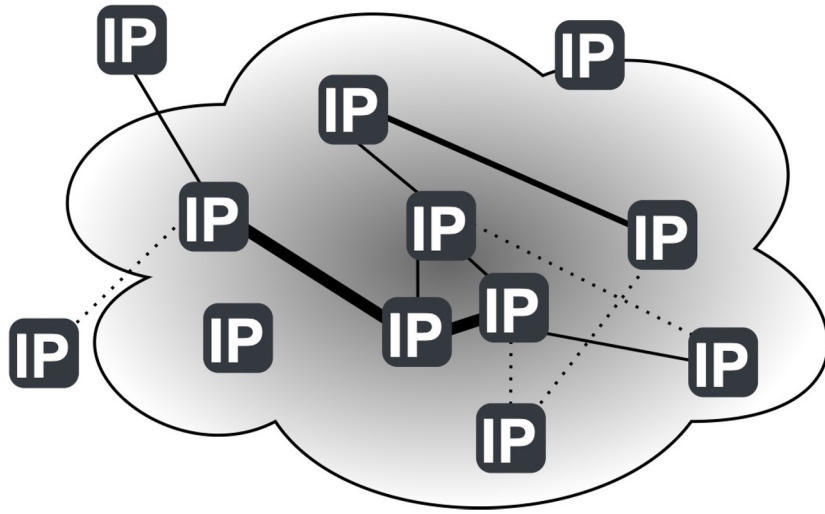
☆ 2.5k stars 🔗 634 forks 🌿 Branches

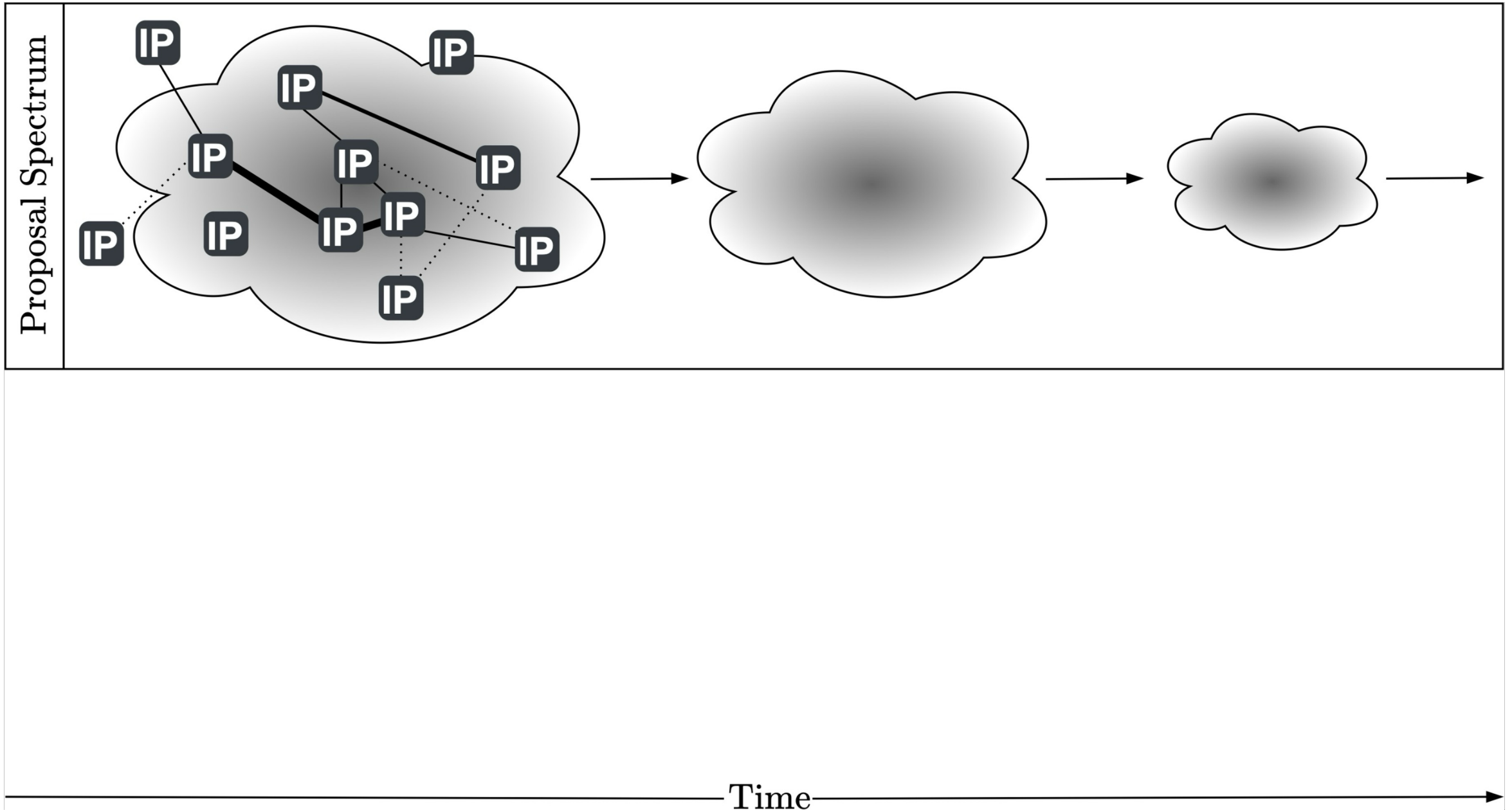
§ NIPs

A Nostr Implementation Possibility, or NIP for short, exist to document what **MUST**, what **SHOULD** and what **MAY** be implemented by Nostr-compatible relay and client software.

NIPs are the documents that outline how the Nostr protocol works.

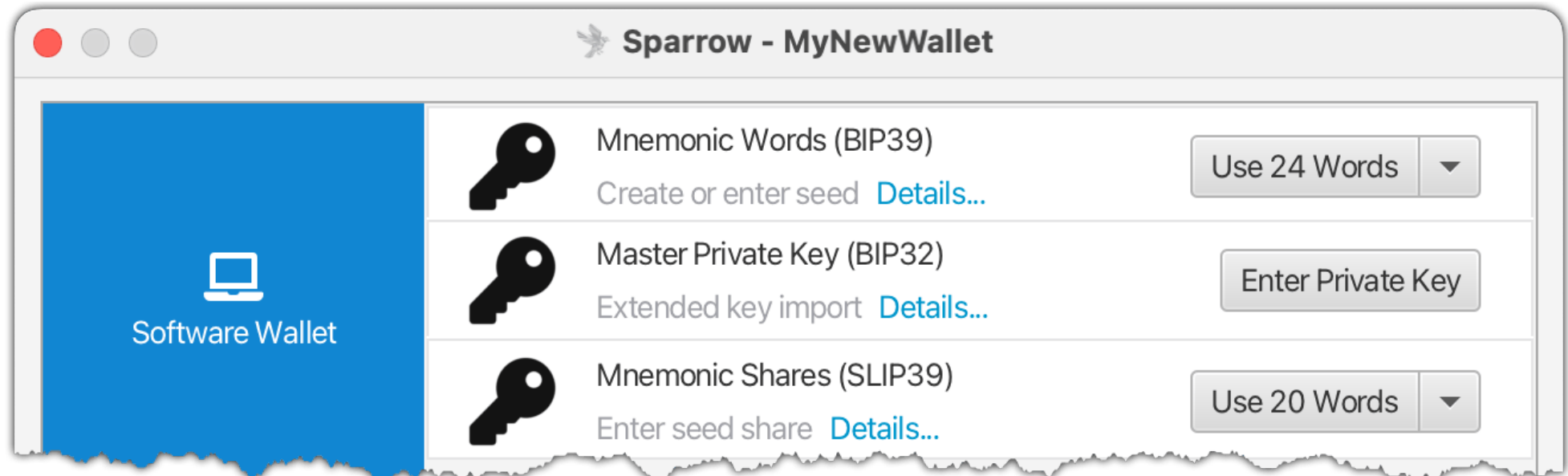
[nostr.how](#)

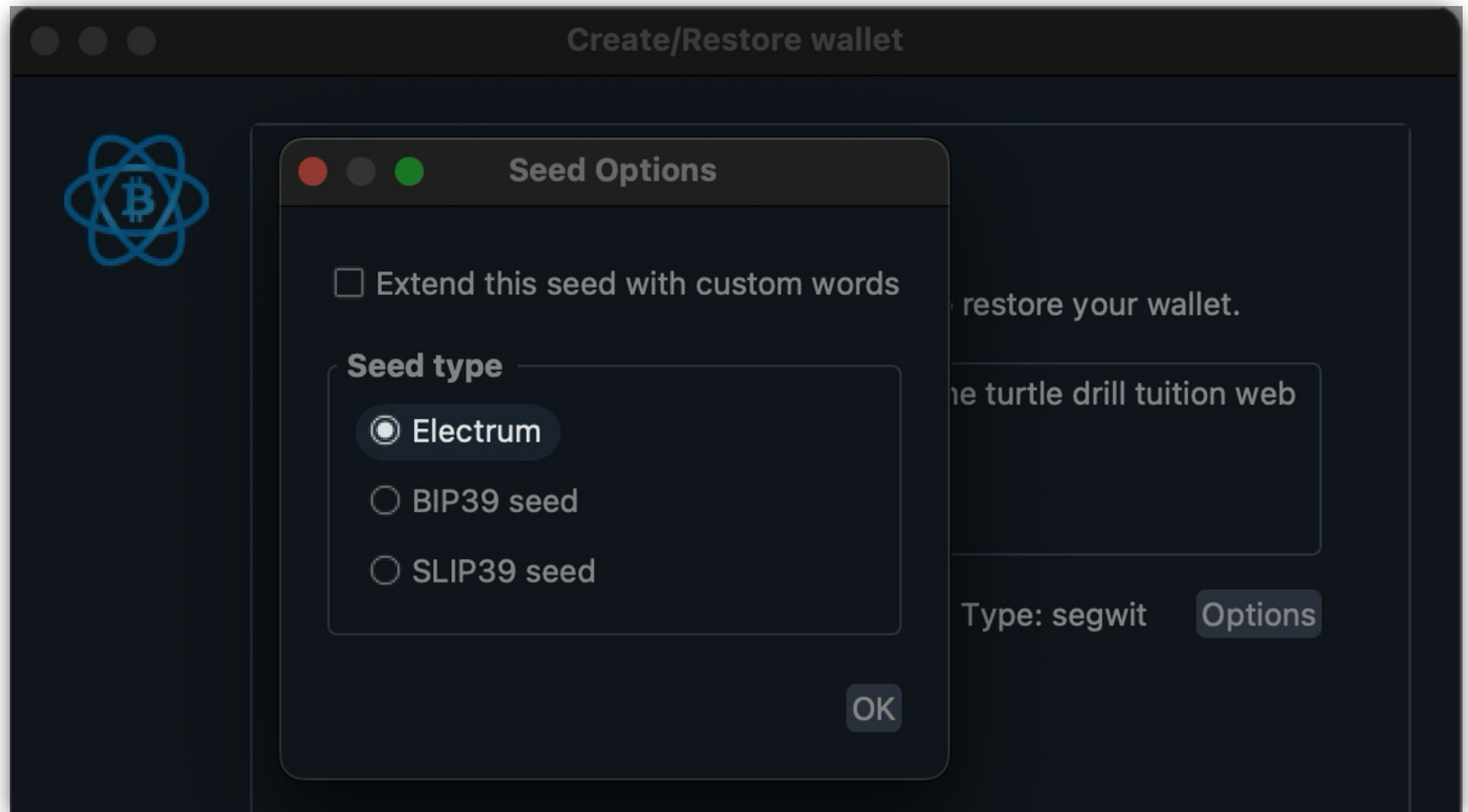


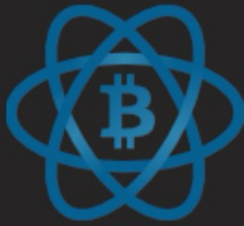


...but Standards Based

Sparrow tries **wherever possible to adhere to commonly accepted standards** in order to **have as wide an interoperability as possible**. In particular, it has been built to support Partially Signed Bitcoin Transactions (PSBTs) from the ground up, influencing everything from the keystore design to the transaction editor.







- ⊕ Verifying GPG signature of Electrum using Linux command line

DAEMON AND COMMAND LINE

- ⊕ Command Line
- ⊕ How to configure SSL with Electrum
- ⊕ How to accept Bitcoin on a website using Electrum
- ⊕ How to setup a watchtower

JSONRPC interface

FOR DEVELOPERS

The Python Console

Simple Payment Verification

Electrum Seed Version System

Description

Motivation

Version number

List of reserved numbers

Seed generation

Electrum Seed Version System

This document describes the Seed Version System used in Electrum (version 2.0 and higher)

BIP39 was introduced two years after Electrum. BIP39 seeds include a checksum, in order to help users figure out typing errors. However, BIP39 suffers the same shortcomings as early Electrum seed phrases:

- A fixed wordlist is still required. Following our recommendation, BIP39 authors decided to derive keys and addresses in a way that does not depend on the wordlist. However, BIP39 still requires the wordlist in order to compute its checksum, which is plainly inconsistent, and defeats the purpose of our recommendation. This problem is exacerbated by the fact that BIP39 proposes to create one wordlist per language. This threatens the portability of BIP39 seed phrases.
- BIP39 seed phrases do not include a version number. This means that software should always know how to generate keys and addresses. BIP43 suggests that wallet software will try various existing derivation schemes within the BIP32 framework. This is extremely inefficient and rests on the assumption that future wallets will support all previously accepted derivation methods. If, in the future, a wallet developer decides not to implement a particular derivation method because it is deprecated, then the software will not be able to detect that the corresponding seed phrases are not supported, and it will return an empty wallet instead. This threatens users funds.

For these reasons, Electrum does not generate BIP39 seeds. Starting with version 2.0, Electrum uses the following Seed Version System, which addresses these issues.

 [nostr-protocol](#) / [nips](#) Public

Nostr Implementation Possibilities

☆ 2.5k stars 🍴 634 forks 🌿 Branches

Clients

Source: github.com/.../awesome-nostr

- [Agora](#)  Stars 17 - Follow your favorite topics in nostr-verse (and even posts from Mastodon, Reddit, Bluesky, and Twitter)
- [Alexandria](#)  Stars 11 - A Knowledge Base and future eReader app. Focuses on the implementation of [NKBIP-01](#)
- [algia-web](#)  Stars 11 - A small resource consumption oriented Nostr web client
- [algia](#)  Stars 188 - A cli application for nostr.
- [alphaama](#)  Stars 27 - alphaama is just a nostr fucking client
- [Amethyst](#)  Stars 1.2k - An Android client for nostr written in Kotlin
- [Anonostr](#)  Stars 14 - Anonostr allows users to send anonymous notes to the Nostr network without revealing their identity. For each note submission, the app generates a new key pair, sends the note through select relays, and then securely burns the key pair.
- [ArcadeCity](#)  Stars 22 - Public group chats and P2P services (WIP) over nostr
- [Asknostr.site](#) - A Quora/StackOverflow Q&A site using the nostr network and #asknostr content
- [Astral](#)  Stars 101 - a branle fork with global feed and UI makeover
- [Attached](#)  Stars 26 - Open-Source ReactNative Expo app for Nostr (iOS, Android).
- [badger](#)  Stars 0 - A NIP58 nostr badge client. Create Badges view Profile Badges and more.
- [Beagle](#)  Stars 30 - Decentralized real-time Text/Audio/Video chat client for iOS, using nostr relays for users to share information and moments by posts of text, image and video.
- [BlazeJump](#)  Stars 6 - A fast web client boilerplate written in C# / Blazor, that uses an in-browser SQLite database.
- [Blowater](#) - A desktop Web client focusing on chat with delightful UX

 [nostr-protocol](#) / [nips](#) Public

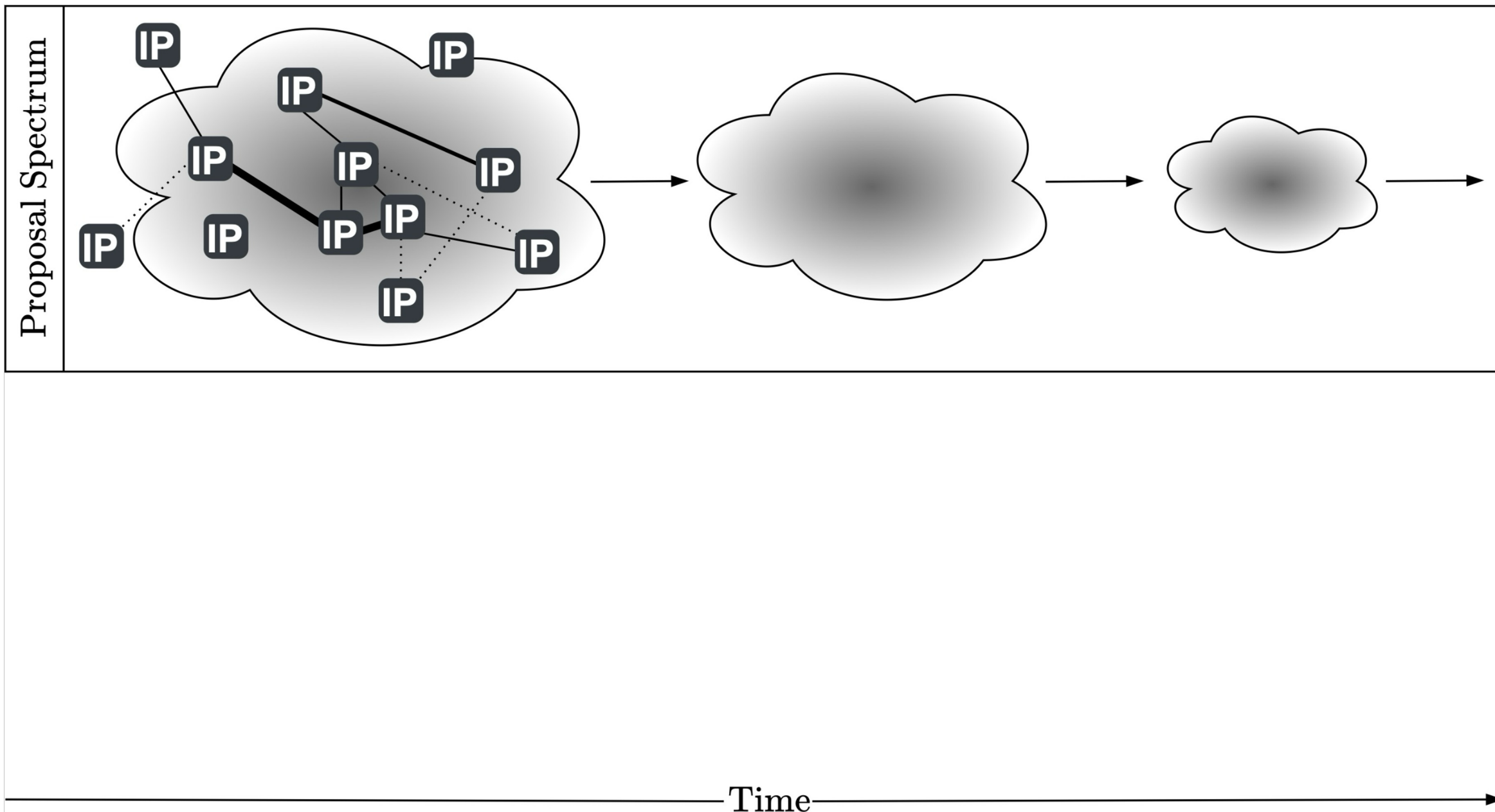
Nostr Implementation Possibilities

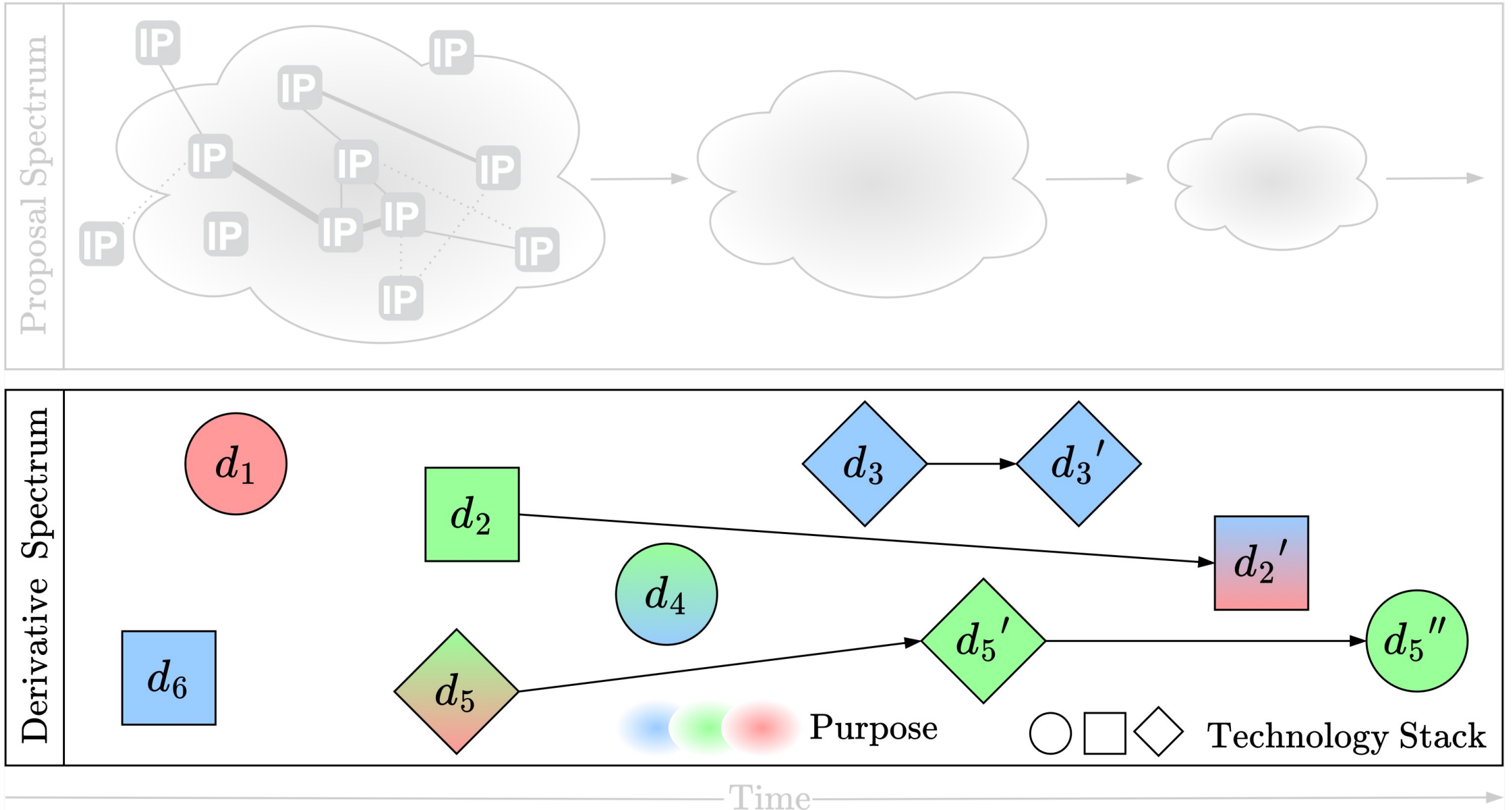
☆ 2.5k stars 🍴 634 forks 🌿 Branches

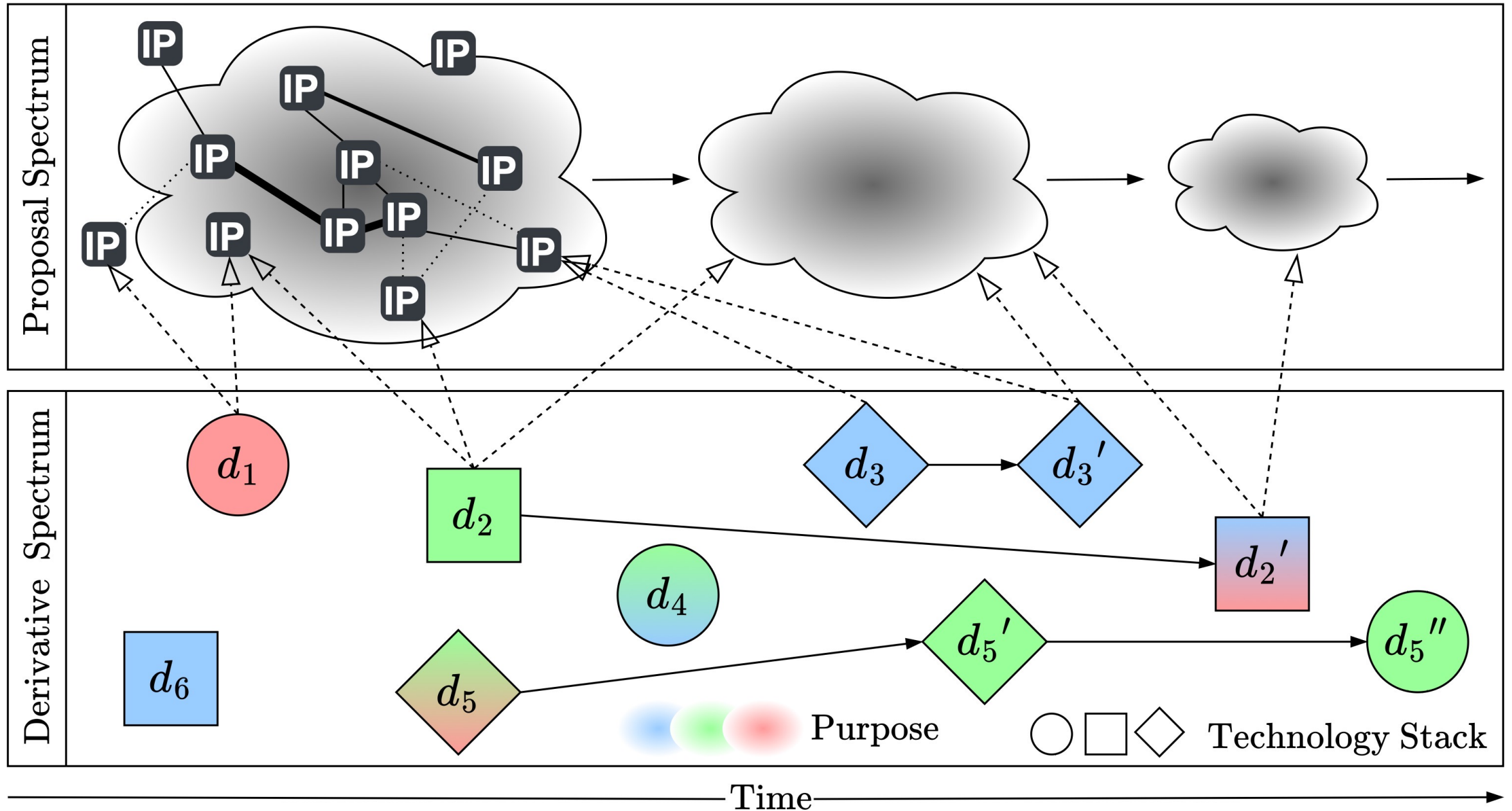
Clients

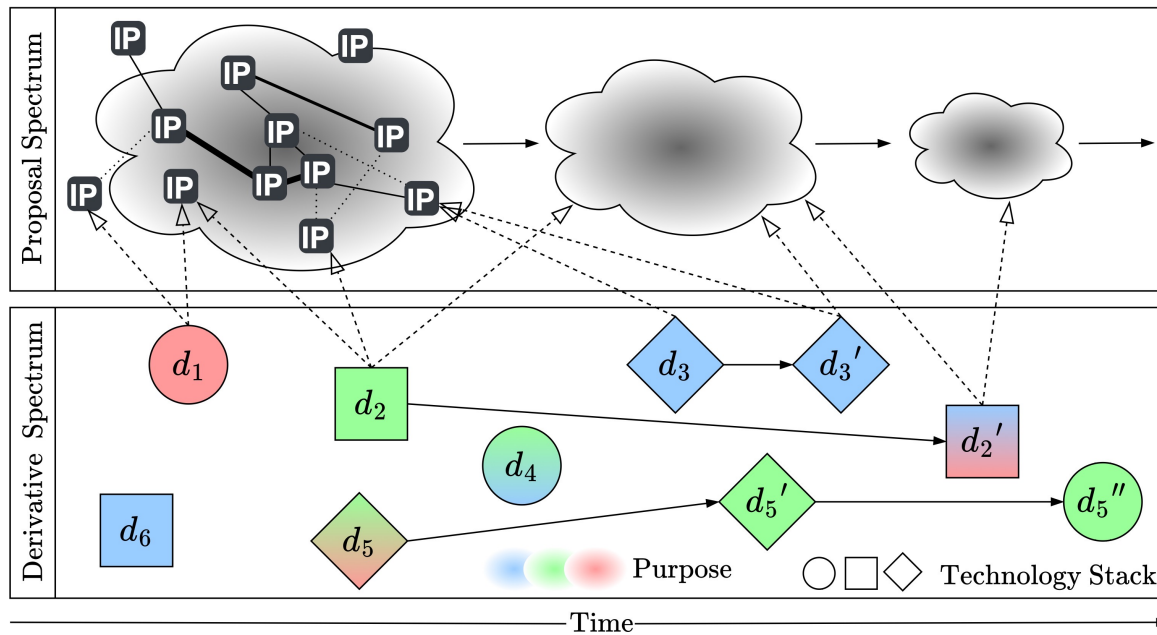
Source: github.com/.../awesome-nostr

- [Agora](#)  Stars 17 - Follow your favorite topics in nostr-verse (and even posts from Mastodon, Reddit, Bluesky, and Twitter)
- [Alexandria](#)  Stars 11 - A Knowledge Base and future eReader app. Focuses on the implementation of [NKBIP-01](#)
- [algia-web](#)  Stars 11 - A small resource consumption oriented Nostr web client
- [algia](#)  Stars 188 - A cli application for nostr.
- [alphaama](#)  Stars 27 - alphaama is just a nostr fucking client
- [Amethyst](#)  Stars 1.2k - An Android client for nostr written in [Kotlin](#)
- [Anonostr](#)  Stars 14 - Anonostr allows users to send anonymous notes to the Nostr network without revealing their identity. For each note submission, the app generates a new key pair, sends the note through select relays, and then securely burns the key pair.
- [ArcadeCity](#)  Stars 22 - Public group chats and P2P services (WIP) over nostr
- [Asknostr.site](#) - A Quora/StackOverflow Q&A site using the nostr network and [#asknostr](#) content
- [Astral](#)  Stars 101 - a branle fork with global feed and UI makeover
- [Attached](#)  Stars 26 - Open-Source [ReactNative](#) Expo app for Nostr ([iOS](#), [Android](#)).
- [badger](#)  Stars 0 - A NIP58 nostr badge client. Create Badges view Profile Badges and more.
- [Beagle](#)  Stars 30 - Decentralized real-time Text/Audio/Video chat client for [iOS](#), using nostr relays for users to share information and moments by posts of text, image and video.
- [BlazeJump](#)  Stars 6 - A fast web client boilerplate written in [C# / Blazor](#), that uses an in-browser SQLite database.
- [Blowater](#) - A desktop Web client focusing on chat with delightful UX



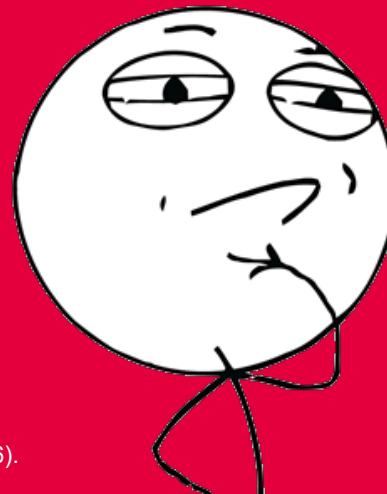




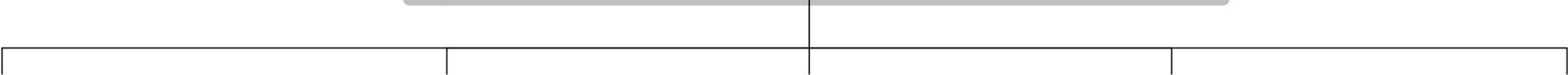


Community-Driven
Variability (CDV)

What constitutes CDV?



Taxonomy of CDV Characteristics



```
graph TD; A[Taxonomy of CDV Characteristics] --- B[ ]; B --- C[ ]; B --- D[ ]; B --- E[ ]
```

Taxonomy of CDV Characteristics

C1: Improvement Proposal

a. de-facto standard

b. interdependencies

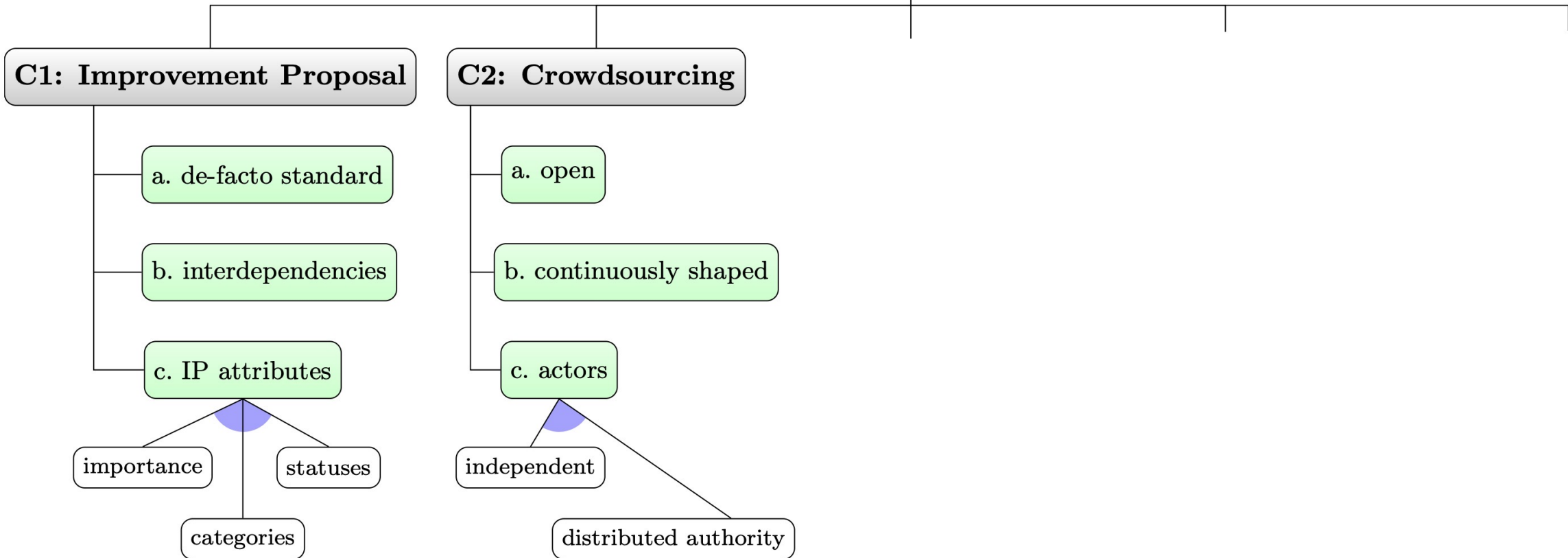
c. IP attributes

importance

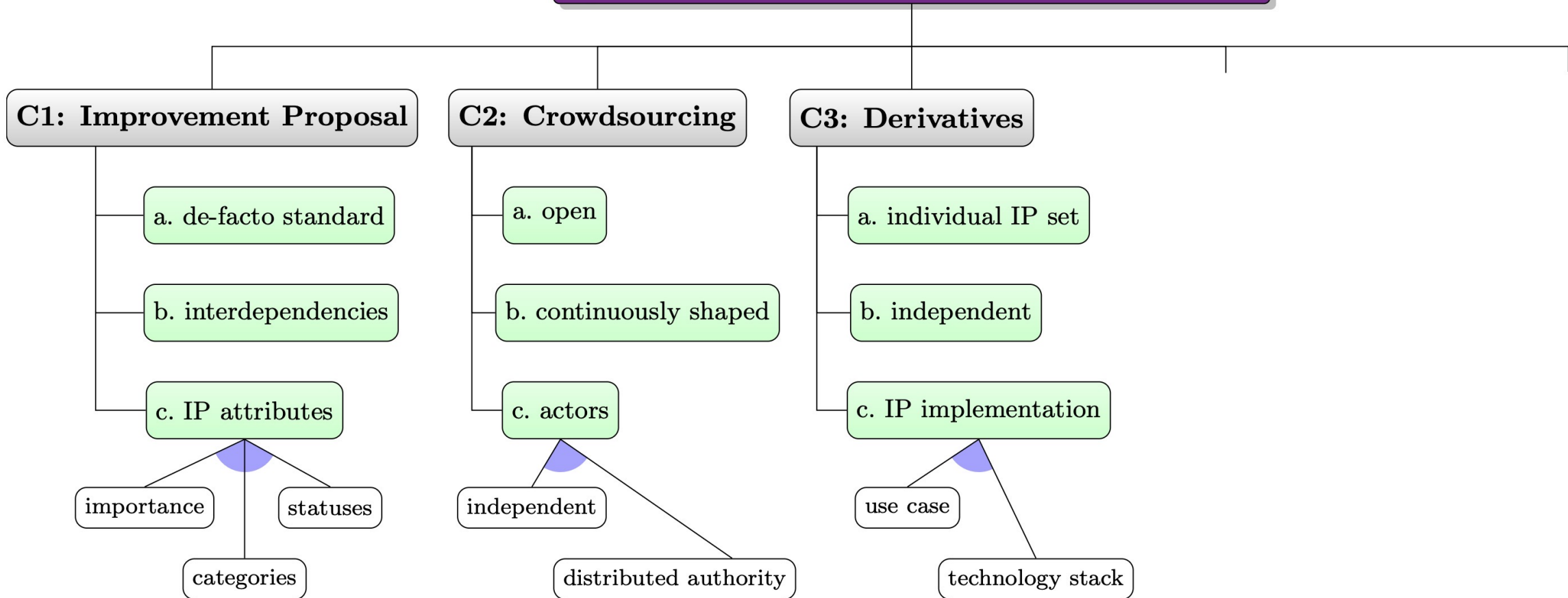
statuses

categories

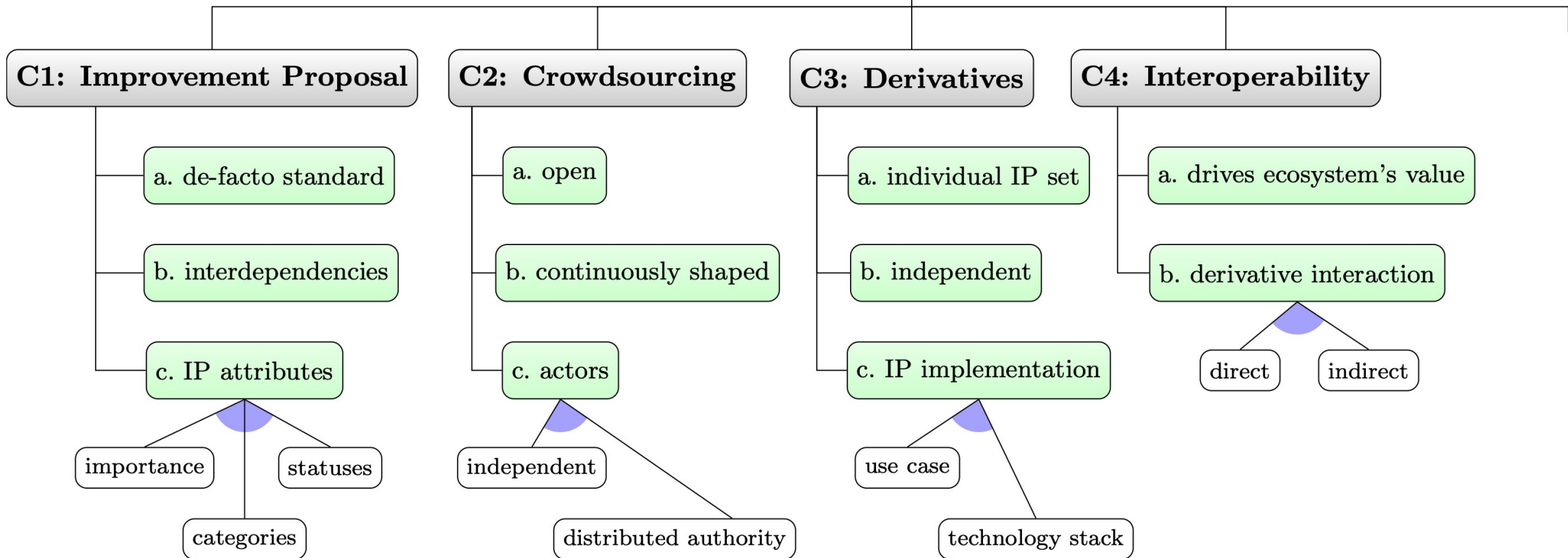
Taxonomy of CDV Characteristics



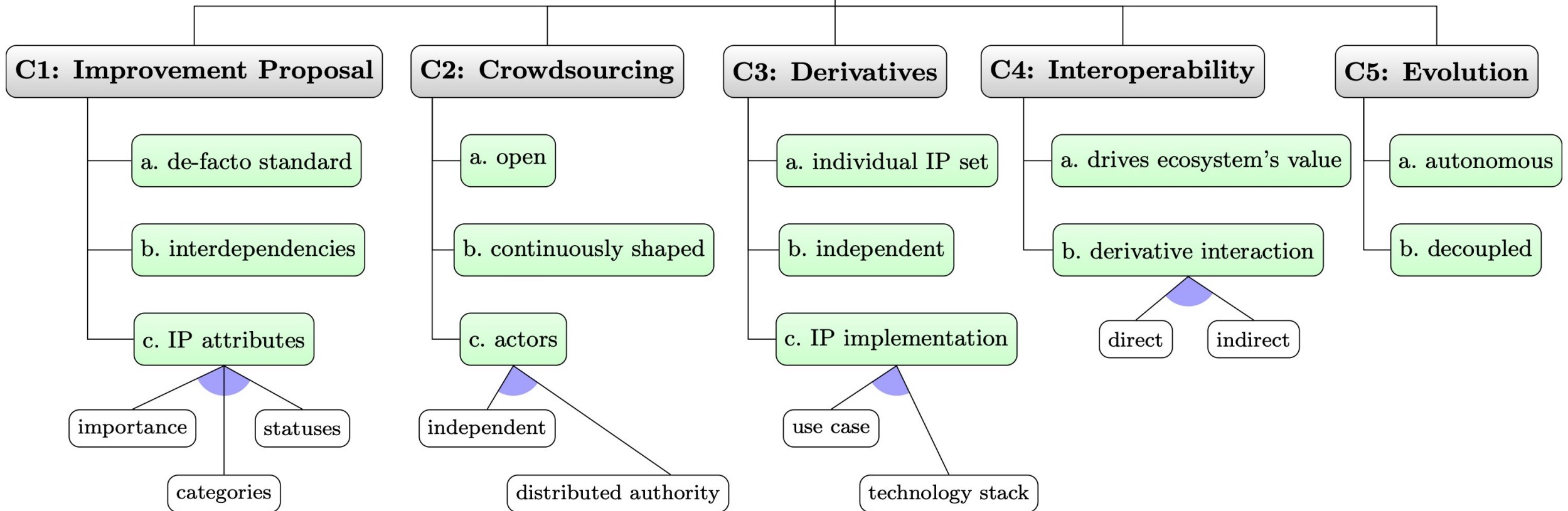
Taxonomy of CDV Characteristics



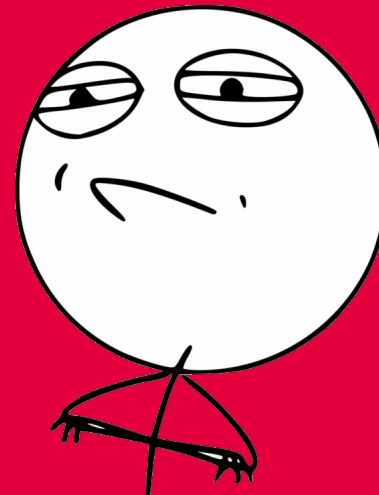
Taxonomy of CDV Characteristics



Taxonomy of CDV Characteristics



Where do we find CDV?



		C1: Improvement Proposal (a:standard, b:interdep., c:IP-attributes)			C3: Derivatives (a:indiv.-IP-set, b:interdep., c:IP-impl.)			C4: Interoperability (a:ecosystem-value, b:interaction)			C5: Evolution (a:autonomous, b:decoupled)	
Paradigm	Ecosystem/Project	C2: Crowdsourcing (a:open, b:cont.-shaped, c:actors)										
		a	b	c	a	b	c	a	b	c	a	b
CDV	Bitcoin [26, 27]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Lightning [61–63]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Ethereum [64, 65]	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓
	Nostr [32, 33]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Tor Protocol [30, 31]	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓
	IPFS [28, 29]	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓
SPL	Linux Kernel [66, 67]	✓	✓	-	✓	✓	✓	-	-	-	-	-
	Eclipse [68]	-	✓	✓	✓	✓	✓	✓	✓	✓	-	-
	BusyBox [69]	✓	✓	-	✓	✓	✓	-	-	-	-	-
Clone-and-Own	Apo-Games [70]	-	-	-	-	-	-	✓	✓	✓	-	✓
	Marlin Forks [71]	-	-	-	✓	✓	✓	✓	✓	-	✓	✓
	Health Watcher [72]	-	-	-	-	-	-	✓	✓	-	-	✓
Other	Home Assistant [73]	-	-	-	✓	✓	-	-	✓	✓	-	✓
	Python [74, 75]	✓	✓	✓	✓	✓	-	✓	✓	✓	-	✓

		C1: Improvement Proposal (a:standard, b:interdep., c:IP-attributes)			C2: Crowdsourcing (a:open, b:cont.-shaped, c:actors)			C3: Derivatives (a:indiv.-IP-set, b:interdep., c:IP-impl.)			C4: Interoperability (a:ecosystem-value, b:interaction)		C5: Evolution (a:autonomous, b:decoupled)	
Paradigm	Ecosystem/Project	a	b	c	a	b	c	a	b	c	a	b	a	b
CDV	Bitcoin [26, 27]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Lightning [61–63]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Ethereum [64, 65]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓
	Nostr [32, 33]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Tor Protocol [30, 31]	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	-	✓
	IPFS [28, 29]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓
SPL	Linux Kernel [66, 67]	✓	✓	-	✓	✓	✓	-	-	-	-	-	-	-
	Eclipse [68]	-	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-
	BusyBox [69]	✓	✓	-	✓	✓	✓	-	-	-	-	-	-	-
Clone-and-Own	Apo-Games [70]	-	-	-	-	-	-	✓	✓	✓	-	-	-	✓
	Marlin Forks [71]	-	-	-	✓	✓	✓	✓	✓	-	-	-	✓	✓
	Health Watcher [72]	-	-	-	-	-	-	✓	✓	-	-	-	-	✓
Other	Home Assistant [73]	-	-	-	✓	✓	-	-	✓	✓	-	✓	-	✓
	Python [74, 75]	✓	✓	✓	✓	✓	-	✓	✓	✓	-	-	-	✓

C1: Improvement Proposal
(a:standard, b:interdep., c:IP-attributes)

C3: Derivatives
(a:indiv.-IP-set, b:interdep., c:IP-impl.)

C4: Interoperability
(a:ecosystem-value, b:interaction)

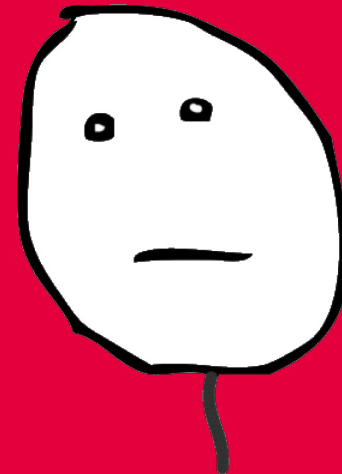
C2: Crowdsourcing
(a:open, b:cont.-shaped, c:actors)

C5: Evolution
(a:autonomous, b:decoupled)

Paradigm	Ecosystem/Project	C1			C2			C3			C4		C5	
		a	b	c	a	b	c	a	b	c	a	b	a	b
CDV	Bitcoin [26, 27]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Lightning [61–63]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Ethereum [64, 65]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓
	Nostr [32, 33]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Tor Protocol [30, 31]	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	-	✓
	IPFS [28, 29]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓
SPL	Linux Kernel [66, 67]	✓	✓	-	✓	✓	✓	-	-	-	-	-	-	-
	Eclipse [68]	-	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-
	BusyBox [69]	✓	✓	-	✓	✓	✓	-	-	-	-	-	-	-
Clone-and-Own	Apo-Games [70]	-	-	-	-	-	-	✓	✓	✓	-	-	-	✓
	Marlin Forks [71]	-	-	-	✓	✓	✓	✓	✓	-	-	-	✓	✓
	Health Watcher [72]	-	-	-	-	-	-	✓	✓	-	-	-	-	✓
Other	Home Assistant [73]	-	-	-	✓	✓	-	-	✓	✓	-	✓	-	✓
	Python [74, 75]	✓	✓	✓	✓	✓	-	✓	✓	✓	-	-	-	✓

		C1: Improvement Proposal (a:standard, b:interdep., c:IP-attributes)			C2: Crowdsourcing (a:open, b:cont.-shaped, c:actors)			C3: Derivatives (a:indiv.-IP-set, b:interdep., c:IP-impl.)			C4: Interoperability (a:ecosystem-value, b:interaction)		C5: Evolution (a:autonomous, b:decoupled)	
Paradigm	Ecosystem/Project	a	b	c	a	b	c	a	b	c	a	b	a	b
CDV	Bitcoin [26, 27]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Lightning [61–63]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Ethereum [64, 65]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓
	Nostr [32, 33]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Tor Protocol [30, 31]	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	-	✓
	IPFS [28, 29]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓
SPL	Linux Kernel [66, 67]	✓	✓	-	✓	✓	✓	-	-	-	-	-	-	-
	Eclipse [68]	-	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-
	BusyBox [69]	✓	✓	-	✓	✓	✓	-	-	-	-	-	-	-
Clone-and-Own	Apo-Games [70]	-	-	-	-	-	-	✓	✓	✓	-	-	-	✓
	Marlin Forks [71]	-	-	-	✓	✓	✓	✓	✓	-	-	-	✓	✓
	Health Watcher [72]	-	-	-	-	-	-	✓	✓	-	-	-	-	✓
Other	Home Assistant [73]	-	-	-	✓	✓	-	-	✓	✓	-	✓	-	✓
	Python [74, 75]	✓	✓	✓	✓	✓	-	✓	✓	✓	-	-	-	✓

Ok... so what?



P1 Missing overview in proposal spectrum

P2 Missing overview in derivative spectrum

P3 IP change impact assessment

P4 Misalignment of proposal & derivative spectrum

P5 Level of derivative interoperability

P6 Ecosystem forks

RG1

Systematic treatment of
CDV in proposal spectrum

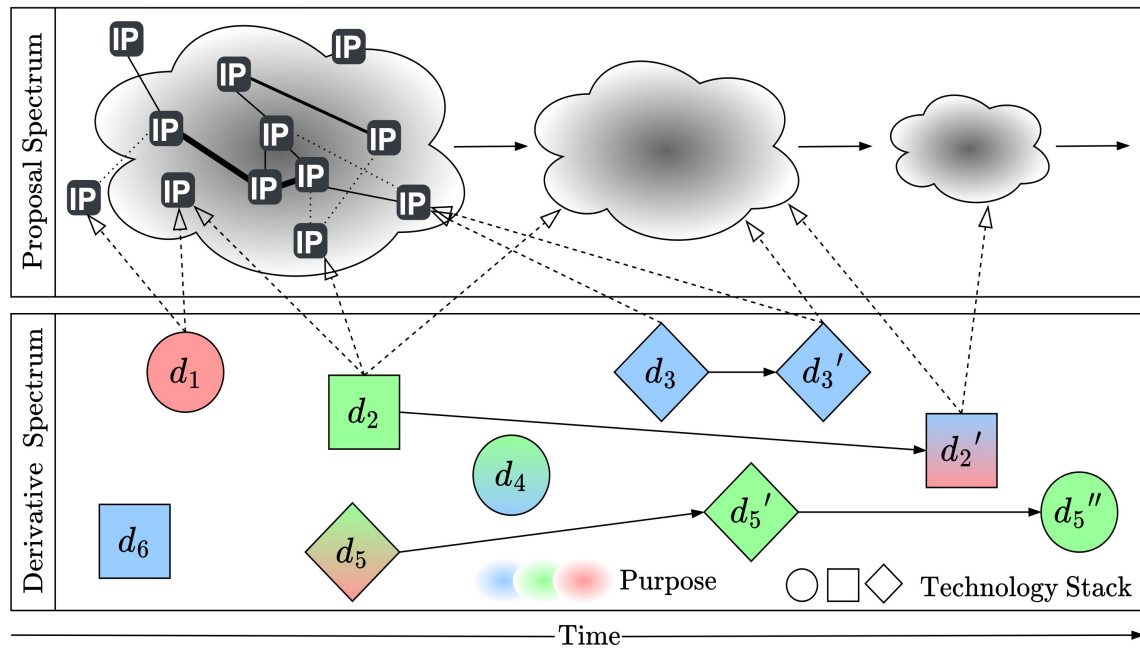
RG2

Supporting cohesive evolution of
proposal and derivative spectrum

RG3

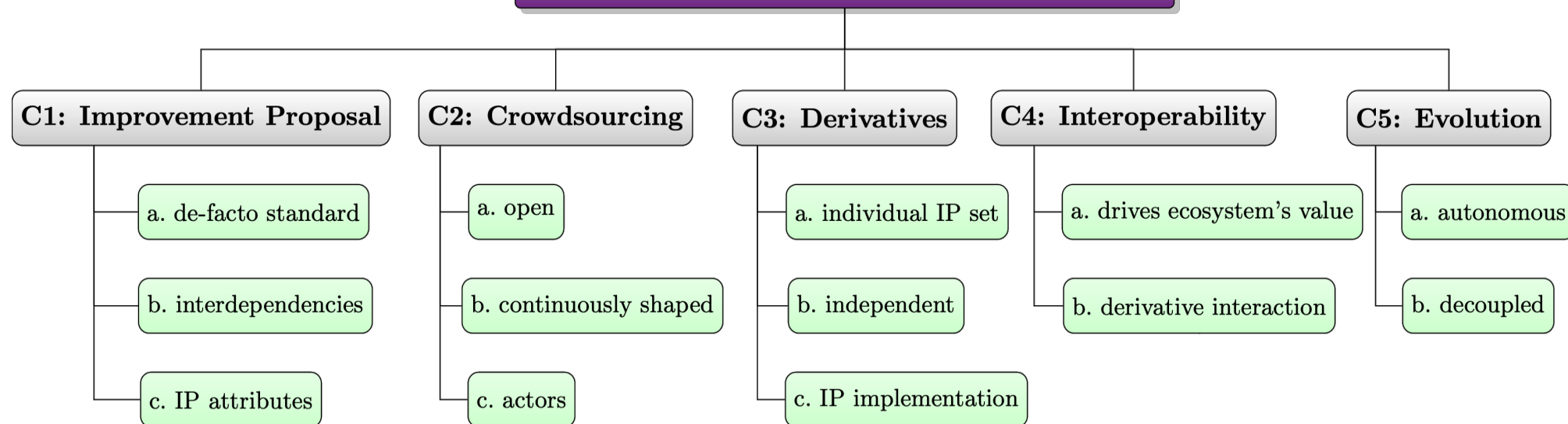
Methodical handling of derivative
interoperability impairment

u^b Community-Driven Variability

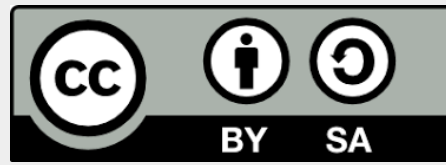


- P1** Missing overview in proposal spectrum
- P2** Missing overview in derivative spectrum
- P3** IP change impact assessment
- P4** Misalignment of proposal/derivative spectrum
- P5** Level of derivative interoperability
- P6** Ecosystem forks

Taxonomy of CDV Characteristics



alturl.com/n8rzz



Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)

You are free to:

Share: copy and redistribute the material in any medium or format

Adapt: remix, transform, and build upon the material for any purpose, even commercially.

Under the following terms:

Attribution You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

ShareAlike If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

No additional restrictions You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

<https://creativecommons.org/licenses/by-sa/4.0/>

u^b

b

**UNIVERSITY
OF BERN**