CDV-Explorer: Navigating Improvement Proposal Spectra in Decentralized OSS Ecosystems

Roman Bögli ✉ and Timo Kehrer

Institute of Computer Science, University of Bern, Switzerland
{roman.boegli,timo.kehrer}@unibe.ch

Abstract. Open-source software (OSS) ecosystems evolve through distributed communities that define features through improvement proposals (IPs), shaping living and heterogeneous feature spaces. As a result, they foster what has recently been described as Community-Driven Variability (CDV). Understanding such proposal spectra is difficult because IP corpora evolve continuously, encode dependencies only partially or implicitly, and distribute authorship across loosely coordinated communities. We present CDV-Explorer, a pipeline for mining, analyzing, and interactively visualizing proposal spectra in CDV-exhibiting ecosystems. We instantiate CDV-Explorer for Bitcoin and Nostr, showing that the pipeline adapts to distinct IP catalogs and supports systematic comprehension and navigation of the respective proposal spectra. A demonstration video is available at youtu.be/56GKRexRuoI.

Keywords: Community-Driven Variability · Improvement Proposals · Mining Software Repositories · OSS Ecosystems · Variability Analysis

1 Introduction

Software engineering increasingly focuses on developing families of related systems rather than isolated products. Classical approaches such as Software Product-Line (SPL) engineering manage variability through explicit variability models, notably feature models, that describe reusable capabilities and variation mechanisms [1]. Community-Driven Variability (CDV) [2, 3] broadens this view to decentralized open-source software (OSS) ecosystems whose software families emerge through collectively curated *improvement proposals* (IPs) rather than centrally managed variability models. These IPs act as guiding specification documents and together form the *proposal spectrum*. Developers then independently adopt IP subsets by implementing them in different software variants denoted as *derivatives* that together form the *derivative spectrum* [3, Fig. 2].

A prominent example of a CDV-exhibiting software ecosystem is Bitcoin with its Bitcoin Improvement Proposals (BIPs) [4], which have evolved for more than a decade and underpin a heterogeneous landscape of interoperable wallets, nodes, watchtowers, and other applications [5, 6]. Similar IP-driven dynamics occur in other software ecosystems such as, e.g., the decentralized messaging protocol Nostr with its *Implementation Possibilities* [7] and more [3, Tab. 1].

⊛ *Preprint – Accepted for publication at VARIABILITY'26* ⊛

The proposal spectrum is the primary source of variability in CDV-exhibiting software ecosystems, yet it remains difficult to comprehend and navigate systematically. Unlike rigorous SPLs, reuse centers on the IPs themselves rather than on code artifacts, as these are intended primarily for human rather than machine consumption. Crowdsourced processes further complicate maintaining an overview of the constantly evolving feature space, while inter-IP relations are often implicit, scattered across body text, or undocumented altogether. Consequently, ecosystem participants struggle with orientation and change impact assessment, both identified as central CDV challenges in prior work [3].

Mining Software Repositories (MSR) infrastructure [8, 9] primarily targets source code, commits, and issue trackers rather than proposal-level artifacts. Likewise, variability research largely assumes explicit variability models or code-centric variability artifacts [10–12], which are absent in decentralized CDV-exhibiting ecosystems. Although existing tools provide partial views of the derivative spectrum, such as wallet comparison platforms [6], they remain handcrafted and rely on ecosystem-specific expert knowledge. Systematic support for exploring the proposal spectrum itself, however, remains unaddressed. We address this gap with CDV-Explorer [13][1], a mining and analysis pipeline that turns IP corpora into a structured, explorable representation. To our knowledge, CDV-Explorer is the first tool designed to systematically process proposal spectra in CDV-exhibiting software ecosystems. In sum, we make the following contributions:

1. We present CDV-Explorer, a four-stage mining and analysis pipeline for exploring improvement proposal spectra in CDV-exhibiting OSS ecosystems.
2. We implement four analysis modules for IP authorship, classification, evolution, and multi-strategy dependency extraction.
3. We instantiate CDV-Explorer for Bitcoin and Nostr, demonstrating adaptability across distinct IP catalogs and surfacing practical insights into proposal evolution, interrelations, and authorship.
4. We provide CDV-Explorer (`v4.1.1`) as a replication package [13], enabling the community to instantiate the tool for further ecosystems and extend the existing Bitcoin and Nostr instantiations with additional analyses. As an accompanying artifact, we also provide a demonstration video [14].

2 Architecture and Workflow

CDV-Explorer is a pipeline for mining, analyzing, and visualizing IP catalogs of CDV-exhibiting software ecosystems. Fig. 1 illustrates its high-level architecture, which comprises four stages: *I. Harvest*, *II. Preprocess*, *III. Analysis*, and *IV. Postprocess*. The first two stages are ecosystem-specific and transform the raw source documents captured at specific points in time from an ecosystem's proposal spectrum into snapshot-specific instances of a structured representation, called the *IP object model*. The latter two stages are ecosystem-agnostic and operate on this IP object model. This separation confines ecosystem-specific logic

[1] Live demo available at `https://seg-unibe.github.io/cdv-explorer/`

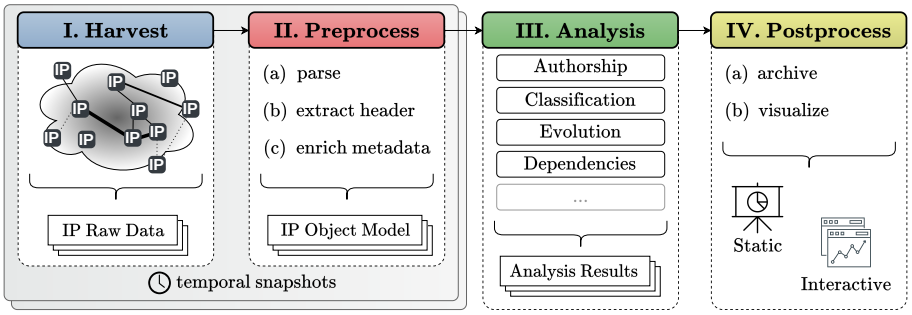

Fig. 1. Conceptual workflow of the CDV-Explorer pipeline.

to data acquisition and preparation and keeps downstream analyses reusable across ecosystems. The remainder of this section details these stages.

I. Harvest. This stage acquires raw IP documents from ecosystem-specific sources via repository cloning, web scraping, or API retrieval. CDV-Explorer captures snapshots by checking out the source at specific dates, enabling longitudinal analysis. For Bitcoin and Nostr, e.g., we clone the public BIP and NIPs repositories [4, 7], respectively, to obtain both current and historical versions.

II. Preprocess. The preprocess stage transforms harvested IP documents into a canonical IP object model stored as per-proposal `JSON` files. Each object comprises three top-level blocks: `raw` (e.g., verbatim preamble and body), `meta` (e.g., parsed fields and Git-derived commit history), and `insights`, a dedicated block reserved for analytical results produced in the next stage. In the Bitcoin case, for example, RFC822-style preamble fields are parsed, body text is extracted, and each BIP is enriched with relevant Git history. Storing the resulting lightweight `JSON` objects ensures that costly extraction runs only once per snapshot. The stable schema makes the IP object model the central interface between ecosystem-specific preprocessing and ecosystem-agnostic analysis, enabling heterogeneous IPs to be processed by the same downstream machinery.

III. Analysis. The analysis stage operates exclusively on the IP object model and is fully ecosystem-agnostic. It computes graph metrics, centrality scores, statistical aggregates, and other derived relations, and persists these results in the `insights` block of each IP object. The current implementation of (`v4.1.1`) comprises four analysis modules that are detailed in Section 3.1. The modular design readily accommodates future extensions, such as proposal conformity checking and analyses linking proposal and derivative spectra.

IV. Postprocess. The final stage compiles analysis artifacts into optimized frontend payloads, retaining only the fields required by each visualization component. The frontend performs no analytical computation at runtime, as all metrics and aggregates are precomputed during the analysis stage and packaged for efficient consumption. This keeps browser-side loading predictable as analytical depth increases. The frontend is further detailed in Section 3.2.

3 Core Capabilities and Extensibility

After introducing the overall pipeline, this section details the reusable capabilities built on top of the IP object model. We describe the implemented analysis modules, interactive frontend, and extension points for additional ecosystems.

3.1 Analysis Modules

The analysis stage (cf. *III. Analysis* in Fig. 1) applies independent modules to the IP object model. Each module targets one aspect of the proposal spectrum motivated by defining characteristics of CDV identified in prior work [3]. Modules expose a uniform interface and emit self-contained artifacts, allowing analyses to be added, removed, or executed independently without changing pipeline orchestration or data loading. The four implemented modules are described below.

Authorship. CDV proposal spectra emerge through crowdsourced contributions rather than centralized ownership [3]. The module builds a co-authorship graph from declared authors, with edge weights denoting joint proposals. It then derives centrality scores and complementary contribution statistics to surface influential contributors, bridges, and other participation patterns that would remain hidden on an individual IP level.

Classification. IPs carry ecosystem-specific attributes that structure the proposal spectrum, such as proposal status (e.g., *draft*, *final*, or *deprecated*) and type (e.g., *core*, *optional*, or *informational*). The module aggregates these classification attributes across snapshots to support longitudinal comparison and provide a holistic view of the ecosystem's IP landscape.

Evolution. CDV implies that proposal spectra evolve continuously as IPs progress through their lifecycle from early draft to deployment, supersession, or retirement [3]. The module reconstructs each IP's status trajectory as a timestamped change log across the available snapshot sequence. This enables the analysis of gradual maturation at both the individual IP and ecosystem levels. At the individual level, it supports detailed status tracking that can inform implementation decisions, while at the ecosystem level, it reveals broader trends that remain invisible when considering isolated snapshots.

Dependencies. Another defining characteristic of CDV-exhibiting ecosystems is that IPs form interrelation networks (cf. IP cloud in stage *I. Harvest*, Fig. 1). IPs reference and build upon one another, making inter-IP relations central for overall comprehension and change impact assessment [3]. The module surfaces such interrelations through three complementary extraction strategies: (i) *preamble-based* extraction of explicitly declared reference fields from IP headers, (ii) *regex-based* extraction of identifier-pattern cross-references from IP bodies, and (iii) *LLM-based* semantic extraction using OpenAI's GPT-5.4 models to capture implicit interrelations. Together, they expose declared relations, textual mentions, and semantically filtered candidate dependencies, enabling differential analysis that surfaces agreement and divergence across these strategies.

The *III. Analysis* stage is intentionally modular, allowing future analyses such as IP–derivative linkage to plug into the same pipeline. Each module persists

granular `CSV` artifacts alongside precomputed `JSON` results. In *IV. Postprocess*, the pipeline materializes interactive visualizations (payloads) from these artifacts while preserving the analysis results for alternative downstream uses.

3.2 Interactive Frontend

The frontend exposes the generated per-ecosystem analysis results through unified visualizations.[2] Realized with `React` and `D3.js`, the dashboard spans four sections that align with our analysis modules: *Authorship*, *Classification*, *Evolution*, and *Dependencies*. A global snapshot picker reloads all views simultaneously, enabling temporal comparison across the dashboard. Visual elements provide hover tooltips to enrich the contextual data with IP identifiers and other relevant information. Where IPs are referenced, users can directly open the corresponding IP document, either in the historical state matching the selected snapshot or in its latest version from the upstream repository.

The browser lazily fetches only the code and payloads each view requires, keeping the initial load lightweight and rendering independent of expensive analysis steps. The application is served as a static site and deployed to `GitHub Pages` through `GitHub Actions`. This serverless, precomputed delivery lowers the adoption barrier and supports replication, forking, and further development.

3.3 Instantiating Additional Ecosystems

CDV-Explorer is implemented in `Python` and exposes a `Typer`-based command-line interface (CLI). The CLI executes end-to-end pipeline runs for selected ecosystems and snapshots, lists available snapshots, and scaffolds ecosystem configurations. It also serves as the primary entry point for reproducible execution, allowing users to regenerate analyses for arbitrary ecosystems and snapshots through a uniform command interface. Artifacts are persisted as `JSON` and `CSV` files. Each ecosystem is described by a `YAML` configuration defining proposal metadata, classification dimensions, conformity rules, and source repositories. Only logic concerning the stages *I. Harvest* and *II. Preprocess* (cf. Fig. 1) requires ecosystem-specific implementation. Adding a new ecosystem to CDV-Explorer requires addressing three engineering concerns:

1. **Ecosystem Configuration.** Defines ecosystem-specific configurations as `YAML` files. This includes source repositories, classification dimensions, and aliases. The configuration drives both the pipeline and the frontend.
2. **Harvester Adapter.** Connects external IP sources to the pipeline by acquiring raw proposal documents and organizing them into snapshot-specific working directories (stage *I. Harvest*). Depending on the OSS ecosystem, this may involve repository cloning, API access, or web scraping.
3. **Preprocessor Adapter.** Connects ecosystem-specific IP document formats to the canonical IP data model by extracting and normalizing metadata such

[2] Live demo available at https://seg-unibe.github.io/cdv-explorer/

as identifiers, classification labels, interrelations, and authorship information (stage *II. Preprocess*).

The IP object model serves as the single interface between ingestion and downstream processing. Any ecosystem that can be mapped to this schema can reuse the existing analysis modules and visualization components. The CLI further supports this extension workflow by listing registered ecosystems, showing their configuration, and scaffolding new ecosystem `YAML` files.

4 Demonstrating Utility Across Ecosystems

This section demonstrates the utility of CDV-Explorer across ecosystems while clarifying the scope of this tool paper. We defer the deeper empirical analysis to a separate research paper and focus here on the reusable pipeline, which we complement with a second instantiation for Nostr and its Implementation Possibilities (NIPs) [7]. Using Bitcoin and Nostr as two contrasting CDV-exhibiting ecosystems allows us to demonstrate both the portability of the pipeline and the kind of ecosystem-level insights it can surface.

4.1 Multi-Ecosystem Instantiation

To emphasize this reusable scope, we instantiate CDV-Explorer for Nostr as a second ecosystem alongside Bitcoin. This required handling different document structures, metadata fields, proposal identifiers, and classification schemes before mapping both corpora to the canonical IP object model. For example, the two ecosystems differ in file naming conventions, metadata representations, and the consistency of status and classification attributes.

Once mapped to the uniform IP object model, both ecosystems are processed by the same analysis modules, postprocessing stage, and frontend views. This confirms that the reusable analysis layer is not Bitcoin-specific, even though Bitcoin is the more mature and structurally richer corpus. More broadly, it validates the architectural separation introduced in Section 2: ecosystem-specific concerns remain confined to data acquisition and preparation, while downstream analyses and visualizations are reusable across ecosystems.

4.2 Gaining Insights into IP Spectra

CDV-exhibiting ecosystems face several unaddressed challenges, including missing ecosystem overview and limited change impact assessment [3, Sec. 6]. Consistent with this tool-paper scope, we report first representative findings from Bitcoin and Nostr that illustrate the structural knowledge CDV-Explorer makes explicit, leaving a detailed empirical investigation for future work.

For Bitcoin, the BIP authorship graph reveals a long-tail contribution structure with many isolated authors and few structurally central contributors. Classification and evolution views expose how the BIP corpus is distributed across

types and lifecycle states and how governance changes, such as the BIP3 taxonomy revision, reshape the corpus over time. The same analyses surface different characteristics in Nostr: its protocol creator authored one third of all NIPs, while more than 80% remain in draft status, indicating strong authorship concentration and a rapidly evolving proposal spectrum. Together, these views provide a compact overview that would otherwise require reconstruction from proposal documents and repository history.

Furthermore, CDV-Explorer surfaces interrelations through the three extraction strategies described in Section 3.1, allowing users to inspect where declared, syntactic, and semantic dependency signals overlap or diverge. This helps reveal undocumented dependencies and feature interactions at scale. For instance, we identified a documentation gap in BIP44, where dependency references to other BIPs were missing from the formal metadata, resulting in a corrective pull request to the upstream repository [15]. Centrality measures over the resulting dependency networks also consistently surfaced historically important BIPs. This makes dependency networks useful navigation aids for newcomers and maintainers by externalizing knowledge otherwise embedded in proposal documents and community expertise. For researchers, CDV-Explorer makes decentralized feature spaces observable as analyzable structures. For practitioners, it supports ecosystem navigation, implementation prioritization, and anticipation of downstream change propagation.

5 Conclusion

We presented CDV-Explorer, a pipeline for mining and analyzing proposal spectra in OSS ecosystems exhibiting Community-Driven Variability (CDV). Its unified Improvement Proposal (IP) object model separates ecosystem-specific acquisition and preprocessing from reusable analysis and postprocessing, turning heterogeneous feature specifications into structured insights exposed through an interactive web frontend.

The Bitcoin and Nostr instantiations show that CDV-Explorer adapts to distinct IP catalogs while reusing the same downstream analyses and visualizations. The resulting interactive views support exploration of proposal evolution, dependency networks, and authorship structure, helping mitigate the CDV challenges of missing ecosystem overview and limited change impact assessment. Together, CDV-Explorer makes the proposal spectra of CDV-exhibiting ecosystems more explicit, thereby improving comprehension and navigation for both researchers studying decentralized feature spaces and practitioners working with IPs.

For future work, we plan to integrate additional CDV-exhibiting ecosystems and extend the analysis stage toward the derivative spectrum, e.g., by linking IPs to concrete implementations [16]. We also plan a field study with ecosystem developers to empirically evaluate the tool's usefulness and identify missing capabilities. In this way, we aim to further strengthen support for navigating CDV-exhibiting OSS ecosystems and mitigating CDV-induced challenges.

References

[1] Apel, S., Batory, D. S., Kästner, C., and Saake, G. *Feature-Oriented Software Product Lines - Concepts and Implementation*. Springer, 2013. DOI: 10.1007/978-3-642-37521-7.

[2] Bögli, R., Boll, A., Schultheiß, A., and Kehrer, T. "Beyond Software Families: Community-Driven Variability". In: *FSE Companion*. 2025, pp. 571–575. DOI: 10.1145/3696630.3728501.

[3] Bögli, R., Boll, A., Schultheiß, A., and Kehrer, T. "Community-driven variability: characterizing a new software variability paradigm". In: *Autom. Softw. Eng.* 33.2 (2026), p. 67. DOI: 10.1007/S10515-026-00594-0.

[4] *Bitcoin Improvement Proposals*. URL: https://github.com/bitcoin/bips (visited on Mar. 16, 2026).

[5] Bitcoin Optech. *Bitcoin Feature Matrix: tracking interoperability of Bitcoin products & services*. URL: https://bitcoinops.org/en/compatibility (visited on Feb. 25, 2026).

[6] The Bitcoin Hole. *Software Wallets: Comparing 25 Bitcoin Software Wallets Feature by Feature*. The Bitcoin Hole. URL: https://thebitcoinhole.com/software-wallets (visited on July 20, 2025).

[7] *Nostr Implementation Possibilities (NIPs)*. URL: https://github.com/nostr-protocol/nips (visited on Feb. 25, 2026).

[8] Hemmati, H., Nadi, S., Baysal, O., Kononenko, O., Wang, W., Holmes, R., and Godfrey, M. W. "The MSR cookbook: mining a decade of research". In: *MSR*. 2013, pp. 343–352. DOI: 10.1109/MSR.2013.6624048.

[9] Kotti, Z., Kravvaritis, K., Dritsa, K., and Spinellis, D. "Standing on shoulders or feet? An extended study on the usage of the MSR data papers". In: *EMSE* 25.5 (2020), pp. 3288–3322. DOI: 10.1007/s10664-020-09834-7.

[10] Martinson, J., Hermann, K., Houbbi, R., Stechow, D., and Berger, T. "Lightweight Visualization of Software Features with HAnS-viz". In: *SPLC*. 2025, pp. 31–34. DOI: 10.1145/3748269.3748487.

[11] Schwarz, T., Mahmood, W., and Berger, T. "A Common Notation and Tool Support for Embedded Feature Annotations". In: *SPLC*. 2020, pp. 5–8. DOI: 10.1145/3382026.3431253.

[12] Andam, B., Burger, A., Berger, T., and Chaudron, M. R. V. "FLOrIDA: Feature LOcatIon DAshboard for extracting and visualizing feature traces". In: *VaMoS*. 2017, pp. 100–107. DOI: 10.1145/3023956.3023967.

[13] Bögli, R. and Eglil, M. *CDV-Explorer*. July 2026. DOI: 10.5281/zenodo.21281061. URL: https://github.com/SEG-UNIBE/cdv-explorer.

[14] Bögli, R. "CDV-Explorer: Navigating Improvement Proposal Spectra in Decentralized OSS Ecosystems [DEMO]". YouTube. Video. June 1, 2026. URL: https://youtu.be/56GKRexRuoI.

[15] Eglil, M. *Bip-0044: Add Requires Header for BIP32 and BIP43*. Merged PR. Jan. 5, 2026. URL: https://github.com/bitcoin/bips/pull/2072.

[16] Bögli, R. "Towards Systematic Treatment of Community-Driven Variability". In: *ICSE Companion*. Apr. 2026. DOI: 10.1145/3774748.3787644.