Parsing the Distributed Web: Techniques for Crawling and Indexing Client-Side Rendered and SPA-Based Decentralized Websites

Dr. Gevorg Margarov[1*], Artyom Harutyunyan[2]

*[*1]Professor, Head of Department, Information Security and Software Development Department, National Polytechnic University of Armenia, Yerevan, Armenia*
[2]Team Lead, Blockchain Architect, Blockstars LLC, Yerevan, Armenia
*[*1]E-mail: gmargarov@gmail.com*
[2]E-mail: artyomharutyunyans@gmail.com

Abstract

The rise of decentralized web protocols—such as IPFS, ENS, UNS, and BNB NS—has enabled the creation of censorship-resistant, privacy-preserving websites. However, the prevalence of Single Page Applications (SPAs) and client-side rendered architectures presents significant challenges for traditional crawlers, which are incapable of executing JavaScript or resolving content stored across distributed networks. As a result, a large portion of Web3 content remains unindexed and effectively invisible. This paper presents Web3Compass, a fully operational indexing system purpose-built for the decentralized web, capable of dynamically rendering SPA-based websites through headless browsers and integrating with decentralized domain resolution protocols. Web3Compass monitors blockchain-based registries in real-time, resolves domains to IPFS content identifiers (CIDs), and retrieves content through a self-hosted IPFS infrastructure with pinning support. When SPA content is detected, it triggers a dynamic rendering pipeline using Puppeteer to extract the fully rendered DOM for indexing. The system also incorporates a censorship-resistant gateway, dweb3.wtf, to ensure reliable access to indexed content without dependency on centralized IPFS gateways. Web3Compass currently indexes over 700,000 decentralized domains and supports a wide array of naming protocols. This paper details its architecture, implementation, and evaluation—demonstrating improved indexing completeness, access resilience, and visibility of dynamic decentralized content.

Keywords: *Web3 search, decentralized web indexing, SPA crawling, client-side rendering, IPFS, ENS, dynamic rendering, Puppeteer, content discovery, censorship resistance, headless browser, decentralized domain resolution, IPFS pinning, blockchain domain monitoring*

1. Introduction

The emergence of the decentralized web, commonly referred to as Web3, signifies a fundamental shift in how websites are hosted, resolved, and accessed. Unlike the traditional Web2 architecture, which relies on centralized servers and domain name systems, Web3 leverages distributed protocols such as the InterPlanetary File System (IPFS), Ethereum Name Service (ENS), Unstoppable Domains (UNS), and BNB Name Service (BNS) to publish, resolve, and serve content across a decentralized infrastructure. While these technologies aim to promote privacy, censorship resistance, and open access, the lack of robust search and discovery mechanisms remains a critical bottleneck to usability and adoption.

A growing number of decentralized websites are built as Single Page Applications (SPAs), relying heavily on client-side rendering (CSR) using JavaScript. Such websites typically return minimal or empty HTML from the server and render content dynamically in the browser. Traditional crawlers, designed for static or server-rendered pages, are inherently incapable of parsing and indexing these SPA-based sites. This results in significant visibility gaps, rendering much of the decentralized web inaccessible to users unless a direct link is known.

Moreover, the challenge is compounded in the context of decentralized content storage. Systems like IPFS and Arweave distribute website data across peer-to-peer nodes without guaranteeing availability through centralized endpoints. Access often depends on third-party gateways (e.g., ipfs.io), which introduce vulnerabilities related to censorship, reliability, and regulatory compliance.

This paper addresses the limitations of traditional crawling techniques in decentralized environments by focusing on dynamic rendering using headless browsers such as Puppeteer. Specifically, it presents the architecture, implementation, and evaluation of a real-world system Web3Compass which integrates dynamic rendering into its indexing pipeline to process and catalog decentralized SPAs. By executing JavaScript and extracting the fully rendered DOM, the system enables accurate indexing of client-side content previously invisible to crawlers.

The core research question explored is:

How does dynamic rendering using headless browsers improve the completeness, reliability, and censorship resistance of crawling and indexing client-side rendered and SPA-based decentralized websites compared to traditional crawling methods?

Through this investigation, the paper contributes practical insights into building scalable and censorship-resistant indexing infrastructure for the distributed web, with emphasis on search completeness, access resilience, and content availability in JavaScript-heavy decentralized environments.

2 Related Work and Literature Review

2.1. Traditional Web Crawling and Indexing

Traditional web crawlers function by systematically traversing hyperlinks across websites, fetching documents, and extracting content for indexing, using algorithms like Depth-First Search (DFS) [1]. Algorithms such as Depth-First Search (DFS) have been widely adopted for link traversal, with crawlers extracting titles, meta descriptions, and keywords from server-rendered HTML to populate search engine indices. These crawlers are optimized for static websites or those that generate full HTML responses server-side, enabling efficient and scalable indexing pipelines, but they struggle with JavaScript-heavy SPAs [2].

However, with the proliferation of JavaScript-heavy websites, particularly Single Page Applications (SPAs), these traditional methods encounter fundamental limitations. SPAs dynamically construct content in the browser using JavaScript after an initial minimal HTML payload is served. Consequently, conventional crawlers fail to capture or interpret such content, resulting in incomplete or inaccurate indexing.

2.2. Decentralized and Distributed Crawling Architectures

To address the challenges of scalability and fault tolerance in traditional centralized crawling systems, distributed web crawlers have been proposed. One such example is Apoidea, a peer-to-peer (P2P) crawling framework that distributes crawling tasks across nodes using Distributed Hash Tables (DHTs) and bloom filters to avoid duplication. This decentralized approach mitigates single points of failure and improves efficiency by leveraging geographical proximity and network diversity.

Beyond crawling, distributed indexing engines have been layered atop DHTs to create scalable keyword search systems without centralized infrastructure, as demonstrated by frameworks like Minerva. [3] These systems demonstrate the feasibility of building decentralized search architectures but fall short in addressing the complexities introduced by modern client-side rendering.

2.3. Challenges Specific to Decentralized Web (Web3)

Crawling and indexing within Web3 ecosystems introduces several novel challenges:

- Domain Discovery and Resolution: Unlike traditional domain name systems, decentralized name services such as ENS, UNS, and BNS operate via smart contracts. Detecting new registrations and resolving domains requires continuous event monitoring and direct interaction with blockchain registries.
- Content Retrieval: Content on Web3 is often hosted on storage networks like IPFS or Arweave. These systems use content-based addressing (e.g., CIDs) and rely on

distributed peers for data retrieval. This introduces variability in latency, availability, and reliability.

- Dynamic Content Rendering: Many decentralized sites are built as SPAs, using frameworks like React or Vue, which render content entirely in-browser. Without the ability to execute JavaScript, traditional crawlers are blind to such content, a challenge well-documented in dynamic web indexing.
- Censorship Resistance: Despite being hosted on decentralized protocols, access to these websites frequently occurs through centralized gateways like ipfs.io. These access points can be restricted or censored, undermining the decentralized promise of Web3.

2.4. Incentive Mechanisms and Content Freshness

Maintaining up-to-date indexes in decentralized networks remains an unsolved problem. Researchers have proposed incentive mechanisms to encourage site owners to publish updates via smart contracts, triggering re-indexing workflows. Balancing incentives across publishers and indexers is an active area of exploration, particularly as traditional mechanisms like sitemaps and robots.txt are not applicable to decentralized architecture.

2.5. Dynamic Rendering Using Headless Browsers

To bridge the gap between dynamic client-side content and static crawler expectations, Web2 systems have adopted headless browsers — notably Puppeteer — to load pages in a full browser environment, execute JavaScript, and extract the final rendered DOM. This method has proven effective for indexing SPAs and dynamic content in centralized contexts. Transferring this technique to decentralized systems enables visibility into a class of websites that were previously inaccessible to crawlers.

Dynamic rendering thus emerges as a practical solution to the inherent limitations of static crawling. When paired with blockchain-based domain monitoring and IPFS retrieval strategies, it allows for comprehensive indexing of the decentralized web — including content that is otherwise hidden behind JavaScript-rendered frontends.

3. Challenges in Crawling Decentralized SPAs

T

he architecture of the decentralized web introduces structural and technical constraints that make traditional crawling impractical. Unlike static HTML pages hosted on centralized servers, decentralized websites often exist as JavaScript-heavy SPAs stored on peer-to-peer networks. Their discovery, resolution, retrieval, and rendering require specialized mechanisms not found in conventional search engines. Web3Compass addresses these challenges through a tightly integrated architecture designed to operate in the absence of centralized DNS, server-side rendering, and standardized metadata.

3.1. Domain Discovery

The first challenge is identifying which decentralized domains are active and point to actual websites. Traditional web crawlers rely on sitemaps or backlinks to discover new pages. In contrast, Web3 domains such as .eth, .crypto, .polygon, and .bnb are registered on-chain through smart contracts. Web3Compass implements continuous event monitoring across multiple decentralized registries including ENS, UNS, and BNB Name Service to detect new registrations and modifications. Each name service uses a unique contract structure, necessitating custom logic to decode registration data and extract metadata such as domain names and associated resolver addresses.

This process is decentralized, asynchronous, and requires robust monitoring infrastructure to track state changes on-chain in near real-time. Domains pointing to traditional Web2 content or inactive placeholders are filtered out during this phase, focusing indexing efforts solely on active, decentralized Web3 websites.

3.2. Content Resolution

Once a domain is discovered, the next step is to resolve it to a content address — typically a content identifier (CID) in the IPFS network. This is done by querying the resolver smart contract associated with the domain to extract fields like contenthash, which stores the CID in an encoded format. This process varies across registries: ENS uses a layered resolver model where each domain points to a resolver contract that implements a contenthash() function. Other systems such as UNS and BNS have different schemas, but the general logic remains: extract the CID that points to decentralized storage.

Domains may use public resolvers or custom ones, the latter requiring validation against known interfaces to ensure compatibility. Web3Compass tracks each domain’s resolver state individually, enabling automatic re-resolution and re-indexing when the contenthash changes.

3.3. Content Retrieval

Retrieving content from decentralized networks like IPFS poses another challenge. Unlike traditional web servers, IPFS nodes are ephemeral, and content availability depends on whether any node on the network is currently hosting the data. Public gateways like ipfs.io may cache or drop content arbitrarily, creating uncertainty in access. Web3Compass mitigates this by operating its own dedicated IPFS infrastructure, hosting and pinning critical content to ensure retrieval regardless of third-party gateway status.

Pinning guarantees content persistence by instructing the local IPFS node not to garbage-collect the data. For scalability and resilience, Web3Compass deploys multiple IPFS nodes. If a node becomes unresponsive, others in the cluster maintain availability. In

cases where content is not locally pinned, fallback retrieval is attempted via public gateways, though such paths are deprioritized due to reliability concerns.

3.4. Content Retrieval

Perhaps the most significant technical hurdle arises when the resolved CID points to a SPA or client-side rendered application. These websites, common across Web3, return minimal server-side HTML and rely entirely on JavaScript to populate the page content. Traditional crawlers, which do not execute scripts, index only an empty shell — resulting in broken or meaningless search results.

Web3Compass addresses this by dynamically detecting whether a website is client-rendered. If so, it triggers a headless browser session using Puppeteer to simulate a real browser environment. The browser loads the page, executes its JavaScript, and waits for the DOM to fully render. Once complete, the browser extracts the final HTML content, which is then parsed and indexed.

This dynamic rendering approach not only enables full visibility into SPA content but also ensures that semantic structures (headers, paragraphs, links) are preserved, enabling meaningful full-text search. Without this step, the decentralized web would remain largely opaque to users.

3.5. Censorship Resistance

The final challenge lies in maintaining access to decentralized content in a censorship-resistant manner. Many users reach IPFS-hosted content through centralized gateways such as ipfs.io or cloudflare-ipfs.com. However, these services are subject to jurisdictional controls and can be compelled to block specific CIDs, undermining the censorship-resistant premise of Web3.

To counter this, Web3Compass operates its own rendering gateway — dweb3.wtf. This service serves the dual purpose of rendering client-side content and bypassing centralized restrictions. Unlike other redirector services that filter or restrict certain content, dweb3.wtf is designed to be permissive and resilient. It supports all major decentralized domain types and delivers faster performance by serving cached, pre-rendered pages when appropriate. The integration of this gateway into the indexing workflow ensures users can access the content they find, regardless of external constraints.

4. System Architecture

The architecture of Web3Compass is designed to operate effectively within the constraints of decentralized infrastructure. It integrates blockchain-based domain monitoring, resilient content retrieval, dynamic client-side rendering, and privacy-respecting indexing into a unified pipeline. This section outlines each core subsystem and

their interactions in the process of crawling and indexing SPA-based decentralized websites.

4.1. Domain Monitoring and Resolution

Web3Compass continuously monitors multiple blockchain-based domain name systems, including ENS, UNS, BNB NS, and others. For each name service, the system runs dedicated event listeners that capture new domain registrations, resolver updates, and content hash changes in real time. The logic for each registry is adapted to its smart contract schema.

Once a domain is detected, the resolution subsystem queries the appropriate resolver contract to extract the contenthash. This value, often encoded, is decoded to retrieve the CID pointing to the content on IPFS. Domains using custom resolvers must implement a standard interface (e.g., contenthash()) to be compatible. Web3Compass validates each resolver's structure before attempting resolution.

4.2. Content Access and Pinning

After a CID is resolved, the system attempts to fetch the corresponding content from IPFS. Web3Compass operates its own IPFS nodes to ensure independence from public gateways and improve reliability. These nodes are configured to pin content under the following conditions:

- The content is HTML or text-based and under 100MB in size.
- The CID has not already been indexed.
- The content passes initial filtering (e.g., is not an image, video, PDF, or other binary format).

Pinning ensures long-term availability by preventing the garbage collection of important data. If the content is not pinned or temporarily unavailable from internal nodes, fallback access is attempted using public IPFS gateways such as ipfs.io, though this is treated as a last resort.

4.3. Dynamic Rendering Pipeline

For client-side rendered websites — common among decentralized applications — traditional crawlers fail due to their inability to execute JavaScript. Web3Compass overcomes this by incorporating a dynamic rendering pipeline.

When a page is detected as a SPA, the system automatically launches a headless browser instance using Puppeteer. The browser loads the page as a real user would, waits for the

DOM to fully render, and then captures the final HTML content. This output is passed to the indexing engine.

This dynamic rendering is essential for visibility into decentralized websites that depend on JavaScript frameworks. Without this component, a significant portion of meaningful content would be excluded from the index.

4.4. Filtering and Indexing

Once content is retrieved and rendered, Web3Compass filters it to exclude non-textual assets. Only content that can be semantically indexed — HTML, text, and metadata — is processed. The indexing pipeline is built on Meilisearch, a lightweight, privacy-respecting search engine optimized for full-text search.

Each indexed entry includes metadata such as:

- Domain name
- CID
- Timestamp of last index
- Resolver type
- Extracted content

Search results are ranked using a combination of relevance metrics, including term frequency, proximity, attribute weighting (e.g., CID vs. content body), exactness, and support for synonyms. The system explicitly avoids behavioral signals like click tracking or cookies to preserve user privacy.

4.5. Censorship-Resistant Gateway

To ensure that indexed content remains accessible, Web3Compass deploys its own rendering gateway at dweb3.wtf. This service acts as a replacement for centralized redirectors like ipfs.io, .limo, or .link, which are prone to filtering, throttling, or takedown.

dweb3.wtf fetches content directly from internal IPFS nodes or public fallbacks and renders it in a standard browser environment. It is integrated with the dynamic rendering pipeline and caches rendered outputs for performance. Importantly, it supports all major decentralized domain formats and does not enforce third-party content restrictions.

The result is an access layer that matches the censorship resistance of the underlying storage while maintaining usability for Web2 and Web3 audiences alike.

5. Implementation Details

This section details the practical implementation of Web3Compass, describing specific technologies, methods, and workflows involved in crawling, indexing, and serving decentralized client-side rendered and SPA-based websites.

5.1. Discovery and Monitoring Logic

Web3Compass employs dedicated monitoring logic tailored to each supported decentralized name service. Real-time event listeners detect domain registrations, modifications, and resolver updates directly from blockchain smart contracts, using tools such as the Alchemy API and the ENS subgraph for efficient resolution. When a domain's content hash is updated, Web3Compass automatically triggers re-indexing, ensuring the freshness of indexed data without relying on conventional sitemaps or web crawling methods.

5.2. Content Retrieval Strategies

For content retrieval, Web3Compass operates a self-hosted IPFS node infrastructure. This internal network of nodes ensures consistent availability and redundancy. Content identified as relevant (HTML or text-based files under 100MB) is pinned to prevent garbage collection and ensure rapid retrieval. If internal nodes cannot immediately fetch content, the system temporarily utilizes public IPFS gateways like ipfs.io as a fallback, though with lower reliability expectations.

5.3. Dynamic Rendering Workflow

To handle JavaScript-heavy decentralized SPAs, Web3Compass integrates Puppeteer-based dynamic rendering. The workflow includes:

- SPA Detection: Initial lightweight checks detect whether retrieved HTML is minimal or empty, indicating potential client-side rendering.
- Headless Browser Rendering: Detected SPAs trigger Puppeteer, which loads the page, executes JavaScript, waits for full DOM rendering, and extracts finalized HTML content.
- Content Extraction: Rendered content is then parsed for indexing, ensuring full capture of dynamic site content.

This approach ensures accurate and comprehensive indexing of decentralized sites that are inaccessible to traditional static crawlers.

5.4. Indexing and Search Infrastructure

Indexed content and metadata are managed by Meilisearch, a privacy-preserving search engine. Each indexed document includes the following metadata:

- Domain name and resolved CID.
- Resolver type and blockchain source.
- Timestamps indicating the most recent indexing.

The Meilisearch ranking algorithm prioritizes search results based on matched query terms, proximity, attribute weighting, and exactness. Synonym mappings (e.g., "DeFi" → "decentralized finance") further improve semantic search relevance. Importantly, Web3Compass avoids any user behavioral tracking, maintaining alignment with Web3 privacy principles.

6. Security and Integrity Measures

Decentralized content indexing involves distinct security considerations. Unlike traditional dynamic content, decentralized sites identified by CIDs are immutable—any content change produces a new CID. Web3Compass leverages this characteristic to implement security validation:

- Sites indexed by Web3Compass carry security scores from Hexens, a security audit provider. This assessment confirms the content's security at the moment of indexing.
- Due to content immutability, any subsequent content alteration results in a new CID requiring a fresh security evaluation, ensuring continuous integrity without repeated scanning.

6.1. Censorship-Resistant Access

To maintain true censorship-resistant access, Web3Compass deploys dweb3.wtf, an independent rendering gateway designed explicitly to bypass centralized gateway limitations. Unlike public gateways prone to filtering or censorship, dweb3.wtf provides users unrestricted access to indexed decentralized content, handling all major Web3 domain systems and offering reliable performance even in the presence of centralized network restrictions.

7. Experimental Evaluation

This section outlines the documented evaluation of Web3Compass in terms of indexing completeness, content availability, dynamic content handling, and censorship resistance. All evaluation points are derived from actual implementation descriptions and outcomes explicitly included in the Web3Compass system documentation.

7.1. Dataset Scope

Web3Compass indexes decentralized websites hosted on IPFS and associated with multiple blockchain-based naming systems. The system has indexed over 700,000 decentralized domains, covering extensions such as .eth, .x, .polygon, .crypto, .wallet, .bnb, and others. These domains point to content stored on IPFS, including both static and SPA-based websites.

7.2. Indexing Completeness and Content Retrieval

Traditional crawlers are unable to index dynamic or client-rendered content from SPAs, often returning only empty or partial HTML. Web3Compass addresses this by:

- Running continuous on-chain monitoring across ENS, UNS, BNB NS, and other registries to detect new or updated domains.
- Querying resolver contracts to extract content hashes (CIDs).
- Resolving and retrieving IPFS content via its own infrastructure of dedicated IPFS nodes.
- Pinning content under specific criteria (e.g., file type, size under 100MB) to ensure long-term availability.

This enables Web3Compass to index a broader range of decentralized websites compared to systems relying solely on static crawling or public gateway access.

7.3. Dynamic Rendering Support

Web3Compass uses Puppeteer-based headless browsing to render SPAs and client-side rendered content. When a resolved CID points to a JavaScript-based site, the system:

- Detects client-side rendering.
- Launches a headless browser.
- Waits for full DOM rendering.
- Extracts the rendered HTML for indexing.

This process allows indexing of content that would otherwise be invisible to crawlers, ensuring that dynamic Web3 sites are searchable.

7.4. Content Availability and Gateway Dependence

To ensure reliable access to decentralized content, Web3Compass operates its own IPFS node infrastructure and does not rely exclusively on public gateways such as ipfs.io. If internal nodes cannot fetch content, the system uses public gateways as fallback options, though internal retrieval is prioritized. Content that meets the relevance and size criteria is pinned to ensure persistent availability within the system’s infrastructure.

7.5. Censorship Resistance via Independent Rendering Gateway

Centralized gateways like ipfs.io, .limo, or .link are documented as vulnerable to takedowns or restrictions. To overcome this, Web3Compass deploys its own censorship-resistant gateway, dweb3.wtf. This service:

- Retrieves and renders decentralized websites across supported domain types (e.g., ENS, UNS, BNB NS).
- Operates independently from third-party filters or legal constraints.
- Ensures that indexed results remain accessible regardless of external gateway availability.

This approach improves reliability and preserves access to Web3 content under conditions where public gateways may be unavailable or censored.

8. Discussion

The implementation and operation of Web3Compass illustrate a practical solution to one of the core limitations of decentralized web infrastructure: the inability of traditional crawlers to index dynamic, client-rendered content stored across distributed networks. The system demonstrates several strengths, all grounded in its documented architecture and operational logic.

8.1. Strengths

One of the primary strengths of Web3Compass is its ability to perform real-time monitoring and resolution of decentralized domains across multiple registries. Unlike conventional search engines that rely on static crawling schedules or central authority databases, Web3Compass listens directly to blockchain smart contracts to detect domain creation and content changes. This enables timely and automated indexing without relying on traditional metadata formats such as sitemaps.

Another significant advantage is its support for dynamic rendering. By detecting SPAs and using a headless browser like Puppeteer to render content, Web3Compass can index websites that return little or no HTML in their raw form. This addresses a core deficiency of both legacy crawlers and early-stage Web3 search attempts, which were unable to surface meaningful content from JavaScript-based frontends.

The system also demonstrates resilience in content retrieval. By operating its own IPFS nodes and pinning relevant content, Web3Compass avoids the reliability problems of public gateways and ensures persistent availability for indexed data. This architecture minimizes dependence on third-party infrastructure, improving system autonomy.

Perhaps most notably, the platform addresses censorship resistance at the access layer. Although Web3 sites may be hosted on decentralized networks, they are often accessed via centralized gateways vulnerable to external regulation or filtering. Web3Compass resolves this by deploying its own gateway (dweb3.wtf), which serves as an independent, unrestricted rendering endpoint. This design ensures that users can reach indexed content even if public gateways are blocked or unavailable.

8.2. Limitations

Despite these strengths, Web3Compass currently supports content hosted only on IPFS. Other decentralized storage systems such as Arweave, Swarm, and Skynet are not yet integrated. As a result, websites that rely on alternative protocols remain outside the scope of the platform's indexing capabilities.

The system also does not maintain historical versions of content. When a content hash is updated, only the latest version is retained and indexed; prior versions are not preserved or exposed to search. This simplifies storage but limits archival functionality.

While the indexing engine (Meilisearch) is optimized for privacy and performance, the current implementation does not incorporate AI or machine learning-based ranking or relevance scoring. This is acknowledged in the documentation as a future area of enhancement, including potential integration of NLP and semantic search capabilities.

8.3. Privacy Considerations

Web3Compass is explicitly designed to avoid behavioral tracking. Unlike traditional search engines that rely on user clicks, cookies, and engagement metrics to inform ranking, Web3Compass bases its results on semantic analysis, on-chain signals (e.g., resolver type, registration date), and document content. This approach maintains user privacy while delivering relevant results.

8.4. Scalability and Usability

The infrastructure behind Web3Compass, including multiple IPFS nodes, blockchain listeners, dynamic renderers, and the censorship-resistant gateway, reflects a deliberate tradeoff between decentralization and performance. While the system is scalable by design, operating its components at scale requires dedicated resources and monitoring. The search interface is already public-facing, and a documented API allows third-party integration, though occasional downtimes are noted in internal documentation.

9. Conclusion

This paper presented the architecture, implementation, and evaluation of Web3Compass — a decentralized web indexing system designed to address the limitations of traditional

crawling methods in the context of Single Page Applications (SPAs) and client-side rendered decentralized websites.

Through continuous monitoring of blockchain-based domain registries, resolver contract querying, and dedicated content retrieval infrastructure, Web3Compass enables automated discovery and resolution of decentralized domains. Its dynamic rendering pipeline — based on headless browser technology — allows for accurate indexing of JavaScript-rendered content that is otherwise inaccessible to legacy crawlers.

Web3Compass improves content availability by operating its own IPFS nodes, implementing pinning strategies for relevant content, and ensuring fallback access when public gateways are unavailable. It further enhances censorship resistance by providing an independent gateway, dweb3.wtf, which bypasses the restrictions imposed by centralized IPFS access points.

The system has already indexed over 700,000 decentralized domains and supports a growing range of domain types, including ENS, UNS, and BNB NS. Its implementation avoids behavioral tracking, emphasizing privacy, semantic relevance, and protocol-level integration instead of user-based metrics.

While the current system is limited to IPFS-hosted content and does not store historical versions of indexed sites, its modular design allows for future extension. Planned directions include integration with other decentralized storage protocols (e.g., Arweave, Swarm) and enhancements to ranking algorithms using AI and NLP techniques.

In addressing core challenges of content discovery, dynamic rendering, reliable access, and censorship resistance, Web3Compass contributes a practical and operationally verified solution for making the decentralized web more usable, searchable, and accessible.

References

[1] *C. D. Manning, P. Raghavan and H. Schütze, "Introduction to Information Retrieval", Cambridge University Press,* ***(2008)***.

[2] *M. Najork and A. Heydon, "High-performance web crawling", in Handbook of Massive Data Sets, Springer,* ***(2002)****, pp. 25–45.*

[3] *Z. Yao, B. Ding, Q. Bai, and Y. Xu, "Minerva: Decentralized collaborative query processing over InterPlanetary File System", IEEE Trans. Big Data, vol. 11, no. 2,* ***(2025)****, pp. 669–683.*