Web3BlockSet: A Dataset for Empirical Research in Blockchain-Oriented Software Engineering

Pamella Soares
State University of Ceara
Fortaleza, Brazil
pamella.soares@aluno.uece.br

Giuseppe Destefanis
University College London
London, United Kingdom
g.destefanis@ucl.ac.uk

Allan C. N. dos Santos
Fluminense Federal University
Rio de Janeiro, Brazil
allans@midiacom.uff.br

Allysson Allex Araújo
Federal University of Cariri
Juazeiro do Norte, Brazil
allysson.araujo@ufca.edu.br

Raphael Saraiva
Jerffeson Souza
raphael.saraiva@aluno.uece.br
jerffeson.souza@uece.br
State University of Ceara
Fortaleza, Brazil

Abstract

The rapid evolution of blockchain technology has created a diverse ecosystem of platforms, tools, and applications, and understanding how these technologies are adopted and used in practice is essential for advancing Blockchain-Oriented Software Engineering. This paper presents Web3BlockSet, a curated dataset that enables empirical insights into the blockchain ecosystem through the analysis of Mining Software Repositories on GitHub. We collected 391,596 Issues and Pull Requests from organizations that maintain core blockchain technologies and from community developers who create applications. The dataset offers broad categorization across the blockchain stack, a dual perspective of technology vendors and adopters, and rich metadata that supports diverse analytical approaches.

Keywords

Blockchain-Oriented Software Engineering, Mining Software Repositories, GitHub

[1]

1 Introduction

Blockchain-Oriented Software Engineering (BOSE) [14] establishes a dedicated field to the specificities of developing decentralized systems, recognizing that blockchain applications require distinct practices due to characteristics such as immutability, peer-to-peer networks, and enhanced security mechanisms [10]. In particular, the blockchain ecosystem has evolved rapidly, with a proliferation of platforms, tools, protocols, and services that shape how developers build decentralized applications [6].

Understanding this evolution requires examining how blockchain technologies are adopted in practice through developer interactions. In recent studies, Das et al. [7] investigated collaboration patterns and categorized repositories into three broad categories (Tool, Cryptocurrency Application, non Cryptocurrency-based Application). Opu et al. [13] applied topic modeling in issues, identifying 11 subcategories and analyzing resolution patterns. Although this issue-level categorization provides information on the developer challenges, it does not distinguish between technology providers and adopters, nor does it offer repository-level categorization by blockchain stack layers, limiting investigations into how development challenges propagate from platform maintainers to application developers, for example.

To address these gaps, we present Web3BlockSet, a curated dataset that enables empirical insights into the blockchain ecosystem through developer interactions in both official and community repositories. Through a Mining Software Repository (MSR) approach using community-curated sources, we collected 391,596 Issues and Pull Requests from 80 organizations (381 official repositories) and 4,500 community repositories (from 2,787 organizations) spanning 2012-2024. Compared to previous works, Web3BlockSet offers three key advances: (1) dual perspective distinguishing technology providers from adopters, (2) repository-level categorization across blockchain stack layers (protocols, SDKs, DeFi, wallets, tools) enabling domain-specific analyses, and (3) rich metadata with 20+ fields capturing repository characteristics, user interactions, and temporal evolution. This dataset enables empirical investigations into technology adoption patterns, cross-platform development challenges, collaboration dynamics, issue resolution, security vulnerabilities, and evolution of development practices.

2 Methodology

Our methodology follows a Mining Software Repositories (MSR) approach structured in four phases, as shown in Figure 1.

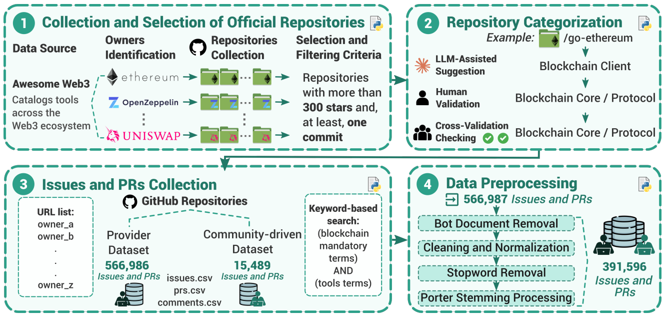
1 Collection and Selection of Official Repositories
Data Source
Owners Identification
Repositories Collection
Selection and Filtering Criteria
Awesome Web3
Catalogs tools across the Web3 ecosystem
ethereum
OpenZeppelin
UNISWAP
Repositories with more than 300 stars and, at least, one commit
2 Repository Categorization
Example: /go-ethereum
LLM-Assisted Suggestion
Blockchain Client
Human Validation
Blockchain Core / Protocol
Cross-Validation Checking
Blockchain Core / Protocol
3 Issues and PRs Collection
URL list:
owner_a
owner_b
owner_z
GitHub Repositories
Provider Dataset
566,986
Issues and PRs
issues.csv
prs.csv
comments.csv
Community-driven Dataset
15,489
Issues and PRs
Keyword-based search: (blockchain mandatory terms) AND (tools terms)
4 Data Preprocessing
566,987 Issues and PRs
Bot Document Removal
Cleaning and Normalization
Stopword Removal
Porter Stemming Processing
391,596
Issues and PRs

Figure 1: Data collection and preprocessing methodology.

[1] Author's version (preprint) (Paper accepted for publication at Mining Software Repository - Data and Tool Showcase Track)

2.1 Step 1: Collection and Selection of Official Repositories

Related to the **Data Source**, we use *Awesome Web3* [2], a community-maintained catalog of Web3 ecosystem tools, as a primary reference to identify repositories and organizations. The *Awesome Web3* repository [3] is curated by blockchain practitioners and had 763 stars, 165 forks, and 83 active contributors at the time of collection. We chose this source because of its quality standards enforced through: a *Code of Conduct* ensuring constructive participation, and *PRs-based validation* with multi-contributor review, which can reduce individual bias and ensures the relevance of cataloged tools.

In **Owner Identification and Repository Collection**, from the *Awesome Web3* list of tools, we identified the tool (e.g. *Brownie*) under a category (e.g. "Development Environment"), accessed its repository [8] and the corresponding organization [9]. We repeated this process for all listed tools to build a list of representative Github URL owners in the Web3 ecosystem, resulting in 80 organizations. We implemented an automated script to download repository metadata for each identified organization. From this collection, in **Selection and Filtering Criteria**, we filtered repositories with more than 300 stars and at least one commit, focusing on actively maintained and community-recognized projects. The 300-star threshold captures established projects while filtering out experimental repositories with minimal community engagement.

2.2 Step 2: Repository Categorization

The *Awesome Web3* repository provides an initial classification of tools into categories such as Protocol, DevOps, Testing, Security, and others. Since we expanded the repository set by including all organizational repositories, we applied a second categorization focused on the *nature and purpose of each repository*. We adopted a hybrid approach with three steps: **(1) LLM-Assisted Suggestion:** We used LLM[2] to propose an initial classification based on each repository's *name*, *description*, and *organization*, which has been demonstrating the effectiveness for automated labeling tasks without fine-tuning, showing considerable agreement with human annotators [1, 5]. **(2) Human Validation:** An experienced researcher with 5+ years of experience in BOSE manually reviewed and adjusted automatic classifications when necessary. Ambiguous cases were inspected by accessing the repository directly. For example, the LLM classified the `go-ethereum` repository as a "Blockchain Client", but the researcher broadened this class to include other repositories with similar purposes (see Figure 1). **(3) Cross-Validation by Experts:** To ensure the reliability of the classifications, we divided the 381 repositories among 4 other experienced BOSE researchers for validation. Each expert received approximately 95 repositories with their proposed categories and indicated agreement or suggested alternatives. Agreement was measured against the initial categorization from step 2, with disagreements resolved through discussion to reach consensus. The final categorization (overall agreement of 92.4%) was stored in the column `repository_classification`.

2.3 Step 3: Issues and Pull Requests Collection

We adopted a two approach to compiling the dataset to allow for an understanding of both the technologies maintained by official organizations and their practical use by the community. For the **Provider Dataset**, using the URLs of the respective organizations captured in Step 1, we used a script to collect all their Issues and PRs. For the **Community Dataset**, we used a *keyword-based search strategy* based on the technologies collected in the Provider Dataset. The search strategy employed a structured keyword file [17] combining mandatory blockchain-related terms ("blockchain", "web3", "smart contract", among others) with organization-specific patterns (tool names, library names, etc.) to ensure relevance and capture diverse technology mentions. This approach filters for adopter reduces false positives from generic mentions. We identified community projects that explicitly mention these tools in the *titles* or *bodies* of Issues or PRs. Unlike the Provider Dataset, we did not apply the popularity criterion (more than 300 stars), as our goal was to capture specific discussions rather than repository completeness. Hence, the Community Dataset includes smaller or emerging projects that demonstrate interactions with ecosystem technologies.

2.4 Step 4: Data Preprocessing

We observed a large number of automated messages from bots and automated tools. Since our focus is on developer challenges, we only filter Issues and PRs by *title*, *body*, and *comments* that did not contain bot-related keywords, to prioritize genuine human discussions (e.g., "bot", "snyk" "copilot", or "graphite").

To improve data quality and consistency, we also performed a cleaning and normalization process. We removed Markdown elements (e.g., code blocks and snippets), applied Unicode Normalization (NFKD), converted characters to ASCII, and others terms with patterns specific to technical discussions (such as hexadecimal addresses, long hash strings, etc.). During stopword removal, we used the NLTK [18] standard set, expanded with developer interaction terms. We also applied the Porter stemming process [15], reducing derived words to their root forms (e.g., "language" and "languages" to "languag"). Finally, we discarded documents with fewer than 15 words after preprocessing to ensure minimal semantic density, filtering out very short messages that typically lack sufficient context for meaningful analysis (e.g., "+1", "fixed", "thanks").

3 Characterizing the Web3BlockSet

Web3BlockSet comprises **391,596 Issues and PRs**, mapping the blockchain ecosystem from official foundations to community-level applications. Providing an **overview of official repositories**, on average, they received 2,426 stars (median of 663) and 832 forks (median of 320). Moreover, according to repository metadata, the five most used programming languages are TypeScript (26.8%), JavaScript (17.2%), Rust (13.2%), Solidity (11.5%) and Go (9%). 20.5% of repositories are archived, which were retained to provide insights into maintainability patterns. Specifically, the repository with the most stars (62,135) is Sway[3], *"a language developed for the Fuel blockchain"* by the *FuelLabs* organization, with 5,419 forks.

[2] Claude by Anthropic (claude-3-5-sonnet). The complete categorization prompt and validation methodology are documented in the repository [17].

[3] https://github.com/FuelLabs/sway

Figure 2 shows the distribution of repository categories for **provider organizations**. The dataset includes 40 repositories classified as *Educational Resource/Boilerplate*, followed by *SDK/Library* (37) and *Blockchain Core/Protocol* (34). Examples of organizations in some of the categories include: SDK/Library (web3, ethers-io, wevm, OpenZeppelin); Blockchain Core/Protocol (Ethereum, Solana, Near, Algorand, Avalanche, BNB Chain, Polkadot, FuelLabs, Neo); DeFi Protocol (Uniswap, Aave, Compound, Curve Finance, Sablier). Development Platform/Framework (Foundry, Hardhat, Truffle, Embark); Wallet Dapp/Library/Browser (MetaMask, Trust Wallet, Rainbow, Coinbase Wallet, WalletConnect).

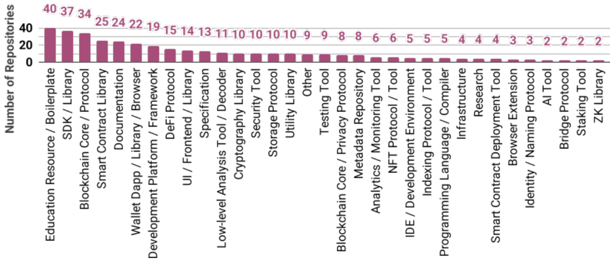

Figure 2: Repository classification distribution.

Web3BlockSet captures developer interactions through **Issues and Pull Requests** across both datasets. For the Provider Dataset, we collected discussions from 381 official repositories across 80 organizations, totaling 383,387 documents (157,452 Issues and 225,935 PRs). Additionally, the Community Dataset contains 8,209 documents (3,953 Issues and 4,256 PRs). Figure 3 shows the temporal distribution of Issues and PRs created over the years.

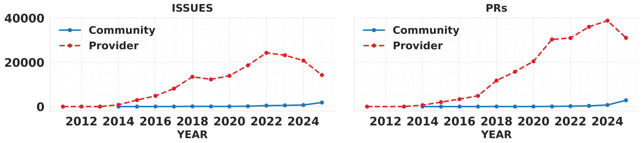

Figure 3: Issues and PRs created over the years.

Concerning the **schema and data structure**, Web3BlockSet is distributed in two CSV files as described in Table 1. The `data_source` field in issues_prs file distinguishes between *provider* (383,387 documents from 381 official repositories) and *community* (8,209 documents from 4,500 adopter repositories). The **issues_prs** file provides levels of text processing in separate columns: (1) `raw_text` preserves original issue title, body, and comments; (2) `clean_text` applies cleaning and normalization; and (3) `stemmed_text` additionally applies Porter stemming. This design allows researchers to access original discussions or work with normalized text for computational linguistics applications.

4 Use Case: Blockchain DEx Analysis

Web3BlockSet has been applied in an ongoing study investigating Blockchain Developer Experience (BcDEx) [16] in the context of Blockchain as a Service (BaaS) platforms. The study examines how developers interact with BaaS platforms by analyzing over 12,800 issues and PRs from provider and community repositories. Using BERTopic [12] for topic modeling and the SPACE framework [11] for productivity analysis, the research identifies development-related topics and characterizes BcDEx across five dimensions: Satisfaction and Well-Being, Performance, Activity, Communication and Collaboration, and Efficiency and Flow. Results show complementary specialization patterns, with provider repositories focusing on infrastructure abstraction (wallet integration, network RPC) while community repositories emphasize implementation challenges (smart contract development, network configuration).

This application illustrates how Web3BlockSet enables the empirical investigation of developer practices, technology adoption patterns, and productivity factors in blockchain development. The dataset's dual perspective, combining official platform repositories with community applications, provides researchers with insight into the challenges in providing and using the technologies in practice. Although this study focuses on BaaS platforms, the dataset's structure and methodology can be adapted to investigate any other repository domain classified in Figure 2.

Table 1: Web3BlockSet Schema and Data Structure.

File	Fields
repositories.csv (381 records)	organization, description, stars, forks, language, created_at, updated_at, archived, size_kb, has_issues, open_issues, default_branch, clone_url, html_url, repository_category
issues_prs.csv (391,596 records)	repository, owner, issue_id, issue_number, issue_title, issue_body, state, created_at, updated_at, closed_at, author, author_id, locked, comments_count, commits_count, labels, type, concatenated_comments, comment_authors, comments_count_filtered, year, raw_text, clean_text, stemmed_text, data_source, repository_category

5 Ideas for Future Research Questions

Web3BlockSet combines provider and community repositories, temporal coverage (2012–2024), and a repository categorisation distinguishing development platforms, blockchain protocols, educational resources, and others. This structure creates valuable opportunities for empirical studies that connect platform development with practical adoption, presented below as open research questions.

(1) Technology adoption and platform migration. **(a) Which blockchain platforms and tools are adopted by community developers, and how do adoption patterns change over time?** Existing BOSE studies focus on theoretical platform comparisons but lack empirical adoption data. To address this, the community dataset records explicit tool mentions in the `raw_text` or `owner_used` column across 2,787 repositories, with `created_at` timestamps tracking adoption chronology. **(b) What factors predict blockchain tool abandonment, and how long do projects remain maintained?** To investigate this, the Provider Dataset includes 20.5% archived repositories with `archived` flags, combined with `created_at` and `updated_at` timestamps measuring lifespan.

(2) Cross-platform development challenges. **(a) Do different blockchain platforms exhibit distinct developer challenges and issue types?** Prior work examines blockchain development practices generally [4, 10] but does not systematically compare how platform architecture affects developer problems. For this purpose, the repository classification distinguishes 34 Blockchain Core/Protocol

repositories (Ethereum, Solana, Near, Algorand, Avalanche, Polkadot, and others). The repository_classification column combined with labels and clean_text across provider documents enables cross-platform issue comparison. **(b) How does programming language choice correlate with issue resolution time and developer sentiment across blockchain platforms?** To explore this, the language metadata combined with created_at and closed_at timestamps addresses this question, as the language distribution (see Section 3) reflects distinct platform ecosystems.

(3) Developer collaboration and contribution patterns. **(a) How do collaboration networks differ between core maintainers and community contributors across blockchain platforms?** Empirical evidence on contributor roles, interaction patterns, and retention in blockchain projects remains limited. To answer this, author, author_id, owner, and comment_authors fields can capture interaction patterns. author and owner can indicate organizational affiliation, and activity frequency (author_id across created_at) distinguish core maintainers from occasional contributors. The comment_authors field captures multi-party discussions and collaboration networks. **(b) What factors predict long-term contributor retention in blockchain projects?** For this analysis, repeated author_id values across created_at timestamps measure sustained contribution over the temporal span (2012–2024).

(4) Issue resolution and maintainability patterns. **(a) How does issue resolution time vary across repository categories and organizational size?** To examine this, the 383,387 provider documents include number_of_comments, created_at, updated_at, and closed_at fields measuring maintainer responsiveness. The dataset includes 40 Educational Resource/Boilerplate repositories, 37 SDK/Library repositories, 34 Blockchain Core/Protocol repositories, and other categories representing different maintenance contexts. The repository_classification combined with temporal fields measures resolution patterns across categories. **(b) Does repository popularity correlate with maintainer responsiveness and code contribution acceptance rates?** To investigate this, the stars and forks metrics can indicate project popularity, and the relationship between popularity metrics and pull request ratios measures contribution acceptance.

(5) Security vulnerability disclosure and response. **(a) How do vulnerability disclosure and resolution practices differ across blockchain platforms and repository types?** Security is paramount in blockchain development due to immutability and financial stakes, but empirical evidence on disclosure practices remains limited. To address this, the labels column captures security-related tags across issues and pull requests, combined with created_at and closed_at fields measuring time-to-resolution for security issues across the 381 provider repositories. **(b) How do security vulnerabilities in provider repositories propagate to community projects?** For this analysis, the community dataset's explicit references to provider tools (captured in raw_text) allow tracking whether security discussions in provider repositories appear in the 2,787 community repositories, enabled by the dual dataset structure.

(6) Evolution of development terminology. **(a) How does blockchain development terminology evolve, and which terms predict emerging technologies or practices?** To explore this, The temporal dimension combined with clean_text enables longitudinal vocabulary analysis from 2012 to 2024, capturing discussions through periods of ecosystem evolution and providing evidence for studying how developer language adapts to technological change.

These research questions unravel specific dataset columns, categorization schemes, and temporal coverage to address gaps in BOSE empirical research. The repository and Issues/PR metadatas provide the measurement infrastructure for investigating these questions.

6 Final Remarks

This paper presents Web3BlockSet, a curated dataset that maps the blockchain ecosystem through Mining Software Repositories techniques applied to GitHub. By analyzing 80 organizations with 381 official repositories and 4500 community repositories, we bridge the gap between theoretical blockchain software engineering research and empirical evidence of industry practice. Our dual-perspective approach captures both the maintainers' viewpoint through official repositories and the developers' perspective through community applications, analyzing Issues and Pull Requests to present real-world development practices, challenges, and technological trends.

Regarding the *limitations* of this work, the provider dataset relies on the Awesome list and the community dataset based on search strings, and may not encompass all relevant projects in the blockchain ecosystem, although we have recognized some well-known projects within the dataset. Additionally, repository classification was validated by experts reviewing proposed categories rather than performing independent categorization, which may affect inter-rater agreement interpretation. However, the high agreement (92.4%) and consensus-based resolution of disagreements suggest reasonable categorization quality. The community dataset (8,209 documents) is substantially smaller than the provider dataset (383,387 documents), which may limit certain comparative analyses between adopter and provider perspectives, though it still captures meaningful adoption patterns across 4,500 repositories.

As *ideas for future improvements*, we intend to incorporate additional repository discovery methods beyond Awesome lists and search strings. Furthermore, we plan to add sentiment analysis scores, code metrics, and dependency network data. The dataset will be updated quarterly to maintain temporal relevance and capture the evolution of blockchain technologies.

Artifacts Availability

To promote transparency and reproducibility, all study materials are openly available in our public repository [17].

Acknowledgments

This work received partial funding from CNPq-Brazil, Universal grant 404406/2023-8, and support from CAPES - Funding Code 001.

References

[1] Gabriel Aracena, Kyle Luster, Fabio Santos, Igor Steinmacher, and Marco A Gerosa. 2025. Applying Large Language Models to Issue Classification: Revisiting with Extended Data and New Models. *Science of Computer Programming* (2025), 103333.

[2] Awesome Web3. [n. d.]. Awesome Web3. https://awesome-web3.com/

[3] Ahmet Aygün. [n. d.]. Awesome Web3 Repository. https://github.com/ahmet/awesome-web3

[4] Amiangshu Bosu, Anindya Iqbal, Rifat Shahriyar, and Partha Chakraborty. 2019. Understanding the motivations, challenges and needs of Blockchain software developers: a survey. *Empirical Software Engineering* 24, 4 (2019), 2636–2673.

[5] Giuseppe Colavito, Filippo Lanubile, Nicole Novielli, and Luigi Quaranta. 2024. Leveraging gpt-like llms to automate issue labeling. In *Proceedings of the 21st International Conference on Mining Software Repositories*. 469–480.

[6] Collin Connors and Charles P Shelton. 2023. Survey of prominent blockchain development platforms. *Future Generation Computer Systems* 146 (2023), 195–223. doi:10.1016/j.future.2023.04.019

[7] Ajoy Das, Gias Uddin, and Guenther Ruhe. 2022. An empirical study of blockchain repositories in GitHub. In *Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering*. 211–220.

[8] Eth-Brownie. [n. d.]. Brownie: A Python-based development and testing framework for smart contracts. https://github.com/eth-brownie/brownie.

[9] Eth-Brownie. [n. d.]. Eth-Brownie Organization. https://github.com/eth-brownie.

[10] Mahdi Fahmideh, John Grundy, Aakash Ahmad, Jun Shen, Jun Yan, Davoud Mougouei, Peng Wang, Aditya Ghose, Anuradha Gunawardana, Uwe Aickelin, et al. 2022. Engineering blockchain-based software systems: Foundations, survey, and future directions. *Comput. Surveys* 55, 6 (2022), 1–44.

[11] Nicole Forsgren, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck, and Jenna Butler. 2021. The SPACE of Developer Productivity: There's more to it than you think. *Queue* 19, 1 (2021), 20–48.

[12] Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794* (2022).

[13] Md Nahidul Islam Opu, Md Shahidul Islam, Sara Rouhani, and Shaiful Chowdhury. 2025. Understanding the Issue Types in Open Source Blockchain-based Software Projects with the Transformer-based BERTopic. *arXiv preprint arXiv:2506.11451* (2025).

[14] Simone Porru, Andrea Pinna, Michele Marchesi, and Roberto Tonelli. 2017. Blockchain-oriented software engineering: challenges and new directions. In *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*. IEEE, 169–171.

[15] Martin F Porter. 2001. Snowball: A language for stemming algorithms.

[16] Pamella Soares, Allysson Allex Araújo, Raphael Saraiva, and Jerffeson Souza. 2024. Towards Blockchain Developer Experience (BcDEx): Exploring Dimensions of Developer Experience in Blockchain-oriented Software Engineering. In *Simpósio Brasileiro de Engenharia de Software (SBES)*. SBC, 616–622.

[17] Pamella Soares, Giuseppe Destefanis, Allan C. N. dos Santos, Allysson Allex Araújo, Raphael Saraiva, and Jerffeson Souza. 2025. Supplementary material for "Web3BlockSet: A Dataset for Empirical Research in Blockchain-Oriented Software Engineering". doi:10.5281/zenodo.17576503

[18] Jalaj Thanaki. 2017. *Python natural language processing*. Packt Publishing Ltd.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009