Resilient Retrieval of Decentralized Content: An Architecture-Centered Study of Web3Compass Content Access Methods through IPFS

Gevorg Margarov
National Polytechnic University of Armenia
Yerevan, Armenia
e-mail: gmargarov@gmail.com

Artyom Harutyunyan
Blockstars LLC
Yerevan, Armenia
e-mail: artyomharutyunyans@gmail.com

***Abstract*— As Web3 matures, decentralized naming and storage systems, such as ENS, Unstoppable Domains, and IPFS, offer new paradigms for publishing and accessing web content without relying on centralized infrastructure. However, the process of retrieving content in such an environment remains fragmented, often dependent on vulnerable public gateways or centralized APIs. This paper investigates the resilience of content retrieval in decentralized systems, using Web3Compass as a case study. The system integrates real-time registry monitoring, on-chain name resolution, and direct access to decentralized storage via self-hosted IPFS nodes. By avoiding reliance on third-party resolution services and fallback gateways except when necessary, Web3Compass provides a robust method for discovering and rendering Web3 websites. We detail the system's architecture, including resolver logic, node infrastructure, and content validation policies, and evaluate its robustness against gateway failure, incomplete pinning, and resolution inconsistencies. Our findings indicate that proactive pinning, resolver-specific logic, and local node infrastructure significantly improve access reliability, even under constrained network conditions.**

***Keywords*— Decentralized Content Retrieval, IPFS, Web3, Resilience, Decentralized Web (DWeb), On-chain Resolution, Ethereum Name Service (ENS), Unstoppable Domains (UD), Headless Rendering, Proactive Pinning.**

I. Introduction

The decentralized web ecosystem is built on distributed systems such as the InterPlanetary File System (IPFS) for content storage and blockchain-based naming systems like Ethereum Name Service (ENS) and Unstoppable Domains (UD). These technologies promise user sovereignty, censorship resistance, and verifiability. However, while content hosting and naming have become decentralized, the retrieval layer remains a critical point of failure.

Typical access to decentralized content is mediated through public gateways like ipfs.io, which allow standard browsers to fetch IPFS content via HTTP. These gateways, while convenient, introduce centralized bottlenecks that are vulnerable to regulatory takedowns, downtime, or service limitations. Similarly, resolving names like example.eth to IPFS content hashes require tools that may rely on centralized APIs, extensions, or interfaces. As a result, access to decentralized content is often fragile or inconsistent, limiting the practical usability of Web3 platforms.

To address this, Web3Compass implements a resilient content access architecture that minimizes reliance on third-party services. It combines autonomous name resolution with IPFS content retrieval through self-hosted infrastructure, fallback mechanisms, and content-specific routing logic. The system is capable of resolving names across multiple registries, such as ENS and UD, and serving dynamic IPFS-hosted content directly to end users. This paper presents the retrieval architecture, evaluates its robustness, and analyzes its contributions toward a more reliable decentralized access layer.

II. Literature Review

The decentralized web (DWeb) has emerged as a response to the limitations and vulnerabilities of traditional, centralized internet infrastructure, offering greater user sovereignty, censorship resistance, and verifiability through distributed technologies. Core systems such as the InterPlanetary File System (IPFS) for decentralized storage and blockchain-based naming services like Ethereum Name Service (ENS) and Unstoppable Domains (UD) have enabled users to publish and access web content without intermediaries. However, despite these advances, the retrieval layer of the DWeb remains a significant challenge, as most users still access content via public gateways or centralized APIs, introducing bottlenecks and exposing the ecosystem to regulatory takedowns and service outages.

Academic literature highlights that a truly decentralized web requires robust integration between decentralized search engines, name registries, and distributed file systems. Search engines must index DWeb content, curate which resources to include, and determine ranking algorithms, all without introducing centralized points of control. Name registries provide human-readable mappings to content identifiers (CIDs), while decentralized file systems store and serve the actual content. Effective interoperability between these components is essential for seamless content discovery and retrieval, but current implementations often rely on hybrid

architectures that blend decentralized protocols with fallback mechanisms to maintain usability and performance [1].

Recent research points to several persistent challenges in decentralized content retrieval. Achieving performance and reliability comparable to centralized systems remains difficult, particularly as decentralized networks can suffer from inconsistent availability and latency. The scarcity of structured metadata in decentralized sites further complicates indexing and ranking, while the increasing prevalence of dynamic, JavaScript-heavy applications (such as Single Page Applications) requires advanced rendering and crawling techniques. Privacy and security are also ongoing concerns, as even decentralized systems can inadvertently leak user data if not carefully designed [1].

Innovative solutions like Web3Compass address these issues by implementing a hybrid, protocol-aware retrieval architecture. By autonomously monitoring multiple blockchain-based registries, performing on-chain name resolution, operating self-hosted IPFS clusters, and integrating headless browser technology for a dynamic site rendering, such systems minimize reliance on third-party services and enhance resilience. These approaches demonstrate that robust, privacy-preserving, and censorship-resistant content retrieval is technically feasible, though ongoing research is needed to further improve usability, governance, and interoperability across the decentralized web ecosystem [2].

III. System Architecture and Design

The decentralized web, while conceptually robust, poses unique engineering challenges for content retrieval. The Web3Compass system addresses these challenges through a hybrid retrieval architecture, combining on-chain monitoring, protocol-aware resolution, and resilient access to decentralized storage networks. This section outlines the system's layered design, its content retrieval mechanisms, and the architectural principles that guide its operation.

A. Monitoring Decentralized Domain Registries

Web3Compass monitors multiple domain name systems that operate across different blockchain networks, including the Ethereum Name Service (ENS), Unstoppable Domains (UNS), and the .bnb namespace on BNB Chain. Each registry employs its own naming protocol and storage mechanisms, which require tailored approaches to interface with them.

The system establishes persistent connections with blockchain networks using APIs such as Alchemy and Chainstack, along with The Graph's subgraph interfaces, when available. ENS domains, for instance, are resolved via calls to the resolver contracts deployed on Ethereum, using the standardized contenthash() function to obtain IPFS hashes. For UNS domains, which are hosted on Polygon, the system retrieves resolution data using both direct smart contract interactions and Unstoppable Domains' publicly available resolution APIs. Similarly, the SpaceID-based BNB NS domains are monitored by parsing event logs from BNB Chain nodes or through dedicated indexing services.

Web3Compass subscribes to on-chain events such as domain registrations and resolver updates using event listeners. These events are parsed and matched with known interface definitions to extract the most recent content identifiers. Upon retrieval, the content mapping is cached and synchronized with the core index to maintain high retrieval speed and reduce query latency.

Resolution and Hash Extraction. Once a new or updated domain is detected, the system performs a two-step resolution: first, it determines the active resolver associated with the domain, and then it queries the resolver contract for the associated content hash. The resolution logic is modular to support multiple registry standards. For instance, ENS follows EIP-1577 for content resolution, while UNS and BNB NS use slightly different structures, necessitating protocol-aware resolution modules.

Extracted content hashes are interpreted as IPFS-compatible CIDs, though some domains may point to Arweave transaction hashes or other decentralized locations. The system normalizes these identifiers and verifies format compatibility before proceeding to content retrieval. Domains without valid or accessible hashes are logged but excluded from the content pipeline.

B. IPFS Access Layer

Web3Compass operates its own self-hosted IPFS cluster, comprising multiple full nodes distributed across independent hosting environments. This infrastructure supports direct content retrieval from the IPFS network without reliance on public gateways such as ipfs.io or Cloudflare IPFS. By maintaining its own nodes, the system ensures low-latency access and greater control over availability and persistence.

To enhance fault tolerance, the content retrieval module falls back to third-party IPFS gateways only if local retrieval fails. A retry policy and latency-based routing logic guide this fallback behaviour, ensuring that even in cases of node unavailability, users can still access the desired content. However, the system always prioritizes self-hosted nodes to uphold its censorship-resistant design.

C. Pinning and Content Curation

In the absence of reliable metadata or curation signals in decentralized networks, Web3Compass applies automated filtering and pinning logic to ensure the quality and retrievability of indexed content. Once a CID is resolved and fetched, the content is analysed for MIME type and structure.

The system prioritizes HTML-based websites for inclusion in the searchable index. Non-relevant formats, such as videos, PDFs, ZIP archives, and encrypted payloads, are excluded from indexing. Retrieved content under 100MB that meets the inclusion criteria is automatically pinned to Web3Compass's IPFS nodes to guarantee persistence. This selective pinning process enhances content availability over time and reduces dependency on ephemeral peer availability.

D. Handling Single Page Applications and JavaScript-Heavy Sites

A key innovation in the Web3Compass architecture is its support for Single Page Applications (SPAs) and client-rendered JavaScript sites, which are increasingly common in the Web3 ecosystem. Unlike traditional HTML sites, SPAs require script execution to populate the Document Object Model (DOM) with meaningful content.

To accommodate these use cases, Web3Compass integrates a headless rendering engine based on Puppeteer. When a site is flagged as requiring JavaScript execution, determined through a lightweight pre-scan, the headless

browser module renders the full page in a virtual browser environment. The resulting DOM is then scraped and converted into a static HTML snapshot, which is passed to the indexing engine. This allows Web3Compass to deliver meaningful search results even for dynamic or JavaScript-dependent decentralized sites.

E. Unified Routing and Domain Coverage

The Web3Compass gateway, exposed via dweb3.wtf, implements a unified routing scheme that maps content hashes and domains into accessible URLs. For example:

https://dweb3.wtf/ipfs/<CID>/

https://dweb3.wtf/<domain.extension>/

This structure provides human-readable access while maintaining underlying decentralization. The gateway supports a wide array of domain extensions, including .eth, .crypto, .wallet, .polygon, .nft, .dao, .zil, .x, .tomi, and .bnb, ensuring broad compatibility across decentralized naming standards.

Routing logic dynamically selects the appropriate resolver path and storage backend, depending on the domain type. All rendered content is served from self-hosted infrastructure, and requests are anonymized to avoid leakage of access patterns or user metadata.

IV. Evaluation and Results

Evaluating a decentralized content retrieval system like Web3Compass requires focusing on its functional resilience, operational efficiency, and architectural consistency rather than speculative benchmarks. This section provides a grounded assessment of the system's architecture in action, covering its robustness in domain resolution, reliability in decentralized content access, and effectiveness in maintaining uninterrupted retrieval without third-party reliance.

A. Resolution Robustness Across Registries

The system was designed to interoperate with multiple decentralized domain name systems. ENS and UNS registries, based on Ethereum and Polygon, respectively, expose standardized resolution interfaces, primarily via the contenthash() function or equivalent APIs. Web3Compass successfully integrates these resolution standards using modular logic and can autonomously detect and cache content hash changes.

For .bnb domains (now referred to as BNB NS or Space ID), resolution is achieved through a direct smart contract interaction on BNB Chain. While the structure and format differ slightly from ENS or UNS, the system's registry-agnostic architecture accommodates this via separate decoding paths. The continuous monitoring of blockchain events (e.g., NewURI, SetContentHash) ensures that Web3Compass maintains an updated mapping of domain-to-content relationships without relying on external resolvers or user input.

In internal testing, the system consistently resolved new or updated domains within seconds of block confirmation across all three registries. Even when a registry had temporary RPC availability issues (e.g., network delays on BNB Chain), the failover architecture, using multiple RPC endpoints per chain, ensured that resolution continuity was not compromised.

B. Retrieval Reliability from IPFS

Content retrieval via IPFS forms the backbone of Web3Compass's delivery model. The use of self-hosted IPFS nodes allowed the system to bypass rate limits, downtime, and compliance constraints imposed by public gateways. By pinning relevant HTML content and maintaining local access, the system achieved persistent availability without relying on third-party infrastructure.

In scenarios where the requested content was not locally pinned or unreachable via Web3Compass's nodes, the system dynamically fell back to public IPFS gateways. This fallback mode preserved access continuity without introducing tracking or third-party API dependencies, as content was still retrieved using decentralized addressing.

The fallback retrieval approach was selectively activated and monitored to prevent over-reliance. This layered design contributed to high resilience and ensured that even low-frequency or older content could be served without manual re-pinning. Furthermore, caching mechanisms allowed frequently accessed sites to be retrieved faster during repeat queries, improving perceived responsiveness.

C. SPA Support and Headless Rendering Efficiency

Many modern decentralized websites are built using frameworks like React or Vue, which require JavaScript execution to render content. Web3Compass addressed this by integrating Puppeteer into its retrieval pipeline. When a site was flagged as JavaScript-dependent, the system launched a headless browser instance to render the full DOM.

This capability enabled the inclusion of Single Page Applications in the search index, something not possible with static IPFS snapshots alone. In practice, this added a few seconds of latency during initial content ingestion, but the rendered output was cached and reused for subsequent accesses. This ensured that client-rendered sites became fully indexable and searchable through the system.

The rendering logic also accounted for timeouts and malformed scripts. If rendering failed (e.g., due to infinite loading or script errors), the system gracefully skipped that domain's dynamic content while retaining the basic hash-to-domain association.

D. Privacy, Filtering, and Ethical Safeguards

Web3Compass's commitment to privacy is structurally enforced. No user data is collected, stored, or tracked during query execution or content access. Unlike traditional search engines that use cookies, behavioral profiling, or IP logging, Web3Compass does not implement any analytics or identifiers. Search queries are handled entirely server-side and anonymized through internal proxying.

On the content side, automatic filtering ensures that only web-relevant resources are indexed. Binary files, multimedia formats, and encrypted blobs are excluded from rendering. This both improves relevance and mitigates the risk of indexing malicious or non-human-readable data. HTML files with interpretable structure are favored, and the presence of metadata (e.g., <title> tags, meta descriptions) is used to enrich indexing when available.

For potentially harmful content (e.g., phishing sites, malware payloads), Web3Compass implements hash-based blacklisting. Domains linked to flagged hashes are labeled and optionally suppressed during search output, though the

content remains technically retrievable by CID. This approach aligns with Web3's principles of neutrality and transparency while enabling basic safeguards against abuse.

E. System Limitations and Trade-Offs

While the system achieves high levels of resilience and coverage, some limitations persist due to the underlying constraints of Web3 architecture. For example:

- Metadata Scarcity: Most decentralized sites lack structured metadata. This reduces indexing quality compared to Web2 pages with rich SEO annotations.
- Network Latency Variance: Although self-hosted IPFS nodes reduce dependency, their connectivity to the global IPFS network can vary. Retrieval times are generally acceptable, but not always uniform.
- Registry Event Complexity: Certain blockchain events (e.g., ENS subdomain updates or multi-record mutations) introduce complexity in resolution. Web3Compass resolves this by focusing on top-level domains and caching disambiguated records.

These limitations are addressed through continuous monitoring, layered fallback logic, and selective indexation, but they remain inherent trade-offs when building in decentralized environments.

V. Conclusion

This paper examined the engineering strategies employed by the Web3Compass platform to ensure resilient retrieval of decentralized web content. Although the decentralized web offers a censorship-resistant and open alternative to traditional internet infrastructure, its content remains largely inaccessible to users without specialized tools. Web3Compass addresses this by combining real-time domain resolution, IPFS-based retrieval, and dynamic rendering under a unified system architecture designed for robustness and autonomy.

Through self-hosted infrastructure, the system avoids reliance on public IPFS gateways and centralized redirectors. The dweb3.wtf gateway plays a central role in this effort, providing an independent rendering layer that supports SPA-based websites and a wide array of decentralized domain extensions. By pinning relevant content and integrating headless rendering, it bridges gaps left by traditional gateway services, many of which are prone to legal takedowns, regional filtering, or technical outages.

This retrieval architecture is tightly coupled with a protocol-aware domain monitoring system that indexes content from registries such as ENS, Unstoppable Domains, and BNB Name Service (CNS). Each domain is resolved to its content identifier, fetched using dedicated infrastructure, parsed, and rendered for end-user access. In doing so, Web3Compass ensures that decentralized websites are not only indexed but also viewable, regardless of whether users have browser extensions or native Web3 support.

Importantly, these features are implemented without behavioral tracking, centralized ranking algorithms, or reliance on third-party APIs. The platform's architectural choices reflect a commitment to Web3 values: decentralization, privacy, and autonomy. In this context, resilient retrieval is not only a technical necessity but a foundational requirement for the usability and growth of decentralized information systems.

The core retrieval architecture, grounded in on-chain resolution, IPFS access, and independent rendering, provides a solid baseline for sustainable, censorship-resistant Web3 access. The work presented here demonstrates that robust content retrieval in decentralized environments is both technically feasible and practically deployable, provided it adheres to the engineering constraints and values described.

References

[1] N.V. Keizer, O. Ascigil, M. Król, D. Kutscher, G. Pavlou, "A Survey on Content Retrieval on the Decentralised Web", *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–38, 2024.

[2] S. Wang, et al. "QueenBee: Decentralized Search on Decentralized Web", *Proceedings of CIDR*, pp. 1–9, 2019.