applied sciences

MDPI

Review

A Survey on Traditional DNS and Blockchain-Based DNS: Comparative Analysis, Challenges, and Future Directions

Juseong Jeon and Sejin Park *

Department of Computer Engineering, Keimyung University, Daegu 42601, Republic of Korea; mo66jyp@gmail.com
* Correspondence: baksejin@kmu.ac.kr

Abstract

Although DNS has been continuously extended to improve usability and security, its centralized, server-based architecture leaves fundamental limitations unresolved, including single points of failure (SPOF), susceptibility to censorship, and exposure to DDoS. This study examines blockchain-based DNS (BDNS) as an alternative proposed to mitigate these structural issues. We first synthesize prior research and systems on BDNS, and then conduct a comparative analysis using practical deployability as the primary criterion. Specifically, we selected seven representative BDNS projects, including Namecoin, Handshake, and Ethereum Name Service (ENS), and evaluated them under a common set of criteria: (i) the record model, finality, and TTL semantics; (ii) friction along real resolution paths involving resolvers, browsers, and gateways; and (iii) interoperability with the legacy DNS, including DNSSEC and DNS over TLS(DoT)/DNS over HTTPS(DoH), together with migration scenarios. The analysis indicates that many systems rely on gateways and client-side extensions, limiting native resolution paths. It further finds that finality latency, dependence on off-chain indexing and availability, and the interplay of key management and tokenomics design increase operational complexity and raise barriers to adoption. Building on these findings, the paper derives operational requirements and proposes a coexistence-first, five-layer migration framework that incrementally integrates BDNS while retaining the legacy DNS. This provides an incremental path toward a more resilient, inclusive, and secure global naming infrastructure.

Keywords: blockchain; blockchain-based DNS; censorship resistance; consensus mechanisms; decentralization; Domain Name System; DNS security; smart contracts; Web3

Check for updates

Academic Editor: Zhengjun Liu

Received: 27 November 2025
Revised: 2 January 2026
Accepted: 4 January 2026
Published: 7 January 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Internet connectivity is an indispensable element of modern computing, and the DNS, the human-readable interface to networked services, is also essential. Since its introduction in 1983, Domain Name System (DNS) has steadily evolved in security and usability: Domain Name System Security Extensions (DNSSEC) protects integrity and origin authentication [1], while DNS over TLS (DoT) and DNS over HTTPS (DoH) encrypt resolver connections [2,3]. Usability has also improved through DDNS, internationalized domain names, DNS-based Authentication of Named Entities (DANE), and DNS64/NAT64 [4–6]. However, structural centralization continues to expose the DNS to SPOF and policy control, leaving it susceptible to censorship and to certain classes of attacks that protocol extensions cannot fully mitigate.

These limitations are particularly significant in emerging computing paradigms. The rise of Web 3.0 has created a paradox in which decentralized applications (DApps) still depend on centralized DNS, inheriting the same vulnerabilities related to censorship and SPOF. This mismatch underscores the urgent need for naming infrastructure that aligns with the decentralized nature of Web 3.0.

Furthermore, traditional DNS security mechanisms pose implementation challenges for the rapidly expanding Internet of Things (IoT). DNS attacks such as redirection to malicious services or fingerprinting and disclosure of device types and vulnerabilities can be devastating for IoT devices that often do not support robust security extensions like DoT or DoH [7].

Meanwhile, existing surveys on BDNS have provided an important foundation for understanding the BDNS landscape by broadly synthesizing diverse systems and comprehensively discussing key issues—governance, security, performance, and interoperability (e.g., Giamouridis et al. [8]).

However, from a practical standpoint addressing whether immediate operation is feasible and which paths enable coexistence and migration, studies that normalize the decisive factors under a common set of criteria for like-for-like comparison are relatively scarce. These factors include (i) operational consistency across the record model, finality, and Time-to-Live (TTL); (ii) friction encountered along real resolution paths involving resolvers, browsers, and gateways; (iii) interoperability with the legacy DNS (including DNSSEC and DoH/DoT) and staged migration strategies; and (iv) operational realities such as fees, operational overhead, and incident response. Moreover, approaches that translate such comparative results into immediately actionable checklists and deployment scenarios from stakeholder perspectives (operators, service providers, and policymakers) are even rarer.

Accordingly, this survey revisits BDNS proposals from the standpoint of practicality and deployability to close this gap. Rather than merely enumerating technical features, we define a unified comparative framework that covers governance/decentralization, security, expected performance/latency, interoperability with the current DNS, and operator/user experience, and we apply it consistently to representative systems (e.g., Namecoin, Handshake, Blockstack, Ethereum Name Service (ENS), Unstoppable Domains, and EmerDNS). The contributions of this paper are as follows.

- We articulate a practicality-oriented comparison framework and a harmonized set of evaluation criteria for BDNS vs. traditional DNS.
- We provide a taxonomy of BDNS designs (e.g., Unspent Transaction Output (UTXO) vs. smart-contract models) mapped to DNS requirements (finality, availability, governance, interoperability).
- We present consistent comparative tables and a scenario-based analysis (integration/migration paths, operator considerations, client support) that resolve prior surveys' inconsistency in criteria.
- We synthesize actionable insights for engineers and policymakers (what is feasible today, what remains research, and where standards are needed).

Section 2 reviews DNS fundamentals and historical evolution and provides a concise blockchain primer for naming. Section 3 surveys related work and clarifies open gaps. Section 4 analyzes the traditional DNS architecture and threat surface. Section 5 deconstructs BDNS building blocks and their operation (e.g., consensus, data/record models, smart contracts, and resolution flow). Section 6 offers a side-by-side comparative analysis of traditional DNS vs. BDNS along practicality and deployability axes. Section 7 applies a unified template to representative BDNS systems. Section 8 outlines future directions, and Section 9 concludes.

2. Background

In this section, we outline the technical evolution of the Domain Name System (DNS) and summarize the core principles of blockchain technology relevant to decentralized naming. In Section 2.1, we trace DNS's historical development from early host tables to the modern centralized hierarchical model, and identify key architectural limitations related to security, scalability, and domain-name scarcity. In Section 2.2, we review blockchain technology as a potential alternative, focusing specifically on how different consensus mechanisms impact key DNS performance metrics such as query latency and transaction finality. This foundational analysis supplies the necessary context for the subsequent sections that compare traditional DNS and blockchain-based DNS architectures.

2.1. Historical Development of DNS

Even before DNS was developed and deployed, there was a consistent demand for more convenient methods of accessing services, as manually entering IP addresses was cumbersome. During the early days of ARPANET, the predecessor of the internet, the number of hosts was extremely limited to only around 23, which made it feasible to use the IP-based addressing system without modification.

However, as computer adoption began to increase, it became difficult for users to memorize network addresses directly. In response, during the 1970s, users introduced a file called `HOSTS.TXT` for convenience. This file was centrally managed by the Stanford Research Institute (SRI) and played a role similar to modern DNS by mapping host names to IP addresses.

The `HOSTS.TXT` file was a plain text file similar to a modern DNS zone file. It stored mappings between IP addresses and their corresponding hostnames. In contemporary DNS, a zone denotes a portion of the DNS namespace managed by a specific organization or operator and functions as an independent administrative unit for managing a domain and its subdomains. Zones are typically categorized into forward and reverse zones: a forward zone resolves domain names to IP addresses, whereas a reverse zone resolves IP addresses to domain names.

The forward zone is used for typical domain name resolution, whereas the reverse zone is leveraged for security mechanisms maintained via reverse DNS (rDNS) lookups by mail servers, as well as for visitor tracking and logging. Information about these zones is managed through zone files stored on DNS servers. A zone file is a human-readable text file that specifies the DNS records contained within the zone. This file serves as the authoritative data source that provides domain-related information to the DNS server.

Compared with modern DNS zones, SRI's `HOSTS.TXT` was far simpler. Each networked host maintained a locally cached copy; when a user entered a hostname, the system consulted the file to resolve it to an IP address and establish the connection. The file was distributed via the File Transfer Protocol (FTP), and hosts periodically fetched the latest `HOSTS.TXT` from SRI and installed it locally [9].

However, as the number of internet hosts and users continued to grow, maintaining synchronization across all `HOSTS.TXT` files became increasingly difficult, leading to inconsistencies between the actual network state and the file contents, as well as conflicts when identical host names were used. In addition, the file itself grew in size, which, combined with increased traffic, placed a burden on network resources. As a result, in 1983, `HOSTS.TXT` was eventually replaced by the DNS, designed by Paul Mockapetris [10]. The newly introduced DNS was implemented based on a distributed database architecture, replacing the previous centralized management approach. Between 1983 and 1984, RFC 882 [11], which detailed the conceptual framework and functional components of DNS, and RFC 883 [12], which described its implementation specifications, were published.

These RFCs laid the foundation for fundamental concepts of DNS that are still in use today, including the name space, resource records, the operation of name servers, and resolvers. In 1984, the first TLDs such as .com, .org, and .net were introduced, along with country-code domains like .kr and .jp. In 1987, the foundational concepts and specifications of DNS were formally defined through RFC 1034 [13] and RFC 1035 [14], which replaced the earlier RFC 882 and RFC 883. These documents provided comprehensive explanations of DNS operations, including conceptual descriptions, definitions of resource records, message formats, and the structure of master files.

By the early 1990s, the Internet began to see widespread public adoption, accompanied by an explosive increase in the number of hosts. In particular, the growing popularity of `.com` led to intense competition among companies to secure domain names under this TLD, driving significant growth in the domain-registration industry. To enable more systematic management, the Internet Corporation for Assigned Names and Numbers (ICANN), a nonprofit organization, was established to oversee policy development and overall coordination. As a result, the responsibility for DNS management shifted from the Internet Assigned Numbers Authority (IANA) to ICANN [15].

From a technical standpoint, the 1990s also saw numerous improvements. Full zone transfer (AXFR) replicated entire zone files between primary and secondary authoritative servers; as zones grew, such full transfers increasingly caused bandwidth and timing bottlenecks. To address this issue, incremental zone transfer (IXFR) was introduced, transmitting only the differences relative to the previous version of the zone [16]. In addition, Dynamic DNS (DDNS) improved usability by enabling effective use of DNS in environments with dynamic IP addresses. Furthermore, the traditional polling model, in which a DNS secondary server periodically queried the primary for changes, was enhanced by the DNS NOTIFY mechanism defined in RFC 1996 [17], reducing unnecessary traffic and delays.

As the internet experienced unprecedented exponential growth in the 2000s, the vulnerabilities of DNS became increasingly apparent, and these factors together began to raise serious security concerns. Various attacks such as DNS cache poisoning [18], Distributed Denial of Service (DDoS) [19], and DNS spoofing [20] began to undermine the availability of DNS [21], prompting the introduction of DNSSEC as a countermeasure. In addition, technologies such as DoH, DoT, and DANE were introduced sequentially, further enhancing the stability of DNS. The introduction of internationalized domain names enabled the use of non-ASCII characters in domain names, allowing non-English-speaking countries to register domain names in their native languages [22]. Furthermore, technologies such as DNS64 and NAT64 are being adopted to support IPv6, the next-generation addressing system following IPv4.

In addition to security concerns, domain name scarcity is an inevitable consequence of the Internet's growth and its structural constraints. Although the number of TLDs increased from roughly 650 in the 1980s to 1443 by 2022 [23], this expansion has not alleviated saturation within the major generic TLDs (gTLDs). As noted in [24], popular gTLDs such as `.com` are effectively saturated with the most valuable domain names, namely two- and three-letter strings and memorable, high-quality dictionary words.

This scarcity stems from the fact that the initial structure of the DNS protocol was developed under the assumption of limited resource allocation [10]. In particular, the hierarchical structure of domain names and the caching mechanism inherently contain scalability constraints.

At the naming layer, to mitigate name scarcity rather than protocol-level constraints, adoption of generic-leaning ccTLDs such as `.io` and `.co` expanded through the 2010s (with `.co` opened to global registration in July 2010), and `.io` saw renewed growth in 2022–2023. In parallel, the 2012 ICANN round introduced a large cohort of new gTLDs, yet public

recognition for newly introduced TLDs has continued to diffuse only gradually. The history of DNS summarized so far can be seen at a glance in Figure 1.

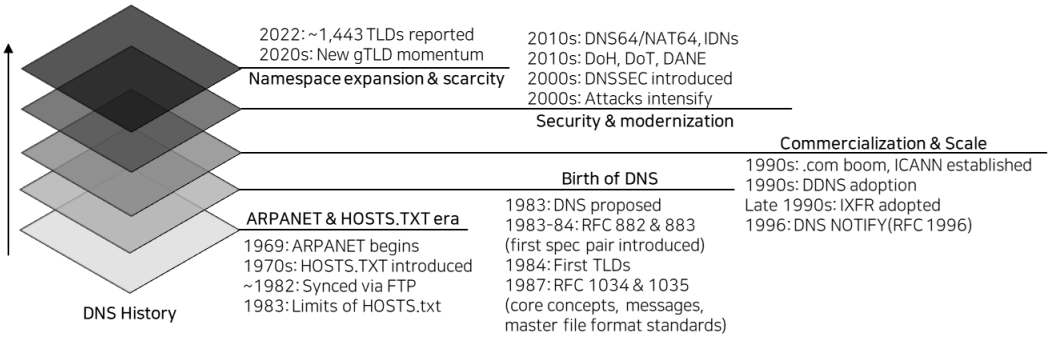

Figure 1. Timeline of DNS development.

2.2. Overview of Blockchain Technology

Blockchain is a distributed ledger technology (DLT) that validates and records transactions through consensus among network participants without a trusted intermediary. In the context of BDNS, blockchain functions as an immutable, distributed storage that replaces conventional centralized root DNS servers and TLD registry databases. This architecture removes SPOF and ensures data integrity, but system performance, particularly DNS query response latency and the finality of record updates, depends on the underlying blockchain's consensus algorithm and the smart contract state-management design. In this section, rather than reviewing general blockchain theory, we concentrate on the consensus mechanisms and state-management logic that directly affect BDNS performance metrics of latency and finality.

2.2.1. Consensus Algorithms: Impact on DNS Latency and Finality

The consensus algorithm is the core protocol that ensures distributed nodes reach agreement on the state of DNS records, such as ownership registrations and IP updates. When designing BDNS, the choice of consensus algorithm determines the trade-off between transaction throughput and finality, which directly affects user experience.

Practical Byzantine Fault Tolerance (PBFT): This mechanism reaches consensus through message exchanges rather than computational competition, providing immediate finality. This means that domain ownership transfers and record updates are finalized as soon as a block is produced, which reduces the risk of DNS cache poisoning and helps ensure read consistency. Experimental results show that PBFT achieves a high throughput of approximately 61,000 operations per second (ops/s) in the $f = 1$ setting, making it suitable for handling high DNS query loads. However, because its communication complexity limits the number of participating nodes, PBFT is primarily used in permissioned BDNS models such as Blockzone.

Proof of Work (PoW): Exemplified by Bitcoin, it provides a high degree of decentralization and censorship resistance, forming the basis for early BDNS systems such as Namecoin and Handshake. However, because PoW exhibits probabilistic finality, domain registrations suffer long latency as they require waiting for multiple block confirmations before becoming effectively final. In addition, Bitcoin's low throughput of approximately 3.5–7 Transactions Per Second (TPS) creates a bottleneck for handling large-scale DNS transactions, imposing structural limits on DNS services that demand real-time responsiveness.

Proof of Stake (PoS): By allocating validation rights based on staked assets, this mechanism offers significantly higher energy efficiency and fast finality compared to PoW. These characteristics reduce operating costs for always-on DNS services and minimize the latency associated with record updates. PoS chains such as post-merge Ethereum and Polygon, which support systems like ENS and Unstoppable Domains, provide scalability on the order of thousands of TPS, establishing themselves as the foundational infrastructure for mainstream BDNS adoption.

Proof of Authority (PoA): Relying on a limited set of trusted validators for block generation, this mechanism ensures effectively deterministic finality and high throughput (often exceeding 2000 TPS). These attributes make PoA highly suitable for environments demanding enterprise-grade performance, such as the VeChain Name Service [25]. However, since validator centralization compromises censorship resistance, PoA is optimally deployed in consortium settings where performance and stability take precedence over the permissionless decentralization typical of public blockchains.

In summary, the choice of consensus algorithm for BDNS is a strategic trade-off between censorship resistance (PoW) and performance and immediacy (PBFT/PoA/PoS), and the characteristics of each algorithm are summarized in Table 1.

Table 1. Comparative Analysis by Consensus Algorithm.

Consensus Algorithm	Transaction Finality	Degree of Decentralization	Representative Blockchain	Transaction Per Second (TPS)	BDNS Examples
PBFT	Immediate	Low (permissioned)	Hyperledger Fabric, Tendermint	10^3–10^4 TPS (network-size-, batch-size-, crypto-option- and fault-dependent)	Blockzone [26]
PoW	Probabilistic finality	Very high	Bitcoin, Ethereum before the PoS transition	~3.5 TPS	Namecoin [27], Handshake [28]
PoS	Fast finality	High	Current Ethereum	16–1300 TPS	ENS [29], Unstoppable Domains [30], EmerDNS [31]
PoA	Immadiate or fast finality	Low (permissioned)	VeChain [25], Rinkeby/Kovan (Testnet)	46–2000 TPS	VeChain Name Service (VNS) [25]

2.2.2. Smart Contract

Smart contracts are software modules that encode programmable rules on a blockchain (or compatible DLT) and are automatically executed, verified, and enforced. They take transactions submitted by network participants as input, evaluate the contract logic deterministically, and, under network-wide consensus, commit the results to the shared state. Because contract terms are enforced automatically without a central operator or trusted intermediary, they achieve both data integrity and transparency of execution.

Representative Ethereum Virtual Machine (EVM)-based platforms support the deployment, invocation, and interaction of contracts written in languages such as Solidity or Vyper via a bytecode interpreter and a standardized Application Binary Interface (ABI). Smart contract systems incorporate the following core mechanisms to ensure secure and predictable execution in distributed environments:

- **State Machine & Transaction Execution Model**: A blockchain's global state is represented by contract storage (e.g., key–value maps) and account balances. Each transaction invokes a contract function, triggering a state transition, and all nodes compute the same result for the same input. This determinism, when combined with consensus, ensures consistency across the entire network.

- **Gas/Resource Accounting**: To prevent infinite loops and excessive computation and to internalize resource usage via price signals, platforms such as the EVM charge gas for computation, storage, logging, and related operations. Callers specify a maximum gas limit; if execution exceeds this limit, the transaction halts, the state is reverted, and the fee is still consumed.
- **Interface Standards and Composability**: Standardized interfaces, such as tokens (Ethereum Request for Comments (ERC)-20/721/1155), delegated authorization via typed-data signatures (Ethereum Improvement Proposal (EIP)-712), and access control (Ownable/role-based), enable safe modularity and composition across heterogeneous contracts. Inter-protocol composition is achieved through asynchronous notifications via event logs and through cross-contract invocations or message calls.
- **Security Model and Verification**: Although execution is public and verifiable, defects in contract code can directly impact on-chain assets and state. To address representative vulnerabilities, including reentrancy, integer overflows/underflows, missing access controls, and time-of-check-to-time-of-use (TOCTOU), practitioners employ formal verification, static analysis, fuzz testing, and governance for controlled upgrades.
- **Oracles and Off-Chain Integration**: Facts external to the blockchain, such as prices or identities, cannot be accessed directly on-chain due to determinism and execution isolation requirements imposed by consensus. Accordingly, observations made off-chain are conveyed into the chain via an intermediary layer (an oracle) that supplies them in a verifiable form. Because oracles entail trust assumptions, it is common to design economic incentives and accountability using committees, multisignatures, and slashing to mitigate risk.

Through these properties, smart contracts provide intermediary-free automation and auditability, and they move diverse domain logic, such as asset issuance, rights management, access control, and name resolution, on-chain.

Conversely, there are operational and performance trade-offs, including security exposure and upgrade-governance complexity as well as gas costs and throughput constraints. Adopting appropriate standards, performing rigorous security verification, optimizing resource usage, and explicitly specifying the oracle trust model help mitigate these limitations, enabling DApps and infrastructure-level functionality to be implemented reliably.

3. Related Works

This section reviews several papers that survey BDNS. First, Section 3.1 describes the procedure used to select the papers discussed in this section. We then provide a brief overview of the analyses conducted on the selected papers in Section 3.2 Subsequently, the analysis objectives of each paper, along with the merits and demerits identified in their evaluations, are examined and summarized in Table 2. Based on this summary, the contribution of this paper to the survey of BDNS is then discussed.

3.1. Search and Selection Procedure for Research Papers

This section aims to provide a structured synthesis of the representative studies in the blockchain-based naming/BDNS domain. Rather than pursuing a comprehensive systematic literature review (SLR/PRISMA) that exhaustively covers all publications, we adopt a lightweight scoping approach centered on design, operations, and security issues. Accordingly, the set of papers included in Section 3.2 was derived based on the research questions and selection criteria outlined below.

3.1.1. Research Questions (RQs)

- **RQ1 (Architecture/Design)**: What are the system architecture and the designs for registration, storage, and resolution?
- **RQ2 (Security & Abuse Measurement)**: What security threats and patterns of abuse (e.g., squatting, phishing, malware) are observed, and what are the empirical findings?
- **RQ3 (Interoperability, Governance & Adoption)**: How do coexistence/interoperability with DNS, name collisions, and governance/operational factors influence adoption?

3.1.2. Search Databases and Time Frame

We used Google Scholar as the primary index (search date: 12 December 2025), taking into account its comprehensive coverage of diverse sources, including ACM, IEEE, USENIX, Springer, Elsevier, and arXiv. To mitigate the risk of omissions, we additionally conducted backward reference tracing from key seed papers and citation-based forward snowballing.

Search Queries (Query Strings)

For the searches, we used keywords such as "blockchain DNS," "decentralized naming," "blockchain-based naming system," "Namecoin," "ENS, Ethereum Name Service," and "Handshake" as the base, while combining auxiliary keywords such as "survey/review/overview," "measurement/analysis," "security," "typosquatting/phishing/malware," and "(TLD) collision/interoperability" to construct multiple queries. As detailed query examples, we conducted the search on Google Scholar using a total of three queries, including the following: `(‘‘blockchain DNS’’ survey) OR (‘‘blockchain DNS’’ review) OR (‘‘blockchain-based DNS’’ survey) OR (‘‘decentralized DNS’’ review)`.

3.1.3. Inclusion/Exclusion Criteria

Inclusion criteria were defined as: (i) studies that directly address the architecture of BDNS/blockchain-based naming or the name-resolution workflow; (ii) empirical analyses of Real-world systems/ecosystems (e.g., measurement, abuse analysis, security analysis); and (iii) studies discussing interoperability with legacy DNS, name collisions, and governance/operational issues. Exclusion criteria were set as: (i) duplicate records; (ii) general blockchain studies not directly related to naming/resolution mechanisms; (iii) non-scholarly sources lacking sufficient technical grounding; and (iv) publications that do not align with this paper's comparative perspective (RQ1–RQ3).

3.1.4. Screening Procedure and Record Counts.

The initial search returned a total of $N_0 = 442$ records. Using a reference-management tool, we merged duplicate entries and removed 188, leaving $N_1 = 332$. To complement potential omissions from keyword-based queries, we conducted a title-targeted search and obtained an additional $N_{\text{add}} = 1$ record (e.g., [32]). We then excluded $N_{\text{ex1}} = 246$ records during the first-stage screening based on titles and abstracts, and a further $N_{\text{ex2}} = 76$ during the second-stage screening based on full-text assessment, resulting in a final set of $N_{\text{final}} = 10$ records selected for discussion in this chapter. As summarized in Figure 2, these steps form the overall search-and-screening flow and record counts used in this study.

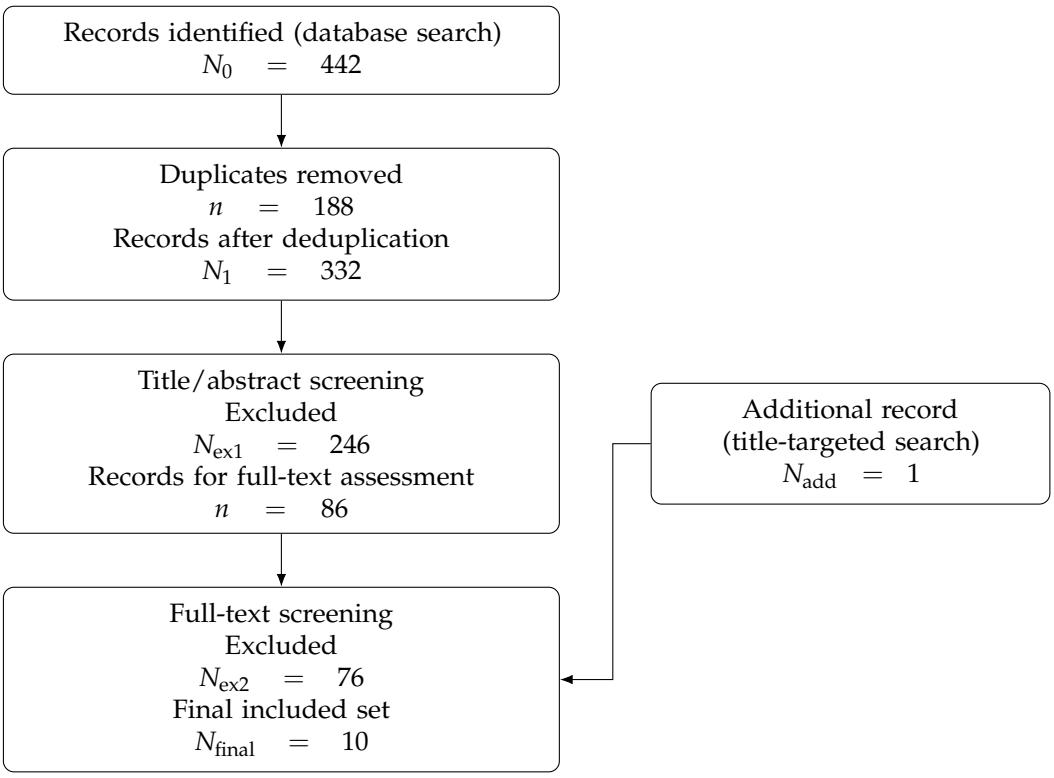

Figure 2. Paper search and screening flow (PRISMA-style).

3.2. Analysis of Prior Research on Blockchain-Based Naming Systems

The study by Kalodner et al. [33] provides an in-depth analysis of the name system in Namecoin. It was found that the Namecoin domain ecosystem is largely underutilized, with a high prevalence of domain squatting and a significant number of domains that see little or no active use. Domain squatting refers to the preemptive registration of Internet domain names that incorporate another party's trademark, service name, or other widely recognized identifiers. It constitutes a form of bad-faith domain registration intended either to obstruct the rightful owner's use of the domain or to resell the domain at an inflated price for profit.

In this regard, among approximately 120,000 registered .bit domains, only 28 hosted substantive content. The authors observed that there was virtually no domain trading or transfers in the secondary market, indicating that market functionality was nearly nonexistent. They argue that the current design of decentralized name systems fails to incorporate effective incentive mechanisms, and propose a utility-based evaluation model to suppress domain squatting and encourage meaningful use of domain names.

The study by Ali et al. [34] describes the architecture and design of Blockstack. The authors operated a decentralized Public Key Infrastructure(PKI) system on Namecoin, handling 33,000 names and over 200,000 transactions, and encountered serious issues related to reliability, throughput, and security. For example, there was a period during which a single miner controlled more than 51% of the hash power for several months. To address these issues, Blockstack transitioned from Namecoin to a Bitcoin-based platform and introduced a new abstraction layer called virtualchain, which separates name control logic from data storage. Large-scale data is stored off-chain, which improves overall scalability and stability, and increases the potential for integration with DNS.

The study by Wang et al. [26] proposes a structure called BlockZone, in which DNS servers are implemented as blockchain nodes without modifying the existing DNS protocol. BlockZone operates in a permissioned network, where DNS name servers store the entire zone data on the blockchain and utilize the PBFT consensus algorithm. This approach

preserves the traditional DNS query mechanism while implementing a decentralized DNS architecture that eliminates censorship and SPOF. It is also designed to withstand DNS query loads through performance optimization.

Here, Byzantine fault tolerance (BFT) refers to the property of a distributed system that guarantees correct operation by reaching consensus even when some participating nodes act maliciously or exhibit arbitrary, unpredictable failures. The term originates from the "Byzantine Generals Problem," which considers how allied generals can reach unanimous agreement on an attack plan in the presence of potential traitors, and it plays a central role in preserving data integrity in decentralized networks where some participants cannot be trusted.

Meanwhile, the study by Patsakis et al. [35] explores the security threats and societal challenges that may arise as decentralized DNSs depart from the traditional centralized DNS model, while features such as censorship resistance and immutability can be considered advantages, they may also be exploited for malicious activities such as malware distribution, phishing, and hosting illegal content. For example, Handshake reserved popular Alexa domains in advance to prevent domain squatting; however, issues such as TLD collisions and dispute resolution still remain. Moreover, in decentralized systems, traditional legal actions or dispute arbitration are either infeasible or extremely difficult, while BDNS enables forensic tracking to some extent, its anonymity and structural complexity make it more challenging to trace attackers.

The study by Casino et al. [32] analyzes malicious activities associated with the .bit and .emc domains operated by Namecoin and EmerDNS. Through taint analysis and blacklist-based domain tracking, the authors demonstrate that a significant number of these domains were registered for purposes such as malware distribution, phishing, and hosting illegal content. The authors argue that the high proportion of malicious domains is directly linked to the censorship resistance inherent in decentralized systems, and they conclude that legitimate use cases are relatively rare.

The study by Randall et al. [36] analyzes how various BDNSs, such as Namecoin, EmerDNS, ENS, Handshake, and Unstoppable Domains, are being leveraged by malware. The authors point out that architectures which appear to be fully decentralized still contain semi-centralized components, such as plugins, proxies, and Simple PAyment Verification(SPV) nodes, making not only attacks possible, but also enabling defensive measures. In practice, some malware was found to operate domains through these intermediate infrastructures, allowing security researchers to trace or block them. In conclusion, no current system can be considered a truly "unstoppable" domain system.

The study by Xia et al. [37] tracked the ENS system over a period of four years to analyze the usage and security issues of .eth domains. The results revealed the presence of squatted domains, scam address associations, and record-keeping vulnerabilities, along with numerous instances of actual domain hijacking. Approximately 3.7% of ENS names were re-registered by others after expiration, effectively resulting in domain hijacking, which was identified as a design vulnerability. The study highlights that while ENS successfully achieves decentralization, improvements are needed in terms of security and reliability.

The study by Giamouridis et al. [8] provides a comprehensive comparison of various systems, including Namecoin, Blockstack, EmerDNS, Handshake, and ENS, addressing their design goals, architecture, usability, security, and interoperability. The study identifies several factors that hinder the adoption of BDNS by general users, including scalability limitations, low interoperability with existing DNS infrastructure, and the absence of legal and governance frameworks. For such systems to become commercially viable, improvements in user experience, performance, and regulatory compliance are essential.

The study by Ito et al. [38] conducts an analysis of TLD conflicts across systems such as Namecoin, EmerDNS, ENS, Handshake, Unstoppable Domains, and DecentraWeb. For instance, the `.wallet` TLD has been registered and operated by both Handshake and Unstoppable Domains, leading to actual conflict cases and legal disputes. The paper provides a detailed comparison of the namespace metadata, domain expiration handling, and DNS record support across different systems, highlighting the risks of duplicate registrations and potential user confusion. Through this analysis, the study objectively confirms the issue of TLD conflicts inherent in BDNSs.

The study by Muzammil et al. [39] presents a detailed analysis of typosquatting across ENS, Unstoppable Domains, and ADA Handles (a naming system on the Cardano blockchain; its native cryptocurrency, ADA, is named after Ada Lovelace). Typosquatting, also known as URL hijacking, is an attack technique that exploits typographical errors users often make when entering web addresses. Attackers pre-register look-alike domains to divert mistyped traffic to malicious landing pages, typically to distribute malware or conduct phishing attacks.

By collecting data on approximately 5 million domains, the authors found that about 37% of popular names had typosquatted variants registered. Legitimate users rarely pre-register typo variants themselves; instead, a small number of squatter addresses hold most of them in bulk. Some of these typo domains were actively used to impersonate cryptocurrency addresses and steal user assets. These findings indicate that BDNS are particularly vulnerable to domain squatting.

Based on the analysis of related works, this paper confirms that while the technology (BDNS) offers potential to overcome the centralization and censorship issues of traditional DNSs, significant limitations still remain in terms of practical commercialization and widespread adoption. Studies by Kalodner et al. [33] and Xia et al. [37] have shown that in systems such as Namecoin and ENS, domain squatting and functionally meaningless registrations are frequently observed, while active domain trading and meaningful usage remain extremely limited. Ali et al. [34], through the case of Blockstack, introduced technical efforts to improve the security and scalability issues of existing systems; however, challenges related to reliability and structural complexity still persist.

In addition, studies by Casino et al. [32], Randall et al. [36], and Muzammil et al. [39] highlight the dual nature of decentralization, which not only enables resilience but also facilitates security threats and malicious activities such as malware, phishing, and typosquatting. These studies point out the inherent difficulties that BDNSs face in terms of legal and societal response. Giamouridis et al. [8] and Ito et al. [38] identify TLD conflicts across systems, lack of interoperability, and regulatory gaps as major obstacles, and suggest that multifaceted improvements such as enhanced user experience, governance models, and integration with existing DNS infrastructure are necessary for commercialization.

To systematically analyze the current landscape, Table 2 applies a unified rubric across these prior BDNS surveys. We rate each work along common dimensions, including type, systems covered, data rigor, and coverage of security, performance/scalability, interoperability, governance/legal, and user aspects, using consistent symbols (✓/△/—). This normalization replaces ad hoc 'merits/demerits' and enables direct, like-for-like comparisons.

Table 2. Unified Evaluation of Existing Surveys on BDNS.

Authors	Year	Type	Systems	Data Rigor	Security	Perf/Scal	Interop	Gov/Legal	UX
Kalodner et al. [33]	2015	Empirical	Single (Namecoin)	Strong (120,000 domains)	—	—	—	✓	—
Key Findings		- Among 120 k registered .bit domains, only 28 hosted substantive content → extremely low active use. - Virtually no trading/transfers in the secondary market → market functionality is nearly nonexistent. - Proposes a utility-based incentive model to deter squatting and promote meaningful use.							
Ali et al. [34]	2016	Arch + Ops	Single (Blockstack)	Moderate (33 k names; 200 k tx)	✓	✓	△	—	—
Key Findings		- Operating a decentralized PKI on Namecoin (33 k names; 200 k tx) exposed reliability, throughput, and security issues (e.g., a single miner held >51% hash power for months). - Blockstack migrated to Bitcoin and introduced virtualchain to separate name control from storage; large data moved off-chain. - Off-chain design improved scalability/stability and opened a path to DNS integration.							
Wang et al. [26]	2019	Arch/Design	Prototype (BlockZone)	Limited	✓	✓	✓	—	—
Key Findings		- Implements DNS semantics over a permissioned blockchain using PBFT, without changing the DNS protocol. - Aims to remove censorship and SPOF; optimized for DNS query loads. - Preserves interop with legacy resolvers while decentralizing authoritative data.							
Patsakis et al. [35]	2020	Threat Analysis	Multi	Limited	✓	—	—	✓	—
Key Findings		- Decentralization's benefits (censorship resistance, immutability) also enable abuse: malware distribution, phishing, illegal content. - Legal recourse and dispute arbitration are hard or infeasible in BDNS contexts. - Even with measures (e.g., Handshake's pre-reservation of popular Alexa names), TLD collisions and disputes persist.							
Casino et al. [32]	2021	Empirical Security	Multi (.bit/.emc)	Strong (taint + blacklist)	✓	—	—	—	—
Key Findings		- Taint analysis and blacklist tracking show a high share of malicious domains in .bit and .emc. - Censorship resistance can be exploited for abuse; legitimate use cases are relatively rare.							
Randall et al. [36]	2022	Arch+Threat	Multi (Namecoin/ENS/HNS/UD/...)	Limited	✓	—	—	—	—
Key Findings		- "Fully decentralized" systems retain semi-centralized components (plugins, proxies, SPV nodes). - These components create both attack surfaces and defender intervention points. - No current solution qualifies as a truly "unstoppable" domain system.							
Xia et al. [37]	2022	Empirical	Single (ENS)	Strong (4-year longitudinal)	✓	—	—	—	△
Key Findings		- Four-year ENS study finds squatting, scam associations, record-keeping weaknesses, and numerous hijacking incidents. - 3.7% of ENS names were re-registered by others after expiration (effective hijack). - ENS achieves decentralization but needs security and reliability improvements.							
Giamouridis et al. [8]	2024	Survey+Policy	Multi	Moderate	✓	—	✓	✓	△
Key Findings		- Balanced analysis across technical, legal, and social dimensions; proposes concrete technical countermeasures. - Adoption is hindered by interoperability gaps, regulatory uncertainty, and UX shortcomings. - Stresses the need for governance models and structured integration with DNS.							
Ito et al. [38]	2024	Interop/Policy	Multi	Moderate (namespace comparison)	△	—	✓	✓	—
Key Findings		- Documents TLD conflicts across systems (e.g., .wallet in Handshake vs. Unstoppable Domains) leading to real disputes. - Compares namespace metadata, expiration handling, and record support; highlights risks of duplicate registrations and user confusion. - Argues for interop frameworks and dispute-resolution mechanisms.							
Muzammil et al. [39]	2024	Empirical (Typosquatting)	Multi (ENS/UD/ADA)	Strong (5,000,000 domains)	✓	—	—	—	—
Key Findings		From 5 M domains, ≈37% of names have typosquatted variants; a small set of squatter addresses holds most variants. - Some typo domains actively impersonate crypto addresses to steal assets. - Demonstrates that BDNS is particularly susceptible to squatting/typosquatting.							

Legend—Systems: "Single/Multi/Prototype/Broad"; Data Rigor: None < Limited < Moderate < Strong. Dimensions Covered: ✓ = covered, △ = partially, — = not covered.

However, despite these existing contributions, a closer examination reveals several limitations, which are outlined below:

1. The total number of studies is still relatively small or heavily focused on specific systems such as Namecoin and ENS, resulting in a lack of comprehensive analysis across diverse implementations.
2. Existing reviews often lack a clear or consistent classification framework for the systems analyzed, and fail to provide systematic comparisons of core elements such as domain registration mechanisms, governance structures, and incentive models. For example, some studies focus solely on specific security threats or individual case analyses, resulting in insufficient in-depth examination of the overall architecture and technical stack.
3. Consideration of the legal, policy, and societal challenges associated with BDNS remains limited, and aspects such as user experience and accessibility from the perspective of Real-world users have been largely overlooked.

Therefore, to address these gaps, this study aims to incorporate as many diverse cases of BDNS as possible and conduct a consistent analysis from multiple perspectives. In particular, aspects such as their interoperability with existing DNS infrastructure, analysis of domain squatting, governance models and the resulting degree of decentralization, security, and censorship resistance must be taken into account to derive constructive insights for improving and envisioning the future of BDNS. Moreover, for BDNS to become not merely a technical alternative but an integral part of the internet infrastructure, a multidimensional approach is required.

4. Analysis of Traditional DNS

The traditional DNS is based on a hierarchical namespace organized along the Root–Top-level Domain (TLD)–Second-Level Domain (SLD)–subdomain structure, and coordination of the top level of the namespace relies on a centralized coordination regime [40]. Although root servers are globally distributed and operated via anycast, the structural characteristic of concentrating policy and coordination authority at the top level is preserved [41].

TLDs beneath the root are classified into categories such as gTLDs and ccTLDs [42], and because actual domain registration and operation are performed through intermediary structures such as registries and registrars, users access services based on the state provided by the upper layers and these intermediaries [43–45], while this delegation-based architecture offers scalability and operational efficiency, it inherently entails structural constraints in that the locus of trust converges on specific layers and administrative entities.

Modern DNS heavily relies on caching and recursive resolvers to handle large query volumes [46], and in this process it is difficult for end users to independently validate the authenticity of resolution results. From an operational perspective, managed DNS services such as Cloudflare [47] and Amazon Route 53 [48] provide operational convenience through global load balancing and automated failover, helping to reduce propagation delay and improve availability.

However, as traffic and operations become concentrated on the infrastructures of a small number of resolvers and providers, the scope of impact of failures at specific points and policy interventions tends to increase. In this section, from this perspective, we first examine the security vulnerabilities of the traditional DNS in Section 4.1, and then provide an overview of the techniques introduced to mitigate them in Section 4.2.

4.1. DNS Security and Vulnerabilities

DNS operates in an open environment and is frequently targeted by attacks that manipulate resolution results, degrade availability, or abuse DNS as a covert channel. We summarize representative threat categories that will later be used to revisit baseline

defenses (Section 4.2) and to re-examine their security implications in the context of BDNS (Section 5.2).

DNS Cache Poisoning: DNS Cache Poisoning is an attack in which an adversary injects malicious data into a recursive resolver's cache to steer users to unintended IP addresses. The canonical threat model assumes off-path, race-style forgery; when the attacker is on-path, the scenario more closely resembles spoofing or a man-in-the-middle (MITM) attack. Typically, the attacker floods forged responses that arrive before the legitimate reply to a bona fide query, causing the resolver to cache the counterfeit record. Subsequent clients attempting to reach the target domain may be redirected to phishing or malware-hosting sites crafted to closely mimic the legitimate service.

Defenses include DNSSEC validation for signed zones, and increasing forgery difficulty via per-query randomization (e.g., transaction ID and source port); additional hardening mechanisms further narrow the acceptance window for forged responses.

DNS Spoofing: DNS spoofing broadly refers to forging DNS responses (on-path or off-path) to steer victims to unintended destinations or enable man-in-the-middle behavior. Unlike cache poisoning, spoofing does not necessarily aim to persistently contaminate a resolver cache.

DNS Reflection and Amplification Attacks: By spoofing a victim's IP as the source of DNS queries, attackers can reflect responses toward the victim. Amplification arises when small queries trigger disproportionately large responses (e.g., via EDNS(0), DNSSEC, or large TXT records), enabling high-volume DDoS. Mitigations include limiting response sizes and applying network-level filtering to reduce IP spoofing [49].

Zone Transfer Exploitation: DNS zone transfer is a mechanism used to synchronize DNS data between a master server and a slave server. However, if this feature is configured without access control, anyone can request and download the entire zone file. This may lead to the exposure of sensitive information such as domain configuration, internal network structure, and the servers in use. An attacker may send an AXFR (zone transfer) request to a publicly accessible DNS server, and if the server responds, they can collect all hostnames, IP addresses, mail server details, and other domain-related information. This technique is frequently used as a reconnaissance step in preparation for further attacks. The attack can be prevented by configuring the DNS server to allow zone transfer requests only from trusted IP addresses and by blocking unnecessary AXFR requests [50].

DNS Tunneling: DNS tunneling is a technique that conceals data within DNS queries and responses to enable unauthorized communication, making the traffic appear legitimate. It is commonly used to bypass security devices such as firewalls, proxies, and intrusion detection systems (IDS), and is frequently employed by malware to communicate with external command-and-control (C2) servers. Typically, DNS tunneling leverages encoded data embedded in the format of domain names within DNS queries. In practice, the malware is executed within the internal system and transmits data by sending DNS requests to attacker-controlled domains. For example, the malware may send a query such as *data.exfil.attacker.com* and receive commands embedded in the response packet to perform remote control. This type of attack can be mitigated by monitoring abnormal query patterns within DNS requests and applying DNS filtering techniques to block known suspicious domains [51].

4.2. Defense Mechanisms for Securing DNS

To defend against the aforementioned types of attacks, DNS has incorporated various security features throughout its development. The most prominent among these is DNSSEC, which has been mentioned multiple times as a countermeasure to the attacks described

above. In addition, several encryption-based technologies such as DoT and DoH have also been introduced.

DNSSEC: DNSSEC is a security extension to DNS that provides data integrity and origin authentication for DNS data by attaching cryptographic signatures (e.g., RRSIG) to DNS records. Essentially, a resolver can verify whether a received response corresponds to signed data from an authoritative zone, rather than simply accepting unsigned data on the basis of trust. Validation is structured as a hierarchical chain of trust: child zones publish their signing keys (DNSKEY), and parent zones vouch for them via DS records, enabling validating resolvers to authenticate records by walking the chain from a trusted starting point. If validation fails, the response is regarded as untrusted and is rejected by the resolver [52].

Despite its effectiveness against forgery-based threats such as cache poisoning, DNSSEC introduces substantial deployment and operational burdens, including significant key management and rollover complexity, increased response sizes, and validation overhead. Moreover, DNSSEC primarily addresses authenticity and integrity; it does not encrypt queries, nor does it resolve the dependence on recursive resolvers along the resolution path, leaving broader operational and privacy considerations to complementary mechanisms. These costs also help explain why DNSSEC, despite long-standing standardization, has seen limited effective deployment in practice.

Deployment reality of DNSSEC: Recent measurement-based reports suggest that DNSSEC adoption and its effective use remain limited in practice. A scan of the Tranco top-1M list found that 9.4% of domains are validly DNSSEC-signed, while only 3.2% of queries (based on Cloudflare 1.1.1.1 logs) target DNSSEC-signed names. APNIC's measurement further indicates that roughly one-third of users sit behind DNSSEC-validating resolvers; combining these figures implies that DNSSEC validation is triggered only around 1% of the time under today's query profile [53]. Figure 3 summarizes these indicative metrics.

DoT, DoH: DoT is a protocol that encrypts DNS queries using the Transport Layer Security (TLS) protocol, operating over a dedicated channel on port 853. This approach provides end-to-end encryption between the client and the DNS resolver, protecting against eavesdropping and man-in-the-middle attacks. A key advantage of DoT is that network administrators can identify and manage traffic based on port usage. However, this also introduces the possibility of access restrictions due to port blocking. In some countries, there have been reports of port 853 being blocked under the pretext of enhancing privacy, as a means of censoring politically sensitive content [54].

DoH encapsulates DNS queries within the HTTPS protocol and transmits them over port 443. Since it is difficult to distinguish from regular web traffic, DoH can easily bypass firewalls. Major providers such as Cloudflare and Google currently operate DoH services. In browsers such as Mozilla Firefox and Chrome, DoH can be enabled through user settings, offering increased convenience for individual users.

However, in enterprise environments, there are concerns that it may undermine DNS monitoring and filtering systems. According to a 2024 study, 14% of DoH traffic has been found to be exploited for communication with malicious C2 servers [55]. In response, AI-based anomaly detection solutions are currently under development [56].

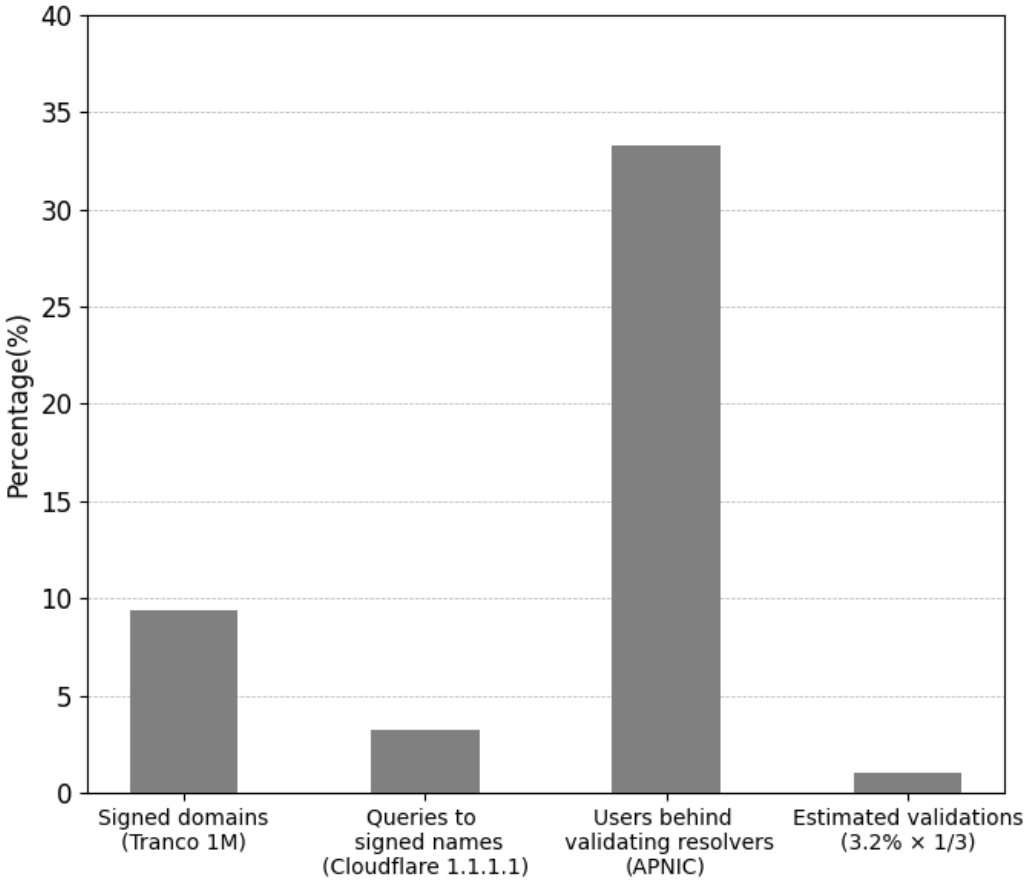

Figure 3. Indicative DNSSEC deployment and estimated effective validation rate (reported by APNIC using Tranco top-1M scans and Cloudflare 1.1.1.1 query logs).

5. Analysis of Blockchain-Based DNS

The traditional DNS suffers from limitations such as SPOF, security vulnerabilities, and susceptibility to censorship arising from its centralized architecture [57,58]. As an alternative, BDNS leverages DLT to introduce a new paradigm for managing domains without central intervention. In this section, we examine the underlying architecture and operational principles of BDNS and analyze the resulting security properties and potential threats.

5.1. Architecture and Operational Mechanism

Traditional DNS has been continuously extended to meet usability and security requirements. However, it remains structurally centralized because policy and coordination authority over the top-level namespace is concentrated in the upper tiers. This centralization has repeatedly been associated with security and censorship vulnerabilities. In practice, attacks targeting central infrastructure have been reported [57], and politically motivated DNS censorship has been observed in some countries [58].

Against this backdrop, BDNS has been proposed as an alternative that can operate without central intervention [33]. BDNS records state changes, such as domain registration, renewal, and delegation, as transactions and replicates them on a distributed ledger. The ledger is append-only, and cryptographic block linking provides immutability, making retroactive modification difficult unless the network's consensus rules are bypassed. Accordingly, BDNS aims to reduce reliance on specific root servers or a single administrative authority, thereby improving resistance to SPOF and arbitrary manipulation [8].

From an implementation perspective, BDNS can be broadly categorized into (i) smart contract–based designs and (ii) UTXO-based designs. Smart contract–based systems (e.g., ENS [59]) encode registration and delegation policies in on-chain logic and store state in contract storage. This yields high expressiveness (e.g., auctions and pricing policies) and strong composability. At the same time, logic bugs and access-control failures can directly compromise domain lifecycle semantics. As a result, these systems require rigorous static and dynamic analysis, external audits, and cautious upgrade mechanisms (e.g., proxies, timelocks, and emergency stops). They must also contend with L1 throughput limits and gas-price volatility, which can increase the cost and latency of registrations and updates. Common mitigations include minimizing on-chain storage and computation, leveraging L2 solutions, and integrating off-chain storage backends.

UTXO-based systems (e.g., Namecoin [27] and EmerDNS [31]) embed name–value data in transaction outputs and treat the blockchain as a distributed key–value store. Policy logic is typically simpler, while advanced features are often delegated to wallets/clients or auxiliary protocols. This approach offers simplicity and predictability, but it limits expressiveness, increases reliance on off-chain components, and largely inherits the performance constraints of the underlying base chain.

These architectural choices also shape name resolution and operational control. Unlike traditional DNS resolution (root–TLD–authoritative-server traversal), BDNS resolution typically performs a direct lookup of a name's state on the blockchain or via a gateway/node API. Clients (e.g., browsers, extensions, or OS resolvers) first identify blockchain-managed namespaces (e.g., by TLD rules) and then query a local full node or a remote RPC/gateway to retrieve records. Because responses are derived from consensus-finalized state, they provide a strong basis for authenticity and integrity. For performance, implementations often combine local caching with event subscriptions. Freshness is commonly tracked using chain-native signals such as block height or timestamps rather than DNS TTL.

Operationally, governance shifts from ICANN–registry–registrar-style enforcement toward protocol rules and community governance, optionally mediated by a DAO. This governance determines upgrades, fees, and policy changes. Token economics, such as consensus rewards and registration fees, also provide incentives for sustaining the network. For backward compatibility, some deployments adopt dual resolution, routing conventional domains to legacy DNS while resolving only selected namespaces via the blockchain path.

5.2. Security Vulnerabilities and Countermeasures in BDNS

Security threats faced by BDNS can largely be categorized into chain-level attacks and user/application-level attacks. By recording domain state on a distributed ledger and finalizing it through consensus, BDNS makes record tampering difficult by providing integrity and immutability, along with public-key-based ownership authentication and authorized updates as first-class primitives. Because it relies less on a central administrative entity, BDNS can also improve censorship resistance against arbitrary blocking or seizure, as well as availability and resilience through replication across many nodes. Since transaction histories are public, BDNS further enables transparency and traceability of registration, renewal, and transfer events, which can aid post-incident auditing and response [8].

Accordingly, the primary security failure modes shift from registrar compromise to consensus security, key custody, contract correctness, and the access infrastructure that mediates resolution. These benefits, however, introduce new assumptions and attack surfaces. In this section, we group BDNS threats into two layers: *chain-level* threats, which arise from the security of consensus and on-chain execution (including smart-contract logic and cross-chain namespace ambiguity), and *user/application-level* threats, which arise from key management, transaction submission and ordering, and the operational middleware

(e.g., gateways and RPC endpoints) that most users rely on to perform resolution. These vulnerabilities motivate the threat and defense discussion presented below.

Among chain-level threats, consensus takeover (e.g., a 51% attack) is a major concern, particularly for smaller or insufficiently decentralized networks. If the network is not sufficiently large, a group of miners or validators with majority hash power or stake may reorganize recent blocks and revert registration/transfer transactions, potentially enabling double-spend–like effects on domain updates. Additionally, logical vulnerabilities within smart contracts, such as reentrancy, flawed commit–reveal or auction logic, signature replay, or improper access control, can be exploited to bypass auction logic, acquire names at artificially low prices, or hijack ownership during registration or expiration processes. Furthermore, namespace collisions caused by different blockchains issuing the same TLD may create confusion about which chain's record to trust, providing a foothold for phishing or man-in-the-middle attacks [38].

At the user and application level, the most realistic threats are private key theft and transaction-ordering manipulation (MEV), including front-running [60,61]. In BDNS, where ownership is determined by possession of the private key, an exposed key allows an attacker to hijack a domain immediately with a single signed update transaction. Front-running, which involves monitoring the mempool in real time and submitting a transaction with a higher gas fee to register the same name first, is often used in "name sniping" attacks targeting popular keywords. Due to low registration fees and common models of permanent ownership, large-scale squatting and typo-squatting using bots are widespread, posing risks of phishing and brand abuse. The censorship resistance of BDNS can also be abused to host malware or illicit content, as noted in prior studies and incident reports. Additionally, because most users rely on lightweight gateways, gateway- or RPC-endpoint DDoS attacks can directly threaten resolution availability.

To mitigate these threats, BDNSs employ a combination of multi-layered defense mechanisms. The risk of consensus takeover is reduced by expanding hash power or stake pools through merged mining or delegated staking, and by verifying validator identities in permissioned consensus models such as PBFT or PoA, albeit with stronger governance and trust assumptions [62,63].

To address the user onboarding friction caused by private key management and the resulting SPOF, next-generation key management technologies that go beyond simple hardware wallets are being actively adopted. For example, threshold/multiparty computation (MPC) wallets avoid creating a private key as a single entity by distributing it across multiple shares and only combining them at signing time; consequently, domain ownership can be preserved even if some shares are exfiltrated [64]. In addition, the social recovery mechanism proposed by Buterin [65] enables a user who has lost their key to reset account access via the consent of a pre-designated majority of a trusted group, thereby dramatically improving the trade-off between security and user experience.

Contract vulnerabilities are addressed by requiring static and dynamic analysis and external security audits prior to deployment, and by adopting proxy patterns to allow for emergency patches. However, upgradeability also introduces an additional trust surface (e.g., administrator keys or governance capture), and thus must be paired with transparent governance, time-locks, and well-scoped upgrade policies. Economic disincentives also play a critical role: front-running and validator misbehavior are deterred through mechanisms such as blind auctions, registration fee burning, and slashing.

As summarized in Table 3, BDNS employs a multi-faceted approach to address security threats compared to traditional DNS. To fundamentally mitigate name-squatting, a sophisticated tokenomics design that goes beyond a simple registration-fee model is essential. One alternative under discussion is the Harberger tax model [66,67]. This system

requires domain owners to pay fees based on a self-assessed value, while allowing third parties to immediately purchase the domain at that price. This structure effectively maximizes the carrying cost of speculative holdings. Similarly, differentiated pricing schemes used by systems like ENS, such as sliding-scale fees that vary by name length or rarity or a price-decay premium implemented via a Dutch-auction style re-registration (where the price for an expired domain starts high and gradually decreases) are representative countermeasures that effectively eliminate the economic incentive for bot-driven front-running and preemptive registration attacks.

Table 3. Security threats and mitigations in Traditional DNS vs. BDNS.

Category	Traditional DNS: Key Security Threats	Traditional DNS: Mitigations/Countermeasures	BDNS: Key Security Threats	BDNS: Mitigations/ Countermeasures
Integiry/Forgery	Cache Poisoning; spoofing(incl. MITM)	DNSSEC (chain of trust); query ID & source-port randomization; DoT/DoH to encrypt client resolver link	Consensus capture (e.g., 51%) enabling rollback/ownership reversal; smart contract logic bugs	Increase hash/stake—economic security; permissioned consensus (PBFT/PoA) with validator identity & governance; contract audits and formal verification
Availability/DDoS	Reflection & amplification DDoS	Response size limiting; anti-spoofing filters (e.g., BCP38); Anycast & managed DNS for resilience	DDoS on gateways/RPC and resolver nodes (edge dependency points despite decentralization)	Scale out nodes & Anycast-style gateways; hybrid on-/off-chain design keeping only core mappings on-chain
Information Exposure/Recon	AXFR revealing internal topology	Restrict AXFR (ACLs/trusted IPs); disable unnecessary AXFR	—(Records are transparent on-chain: an auditability vs. privacy trade-off)	Use transparent ledger for audit/traceability & accountability (forensics-friendly)
Data Exfiltration/ Unauthorized Comms	DNS tunneling for C2 and exfiltration	Monitor anomalous query patterns; domain filtering/blocking; anomaly detection	Similar tunneling via gateways to hide traffic	Combine on-chain tranparency with gateway policy & anomaly detection (ongoing research)
Governance/ Censorship	Centralization → SPoF & censorship vulnerability	Multi-provider setups, Anycast, and operational hardening (not a fundamental fix)	Decentralized design → strong censorship resistance & no SPoF (advantage)	—(Intentional design property)
Namespace/User Abuse	—	—	Cross-chain TLD collisions; squatting/typosquatting; key theft & front-running; illicit content hosting	Standardization/interoperability (mitigate collisions); better commit–reveal/auction and anti-front-running; hardware wallets/multisig/social recovery/timelocks; reputation lists/blacklists & gateway policy; user education

For malicious content and already-registered squatting domains, on-chain immutability is preserved while mitigation is handled through auxiliary measures at the browser or gateway layer, such as Safe Browsing filters, community-maintained blacklists, and preferential re-registration rights upon expiration. Namespace collisions are mitigated through cross-chain reservation status checks or resolver rules that allow users to specify priority. Finally, to ensure availability even under gateway DDoS or Eclipse attacks, measures such as anycast RPC multiplexing, local ledger caching, and subscription to chain events are employed. This layered defense strategy, combining cryptographic access control, economic incentive design, code quality assurance, and network redundancy, constitutes the core of the BDNS security model.

6. Comparative Analysis of Traditional DNS and BDNS

In this section, building on the key characteristics summarized in Sections 4 and 5, we contrast traditional DNS and BDNS along a common set of dimensions. The comparison is conducted from the perspectives of architecture and governance (Section 6.1), security and trustworthiness (Section 6.2), censorship resistance and user autonomy (Section 6.3), scalability and performance (Section 6.4), and integration and compatibility with existing infrastructure (Section 6.5). Through this analysis, we present a balanced account of the benefits offered by both approaches as well as the trade-offs that arise in terms of performance, cost, operational complexity, and compatibility, and finally draw overall conclusions (Section 6.6). Figure 4 presents a side-by-side comparison of the end-to-end name-resolution flows for the two systems and serves as a common reference point for the comparative discussion in the subsequent subsections.

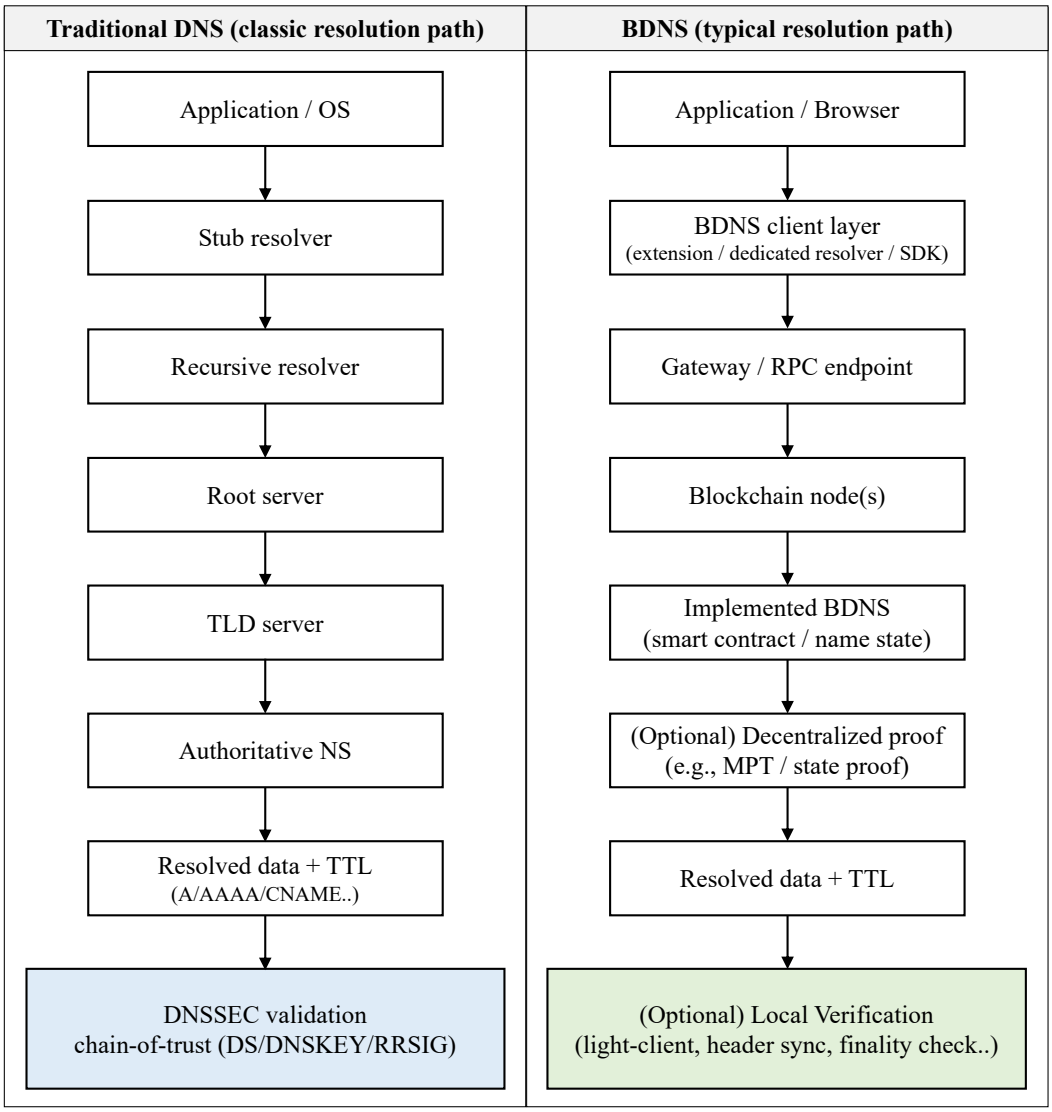

Figure 4. Comparison of Name Resolution between Traditional DNS and BDNS.

6.1. Structure: Centralized vs. Decentralized

As illustrated in Figure 4, traditional DNS resolves names primarily through a hierarchical delegation chain (Root–TLD–Authoritative) mediated by recursive resolvers. Consequently, administrative and coordination authority tends to concentrate in the upper tiers of the hierarchy and in the entities that operate them. Building on the architectural

vulnerabilities identified in Section 4, this hierarchy, rooted in centralized administrative authority, can introduce a SPOF [8,68].

In contrast, the BDNS proposed by Li et al. [69] leverages DLT to replicate naming data among participating nodes and to verify the legitimacy of updates based on consensus and cryptographic ownership, thereby shifting the underlying trust model, while the hierarchical design of traditional DNS has demonstrated scalability and operational efficiency, BDNS emphasizes decentralization and resilience by reducing dependence on centralized control.

Nevertheless, prior work indicates that these structural differences entail trade-offs. For example, ref. [68] reports that synchronization and consensus overhead among geographically distributed nodes can increase the effective query response time. Furthermore, ref. [70] emphasizes that if multiple blockchain-based naming systems coexist without coordination or interoperability mechanisms, the likelihood of name collisions increases, fragmenting the namespace and, in turn, making backward compatibility with the legacy DNS more difficult.

6.2. Security and Trustworthiness

As illustrated in Figure 4, traditional DNS relies heavily, during name resolution, on trust in recursive and authoritative name server operators and in a centralized administrative and coordination framework (reinforced when needed by DNSSEC validation), whereas BDNS establishes trust based on the consensus-finalized ledger state and cryptographic verification of ownership. This difference in the placement of trust anchors leads to differences in attack surface (attack surface), attack cost, and integrity assurance mechanisms, thereby determining the respective strengths and weaknesses of the two approaches in terms of security and reliability.

One of the main vulnerabilities of the traditional DNS lies in its lack of security. In this regard, Li et al. [69] experimentally demonstrated that BDNS offers improved security compared to traditional DNS in terms of attack success rate, attack cost, and attack surface. In particular, by leveraging the immutability of blockchain, once DNS records (or domain states) are verified and finalized through network consensus, it becomes extremely difficult to modify them arbitrarily, thereby enhancing resistance against threats such as record tampering and hijacking.

Meanwhile, ref. [8] argues that a blockchain-based DNS can reduce the risk of SPOF inherent in the centralized architecture of conventional DNS and improve resilience to cyberattacks, while simultaneously enhancing the transparency and accountability of domain events such as registration, renewal, and transfer, by recording domain ownership and associated DNS records on a decentralized immutable ledger.

From the standpoint of cryptographic techniques and consensus procedures, blockchain-based DNSs generally require the agreement of multiple participants to authorize network changes, typically via “voting” or similar consensus mechanisms, and in many cases all nodes, or a sufficiently large subset of nodes, must participate in the approval process [71].

In such settings, the security level is largely determined by the distribution of voting power or stake and by the consensus rules. In environments with highly influential participants (whales), the cost for an attacker to obtain majority influence can increase, whereas when authority is concentrated in a small set of actors, risks such as consensus capture or governance capture must also be considered.

6.3. Censorship Resistance and User Autonomy

As illustrated in Figure 4, the points at which censorship can be inserted differ between the two systems. In traditional DNS, because user requests traverse a recursive resolver

(typically provided by an ISP or a specialized operator), that resolution point can serve as a practical choke point for policy control and filtering. In contrast, although BDNS is designed to reduce reliance on centralized delegation authorities, in practice service access often goes through intermediary infrastructure such as gateways or RPC endpoints, so censorship resistance and user autonomy are determined by the configuration of not only the "ledger layer" but also the "access layer."

Authoritarian governments are increasingly employing technical measures to control the free flow of online information, and in this context the structural properties of traditional DNS have made it an operationally convenient instrument for censorship. Specifically, state-backed censors can tamper with DNS resource records at ISP-operated resolvers to connect users to invalid IP addresses or redirect them to block pages, and ref. [72] reports that such blocking mechanisms are in use in multiple countries.

Ref. [68] introduces projects such as Namecoin, ENS, Blockstack, and Handshake, emphasizing that they aim to provide "censorship-free" naming services independent of central authorities like ICANN. In particular, it argues that in traditional DNS a single administrative framework, namely the root and upper-level delegation structure, can constitute centralized trust and SPOF, leaving the system exposed to risks such as policy intervention, DDoS attacks, and forced mapping changes or spoofing. On this basis, ref. [68] contends that blockchain-based naming may offer potential advantages in terms of censorship resistance.

Ref. [70] further notes that even if operations on domain state, such as create, read, update, and delete, are publicly accessible, it is difficult to directly infer ownership from the identity of the actor performing the operation, implying that pseudonym-based registration mechanisms can enhance user privacy. The study in [73] presents a distributed DNS configuration that combines IPFS with the Phincoin blockchain, describing an approach to mitigate the censorship risks inherent in the centralized structure of traditional DNS and reporting that a decentralized design can increase censorship resistance.

As suggested by Figure 4, even if BDNS weakens central control at the ledger layer, practical censorship resistance and user autonomy can vary depending on the degree of centralization in the access layer (e.g., gateways/RPC endpoints, browser extensions/clients). In other words, the more users secure self-verifiable paths (e.g., operating their own nodes, light-client-based verification, leveraging multiple gateways), the greater their autonomy; conversely, the more they depend on a specific provider's infrastructure, the more likely DNS-like control points will be reconstituted.

6.4. Scalability and Performance

As shown in Figure 4, traditional DNS offers high scalability and performance even at a global query scale, owing to mature operational optimizations built around recursive resolver caching and a hierarchical delegation structure. In contrast, blockchain-based DNS generally underperforms because name resolution entails additional overheads such as ledger access, node synchronization and consensus, and, depending on the design, proof generation and verification as well as finality considerations [8]. Nevertheless, recognizing these limitations, numerous studies have proposed a variety of approaches to improve performance.

First, ref. [74] proposes performance improvements via sharding. Sharding is a technique that increases throughput by distributing data and processing load across multiple nodes or node groups, and it is widely discussed as a means to mitigate scalability bottlenecks in blockchain systems. The study focuses on enhancing scalability through transaction-level load balancing by partitioning nodes or node groups into multiple shards and configuring each shard to handle DNS queries. In simulations, the proposed method

achieved more balanced distribution than conventional load balancing schemes such as round robin and weighted round robin, and system performance tended to improve as the number of replicas increased. These results indicate that the proposed design can improve scalability and yield better load balancing effects [74].

Additionally, ref. [75] proposes a consortium BDNS caching system called DNS-BC, along with a new encrypted DNS protocol, DNS over KCP (DoK). Here, KCP is a fast and reliable Automatic Repeat Request (ARQ) protocol designed to mitigate the performance and security issues of conventional DNS. DoK leverages the fast transport characteristics of KCP to reduce the complex handshake overhead often cited as a performance bottleneck in DoT/DoH, and aims to increase data transfer throughput. The accompanying cache is designed to deliver high real-time performance and accuracy, as well as improved defenses against DoS attacks [75]. As a result, although raw query latency was somewhat longer than in traditional DNS, the user-perceived difference was not substantial; the system reported up to 100% cache accuracy, mitigation of real-time performance degradation due to expired entries, and improved protection of name servers against DoS attacks [75].

However, a substantial subset of these improvement proposals, such as [75], is built on consortium chains or private/permissioned environments rather than public blockchains, or focuses on localized optimizations at the level of specific components (e.g., transport protocols, caches) rather than the system as a whole, which remains a limitation. Consequently, for blockchain-based DNS to see broad adoption, sustained performance improvements are required across the entire end-to-end path, including ledger access and consensus/verification overheads as illustrated in Figure 4.

Separately from performance, blockchain-based DNS shows scalability potential in fundamentally mitigating the domain name exhaustion problem of traditional DNS through decentralized TLD creation mechanisms. For example, the conventional DNS, due to its ICANN-centered administrative framework, permits only a limited number of TLDs (about 1,500), whereas protocols such as Handshake (HNS) provide an open namespace in which users can create arbitrary TLDs (e.g., .bank, .app) [8,28]. In particular, Handshake's auction system introduces a two-phase blind auction procedure to mitigate domain preemption, and empirical studies report a 42% reduction in domain squatting [33].

However, a key challenge for blockchain DNS lies in its limited interoperability with the existing DNS infrastructure. In the case of Handshake, as of 2025 only a small fraction of global DNS resolvers are reported to support Handshake TLDs. This suggests the need for hybrid resolver architectures, which we discuss in the next section.

6.5. Integration and Compatibility

As shown in Figure 4, traditional DNS offers a unified global namespace through a standardized resolution path integrated into the OS/browser networking stack (via stub and recursive resolvers). In contrast, BDNS typically accesses ledger state through additional layers such as extensions or Software Development Kit (SDKs) and gateway or RPC endpoints. This architectural mismatch makes integration and compatibility a central challenge: namely, how heterogeneous naming systems can interface with the existing DNS ecosystem, including resolvers, DNSSEC, OS/browser stacks, and governance and policy frameworks.

Interoperability with traditional DNS is therefore a practical prerequisite for mainstream BDNS adoption. However, BDNS projects span a wide spectrum from coexistence to replacement. For example, Handshake pursues a DNS-grounded decentralized root, whereas Namecoin targets an independent namespace, and limited consensus persists at both technical and governance levels [70].

In this context, ref. [35] summarizes key integration issues faced when blockchain-based naming systems coexist with or attempt to replace traditional DNS. These challenges include (i) domain collisions caused by TLD overlap with existing DNS, (ii) trust erosion induced by namespace confusion and rapid namespace proliferation, (iii) increased malware risk enabled by bulk registration of semantically meaningful, low-entropy names, (iv) escalation of phishing and potential bypass of DNS security mechanisms, and (v) cybersquatting driven by large-scale preemptive registrations [35].

Beyond these general concerns, prior work also highlights concrete operational bottlenecks. Ref. [70] notes the lack of agreement on coexistence versus replacement across BDNSs, while [76] observes that ENS–DNS linkage depends on DNSSEC and can be constrained by mismatches in supported algorithms. For instance, in September 2022 ENS did not support RSA-SHA512, which prevented linking certain ".it" SLDs, illustrating how incomplete alignment on current and future DNSSEC algorithms can hinder real-world interoperability [76].

Importantly, BDNS does not necessarily need to replace traditional DNS. Ref. [59] emphasizes that ENS is designed to operate alongside, rather than supplant, the legacy DNS, motivating adoption strategies centered on coexistence. Complementarily, other approaches replace only specific parts of the DNS pipeline, including the resolver or validation layer, with blockchain-based components. For example, ref. [77] proposes a configuration that detects and defends against forged records induced by DNS cache poisoning, suggesting an incremental integration path via partial substitution. This perspective is consistent with Figure 4, which indicates that BDNS can be coupled to the existing resolution path as an additional layer.

To make such coexistence and partial substitution operational, we consolidate the integration requirements into a practicality-oriented, coexistence-first framework that guides deployment from protocol-level design to staged rollout. Figure 5 provides an overview of this framework as five layers, which are then instantiated by the concrete recommendations in (A)–(E) below.

(A) Record/Protocol-level Interoperability

- *Minimal interoperable RR set*: Support and surface A/AAAA/CNAME/TXT/TLSA as the baseline. When BDNS data is served, attach verifiable proofs (e.g., structured signatures) within TXT/TLSA or via an HTTPS/SVCB advertisement.
- *Cross-advertisement (DNS ↔ BDNS)*: Publish a BDNS authority notice in traditional DNS and reciprocally anchor DNSSEC fingerprints in BDNS.
- *Collision handling and claims*: Maintain a reserved-collision list for overlapping TLDs/labels. When conflicts arise, require a time-bounded *claim-with-proof* window (e.g., DNSSEC-based proof-of-control over the legacy name) before activation in BDNS.

(B) Resolver Behavior and Safety Policies

- *Deterministic query order*: (i) cache, (ii) traditional DNS authoritative path, and (iii) *only if DNS returns* `NXDOMAIN` *(default policy)*, invoke the BDNS verification path. Allow policy-based overrides (e.g., BDNS-preferred subtrees) where appropriate.
- *Monotonic responses under reorgs*: Stage BDNS answers in a shadow cache and promote only after finality to avoid response rollback. If a reorg is detected, pin prior answers until new finality is reached.
- *TTL normalization*: Expose BDNS answers only after finality and cap their lifetime using

$$\mathrm{TTL}_{\mathrm{bdns}} = \min\big(\mathrm{TTL}_{\mathrm{auth}},\ \max(0,\ T_{\mathrm{refresh}} - \Delta T_{\mathrm{finality}})\big), \tag{1}$$

where $\mathrm{TTL}_{\mathrm{auth}}$ is the authoritative TTL, $\Delta T_{\mathrm{finality}}$ is the required finality *delay* (confidence window) before exposing BDNS data, and T_{refresh} is the resolver's refresh horizon. In practice, Equation (1) prevents stale or non-final data from persisting at the edge.

- *Fallback and precedence*: By default, attempt BDNS substitution only on `NXDOMAIN`. Optionally, for explicitly scoped policies (e.g., BDNS-preferred subtrees), query DNS and BDNS in parallel and resolve conflicts via a documented selector (e.g., "prefer a final, verifiable BDNS answer within the TTL budget").

(C) Security and Algorithm Compatibility

- *Algorithm matrix*: Publish a compatibility matrix covering DNSSEC algorithms (e.g., RSASHA256, ECDSAP256SHA256, ED25519, ED448) and BDNS-side signature schemes. Provide a fall-back cross-signing path when a zone uses a not-yet-supported algorithm.
- *Abuse mitigation*: Rate-limit low-entropy mass registrations, require reputation or proof-of-control for sensitive labels, and enable anti-phishing policy flags (e.g., "must-match HTTPS/TLSA SANs") for user agents.

(D) Operational/Governance Integration

- *Gateway–registrar pattern*: Allow legacy registrars to act as delegated issuers/attesters for BDNS entries while aligning disputes (squatting, priority, revocation) with well-known procedures (UDRP/URS-like).
- *Service objectives*: Target end-to-end monthly availability $\geq$ 99.9% and additional p95 resolve-latency overhead $\leq$ 20 ms in mixed DNS+BDNS operation; instrument continuous monitoring and public SLO reports.
- *Interoperability test suite*: Publish reproducible test cases that cover A/AAAA/TXT/TLSA, SVCB/HTTPS hints, DNSSEC anchoring, cache hit/miss paths, and reorg-resilience; require pass status before enabling BDNS-preferred policies.

(E) Phased Roll-out with Clear Exit Criteria

- *Phase 0 (opt-in overlay)*: HTTPS gateway + resolver plug-in; success if added p95 latency $\leq$ 20 ms and failure rate $\leq$ 0.1%.
- *Phase 1 (hybrid resolver)*: Parallel DNS/BDNS queries with policy merge; success if availability stays $\geq$ 99.9% in mixed traffic and no non-monotonic responses are observed.
- *Phase 2 (registrar integration)*: Dual issuance (DNS+BDNS) with atomic state sync for issue/revoke and documented rollback; success if zero dangling states appear during forced failover tests.
- *Phase 3 (selective switchover)*: Enable BDNS-preferred policies for well-scoped domains (e.g., IoT subtrees) with a revert plan; success if no user-visible regressions or outage tickets are filed in the evaluation window.

This guidance turns integration and compatibility into a verifiable plan: concrete records to publish, deterministic resolver behavior (including Equation (1)), algorithm-compatibility guardrails, and phased roll-out criteria. Such specificity allows operators to adopt BDNS incrementally while preserving the safety, predictability, and user trust established by traditional DNS.

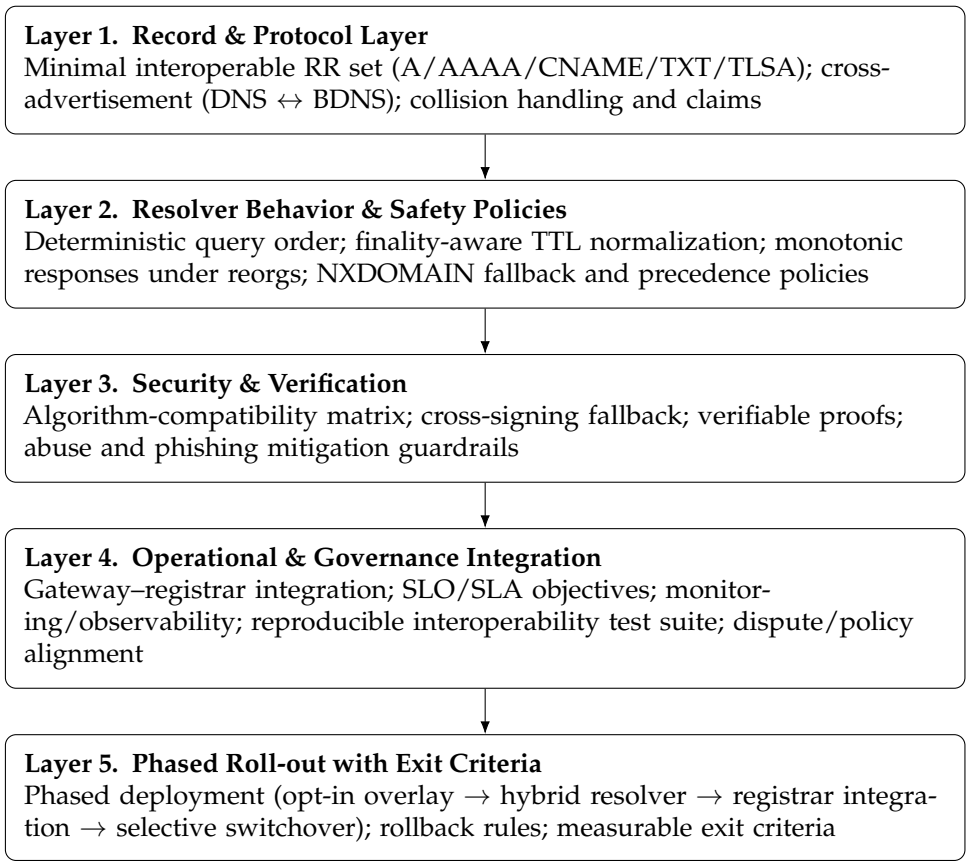

Figure 5. Coexistence-first five-layer integration and migration framework.

6.6. Overall Summary and Conclusion

Unlike the centralized structure of traditional DNS, BDNS offers distinct advantages in terms of security and trustworthiness due to its decentralized architecture. In particular, the immutability of blockchain prevents unauthorized modifications to DNS records, eliminates SPOF, and provides better protection against traditional attacks such as DNS cache poisoning. Additionally, BDNS is recognized for its strong censorship resistance against authoritarian government control and for enhancing user privacy.

However, BDNSs exhibit significant weaknesses in terms of performance and scalability compared to traditional DNS. In particular, delays in data synchronization between nodes can result in longer query response times, and issues such as namespace collisions and limited backward compatibility hinder smooth integration with existing infrastructure, while some studies have attempted to improve performance through techniques such as sharding and caching, these efforts have largely remained limited to specific areas rather than addressing overall system performance.

In addition, issues related to integration and compatibility with the existing DNS remain major challenges that must be addressed for the mainstream adoption and long-term growth of BDNS. These aspects are summarized and compared in Table 4, which provides a clear overview of the key differences. Given these considerations, a more realistic approach is not to fully replace the traditional DNS, but to establish a complementary and coexistent relationship while gradually improving performance and pursuing integration strategies.

Table 4. Traditional DNS vs. BDNS: qualitative comparison.

Comparison Factors	Traditional DNS	BDNS
Governance & Control	Hierarchical delegation (Root–TLD–SLD); global coordination under ICANN/IANA and registry/registrar ecosystem	Naming rules and ownership changes are enforced by protocol/consensus and smart contracts; governance and policies may differ per BDNS project (potential fragmentation)
Trust Anchor & Authentication	Trust is typically placed in recursive/authoritative operators; DNSSEC adds a hierarchical chain of trust (DS/DNSKEY)	Integrity/authentication are anchored to consensus + public-key ownership; however, security depends on consensus assumptions, contract correctness, and key custody
Data Integrity & Auditability	Integrity is optional (DNSSEC deployment/validation varies); limited public audit trail for updates depending on operators	Updates are recorded as transactions, enabling transparent and tamper-evident audit trails; on-chain state is hard to alter without breaking consensus
Update Model & Finality	Updates propagate through zone distribution and caches (TTL); eventual consistency with operational propagation delays	Updates become effective after block confirmation/finality; may require handling reorg/confirmation depth and proof freshness depending on design
Resolution Path & Operational Dependency	Native OS/browser support; resolution via stub → recursive → iterative lookups; anycast and mature operations common	Often requires gateways/RPC endpoints, browser extensions, or dedicated resolvers/light clients; some designs add off-chain indexing/storage (e.g., gateways, IPFS-like layers)
Scalability & Performance	Very high in practice due to caching, anycast, and mature resolver optimizations	Typically higher latency and overhead from chain access, proof generation/verification, and network synchronization; mitigations include caching, light-client proofs, and sharding/L2 research
Failure Modes & Attack Surface	DDoS on authoritative/recursive infrastructure; registrar/account compromise; cache poisoning when validation is absent	Consensus-level attacks (e.g., majority takeover), smart-contract bugs, key compromise/loss; additionally, gateways/RPC endpoints can reintroduce central failure/censorship points
Censorship Resistance	Policy and administrative takedowns feasible due to centralized coordination points (registry/registrar, resolvers)	Higher resistance to unilateral seizure at the naming layer; however, practical censorship can still occur at access infrastructure (gateway/RPC blocking)
Integration & Compatibility	Unified global namespace; broad backward compatibility without application changes	Interoperability is non-trivial; risk of namespace fragmentation/TLD collisions across multiple systems; requires integration layers for legacy applications

Note: BDNS properties vary by implementation (underlying chain, consensus, and access layer). The table summarizes common characteristics and trade-offs.

7. Analysis of Major BDNS Projects

In this section, we select seven representative BDNS projects, either currently in active operation or of significant academic interest, and analyze them in depth. Because the BDNS ecosystem remains in an early stage of development, the number of commercialized projects is limited. Consequently, this study avoids a simple catalogue of cases and instead selects analysis targets using two core criteria: Operational Status and Architectural Representation. To systematically compare the technical approaches and differences in data models adopted by each project, we classify the selected projects into three categories: (i) UTXO-based models (Type I), (ii) smart-contract-based models (Type II), and (iii) standalone-chain and hybrid models (Type III). Table 5 summarizes this architecture-based classification. In the sections that follow, we first analyze the projects corresponding to the UTXO-based model

(Type I) and then examine the smart-contract-based model (Type II) and the standalone-chain and hybrid models (Type III) in turn.

Table 5. Classification of Representative BDNS Projects by Architectural Approaches.

Category	Specification	Projects
Type I: UTXO-based Systems	Based on a Bitcoin fork; limited scripting capabilities; naming data stored in transaction outputs.	Namecoin, EmerDNS
Type II: Smart Contract-based Systems	Runs on Turing-complete chains (e.g., Ethereum); high logic flexibility; enables DAO governance.	ENS, Ustoppable Domains, Blockzone (research prototype, EVM-based)
Type III: Custom L1/Hybrid Systems	Introduces DNS-specific consensus algorithms or off-chain layers to address scalability and root zone issues.	Handshake (UTXO-based, replaces Root Zone, dedicated PoW), Stacks(Blockstack) (virtualchain, off-chain storage)

7.1. Namecoin

Namecoin, the prototypical Type I example, is the first decentralized DNS based on a UTXO model. It was developed in 2011 by forking the Bitcoin codebase [27]. It aims to enhance censorship resistance and privacy within the internet infrastructure and is specialized in managing `.bit` domains and operating an identity system. By supporting an independent namespace that does not rely on central authorities such as ICANN, it offers strong censorship resistance compared to traditional DNS, as neither governments nor corporations can seize or censor domains.

As it was forked from the Bitcoin 0.3 codebase in 2011, the system adopts a SHA-256-based PoW consensus algorithm and supports merged mining, allowing simultaneous mining of both Bitcoin and Namecoin. By sharing the same PoW algorithm across two or more blockchain networks, blocks on the smaller chain, Namecoin, are generated in conjunction with the Bitcoin mining process. As a result, attacking Namecoin would require computational power comparable to the hash rate provided by Bitcoin miners, significantly enhancing its security. In addition, miners can expect additional revenue by mining both chains simultaneously.

Domain registration is performed through specific transaction types such as `name_new`, `name_firstupdate`, and `name_update`, using JSON-formatted data to store information such as IP addresses, subdomains, and TLS certificates. Since `.bit` domains are recorded on the blockchain, the registered information cannot be tampered with and is permanently preserved due to the inherent properties of the blockchain. This immutability helps prevent DNS spoofing and cache poisoning and enables the construction of a trust chain without relying on centralized certificate authorities (CAs) by storing TLS certificates on the blockchain.

However, Namecoin is difficult to use in environments such as mobile devices due to the complexity of its initial setup and the large blockchain size exceeding 4 GB. Furthermore, if the full blockchain is not synchronized, its censorship resistance and security may be weakened. In addition, the issue of domain squatting hinders fair domain allocation. Due to the low cost of domain registration in Namecoin, malicious users often preemptively register large numbers of domains for resale or advertising purposes. This is due to the absence of mechanisms such as auctions or algorithm-based pricing, which ultimately hinders the effective utilization of domains [8]. In addition, the `.bit` TLD is not registered with standardization bodies such as ICANN or IETF, resulting in poor compatibility with the traditional DNS. It is also not natively supported by most web browsers, leading to a limited user experience (UX) [78].

With respect to performance, report [79] notes that Namecoin—the first blockchain-based naming system forked from Bitcoin—has an average block interval of 10 min. Consequently, it exhibits Bitcoin-like throughput, at around 7 tx/s.

A study in 2015 [33] found that out of 196,023 domains registered on Namecoin, only 28 were actively in use, with the vast majority held by squatters. Furthermore, due to its anonymity, the `.bit` domain began to be used as a hub for malware. The inability to contact domain owners led to backscatter effects, and many operators of public Tier 2 servers received service termination notices from providers, domain blacklisting, and email blocking. As a result, OpenNIC discontinued support for `.bit` domains [32]. Consequently, the project is now rarely used in practice.

7.2. EmerDNS

EmerDNS [31] is a decentralized domain name system that leverages Emercoin's UTXO architecture and Name-Value Storage (NVS), and operates under the 'dns' service abbreviation. Thanks to Emercoin's secure and distributed blockchain, domain name records are fully decentralized and censorship-resistant, meaning that no regulatory authority can alter, revoke, or suspend them. This system is specifically designed as an alternative to censorship, such as government or authority-imposed website blocking.

Based on the NVS, this system operates by recording and storing name-value pairs on the blockchain, as its name suggests [80]. Domain records are primarily managed using the Emercoin NVS feature within the Emercoin wallet GUI, or through the `name_new` and `name_update` commands provided by the Emercoin API. These records can be easily retrieved via the Emercoin API using JSON-RPC or the command-line interface, as well as through the standard RFC1034 [13] DNS protocol integrated into all Emercoin wallets, while the system is technically capable of supporting any DNS zone or TLD, it is currently recommended to create records under the `.coin`, `.emc`, `.lib`, and `.bazar` zones to ensure smooth integration with the existing DNS hierarchy. These TLDs are used, respectively, for digital currency and commerce websites, Emercoin project-related sites, knowledge and freedom-related content, and marketplace platforms. Each domain can store all traditional DNS record types, including A, AAAA, CNAME, MX, and TXT.

Domain resolution in EmerDNS follows a three-step process: domain normalization, NVS lookup, and proxy or OpenNIC resolution [81]. During domain normalization, when a user enters a domain such as example.coin, the input is converted to lowercase and validated against the UTS46 standard [82] to ensure the use of valid characters. In the NVS lookup step, the normalized domain name is queried within the Emercoin blockchain's NVS, and the corresponding DNS records are returned. For most cases, the resolution process ends here; however, if the underlying DNS infrastructure does not recognize EmerDNS, the domain is resolved via OpenNIC or a browser extension.

Through its EmerDNS decentralized SSL certificate system, it mitigates MITM attacks and strengthens server access security by anchoring SSH key management on-chain, thereby providing multiple layers of defense. Domain registrations are paid for with EmerCoin (EMC), and, similar to Unstoppable Domains, it adopts a one-time purchase model under which a single purchase confers permanent ownership of the domain name.

However, due to compliance with RFC 1912 and 2181 standards [83,84], the use of CNAME records at the root domain level (e.g., `example.com`) is not permitted, although CNAMEs are allowed for subdomains such as blog.example.coin. This, however, leads to compatibility issues with traditional DNS infrastructure. In addition, since the system depends on the Emercoin blockchain, it inherits the performance limitations of that network. Specifically, the relatively low TPS, resulting from Emercoin's average block generation time of approximately 10 min, can cause performance bottlenecks.

Major browsers such as Chrome and Safari do not offer native support, making it necessary to configure OpenNIC or install browser plugins. Furthermore, due to the nature of blockchain-based systems, there is no recovery mechanism in case of private key loss, which has led to user experience challenges—34% of users in 2024 reported difficulties in key management [85]. These limitations highlight the need for improvements in usability.

Performance is inherently bounded by Emercoin L1 and can be verified via the Emercoin explorer [86]. Although the official Emercoin specification [87] sets the block interval at 10 min, observations from [86] on 11 October 2025 indicate an average of approximately 7.6 min. For TPS, the 24-h transaction count was 422, which computes to about 0.0049 tx/s; however, given the very low current activity, this figure should not be interpreted as the maximum throughput.

Real-world usage examples are presented below.

- **Use of Fbot Botnet Infrastructure:** The Fbot botnet has been found to utilize EmerDNS for managing C2 server addresses. Qihoo 360's NetLab 360 team reported that Fbot made tracking more difficult by using EmerDNS instead of traditional DNS to manage its C2 domains [88].
- **Expanding Global Accessibility Through Collaboration with OpenNIC:** Through a partnership with the decentralized DNS provider OpenNIC, EmerDNS made its custom TLDs such as `.coin` and `.bazar` publicly accessible to global users. Users who configure their systems to use OpenNIC DNS servers can access EmerDNS domains without requiring additional plugins.

7.3. ENS

Moving to the smart-contract-based model (Type II), ENS is a decentralized naming system built on the Ethereum blockchain that maps complex cryptocurrency wallet addresses within Ethereum (e.g., 0x4cbde58...) to human-readable names such as `vitalik.eth` [29]. It maps complex cryptocurrency wallet addresses within Ethereum, such as `0x4cbde58...` to human-readable names like `vitalik.eth`. In addition to Ethereum, ENS allows linking various cryptocurrency wallet addresses, such as Bitcoin, to a single ENS domain. For example, both Ethereum and Bitcoin addresses can be associated with `bob.eth`, enabling the use of a single domain for transactions across multiple cryptocurrencies.

ENS can also be used to host websites by integrating with decentralized storage networks such as IPFS, allowing access to distributed content through domains like `blog.bob.eth` [89]. ENS can also serve as a Web3 profile and digital identity, as it allows the storage of various metadata such as email addresses, social media profiles, and avatars within the domain. This enables users to maintain a consistent digital identity across different applications and services [90].

ENS, which supports such services, is primarily composed of a set of smart contracts, categorized into the Registry, Resolver, and Registrar contracts. As shown in Figure 6, which has been reconstructed from the official ENS documentation for clarity, the name resolution process can be understood through the interaction of these components. The Registry serves as the starting point for all ENS lookups, storing and managing information about the owner, resolver, and TTL for each domain and subdomain. It is responsible for returning the address of the corresponding resolver in response to a query. Through the Registry, a domain owner can configure the resolver, create subdomains, and transfer ownership.

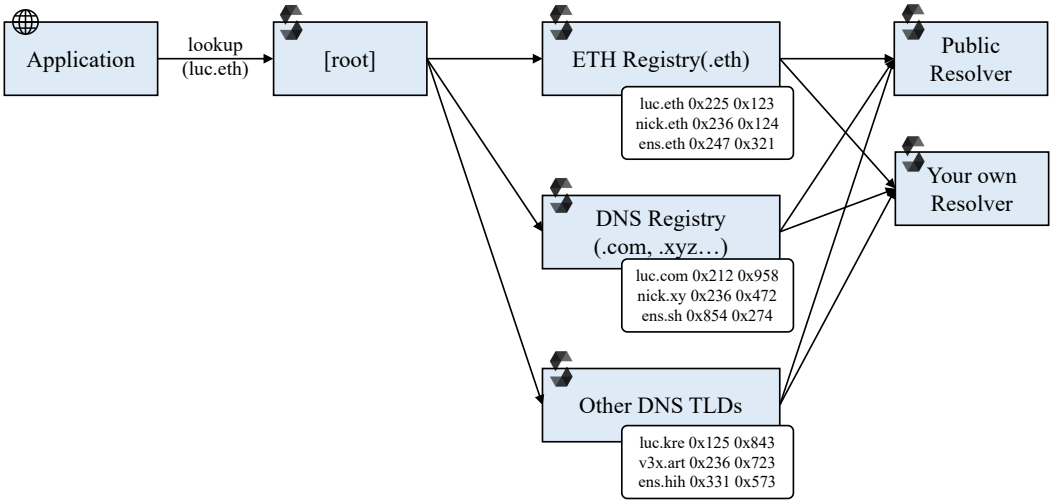

Figure 6. The ENS resolution process. Adapted from [91].

The Registrar is a smart contract responsible for managing ownership of specific TLDs such as `.eth`, and it handles the registration and management of subdomains under those domains. Users can register or renew domains according to the rules defined by the Registrar, which processes such registration requests. The Resolver is a smart contract that returns resources associated with a domain name, such as cryptocurrency wallet addresses or content hashes. Using the resolver specified in the Registry, various types of information about a domain can be queried. Therefore, the Registrar is responsible for domain registration, while the Resolver performs domain name resolution. The Registry, in turn, stores and manages domain-related information throughout this process.

Meanwhile, the domain name resolution process follows a series of steps. First, the ENS domain name to be queried is normalized off-chain according to the UTS46 standard, which removes case sensitivity and handles invalid characters. A 32-byte namehash is then generated. Using this namehash, a query is sent to the ENS Root Registry to obtain the address of the resolver associated with the name. Finally, the resolver contract is queried using the returned address to retrieve the desired information, such as cryptocurrency addresses or text records, thereby completing the resolution process.

ENS enables domain management without reliance on central authorities such as ICANN. Users retain full control over domain ownership through private keys, and the cryptographic properties of the blockchain offer protection against traditional security threats such as DNS spoofing and cache poisoning, while also providing strong resistance to censorship. However, ENS continues to face clear limitations related to cost, scalability, compatibility, and vulnerability to misuse.

First, registering and renewing an ENS domain incurs Ethereum gas fees, and an annual cost ranging from approximately $5 to $640 is applied depending on the length of the domain name [92]. In particular, when the Ethereum network is congested, gas fees can increase significantly, reducing accessibility. Moreover, Ethereum's low TPS capacity also poses a barrier to scalability.

Since ENS was originally developed to improve the usability of Ethereum wallet addresses, compatibility with traditional DNS was not considered in its design. Consequently, it faces significant challenges in evolving into a mainstream DNS solution for general users. In addition, limited native browser support requires users to configure settings or install extensions, which creates an additional barrier to adoption.

Like other BDNSs, ENS is also exposed to emerging threats such as domain squatting, in which malicious actors or organizations register names in bad faith, and the association of domains with harmful websites. According to research, squatting behavior has been

observed since the initial launch of ENS, and at the time of the study, an estimated 7.2% to as much as 67.3% of registered domains were suspected to be cases of squatting, including explicit and typosquatting cases [93].

Since performance is tied to Ethereum, our assessment is based on the Ethereum network. According to empirical measurements in [94], as of 10 October 2025, the average block interval has held steady at roughly 12 s with little variation over the past year. For TPS, ref. [95] reports a recent average of about 1,600,000 transactions per day. Dividing by the number of seconds in a day (86,400) yields approximately 18.5 tx/s.

In summary, the ENS is a blockchain-based decentralized naming system that enhances the readability of cryptocurrency addresses and facilitates digital identity management and content access in Web3 environments. It systematically manages domain registration and data resolution through smart contracts such as the Registry, Registrar, and Resolver, ensuring security and censorship resistance without the involvement of centralized authorities. However, challenges remain for widespread adoption and scalability, including high gas fees on Ethereum, limited scalability, lack of compatibility with traditional DNSs, and issues such as domain squatting and misuse. Real-world applications of ENS are as follows:

- Major Web3 marketplaces such as OpenSea, Rarible, and Uniswap support user identification and asset association through ENS addresses.
- ENS is integrated with wallet services such as Metamask and Rainbow, improving the usability of address input.
- Since ENS-based domains can be traded as Non-Fungible Token (NFT)s, they have gained popularity through community-driven auction systems.

7.4. Unstoppable Domains

Unstoppable Domains [96], like ENS, belongs to the smart-contract-based model (Type II). It operates on the Ethereum and Polygon networks via a proprietary registry protocol and maps complex cryptocurrency wallet addresses to human-readable URLs. This allows users to send and receive cryptocurrency through simplified domain names without needing to know long public key addresses. As of October 2024, Unstoppable Domains supports 23 blockchain networks and ecosystems, including Ethereum, EVM compatible chains, and major Layer 2 (L2) networks [97].

In addition to cryptocurrency address mapping, Unstoppable Domains provides decentralized website hosting functionality. This allows users to access their own services in a censorship-resistant manner. It also functions as a digital identity platform, enabling users to maintain a consistent digital identity across multiple blockchain networks.

Unstoppable Domains operates on the Crypto Name Service (CNS), a set of Ethereum smart contracts. CNS specifies the rules that govern how blockchain domains are created and used. Accordingly, each CNS domain is issued as a non-fungible token (NFT) compliant with ERC-721 [98], Ethereum's standard for creating and managing unique digital assets. Similar to the ENS, the CNS comprises two smart contracts: a registry contract and a resolver [99].

The registry contract is responsible for issuing new cryptocurrency domains and subdomains, as well as managing domain ownership. There is no central administrator in this system, so no entity, including Unstoppable Domains, can manage or transfer a domain without the user's consent. The resolver contract is used to interpret domains, translate human-readable names into cryptocurrency addresses, and store domain records.

More recently, Unstoppable Domains has developed a new registry and set of smart contracts called the Unstoppable Name Service (UNS). Unlike the CNS, the UNS uses a single registry smart contract to handle both domain ownership management and domain

record storage. Its structure is similar to that of the CNS in that domains are owned in an irrevocable manner. Domains do not require renewal and cannot be reclaimed by Unstoppable Domains. Once issued, users have full control over their domains.

In 2024, a significant integration feature called Metamask Snap was released, enabling users to directly resolve Unstoppable domains within the Metamask wallet. This integration significantly expanded MetaMask's name resolution capabilities by allowing users to replace complex alphanumeric wallet addresses with easily memorable domain names across 23 blockchain networks [100].

Since 2021, Ethereum's Layer-2 expansion has offered free minting via Polygon (Polygon PoS) [101], and support was subsequently extended to Base (Optimistic Rollup). According to the minting-chain guide [102], TLDs are currently minted across Ethereum (L1), Polygon (L2), Base (L2), Solana (L1), and Sonic. From a performance standpoint, the Base metrics page [103] indicates that, as of 11 October 2025, the block time is about 2 s and throughput is roughly 90 tx/s. For Polygon, Polygonscan [104] shows, on the same date, a block time of about 2 s and a throughput of roughly 55 tx/s.

Unlike traditional domain name services, cryptocurrency domains obtained through Unstoppable Domains are not leased but owned permanently. Using such a domain, users can transfer, update, and connect their domains to other services without relying on Unstoppable Domains.

However, while permanent ownership aligns with goals such as eliminating renewal fees, enhancing decentralization, and strengthening censorship resistance, it also introduces operational challenges. In particular, it increases the risk of domain squatting and leads to the accumulation of dead domains, which can degrade the overall efficiency of domain usage. In particular, as the system operates over time, it inherently faces a flexibility conflict, where the number of meaningful domain names available for registration may become extremely limited.

Another limitation is that it does not support the types of records commonly used in traditional DNS, making it impossible to assign aliases to domain names—for example, linking blog.example.com to example.com using a CNAME record. In addition, it does not directly support the same types of records used in traditional DNS, and its reliance on custom record formats presents a limitation in terms of legacy compatibility, reducing accessibility for early users.

Compatibility issues also remain a critical challenge. According to ICANN, as of 28 March 2022, over 2.1 million domains had been registered with Unstoppable Domains [105], but these domains are not fully integrated with the existing DNS infrastructure. As a result, users are required to use specialized browsers or browser extensions to access such domains. Real-world usage examples is provided below.

- **Integration with MetaMask**: In October 2024, Unstoppable Domains launched MetaMask Snap, enabling domain resolution directly within the MetaMask wallet. This integration impacts over 4 million Unstoppable Domains users and provides cross-chain functionality across 23 different blockchain networks.
- **Use of Cloudflare's Distributed Web Resolver**: Unstoppable Domains utilizes Cloudflare's distributed web resolver to support blockchain domain name resolution. This allows users to access decentralized websites by entering .crypto domains just like .com addresses, regardless of the browser they use [30].

7.5. Blockzone

Blockzone [16] is a prototype of a blockchain-based DNS (BDNS) that implements domain registration, renewal, and ownership-management logic via Ethereum smart contracts and verifies record updates using a PBFT-family consensus. It aims to mitigate threats aris-

ing from the centralized architecture of conventional DNS, such as SPOF, cache poisoning, and spoofing, and is organized into four layers: data, index, consensus, and network.

The data layer logs DNS records on the blockchain as transactions, while large payloads are stored off-chain in a distributed, IPFS-based storage. The index layer employs a Bloom filter-based indexing structure to accelerate lookups of domain-name-to-record mappings [103]. The consensus layer achieves agreement among nodes using a PBFT variant to provide fast finality, while the network layer handles interoperability between blockchain nodes and traditional DNS resolvers to maintain backward compatibility. Blockzone also deploys smart contracts for ownership and relationship management (e.g., Ownership and Relationship contracts), and transaction processing incurs Ethereum gas fees.

The authors reported, based on a prototype implementation and experiments, an improvement in parsing and authentication efficiency compared to DNSSEC (e.g., 37.8%), and high throughput for the PBFT-variant consensus (e.g., 10,900 TPS) [26]. However, Blockzone requires operating on-chain (Ethereum), off-chain (IPFS/indexing), and gateway components together, which increases system complexity. Due to PBFT's communication overhead among nodes, scalability may be constrained as the deployment grows. Additionally, operators must shoulder both Ethereum gas-price volatility and the operational costs of running PBFT nodes, and the design depends on IPFS availability and performance. Therefore, Blockzone is more appropriately regarded as a research-oriented reference design rather than a solution ready for large-scale commercial adoption.

7.6. Handshake

Unlike earlier models that leverage existing blockchains, Handshake, which falls into Type III as a standalone or hybrid model, is a blockchain-based protocol launched in 2020 that aims to provide decentralized management of the DNS root zone [28]. Similar to Namecoin, its primary goal is to replace the centralized domain management system led by ICANN, thereby enhancing censorship resistance and user sovereignty. Handshake adopts Bitcoin's SHA-256-based PoW algorithm but introduces Proof of Work Name Service(PoWNS) to implement a structure specialized for domain auctions and management.

By storing its own TLD information on the blockchain, domain ownership can be transparently recorded and protected against tampering without the need for a central authority. This allows `.hs` domains to be registered and managed without centralized control by governments or corporations. Additionally, by directly integrating TLS certificates through the DANE protocol, the system eliminates dependency on Certificate Authorities.

To ensure fairness in domain auctions, Handshake adopts the Vickrey auction mechanism, in which the highest bidder pays the second-highest bid price. In addition, a lightweight resolver called Handshake Naming System Daemon (HNSD) is introduced, which enables resolution of Handshake names without running the full node, Handshake Daemon (HSD). This enables efficient operation even in resource-constrained environments and allows users to resolve Handshake domains `.hs` that are not natively supported by standard browsers [28].

In Handshake, TLDs managed by ICANN are designated as a reserved zone because, if the rightful owner does not claim the corresponding TLD on the Handshake blockchain, a root chain conflict between the two systems may inevitably occur. Similarly to Namecoin, Handshake also faces cybersquatting because its anonymity and decentralized nature allow anyone to buy TLDs via HNS-denominated auctions, which has increased cases of malicious actors preemptively acquiring many TLDs and trying to resell them at high prices [8]. In addition, since most browsers do not natively support Handshake domains, users must install separate resolvers or browser extensions, which hinders mainstream adoption.

Regarding performance, the official protocol documentation [106] states that Handshake uses a 10-min block interval and that PoW difficulty is recomputed on every block to target 144 blocks per day. Actual deployment cases of Handshake are presented below:

- Namebase, which provides Handshake-based TLD auction and management functionality and can be integrated with user wallets.
- NextDNS, which recognizes Handshake DNS and demonstrates the potential for naming beyond web browsers.
- A custom DNS resolver that, while experimental, supports interoperability with Web2 environments

7.7. Stacks (Blockstack)

Within Type III, a representative hybrid-layer approach is Blockstack, a decentralized naming and storage system that operates on the Bitcoin blockchain and aims to replace centralized Internet infrastructure [34]. It was rebranded to Stacks in 2020. It emphasizes user sovereignty and data privacy, and, through the Blockchain Name System (BNS), it associates human-readable names with encrypted data, while BNS adopts a DNS-like structure, it manages names on-chain without a centralized root, thereby improving censorship resistance and tamper-evidence. Users obtain a digital identity (Blockstack ID) and store encrypted data through Gaia, a decentralized storage system.

According to [34,107], the architecture comprises four layers: the blockchain layer, the virtualchain layer, the Atlas network, and Gaia. The blockchain layer leverages Bitcoin; name operations (e.g., `name_new`, `name_update`) are embedded in Bitcoin transactions and ordered by Bitcoin consensus. The virtualchain is a new state machine that interprets these operations above the base chain, enabling Stacks-specific logic without requiring base-layer nodes to understand it, thus preserving compatibility.

The Atlas network serves as a routing layer that stores and disseminates zone files over a P2P overlay. Zone files retain a DNS-like structure (e.g., TXT/URI records) but typically point to encrypted data rather than directly to IP addresses. Gaia integrates storage backends such as AWS S3, the InterPlanetary File System (IPFS) [108], and local disks, and keeps data encrypted under user control; integrity is anchored by hashes on the blockchain.

Several limitations have been noted [107]. First, because Bitcoin provides only transaction ordering at relatively low bandwidth, it can become a bottleneck compared to Atlas. Operations such as registration and updates inherit Bitcoin's average 10-min block interval, which degrades timeliness for real-time applications and impacts developer and user experience. Moreover, the Atlas peer network is most effective when datasets are small and globally indexed; scalability can be strained for large or highly dynamic data.

Report [109] further observes that with a 1 MB block size and an average transaction size of 380.04 bytes, combined with a 10-min block interval, Bitcoin's throughput is only about 4.6 tx/s. Because Blockstacks (now Stacks) rides on Bitcoin's cadence, practical throughput is similarly constrained (roughly single-digit tx/s).

Meanwhile, as explained earlier, the Blockstack ecosystem was rebranded as *Stacks* after 2020. Through the Stacks 2.0 (Stacks V2) upgrade, its core operating model evolved from a design centered on a "virtual blockchain" that directly embeds all transactions on Bitcoin. In the new model, Bitcoin serves as the final settlement layer. Transactions and logic are processed on a separate Stacks chain and are periodically anchored back to Bitcoin [110].

Specifically, Stacks 2.0 employs Proof of Transfer (PoX) [111], under which leader election is performed on the Bitcoin chain, while actual block production and state evolution occur on the connected Stacks chain, so that many Stacks transactions are settled on Bitcoin

as a "single hash." Consequently, transaction throughput and application execution can scale on the Stacks layer, while finality is maintained in alignment with Bitcoin's block production cadence. In addition, Stacks 2.0 introduces Clarity-based smart contracts in earnest, enabling application logic that is predictable and audit-friendly while remaining anchored to Bitcoin's security.

Additionally, on the naming side, Stacks V2 replaces the V1 approach of "first-order name operations" with a smart contract implementation of BNS deployed at genesis, thereby formalizing naming operations as contract calls. This implies that name registration, renewal, and updates go beyond simple transaction embedding and can be more tightly coupled with application-layer policies (e.g., renewal rules, access control, additional metadata validation). However, despite these protocol/implementation evolutions, the lack of native DNS compatibility at the web-browser and OS levels (e.g., support for standard RR types and the absence of a native resolution path) and the need for additional client-side configuration (e.g., reliance on extensions/gateways) may persist as barriers to adoption. Practical application cases are summarized below:

- DApp authentication and ID mapping with user-owned `.id` names.
- Integration of names with blogging/chat/file-storage in the Stacks ecosystem and name-browsing and search interfaces.

7.8. Summary of Findings from the Analysis of Currently Available BDNS Projects

This analysis demonstrates that BDNS can minimize the involvement of centralized authorities and enhance data integrity through decentralized governance. In addition, by leveraging the transparency and immutability of blockchain, the risk of data tampering is reduced and overall trust is enhanced. These characteristics significantly improve the censorship resistance and security of BDNS, while also contributing to the protection of user privacy. Furthermore, it suggests the potential to enhance the autonomy and openness of the Internet by enabling domain management without being subject to uniform global regulations or centralized control.

Through this, it has been confirmed that while BDNS offers a promising way to address key limitations of the traditional DNS, such as enabling greater autonomy and openness without being bound by uniform global regulations, it must still overcome practical challenges related to compatibility, accessibility, and economic cost in order to evolve into a mainstream technology. Key future directions for BDNS include the adoption of Layer 2 solutions, enhanced integration with web browsers, more realistic domain management cost structures, the development of fair domain allocation mechanisms, improved support for mobile environments, and the enhancement of user interfaces and overall user experience.

In addition, the adoption of standardized protocols and alignment with international regulatory frameworks are essential for improving the reliability and adoption of BDNS technologies. In particular, it is necessary to explore the potential of hybrid approaches that combine blockchain technology with traditional DNSs.

Before presenting the summary table of the findings organized in this section, we provide a detailed explanation of the evaluation criteria in Table 6, excluding those that are self-explanatory by name. In this evaluation, Blockzone was excluded from the tables because most criteria were either not applicable or too difficult to assess, and because it remains in an experimental stage without implementation or deployment. The analysis results of major BDNS projects, including Namecoin, Handshake, Blockstack, ENS, Unstoppable Domains, and EmerDNS, are summarized in Tables 7 and 8.

Table 6. Description of evaluation criteria for blockchain domain systems.

Criterion	Description
Compatibility with Traditional DNS	Indicates whether the blockchain domain is compatible with the existing ICANN root-based DNS infrastructure, allowing domain resolution without the need for additional plugins or configuration, and functioning like a traditional DNS.
TLS/SSL Support	Indicates whether the domain supports secure communication, such as HTTPS, either by obtaining an TLS/SSL certificate or through alternative methods like DANE.
Decentralized Storage Integration	Indicates whether web content or data can be stored and hosted on decentralized storage systems such as IPFS in association with the domain name, thereby enabling the operation of censorship-resistant websites.
Standard DNS Record Support	Indicates whether the name system supports various record types commonly used in traditional DNS (e.g., A, MX, TXT), allowing functionalities such as email configuration via MX records or the use of TXT records, similar to traditional domains.
Multi-Namespace Support	Indicates whether the system supports not only a fixed domain suffix (TLD) but also the operation of multiple domain zones or the ability for users to create new domain namespaces, such as custom TLDs.
Integration with Existing DNS Domains	Indicates whether existing traditional DNS domain names can be linked to or imported into the blockchain-based name system for use.
Permissionless Registration	Indicates whether anyone can register a domain solely through a blockchain transaction without requiring centralized approval. It reflects whether domain names can be freely registered according to protocol rules, without involvement from specific companies or intermediaries.
Censorship Resistance	Indicates the extent to which domains are protected from arbitrary deletion or access restrictions by governments or third parties. Specifically, it assesses whether domain records are stored in a decentralized manner without a central authority, such that they cannot be revoked or modified without the consent of the original owner.
Native Support in Major Browsers	Indicates whether the domain can be accessed directly in mainstream web browsers such as Chrome, Firefox, and Safari without requiring additional plugins.
Dedicated Extension/App Support	Indicates whether official browser extensions or dedicated applications are provided to enable the resolution and use of the domain. In the absence of such official tools, users must manually configure compatible nodes or DNS settings, which reduces accessibility.
Permanent Ownership Available	Indicates whether a domain, once registered, can be owned permanently without expiration. This means the domain can be used indefinitely without incurring additional fees.
Initial Auction Mechanism	Indicates whether domain names are allocated through an auction mechanism rather than on a first-come, first-served basis. This enables price discovery for contentious names.
Domain Squatting Possibility	Indicates how susceptible the system is to domain squatting. Systems with low costs and permanent ownership are more prone to squatting, whereas recurring fees discourage unnecessary holdings.
Governance Structure (DAO)	Indicates whether the project adopts a decentralized autonomous organization (DAO) for operational and policy decisions, allowing community members to participate through voting.
Governance Structure (Community)	Indicates whether the project's development direction is openly driven by the community, such as having an open-source codebase and allowing protocol changes through consensus.
Average Block Generation Time	Indicates whether the blockchain network has a short block generation interval, enabling rapid confirmation of domain registration and renewal transactions.
TPS Scalability	Indicates whether the network can handle high transaction throughput, allowing domain registration and renewal operations to function smoothly without delays or fee spikes.

Table 7. Comparative Analysis of BDNS: Technical Architecture and Specifications.

Technical Criteria	**Namecoin**	**Handshake**	**Blockstack**
Average Block Generation Time	10 min	10 min	10 min
TPS Scalability	~7 TPS (low)	~5.5 TPS (theoretical)	~7 TPS (low)
Governance Structure	Community	Community	Community
Governance Decentralization	Yes (Open-source)	Yes (Community driven)	Yes (Open-source)
TLS/SSL Support	DANE-based verification using on-chain TLSA records	No native support (TLS managed off-chain)	No support (relies on traditional SSL)
MITM Attack Protection	Yes (on-chain cert pinning prevents CA MITM)	? (depends on client config)	No (no blockchain-based verification)
Decentralized Storage Integration	No (no IPFS integration)	No (naming only)	Yes (uses Gaia decentralized storage)
Standard DNS Record Support	Yes (supports A, MX, etc.)	Yes (via NS, DS; off-chain)	Partial (off-chain zone file)
Multi-Namespace Support	No (only `.bit`)	Yes (register any TLD)	Yes (namespaces programmable)
Subdomain Support	Limited (custom DNS req.)	Yes (traditional DNS mgmt)	Yes (via registrar scripting)
Technical Criteria	**ENS**	**Unstoppable Domains**	**EmerDNS**
Average Block Generation Time	12 s	2 s	~7.6 min
TPS Scalability	~18.5 TPS	~90 TPS (Base)	~0.005 TPS
Governance Structure	DAO	Centralized (Corporate)	Community
Governance Decentralization	Yes (ENS DAO)	No (Corporate decisions)	Yes (NVS structure)
TLS/SSL Support	No native CA; uses IPFS content hashes	Blockchain + IPFS records	EmerSSL (cert hash on blockchain)
MITM Attack Protection	Yes (DNSSEC & content addressing)	Yes (immutable IPFS hashes)	Yes (records immutable)
Decentralized Storage Integration	Yes (IPFS & Swarm)	Yes (native IPFS integration)	Yes (records point to decentralized storage)
Standard DNS Record Support	Yes (ENSIP-6 defined)	Yes (A, MX, TXT, etc.)	Yes (standard DNS records)
Multi-Namespace Support	No (mainly `.eth`)	Yes (`.crypto`, etc.)	Yes (`.coin`, `.bazar`, etc.)
Subdomain Support	Yes (unlimited via contract)	No (not supported)	Yes (supported)

Table 8. Comparative Analysis of BDNS: Usability, Economics, and Ecosystem.

Usability Criteria	**Namecoin**	**Handshake**	**Blockstack**
Compatibility with Traditional DNS	No (separate `.bit` zone)	Yes (uses root zone model)	No (separate BNS)
Integration with Existing DNS Domains	No	Yes (legacy TLDs claimable)	No
Native Support in Major Browsers	No (plugin required)	Yes (e.g., Opera)	No (Blockstack browser)
Dedicated Extension/App Support	Yes (ncdns, FreeSpeechMe)	Yes (Fingertip, etc.)	Yes (Blockstack browser)
Permissionless Registration	Yes	Yes (Vickrey auction)	Yes
Censorship Resistance	Yes (immutable records)	Yes (ownership on-chain)	Yes (blockchain ownership)
Personal Information Required	Wallet only	Wallet only	Wallet only
Permanent Ownership Available	No (renewal required)	No (2-year renewal)	No (renewal required)
Initial Auction Mechanism	No (First-come first-served)	Yes (Vickrey auction)	No
Domain Squatting Possibility	High (cheap & permanent)	Limited (auction model)	High (low cost)

Table 8. *Cont.*

Usability Criteria	ENS	Unstoppable Domains	EmerDNS
Compatibility with Traditional DNS	Yes (via DNSSEC import)	No (requires extensions)	Yes (via OpenNIC)
Integration with Existing DNS Domains	Yes (via DNSSEC proofs)	No	No
Native Support in Major Browsers	Yes (Brave, Opera)	Yes (Brave, Opera)	No (OpenNIC/Plugin)
Dedicated Extension/App Support	Yes (MetaMask, wallets)	Yes (Mobile App)	Yes (CLI/API)
Permissionless Registration	Yes	No (Corporate issued)	Yes
Censorship Resistance	Yes (non-revocable)	Yes (NFT-based)	Yes (key owner only)
Personal Information Required	Wallet only	Wallet & Email (No KYC)	No info required
Permanent Ownership Available	No (annual fees)	Yes (One-time payment)	Yes
Initial Auction Mechanism	No (deprecated)	No (Fixed price)	No
Domain Squatting Possibility	Limited (renewal fees)	High (permanent NFT)	High (low cost)

8. Future Direction

To propose future directions, Table 9 summarizes the pros and cons of traditional DNS and BDNS. Based on this comparison, the key future challenges and research directions for BDNS can be itemized as follows:

- **Scalability and performance.** Public blockchain throughput and synchronization overhead still limit BDNS from matching global DNS-scale query volumes. Future work should further explore performance-oriented designs such as alternative consensus mechanisms, sharding/DAG-based architectures, and hybrid on/off-chain data layouts.
- **Interoperability and coexistence with the DNS ecosystem**: For broad adoption, BDNS must integrate with existing DNS protocols, resolvers, and user applications with minimal friction. Promising directions include protocol-level compatibility (e.g., operating blockchain nodes as DNS servers while preserving standard DNS query flows) and migration pathways that do not disrupt legacy DNS operations.
- **Standardization and namespace-collision mitigation**: Conflicts across independent naming systems (e.g., TLD collisions) can cause user confusion and weaken deployability. Establishing shared naming rules and interoperation protocols is required to support stable coexistence between decentralized and centralized naming systems.
- **Smart-contract correctness and governance robustness**: Since domain ownership and record updates depend on smart contracts, even minor bugs can lead to irreversible asset loss. Future research should emphasize formal verification, systematic audits, and safer upgrade/governance mechanisms for on-chain name management.
- **Security hardening against emerging threats**: Beyond traditional DNS threats, BDNS introduces new risks (e.g., consensus attacks, key compromise, and infrastructure abuse). Continued work is needed on attack-resilient consensus/incentives, DNS-specific threat mitigation (e.g., cache-poisoning-aware defenses), and long-term adoption of post-quantum cryptography.
- **Usability, cost, and user accessibility**: Wallet-based authentication, transaction fees, and operational complexity remain major adoption barriers. Future systems should reduce friction via L2 scaling, more efficient fee models, and user-friendly resolution flows that make decentralized naming as transparent and intuitive as traditional DNS.

Table 9. Comprehensive Comparison of Pros and Cons Between Traditional DNS and BDNS.

Aspect		Traditional DNS	BDNS
Security	Pros	Decades of operational experience with DNS; security extensions like DNSSEC and DoT/DoH are available to improve integrity and privacy.	Uses an immutable ledger to store DNS records, preventing tampering; no SPoF, which enhances resistance to attacks.
	Cons	Vulnerable to attacks such as DDoS and cache poisoning due 1 to centralized points of failure; lacks transparency in handling DNS data changes.	It introduces new attack vectors such as 51% blockchain attacks or smart contract bugs, and offers no recovery mechanism if a private key is lost or compromised, meaning a domain can be irreversibly stolen with no central authority to intervene.
Usability	Pros	Domain name resolution is transparent and automatic for users with standard internet setups; mature ecosystem (browsers, OS, and network providers) natively supports DNS.	Domain management relies on third-party registrars and DNS hosting, which can introduce delays (e.g., DNS propagation times) and occasional configuration errors
	Cons	Domain management relies on third-party registrars and DNS hosting, which can introduce delays (e.g., DNS propagation times) and occasional configuration errors; users have no direct ownership of domains.	Higher technical barrier to entry—users must manage cryptographic keys and use special resolvers or browser plugins to access 6 domains; not widely supported by default in browsers/OS, which complicates usage for non-technical users.
Scalability	Pros	Proven to scale to billions of queries per day worldwide through hierarchical caching and distribution—the system handles global traffic with low latency.	Each node on the blockchain can resolve domain queries from its local copy of the ledger, potentially distributing query load across the network (no dependence on a single hierarchy for resolution).
	Cons	Structure has some inherent scalability limits (e.g., a finite number of root servers and potential bottlenecks under extreme load).	Since public blockchains handle only limited TPS, query resolution is slower than traditional DNS and fully storing DNS data on-chain is resource-intensive.
Governance	Pros	Organizations like ICANN provide well-established governance and root management, using multi-stakeholder processes for disputes and updates, while clear legal frameworks address trademark conflicts and abuse.	BDNS projects use a decentralized governance model in which protocol changes require community consensus often through voting or DAO mechanisms preventing any single entity from seizing domains and making management more democratic.
	Cons	Governance is centralized, so decisions are made by a limited set of authorities; this centralized decision-making can sometimes be slow, politically influenced, or not fully transparent to the broader community.	Without a central authority, disputes or abuses lack clear arbitration paths, as legal processes are hard to apply; evolving governance may cause forks or instability, risking namespace fragmentation.
Resilience to Censorship	Pros	Central oversight lets authorities block or remove malicious domains for law-enforcement takedowns - more a control feature than a resilience benefit.	Because no central authority can remove a domain, BDNS is highly censorship-resistant: as long as the network is reachable, records keep resolving, making it resilient to political or institutional suppression.
	Cons	Vulnerable to censorship: authorities or registries can delete or suspend domains, ISPs can block queries, and authoritarian regimes have censored content via DNS manipulation or shutdowns.	The same censorship resistance makes malicious domains like phishing or malware hard to take down, enabling abuse, and while the blockchain itself is censorship-proof, governments can still target it by blocking access to nodes or related traffic.
Cost	Pros	Domain registration costs are relatively low and predictable; the ongoing fees create a modest barrier that discourages mass squatting by making it expensive to hold many domains without use. DNS query usage is free for end users (costs are borne by operators and ISPs).	Generally, blockchain naming systems charge just a one-time transaction fee (and sometimes a small purchase price) with no annual renewals, so ownership remains permanent until the holder decides to transfer it.
	Cons	Domains require ongoing renewal fees or they expire, popular names command high secondary-market prices, and organizations must cover authoritative-server costs.	Cheap registration encourages cybersquatting, yet high blockchain transaction fees can make domain actions costly, and running a full node for trustless resolution adds hardware and bandwidth expenses for users.
Interoperability	Pros	Ubiquitous standard—traditional DNS is universally supported by existing internet infrastructure, ensuring any compliant device or application can resolve DNS names.	There are ongoing efforts to bridge blockchain domains with the legacy DNS (e.g., integrations that allow resolving blockchain domains via standard DNS resolvers), which could improve interoperability in the future.
	Cons	Not designed to interoperate with fundamentally different naming systems; it cannot natively resolve blockchain-based domains or other alternate roots without special gateways or modifications.	Isolated from traditional DNS, and different BDNS projects do not interoperate, risking name collisions and a fragmented user experience when multiple decentralized naming systems coexist without coordination.

In summary, the future agenda for BDNS can be organized along six axes: scalability, interoperability, standardization, reliability, security, and usability. Addressed systemat-

ically, these challenges could position BDNS as a practical alternative that compensates for structural weaknesses in today's DNS. In particular, in IoT environments where deploying traditional DNS security extensions (e.g., DNSSEC, DoH, DoT) is challenging, BDNS can evolve into a more secure and scalable naming infrastructure by combining with lightweight clients.

9. Conclusions

In this paper, rather than presuming BDNS to be a wholesale replacement, this study conducted a comparative analysis that centers on its deployability in real operational environments and the conditions under which it attains practical value while coexisting with the legacy DNS. To this end, we (i) summarized the trust and operational structure of traditional DNS and its points of weakness; (ii) analyzed how BDNS's consensus-based state recording and public-key ownership model shift the underlying trust assumptions; and (iii) established a common set of criteria, covering the record model and finality, TTL semantics, additional overhead along resolution paths, reliance on gateways and off-chain components, and interoperability and migration scenarios, and compared representative systems against these dimensions. In doing so, this study does not simply assert that BDNS is more secure; rather, it characterizes, from an end-to-end resolution-path perspective, the specific points where trust is strengthened and the points at which new operational burdens emerge.

Based on the comparison, we confirm that BDNS provides a trust model fundamentally different from that of the legacy DNS. Whereas the latter relies on a centralized hierarchical structure (for example, registries, resolvers, and root operators), BDNS maintains domain state on a distributed ledger and validates domain updates through network consensus and cryptographic signatures. As a result, the difficulty of unauthorized record tampering or hijacking increases substantially, and failure modes that converge on central points, such as registry compromise or privilege abuse, are mitigated. In particular, under the threat model synthesized in this study, BDNS demonstrates the potential to reduce core vulnerabilities of traditional DNS, such as cache poisoning, by natively embedding integrity of name state and ownership verification.

Another key advantage of BDNS is censorship resistance and mitigation of SPOF. In traditional DNS, policy or technical actions directed at central registries or authoritative infrastructure can directly result in service blocking or seizure. By contrast, in BDNS, control is distributed across many nodes, making unilateral actions that do not satisfy network consensus rules structurally difficult. However, as this study's comparative analysis indicates, the actual resolution path often depends on gateways, RPC endpoints, browser extensions, and off-chain indexing and availability layers, where operational and access infrastructure can re-concentrate. In other words, ledger-level decentralization alone does not automatically realize the benefits of BDNS. The extent of performance, availability, and censorship resistance depends on how the end-to-end resolution path is designed and operated.

In environments where deploying traditional DNS security extensions is challenging, such as IoT and sensor networks, the BDNS model of integrity and ownership verification can serve as a promising alternative. However, because practical operability on resource-constrained devices is only briefly addressed in the main text, empirical evaluations, including lightweight verification (e.g., light clients) and gateway-based operations, remain important avenues for future work.

In conclusion, current BDNS implementations face practical limitations, including scalability, usability, the absence of standardization, and centralization of access infrastructure, and no single system can be regarded as having achieved a universally "un-

stoppable" domain service. Nevertheless, our literature review and comparative analysis identify (i) integrity and ownership verification afforded by consensus-based state recording, (ii) distributed control that structurally mitigates censorship and seizure risks, and (iii) new bottlenecks stemming from friction and operational complexity along the end-to-end resolution path.

Accordingly, BDNS research and development should converge not on complete replacement, but rather on interoperability, lightweight verification, key management and recovery, incentive-compatible and abuse-resistant economic design, and operational standardization, all premised on coexistence and phased transition. If this agenda advances systematically, BDNS can mature into a practical alternative that strengthens the global naming infrastructure against censorship and SPOF, and it can lay the groundwork for a more resilient naming substrate in IoT environments where the deployment of traditional DNS security extensions is constrained.

Author Contributions: Conceptualization, J.J. and S.P.; methodology, J.J.; software, J.J.; validation, J.J.; formal analysis, J.J.; investigation, J.J.; resources, S.P.; data curation, J.J.; writing—original draft preparation, J.J.; writing—review and editing, J.J.; visualization, J.J.; supervision, S.P.; project administration, S.P.; funding acquisition, S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Digital Innovation Hub project supervised by the Daegu Digital Innovation Promotion Agency(DIP) grant funded by the Korea government(MSIT and Daegu Metropolitan City) in 2025(No.25DIH-32/Development of a Vehicle Operation Data Recording and Trust Verification Platform Based on Micro-Blockchain Technology).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

ABI	Application Binary Interface
ARQ	Automatic Repeat Request
AXFR	Full zone transfer
BDNS	blockchain based DNS
BFT	Byzantine Fault Tolerance
C2	command-and-control
DANE	DNS-based Authentication of Named Entities
DApps	decentralized applications
DLT	distributed ledger technology
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
DoH	DNS over HTTPS
DoK	DNS over KCP
DoT	DNS over TLS
EIP	Ethereum Improvement Proposal
EMC	EmerCoin
ENS	Ethereum Name Service
ERC	Ethereum Improvement Proposal
EVM	Ethereum Virtual Machine
FTP	File Transfer Protocol
gTLDs	generic TLDs

IANA	Internet Assigned Numbers Authority
ICANN	Internet Corporation for Assigned Names and Numbers
IDS	intrusion detection systems
IoT	Internet of Things
IXFR	Incremental Zone Transfer
L2	Layer 2
MEV	transaction-ordering manipulation
MITM	Man-In-The-Middle
NVS	Name-Value Storage
PBFT	Practical Byzantine Fault Tolerance
PoA	Proof of Authority
PoS	Proof of Stake
PoW	Proof of Work
rDNS	reverse DNS
SDK	Software Development Kit
SLD	Second-Level Domains
SPOF	Single Points of Failure
SPV	PAyment Verification
SRI	Stanford Research Institute
TLD	Top-Level Domains
TOCTOU	time-of-check-to-time-of-use
TPS	Transactions per Second
TTL	Time-to-Live
UTXO	Unspent Transaction Output

References

1. Arends, R.; Austein, R.; Larson, M.; Massey, D.; Rose, S. DNS Security Introduction and Requirements. Technical Report RFC 4033, Internet Engineering Task Force (IETF). 2005. Available online: https://datatracker.ietf.org/doc/html/rfc4033 (accessed on 21 April 2025).
2. Kristoff, J.; Wessels, D. DNS Transport over TCP—Operational Requirements. Technical Report RFC 9210, Internet Engineering Task Force (IETF). 2022. Available online: https://datatracker.ietf.org/doc/html/rfc9210 (accessed on 21 April 2025).
3. Hoffman, P.; McManus, P. DNS Queries over HTTPS (DoH). Technical Report RFC 8484, Internet Engineering Task Force (IETF). 2018. Available online: https://datatracker.ietf.org/doc/html/rfc8484 (accessed on 21 April 2025).
4. Vixie, P.; Thomson, S.; Rekhter, Y.; Bound, J. Dynamic Updates in the Domain Name System (DNS UPDATE). Technical Report RFC 2136, Internet Engineering Task Force (IETF). 1997. Available online: https://datatracker.ietf.org/doc/html/rfc2136 (accessed on 21 April 2025).
5. Hoffman, P.; Schlyter, J.; Kumari, W. The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA. Technical Report RFC 6698, Internet Engineering Task Force (IETF). 2012. Available online: https://datatracker.ietf.org/doc/html/rfc6698 (accessed on 21 April 2025).
6. Bagnulo, M.; Sullivan, A.; Huitema, C.; Thaler, D. DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers. Technical Report RFC 6147, Internet Engineering Task Force (IETF). 2011. Available online: https://datatracker.ietf.org/doc/html/rfc6147 (accessed on 21 April 2025).
7. Hesselman, C.; Droms, R.; van der Veen, V.; van Eijk, P.; Banchs, A.; Moura, G.C.M.; Jansen, J.; de Vries, W.B.; van Rijswijk-Deij, R. The DNS in IoT: Opportunities, Risks, and Challenges. *IEEE Internet Comput.* **2020**, *24*, 23–32. [CrossRef]
8. Giamouridis, G.; Kang, B.; Aniello, L. Blockchain-based DNS: Current Solutions and Challenges to Adoption. In Proceedings of the 6th Distributed Ledger Technologies Workshop (DLT 2024), Turin, Italy, 14–15 May 2024; CEUR Workshop Proceedings, Volume 3791, pp. 1–18. Available online: https://ceur-ws.org/Vol-3791/paper16.pdf (accessed on 21 April 2025).
9. Saige. From HOSTS.TXT to DNS: The Story of Internet Addressing. 2025. Available online: https://somedomain.co.uk/from-hosts-txt-to-dns-the-story-of-internet-addressing/ (accessed on 21 April 2025).
10. Mockapetris, P.V.; Dunlap, K.J. Development of the Domain Name System. In Proceedings of the ACM SIGCOMM '88 Conference on Communications Architectures and Protocols, Stanford, CA, USA, 16–18 August 1988; pp. 123–133. [CrossRef]
11. Mockapetris, P. Domain Names: Concepts and Facilities. Technical Report RFC 882, Internet Engineering Task Force (IETF). 1983. Available online: https://datatracker.ietf.org/doc/html/rfc882 (accessed on 21 April 2025).

12. Mockapetris, P. Domain Names: Implementation Specification. Technical Report RFC 883, Internet Engineering Task Force (IETF). 1983. Available online: https://datatracker.ietf.org/doc/html/rfc883 (accessed on 21 April 2025).
13. Mockapetris, P. Domain Names—Concepts and Facilities. Technical Report RFC 1034, Internet Engineering Task Force (IETF). 1987. Available online: https://datatracker.ietf.org/doc/html/rfc1034 (accessed on 21 April 2025).
14. Mockapetris, P. Domain Names—Implementation and Specification. Technical Report RFC 1035, Internet Engineering Task Force (IETF). 1987. Available online: https://datatracker.ietf.org/doc/html/rfc1035 (accessed on 21 April 2025).
15. IBM. What is DNS (Domain Name System)? 2024. Available online: https://www.ibm.com/think/topics/dns (accessed on 21 April 2025).
16. Ohta, M. Incremental Zone Transfer in DNS. Technical Report RFC 1995, Internet Engineering Task Force (IETF). 1996. Available online: https://datatracker.ietf.org/doc/html/rfc1995 (accessed on 21 April 2025).
17. Vixie, P. A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY). Technical Report RFC 1996, Internet Engineering Task Force (IETF). 1996. Available online: https://datatracker.ietf.org/doc/html/rfc1996 (accessed on 21 April 2025).
18. Agarwal, S.; Pramanick, S.; Bhandari, N.; Usha, G. A Case Study Solution to DNS Cache Poisoning Attacks. *Int. J. Adv. Res. Basic Eng. Sci. Technol. (IJARBEST)* **2017**, *3*, 91–97. Available online: https://ijarbest.com/conference/spcl36/1205 (accessed on 21 April 2025).
19. Cloudbric. The Most Notorious DDoS Attacks in History—2021 Update. 2021. Available online: https://www.cloudbric.com/the-most-notorious-ddos-attacks-in-history-2021-update/ (accessed on 21 April 2025).
20. Wei, L.; Heidemann, J. Whac-A-Mole: Six Years of DNS Spoofing. *arXiv* **2020**, arXiv:2011.12978v1.
21. Kim, T.H.; Reeves, D. A Survey of Domain Name System Vulnerabilities and Attacks. *J. Surveill. Secur. Saf.* **2020**, *1*, 34–60. [CrossRef]
22. Fältström, P.; Hoffman, P.; Costello, A. Internationalizing Domain Names in Applications (IDNA). Technical Report RFC 3490, Internet Engineering Task Force (IETF). 2003. Available online: https://datatracker.ietf.org/doc/html/rfc3490 (accessed on 21 April 2025).
23. DNSSEC Deployment Report. 2025. Available online: https://rick.eng.br/dnssecstat/ (accessed on 12 December 2025).
24. Lindenthal, T. Monocentric cyberspace: The primary market for internet domain names. *J. Real Estate Financ. Econ.* **2018**, *57*, 152–166. [CrossRef]
25. Foundation, V.; Group, B.C. VeChain Whitepaper 3.0: Web3 for Better. 2023. Available online: https://files.vechain.org/vechaincontainer/vechain-whitepaper-3-0.pdf (accessed on 2 May 2025).
26. Wang, W.; Hu, N.; Liu, X. BlockZone: A Blockchain-Based DNS Storage and Retrieval Scheme. In *Lecture Notes in Computer Science, Proceedings of the Artificial Intelligence and Security (ICAIS 2019)*; Springer: Cham, Switzerland, 2019; Volume 11635, pp. 155–166. [CrossRef]
27. Namecoin Project. Namecoin. 2025. Available online: https://www.namecoin.org/ (accessed on 21 April 2025).
28. Handshake Project. Handshake White Paper. 2018. Available online: https://handshake.org/files/handshake.txt (accessed on 21 April 2025).
29. Ethereum Name Service. What is the Ethereum Name Service? 2025. Available online: https://docs.ens.domains/learn/protocol/ (accessed on 21 April 2025).
30. Unstoppable Domains. Unstoppable Domains First to Bring Blockchain Domain Access to Mainstream. 2021. Available online: https://www.businesswire.com/news/home/20210216005342/en/Unstoppable-Domains-First-to-Bring-Blockchain-Domain-Access-to-Mainstream (accessed on 21 April 2025).
31. Emercoin. EmerDNS—Blockchain Service. 2025. Available online: https://emercoin.com/en/emerdns/ (accessed on 21 April 2025).
32. Casino, F.; Lykousas, N.; Katos, V.; Patsakis, C. Unearthing malicious campaigns and actors from the blockchain DNS ecosystem. *Comput. Commun.* **2021**, *179*, 217–230. [CrossRef]
33. Kalodner, H.A.; Carlsten, M.; Ellenbogen, P.; Bonneau, J.; Narayanan, A. An Empirical Study of Namecoin and Lessons for Decentralized Namespace Design. In Proceedings of the 14th Workshop on the Economics of Information Security (WEIS), Delft, The Netherlands, 22–23 June 2015; pp. 1–23. Available online: https://www.cs.princeton.edu/~arvindn/publications/namespaces.pdf (accessed on 21 April 2025).
34. Ali, M.; Nelson, J.; Shea, R.; Freedman, M.J. Blockstack: A Global Naming and Storage System Secured by Blockchains. In Proceedings of the Proceedings of the 2016 USENIX Annual Technical Conference (USENIX ATC 16), Denver, CO, USA, 22–24 June 2016; pp. 181–194.
35. Patsakis, C.; Alepis, E.; Douligeris, C. Unravelling Ariadne's Thread: Exploring the Threats of Decentralised DNS. *IEEE Access* **2020**, *8*, 118559–118571. [CrossRef]

36. Randall, A.; Hardaker, W.; Voelker, G.M.; Savage, S.; Schulman, A. The Challenges of Blockchain-Based Naming Systems for Malware Defenders. In Proceedings of the 2022 APWG Symposium on Electronic Crime Research (eCrime), Virtual, 30 November–2 December 2022; pp. 1–14. Access online: https://dblp.org/rec/conf/ecrime/RandallHVSS22 (accessed on 21 April 2025).
37. Xia, P.; Wang, H.; Yu, Z.; Liu, X.; Luo, X.; Xu, G.; Tyson, G. Challenges in Decentralized Name Management: The Case of ENS. In *Proceedings of the 22nd ACM Internet Measurement Conference (IMC), Nice, France, 25–27 October 2022*; Association for Computing Machinery: New York, NY, USA, 2022; pp. 65–82. [CrossRef]
38. Ito, D.; Takata, Y.; Kumagai, H.; Kamizono, M. Investigations of Top-Level Domain Name Collisions in Blockchain Naming Services. In *Proceedings of the ACM Web Conference 2024 (WWW '24), Singapore, 13–17 May 2024*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 2926–2935. [CrossRef]
39. Muzammil, M.; Wu, Z.; Harisha, L.; Kondracki, B.; Nikiforakis, N. Typosquatting 3.0: Characterizing Squatting in Blockchain Naming Systems. *arXiv* **2024**, arXiv:2411.00352v1. [CrossRef]
40. Banday, M.T. Recent Developments in the Domain Name System. *Int. J. Comput. Appl.* **2011**, *31*, 18–25. [CrossRef]
41. Root Server Technical Operations Association. Root Server System. 2025. Available online: https://root-servers.org/ (accessed on 21 April 2025).
42. Revankar, S. TLD Statistics by Facts, Usage, Generic, Country, Market Share, Rank, Trust, Registration and Renewal Prices. 2025. Available online: https://www.coolest-gadgets.com/tld-statistics/ (accessed on 21 April 2025).
43. Q-Success. Usage Statistics and Market Shares of Top Level Domains for Websites. 2025. Available online: https://w3techs.com/technologies/overview/top_level_domain (accessed on 21 April 2025).
44. GoDaddy. Domain Names, Websites, Hosting & Online Marketing Tools. 2025. Available online: https://www.godaddy.com accessed on 21 April 2025).
45. ICANN. The Domain Name System. 2022. Available online: https://www.icann.org/resources/pages/dns-2022-09-13-en (accessed on 21 April 2025).
46. Wallace, R. 2023 DNS Traffic and Trends Analysis. 2024. Available online: https://vercara.com/resources/2023-dns-traffic-and-trends-analysis (accessed on 21 April 2025).
47. Cloudflare. Cloudflare. 2025. Available online: https://www.cloudflare.com (accessed on 21 April 2025).
48. Amazon Web Services. Amazon Route 53 Developer Guide. 2025. Available online: https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html (accessed on 21 April 2025).
49. Rozekrans, T.; Mekking, M.; de Koning, J. Defending Against DNS Reflection Amplification Attacks. Technical Report, University of Amsterdam, System & Network Engineering RP1. 2013. Available online: https://www.nlnetlabs.nl/downloads/publications/report-rrl-dekoning-rozekrans.pdf (accessed on 21 April 2025).
50. Skwarek, M.; Korczynski, M.; Mazurczyk, W.; Duda, A. Characterizing Vulnerability of DNS AXFR Transfers with Global-Scale Scanning. In Proceedings of the 2019 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 20–22 May 2019; pp. 1–6. [CrossRef]
51. Salat, L.; Davis, M.; Khan, N. DNS Tunnelling, Exfiltration and Detection over Cloud Environments. *Sensors* **2023**, *23*, 2760. [CrossRef]
52. Martin, A.R. DNSSEC: How It Works and Key Considerations. 2022. Available online: https://www.akamai.com/blog/trends/dnssec-how-it-works-key-considerations (accessed on 21 April 2025).
53. Huston, G. Calling Time on DNSSEC? APNIC Blog. 2024. Available online: https://blog.apnic.net/2024/05/28/calling-time-on-dnssec/ (accessed on 2 January 2025).
54. Li, R.; Liu, B.; Lu, C.; Duan, H.; Shao, J. A Worldwide View on the Reachability of Encrypted DNS Services. In Proceedings of the WWW '24: Proceedings of the ACM Web Conference 2024, New York, NY, USA, 18 May 2024. [CrossRef]
55. Dawood, M.; Tu, S.; Xiao, C.; Haris, M.; Alasmary, H.; Waqas, M.; Rehman, S.U. The Impact of Domain Name Server (DNS) over Hypertext Transfer Protocol Secure (HTTPS) on Cyber Security. *Comput. Mater. Contin.* **2024**, *80*, 4513–4542. [CrossRef]
56. Jung, W.K.; Kwak, B.I. MTL-DoHTA: Multi-Task Learning-Based DNS over HTTPS Traffic Analysis for Enhanced Network Security. *Sensors* **2025**, *25*, 993. [CrossRef] [PubMed]
57. Csikor, L.; Divakaran, D.M. The Evolution of DNS Security and Privacy. *arXiv* **2023**, arXiv:2312.04577v1. [CrossRef]
58. Hoang, N.P.; Niaki, A.A.; Dalek, J.; Knockel, J.; Lin, P.; Marczak, B.; Crete-Nishihata, M.; Gill, P.; Polychronakis, M. How Great is the Great Firewall? Measuring China's DNS Censorship. In Proceedings of the 30th USENIX Security Symposium (USENIX Security '21), Vancouver, BC, Canada, 11–13 August 2021. Available online: https://www.usenix.org/conference/usenixsecurity21/presentation/hoang (accessed on 21 April 2025).

59. Kraken Learn Team. What is Ethereum Name Service (ENS)? 2023. Available online: https://www.kraken.com/learn/what-is-ethereum-name-service-ens (accessed on 21 April 2025).
60. Chainlink. What Is Maximal Extractable Value (MEV)? 2023. Available online: https://chain.link/education-hub/maximal-extractable-value-mev (accessed on 16 December 2025).
61. Erinle, Y.; Kethepalli, Y.; Feng, Y.; Xu, J. SoK: Design, Vulnerabilities, and Security Measures of Cryptocurrency Wallets. *arXiv* **2023**, arXiv:2307.12874. [CrossRef]
62. Sayeed, S.; Marco-Gisbert, H. Assessing blockchain consensus and security mechanisms against the 51% attack. *Appl. Sci.* **2019**, *9*, 1788. [CrossRef]
63. Vashishtha, G. Consensus Mechanisms for Permissioned Blockchain Protocols. 2023. Available online: https://www.zeeve.io/blog/consensus-mechanisms-for-permissioned-blockchain-protocols/ (accessed on 16 December 2025).
64. Erinle, Y.; Feng, Y.; Xu, J.; Vadgama, N.; Tasca, P. Shared-Custodial Wallet for Multi-Party Crypto-Asset Management. *Future Internet* **2024**, *17*, 7. [CrossRef]
65. Buterin, V. Why We Need Wide Adoption of Social Recovery Wallets. 2021. Available online: https://vitalik.eth.limo/general/2021/01/11/recovery.html (accessed on 16 December 2025).
66. Pyakuryal, S. Vitalik's Harberger Tax Idea for ENS. 2022. Available online: https://web3domains.com/vitaliks-harberger-tax-idea-for-ens-domain-names/ (accessed on 16 December 2025).
67. Posner, E.A.; Weyl, E.G. Property is only another name for monopoly. *J. Leg. Anal.* **2017**, *9*, 51–123. [CrossRef]
68. Raventós, P.K. DNS and Blockchain-Based Solutions. Are They Any Better? Master's Thesis, Universitat Politècnica de Catalunya, Catalonia, Spain, 2024. Available online: https://upcommons.upc.edu/handle/2117/418470 (accessed on 21 April 2025).
69. Li, Z.; Gao, S.; Peng, Z.; Guo, S.; Yang, Y.; Xiao, B. B-DNS: A Secure and Efficient DNS Based on the Blockchain Technology. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 1674–1686. [CrossRef]
70. Afnic. Could Blockchain (Really) Replace DNS? Technical Report, Afnic. 2024. Available online: https://www.afnic.fr/wp-media/uploads/2024/06/Could-Blockchain-really-replace-DNS-Afnic-Issue-Paper.pdf (accessed on 21 April 2025).
71. Supra Academy. Blockchain DNS: Is Blockchain the Future of Domain Names? 2021. Available online: https://supra.com/academy/blockchain-dns-is-blockchain-the-future-of-domain-names/ (accessed on 21 April 2025).
72. Brown, J.; Jiang, X.; Tran, V.; Bhagoji, A.N.; Hoang, N.P.; Feamster, N.; Mittal, P.; Yegneswaran, V. Augmenting Rule-based DNS Censorship Detection at Scale with Machine Learning. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Long Beach, CA, USA, 6–10 August 2023. [CrossRef]
73. Yang, G. Development and Application of a Decentralized Domain Name Service. *arXiv* **2024**, arXiv:2412.01959v2. [CrossRef]
74. Rahman, F.; Titouna, C.; Naït-Abdesselam, F. Scalable Blockchain System for Load Balancing DNS Servers Using Heuristic Sharding. In Proceedings of the 2023 Fifth International Conference on Blockchain Computing and Applications (BCCA), Kuwait, 24–26 October 2023.
75. Gao, T.; Dong, Q. DNS-BC: Fast, Reliable and Secure Domain Name System Caching System Based on a Consortium Blockchain. *Sensors* **2023**, *23*, 6366. [CrossRef] [PubMed]
76. Sheth, S. Responsible Integration of DNS and Blockchain Namespaces. 2023. Available online: https://www.dnib.com/articles/blockchain-DNS-responsible-integration (accessed on 21 April 2025).
77. Fu, Y.; Peng, Y.; Wang, R.; Ye, C.; Wang, J. TI-DNS: A Trusted and Incentive DNS Resolution Architecture Based on Blockchain. In Proceedings of the 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Exeter, UK, 1–3 November 2023; pp. 1414–1421.
78. Lammertink, X.; Davids, M. Namecoin as Alternative to the Domain Name System. Technical Report, SIDN Labs. 2016. Available online: https://www.sidnlabs.nl/downloads/QX1OMUZiRcS1dpAuY4_f3g/60b9727441f6d293d4aef954abcd6bf9/Report_Namecoin_as_alternative_to_the_Domain_Name_System.pdf (accessed on 21 April 2025).
79. Manzoor, S.; Salahuddin, M.; Khan, W.; Dharejo, F.A.; Jawad, M.; Ullah, S. Blockchain-Based Decentralized Domain Name System. *arXiv* **2025**, arXiv:2508.05655v1.
80. Neurocod. I2P Domain Name Registration with EmerDNS. 2025. Available online: https://github.com/emercoin/docs/blob/master/en/020_Blockchain_Services/030_EmerDNS/031_I2P_Domain_Name_Registration_based_on_blockchain.md (accessed on 21 April 2025).
81. EmerDNS Introduction. 2025. Available online: https://emercoin.com/en/documentation/blockchain-services/emerdns/emerdns-introduction/ (accessed on 21 April 2025).
82. *UTS46*; Unicode IDNA Compatibility Processing. Available online: https://unicode.org/reports/tr46/ (accessed on 26 November 2025).
83. *RFC 1912*; Common DNS Operational and Configuration Errors. Available online: https://www.ietf.org/rfc/rfc1912.txt (accessed on 26 November 2025).

84. *RFC 2181*; Clarifications to the DNS Specification. Available online: https://translate.google.com.hk/?sl=zh-CN&tl=en&op=translate (accessed on 26 November 2025).
85. Heo, J.W.; Kim, J.H.; Jun, M.S. Design and Implementation of Blockchain Network Based on Domain Name System. *J. Korea-Acad.-Ind. Coop. Soc.* **2019**, *20*, 36–46. [CrossRef]
86. Emercoin. Emercoin Network Statistics. Available online: https://explorer.emercoin.com/stats (accessed on 11 October 2025).
87. Emercoin. Emercoin Specifications. 2025. Available online: https://emercoin.com/en/documentation/introduction/specifications (accessed on 11 October 2025).
88. Barth, B. Quirky Fbot IoT Botnet Kills Rival, Communicates via Blockchain-Based DNS. 2018. Available online: https://www.scworld.com/news/quirky-fbot-iot-botnet-kills-rival-communicates-via-blockchain-based-dns (accessed on 21 April 2025).
89. Tahelyani, R. Everything to Know About Ethereum Name Service (ENS) & .ETH Domains. 2024. Available online: https://droomdroom.com/ethereum-name-service-explained (accessed on 21 April 2025).
90. Ethereum Name Service. ENS Use-Cases. 2024. Available online: https://support.ens.domains/en/articles/7900302-ens-use-cases (accessed on 21 April 2025).
91. Ethereum Name Service. ENS Documentation: Resolution. Available online: https://docs.ens.domains/resolution (accessed on 16 December 2025).
92. Ethereum Name Service. Fees. 2024. Available online: https://support.ens.domains/en/articles/7900605-fees (accessed on 21 April 2025).
93. Xia, P.; Wang, H.; Yu, Z.; Liu, X.; Luo, X.; Xu, G. Ethereum Name Service: The Good, the Bad, and the Ugly. *arXiv* **2021**, arXiv:2104.05185v1. [CrossRef]
94. YCharts. Ethereum Average Block Time. Available online: https://ycharts.com/indicators/ethereum_average_block_time (accessed on 11 October 2025).
95. Etherscan. Ethereum Daily Transactions Chart. Available online: https://etherscan.io/chart/tx (accessed on 11 October 2025).
96. Coinbase. What is Unstoppable Domain? 2025. Available online: https://www.coinbase.com/learn/crypto-glossary/what-is-unstoppable-domain (accessed on 21 April 2025).
97. Access Newswire. 4 Million Unstoppable Domain Users Gain Cross-Chain Name Resolution on MetaMask. 2024. Available online: https://www.accessnewswire.com/newsroom/en/blockchain-and-cryptocurrency/4-million-unstoppable-domain-users-gain-cross-chain-name-resolution-on-932421 (accessed on 21 April 2025).
98. ERC-721; Non-Fungible Token Standard. Available online: https://eips.ethereum.org/EIPS/eip-721 (accessed on 26 November 2025).
99. Gould, M. Unstoppable Domains: A Web Address for Your Crypto Wallet. 2021. Available online: https://www.gemini.com/cryptopedia/unstoppable-domains-zil-domain-crypto-domain (accessed on 21 April 2025).
100. Unstoppable Domains. UNS Architecture Overview. 2025. Available online: https://docs.unstoppabledomains.com/smart-contracts/overview/uns-architecture-overview/ (accessed on 21 April 2025).
101. Unstoppable Domains. Unstoppable on Polygon: How UD is Supercharging Web3 Identity with a Layer 2 Scaling Solution. 2021. Available online: https://unstoppabledomains.com/en-us/blog/polygon-l2-solution (accessed on 11 October 2025).
102. Unstoppable Domains Support. On-Chain Domains: Which Blockchains Are Our TLDs Minted on? Available online: https://support.unstoppabledomains.com/support/solutions/articles/48001266018-onchain-domains-which-blockchains-are-our-tlds-minted-on- (accessed on 11 October 2025).
103. Chainspect. Base Blockchain Explorer and Analytics. Available online: http://chainspect.app/chain/base (accessed on 11 October 2025).
104. PolygonScan. Polygon PoS Chain Charts & Statistics. Available online: https://polygonscan.com/charts (accessed on 11 October 2025).
105. Durand, A. Challenges with Alternative Name Systems. 2022. Available online: https://www.icann.org/en/system/files/files/octo-034-27apr22-en.pdf (accessed on 21 April 2025).
106. Handshake Developers. Handshake Protocol Summary. Available online: https://hsd-dev.org/protocol/summary.html (accessed on 11 October 2025).
107. Ali, M.; Nelson, J.; Shea, R.; Freedman, M.J. Blockstack: A New Decentralized Internet. Technical Report, Blockstack PBC. 2017. Available online: https://blockstack.org/whitepaper.pdf (accessed on 21 April 2025).
108. Benet, J. IPFS—Content Addressed, Versioned, P2P File System. *arXiv* **2014**, arXiv:1407.3561v1.
109. Iyer, V. An Examination of Stacks Protocol Performance Compared to Other Layer-One Blockchains. 2022. Available online: https://www.stern.nyu.edu/sites/default/files/assets/documents/Iyer_Stacks%20Protocol%20Performance.pdf (accessed on 11 October 2025).

110. Stacks Foundation. Stacks Documentation. 2025. Available online: https://docs.stacks.co/ (accessed on 19 December 2025).
111. Blockstack PBC. PoX: Proof of Transfer Mining with Bitcoin. 2020. Available online: https://uploads-ssl.webflow.com/5ff21113877dd72b9f913b35/6154471366c15dae30133499_pox.pdf (accessed on 19 December 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.