IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 February 2024, accepted 15 March 2024, date of publication 27 March 2024, date of current version 16 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3382115

SURVEY

Internet Identifiers: A Survey of History, Challenges, and Future Perspectives

ANDREW BABAKIAN[1], GEOFF HUSTON[2], ROBIN BRAUN[1], (Life Senior Member, IEEE), AND JUSTIN LIPMAN[1], (Senior Member, IEEE)

[1]School of Electrical and Data Engineering, University of Technology Sydney, Ultimo, NSW 2007, Australia
[2]Asia-Pacific Network Information Centre, Brisbane, QLD 4101, Australia

Corresponding author: Andrew Babakian (andrew.babakian@student.uts.edu.au)

ABSTRACT Identity systems are fundamental to any network, just as a language is shaped by a consistent set of symbols and their interpretation. A network is defined by a consistent set of identities and their usages. However, as pressures mount to customize and adapt to these identity spaces, one ponders how a network sustains its utility through coherence. This study investigates evolving Internet identifiers and their supporting systems. Owing to the multi-disciplinary nature of the topic, this paper draws perspectives from a wide array of sources that contribute to Internet systems and digital library systems. Initially, this paper highlights the dynamism and role of namespaces, focusing on the critical need to maintain coherence in the public domain. It also discusses the impact of mobility and digital cloning on identifiers and explores their influence on identity and location. In addition, key Internet identifiers are analyzed, contrasting them with methodologies adopted by digital library systems to provide deeper insights into various identity models. Furthermore, this study explores the evolution of the Domain Name System (DNS) as an identity system, examining the tensions and adaptations driven by customization demands. Finally, this paper explores alternative namespaces beyond the DNS, considering potential responses to these evolutionary pressures and future implications.

INDEX TERMS Identifier systems, naming systems, namespace, URI, URN, URL, DOI, IP address, DNS.

I. INTRODUCTION

Identifier systems serve as the backbone of the digital environment, facilitating interactions between diverse entities within the digital realm. The continuous adaptation of these identifiers reflects the need for efficient and flexible techniques to identify and interact with resources, services, and users. Analogous to a common language, the Internet relies on a common set of symbols for communication efficacy.

This study explores the evolution of Internet identifiers through a comprehensive survey of the relevant literature. It includes a historical overview, an analysis of existing studies, and discussions on current challenges and prospects in this field. Given the multi-disciplinary nature of the topic, this paper draws perspectives from a range of authoritative sources. These include the World Wide Web Consortium (W3C), Internet Architecture Board (IAB) Workshops, Request for Comments (RFCs), and academic literature. Additionally, it incorporates significant contributions from entities involved in digital library systems, such as the Library of Congress, and the development of key registry systems to support the Digital Object Identifiers (DOIs).

The emphasis is not on an exhaustive list of identifiers but on those critical to the Internet's infrastructure and influential in digital handle systems. The aim is to synthesize a broad range of literature concerning digital identifiers in the internet namespace and consider potential future directions influenced by evolutionary pressures. This synthesis seeks to enhance the understanding of the Internet's developmental trajectory.

Exploration begins by examining the fundamental roles of namespaces in identifying, locating, and classifying entities. Moreover, unique considerations on the Internet

The associate editor coordinating the review of this manuscript and approving it for publication was Diana Gratiela Berbecaru.

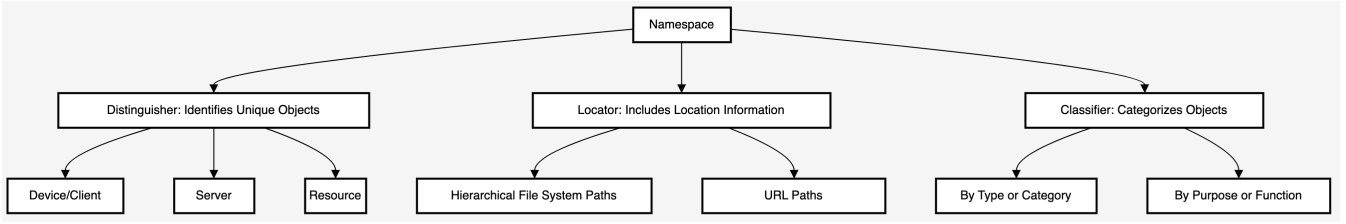


FIGURE 1. Namespace scope and roles.

are given attention, primarily the importance of maintaining coherence in the public namespace. Such coherence hinges on a consistent symbol interpretation, universal reachability, and referential integrity. Subsequently, the critical dimensions of identity and location within the Internet context are examined, exploring how their significance has been influenced by advancements such as mobility and digital cloning.

The narrative progresses to trace and emphasize the roles played by Universal Resource Identifiers (URIs), Universal Resource Locators (URLs), and Universal Resource Names (URNs) within the realm of identifiers and locators. To transcend the boundaries of networked systems, this study's exploration extends to identifier systems utilized in broader contexts, such as library systems. These include the International Standard Book Number (ISBN) used in publishing and Digital Object Identifiers (DOIs) for digital resources. The paper then juxtaposes these identity models with those at the Internet's heart, such as the URI versus URL and the IP Address namespace.

Subsequently, the study examines the thematic topic of search versus identification, exploring the evolution of directory lookup systems. This includes a chronological analysis, commencing with the earlier network namespaces, such as HOSTS.TXT and X.500. Examining X.500 Directory Naming Framework is important to compare its directory system approach with the Domain name system (DNS), particularly the system differences between searchability and uniqueness of resolving names. The study transitions to the emergence and importance of the DNS. The paper draws insights from the DNS predecessors to highlight its evolutionary journey. The successive generation of new resource records highlights this evolution, enhancing the system's flexibility to address emerging challenges.

Furthermore, the study analyzes the shift of DNS from a static lookup system to a dynamic, computation-oriented resource. In addition, this study further examines the evolutionary pressures that catalyzed the creation of other namespaces that operate outside the global scope of the DNS. These alternate namespaces, established for specific purposes, signify another facet of the Internet evolutionary process and highlight the necessity for a cohesive approach across varied namespaces to mitigate the risk of fragmentation.

In conclusion, this study offers insights into the progression of identifier systems in the Internet context, emphasizing the critical role of DNS in supporting Internet identifiers. It also highlights the ongoing transformations driven by evolving market needs and technological advancements. This discussion clarifies the complex interplay between identifier systems, the necessity of namespace cohesion, and the adaptation to evolutionary pressures.

II. BACKGROUND

Various factors contribute to the functionality and design of the complex field of Internet identifiers, making it essential to understand these elements thoroughly. This section establishes the foundation of the paper by tracing the developmental history of Internet identifiers, discussing namespaces and their role in structuring digital communication, and acknowledging their use across multiple disciplines. It clarifies the concept of a namespace, which is crucial for grasping the discussions that follow in this paper. Moreover, this section investigates the specific considerations and challenges within Internet identifiers, focusing on maintaining coherence and referential integrity. Finally, this section examines the interplay between identity and location, clarifying these concepts in the context of Internet infrastructure and library systems.

A. DEVELOPMENT AND HISTORY OF INTERNET IDENTIFIERS

The Internet's development was not in isolation; it was influenced by contemporaneous network protocols and the interplay between the Internet, DECnet and other proprietary network protocols. A common feature of these protocols was their limited scope. As a result, the inclination to associate natural language alias with network-connected hosts was not considered a significant issue. The prevailing solution was to associate a 'word' with a protocol address, assemble this information for all hosts, and distribute a copy to every connected host.

Two methods ensured uniqueness in constructing identifiers: one involved using a statistically unique random number, and the other was a managed identifier space through a hierarchical system, which is further discussed in section II-C. In the case of DECnet, scalability challenges were encountered early in its development, leading to the adoption of a hierarchical, two-level routing structure comprising of 'areas' and 'local networks' and a corresponding two-level name alias system. The system used a namespace format of two labels separated by a double

colon (AREA::NAME), tightly integrated with the network's connection topology [6].

In parallel, the rise of store/forward messaging systems introduced additional requirements. These systems, often not permanently connected, evolved without central coordination and rapidly expanded. Initial attempts to use single-level naming faced collision issues, prompting a shift to a generalized approach to the two-layer naming system using an n-layer hierarchy. The Internet adopted this hierarchical naming approach coinciding with the transition from the HOSTS.TXT file system to the Domain Name System (DNS), which is further detailed in sections IV-A and IV-C.

B. THE ROLE OF A NAMESPACE

Names categorize the world by assigning unique identifiers to concepts and entities. These names serve as a crucial means of conveying a shared understanding of a concept in communication, serving as essential components in maintaining referential integrity and enabling clear references. A namespace is the total set of allowable names within a broad class. Namespaces enable the establishment of defined and limited environments in which the uniqueness of names is ensured within a certain context. For instance, a person's first name may be sufficient to distinguish them within a close-knit community of friends, family, and coworkers. However, when the context expands to a larger population, such as a city, state, or country, additional information, such as the middle and last name, is necessary to guarantee the uniqueness of the name. In human use contexts, a significant level of ambiguity is often tolerated as opposed to automated systems.

Automated systems are no exception regarding the importance of the role and coordination of naming within namespaces. A namespace creates a boundary for the unique use of names and requires strict constraints to eliminate ambiguities. These namespaces are often formally specified to ensure consistency, ensuring that names are unique within a certain context. However, the conflation of multiple roles of names to indicate 'who,' 'where,' and 'what' in automated systems has led to an overload of their semantic role [13]. This study highlights the three roles played by a name in a namespace for automated systems. Please refer to Figure 1, which illustrates these multiple roles.

First, a name identifies and refers to a specific object, including a broad set of networked entities such as devices/clients, content, resources, or servers. They are used to distinguish distinct instances of objects within a broad class [25]. Techniques to disambiguate larger data sets concerning Internet considerations are discussed in Section II-D.

Second, the scope of identity in a namespace may include a locator. This is often observed in hierarchical file systems, where the prefix before the file name functions as a location path [3], [131]. However, it may not be associated with a location context, such as the IMEI in mobile networks, which has no location context. It is important to note that incorporating location into an identity name can assist in the uniqueness of the name, particularly for automated systems. However, this is a convenient shortcut for automated systems, whereas, for human systems, there is an implicit location context of 'here' and 'now' in everyday conversation. The significance of this distinction is discussed in Section II-E to compare and contrast the Internet challenges in terms of identity and location.

Finally, including a classifier in the namespace is optional and depends on the context of use. If the identity system is designed for a specific purpose, the context of its use is already defined. However, if the system is intended for multiple-use contexts, a classifier or 'what' parameter may be required. The classifier can be included as an attached qualifier or embedded in the identity token, depending on the system's requirements and efficiency. While embedding the classifier may be cheaper and more efficient in automated systems, it is not a mandatory feature of identity systems but rather a convenient shortcut in certain situations. The primary role of the classifier is to categorize objects across broad sets of classes using attributes within the namespace [68]. For example, library systems use classification techniques [17], [134] to logically categorize physical and electronic documents based on the subject and discipline, allowing users to browse and locate them.

C. FLAT AND HIERARCHIES

To ensure that identifiers are uniquely allocated to entities within a flat namespace, a fundamental operation involves enumerating each identifier across the namespace to prevent conflict. A conflict arises when multiple entities are assigned the same identifier within a namespace. When this occurs, it becomes impossible for the namespace to resolve the identifiers unambiguously, as there are no distinguishing factors between conflicting entities. Despite these limitations, certain systems have found ways to operate efficiently within flat namespaces. For instance, The Onion Router (TOR) [43] and Distributed Hash Table (DHT) [14] are examples of systems that have managed to maintain the uniqueness of identifiers even as the size of the namespace grows. However, as they become larger and more complex, the cost of administration and maintenance of identifiers and their bindings becomes non-trivial. Initially, Flat namespaces were not considered because of concerns about their ability to preserve uniqueness under scaling pressure. Hierarchies have emerged as a solution to the challenges posed by flat namespaces in maintaining the uniqueness of identifiers as the environment grows [78].

A hierarchical system is composed of interrelated subsystems, with each subsystem being hierarchic in structure until the lowest level of the elementary system is reached [120]. A divide-and-conquer approach was employed using a hierarchical tree to manage large namespaces. This approach allows the namespace to grow beyond the constraints of any single node. This contrasts with the finite size of flat namespace environments and effectively circumvents any limitation on the number of names that a single node can hold.

The identifiers in this context contain a discernible structure by grouping objects that represent their positions in the hierarchy [24]. Hierarchy has several advantages over flat spaces. First, the separation of concerns in large namespaces can be achieved by creating smaller administrative domains, each of which can allocate identifiers from a namespace that does not collide with others in the hierarchical tree [89]. This also enables the independent governance of the namespace subtrees for administrative and scalability purposes.

Second, while resolution through the hierarchical technique may sometimes be slow, it provides an efficient search for identifiers and their associated pointers in the namespace. This hierarchy allows for simple search functions that do not require the querier to assemble a local copy of the entire space for resolution. During a query, only the identity of the name server for the queried label must be locally stored.

Its tree-like structure allows navigation to traverse each parent and child node until the identifier is matched. Unlike hierarchical namespaces, flat namespaces inherently store identifiers sequentially. However, the task of searching in an unstructured flat space may not pose a significant challenge, given that an appropriate sorting mechanism is employed. The transformation of an unstructured data set into a structured or sorted set serves as the key to efficient search operations. The index can then be used for searches, thereby reducing the time required to find an identifier. By contrast, hierarchies provide an alternative approach that does not require the assembly of all data at a single point in time.

Finally, the reusability of the identifiers in a hierarchical structure is an important advantage. Each part of a hierarchical name is resolved relative to a different context of relatively small size. The same identifier can be used with different semantic meanings in different contexts to suit a variety of uses [24]. In the DNS context, it is important to note that there is a distinction between names and labels. A name is an ordered sequence of labels, where each label acts as a node in the name hierarchy. The equivalence of two names is established if they both comprise the same set of labels in the same sequence. This facilitates the utilization of a single label for multiple names without conflicts or ambiguity.

D. INTERNET CONSIDERATIONS

There are two essential preconditions for effective communication between the two entities. First, communication is defined by a common symbol space and its usage. Second, a common semantic interpretation of the composed symbols is essential for preserving the intended or original meaning of the message. Failure to satisfy these two preconditions results in communication failure [56], [116]. A communication network relies on a unique symbol space with consistent semantic interpretation to ensure that any entity can understand the intended meaning of a message received. These strict rules are crucial to prevent symbols from being interpreted differently, leading to the loss of the original meaning [56].

The uniqueness of a common symbol space, the namespace, is crucial for ensuring the unambiguous usage of Internet identifiers, allowing them to be passed on, referred to, and reused without semantic ambiguity. However, the growth and scaling of the Internet have led to the evolution of its name system mapping functions, allowing a name to refer to a constant digital object with a location relative to the querier rather than a single global locator. In addition, when a namespace avoids enclosing a unique location into its name value, issues related to the replication and distribution of digital objects become easier to reconcile. For example, if a digital object is assigned a unique identifier that is not location-specific, replication and distribution can occur without updating it.

Referential integrity in Internet namespaces ensures a reference, when transferred from one entity to another, consistently signifying the same object across the Internet. This objective eliminates the ambiguity in using these references and ensures a shared understanding of communicated terms as valid references. The system's capacity to accommodate Internet dynamism is of equal importance by regularly updating association pointers relevant to the location of the querying entity [36], [37].

For the Internet, as a public communications medium, to maintain its utility as a public service, its associated identifier namespace must be operated as a coherent public namespace [37]. In this context, fragmentation refers to the potential division of this unified namespace into smaller, disjointed parts. Such fragmentation poses a significant risk to the future of the Internet as it could lead to inconsistencies in symbol interpretation across different systems. Specifically, fragmentation can challenge the coherence of the Internet, which hinges on universal reachability, consistency in symbol interpretation, and referential integrity [53]. As described in Figure 2, consider the practice of using the '.local' top-level domain, a common strategy employed by Apple to create a variety of context-specific local namespaces. These namespaces are intentionally limited in visibility and cannot be accessed by standard DNS queries into the public namespace. This approach offers a unique method for creating application-specific namespaces yet concurrently introduces complexities in delineating the global public namespace of DNS from these local domains.

These complexities intensify when the constraints on these local namespaces fail or in scenarios contemplating the delegation of the '.local' domain as the global scope domain. Therefore, ensuring name coherence in local and global network environments is a substantial factor. The issue of namespace fragmentation highlights the significance of identifiers in a broader context. Name systems have traditionally existed in a realm associated with a common infrastructure, where any application can use this common identity system, irrespective of the context of the query into the system, the result the name system provides in response is the same. 'Who' is asking does not materially alter the response. However, adding context and qualification to a

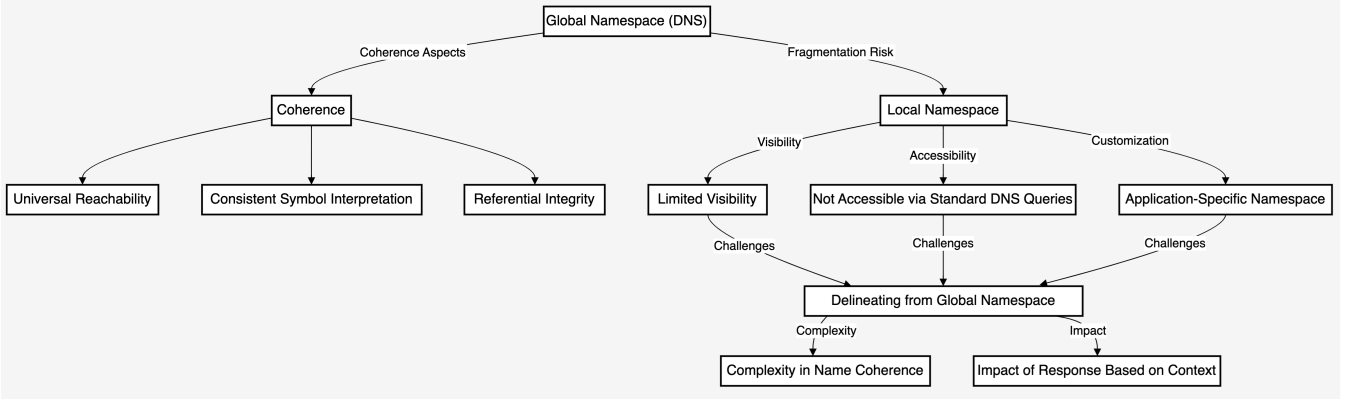


FIGURE 2. Global namespace coherence and fragmentation risks.

query in an identification system can affect the response received because the identity of the querier is a factor, similar to how the 'Client Subnet' in the DNS [23] can impact the response.

It is debatable whether a single infrastructure name system can effectively handle various needs and applications or if a customizable namespace is required. Customized identification systems can be more specific to the realm of use and provide more useful utility models for automated applications. However, the translation problem persists when attempting to transfer an identifier from one context to another.

E. TECHNOLOGICAL INFLUENCES ON IDENTITY AND LOCATION

Identity and location have gained increasing importance in the digital realm in contrast to their traditional usage. This can be attributed to two key factors, mobility and cloning. The early digital environment consisted of mainframe computers dispersed across various floors and buildings, with network connectivity predicated on a fixed model that firmly enmeshed identity and location. However, with the advent of portable computers, the Internet has needed to support an always-connected model that accommodates endpoint mobility. The evolution of mobile cellular services exposed an aspirational goal for endpoints to retain a constant service and context, even when transitioning between different network connections, each with a unique IP context.

This led to a challenge in which IP addresses were viewed interchangeably by different network elements, making them difficult to distinguish. For example, from the routing system perspective, an IP address serves as a locator, whereas from the Domain Name System (DNS) [79] perspective, names are isomorphic to IP addresses, and the IP address acts as an identity that is treated as such by applications [77]. In scenarios where mobility is a critical attribute of the network architecture, the care-of addresses serve as temporal locators only, as endpoints transition between locations and providers, necessitating a new IP address for the device. However, this frequent change in IP addresses disrupts the continuity of ongoing sessions, thereby introducing a unique challenge for mobile networks. Section III-D discusses this challenge and explores strategies to address frequent changes in IP addresses, considering the duality of identity and location inherent in mobile networks.

In contrast, when a file is copied between two systems, it generates two separate named duplicates. Each file's identification incorporates a distinct path namespace, leading to unique references despite the identical content. This raises challenges in identifying digital objects in a world where duplication and replication are necessary to optimize content delivery on the Internet.

In the digital world, the proliferation of cloning has raised challenges in identity and location. For instance, within a library system, there may be numerous physical or digital replicas of the same book, all of which are referred to as a single instance. Regardless of their location, these books are identified by a common identifier, often as a persistent URL. Library systems commonly employ key registry systems such as the International Standard Book Number (ISBN) and Digital Object Identifier (DOI).

Although the URL model is more intrinsically useful for locating the object being sought out, it is less useful when the content might exist at multiple locations. Content Delivery Networks (CDNs) address this challenge by replicating content in proximity to user requests. This involves employing aggressive caching, geo-replication, and traffic engineering [91]. The fundamental challenge in digital environments is to maintain multiple instances of the same entity at different locations. This issue is compounded when the aim is to steer clients optimally towards the instance of a digital object that delivers the best efficiency or speed for each individual client.

Two strategies have been developed to address this problem. The first strategy involves a selective response to DNS queries for the named digital object. The system returns the IP address corresponding to the 'optimal' instance of the object for the client. Akamai illustrated this method well,

using DNS as a geo-locator and modifying the response depending on the assumed identity of the queried [114]. In this context, IP addresses serve as geographical selectors.

On the other hand, Cloudflare's approach represents the second strategy. This involves using a small set of IP addresses announced into the routing system from multiple origination points. This strategy, known as anycast, allows the routing system to locate the optimal instance of a named object, leaving the DNS unaltered [15]. Both strategies enable URLs to mimic and behave like Uniform Resource Identifiers (URIs) [5]. This allows the establishment of a persistent identity while allowing the domain portion of the URL to serve as a dynamic locator.

In summary, the concepts of location and identity can often be interchanged using various mapping system functions. For example, in a mobility scenario, a device maintains its constant identity while its location changes. However, when digital objects are cloned or replicated, they simultaneously maintain a consistent identity across multiple locations. To support location and identity on the Internet, the burden shifts to different systems. For instance, in the case of mobility, updating the DNS to reflect the new IP address mapping when a device changes its location or informing the routing system of the device's new location helps to retain its IP address. In a scenario in which content entities are cloned or exist in multiple locations, certain strategies are necessary to maintain the URL as a persistent identifier. One strategy involves DNS to provide customized responses to clients' queries. This can include appending meta-data to the query resolution, such as using the source IP address to infer the querier's location. An alternative strategy is the anycast approach, in which the routing system independently guides the client to the optimal instance of the content entity. Each discussed approach for mobility and cloning has its own design trade-offs, making it suitable in certain contexts and not in others.

III. IDENTIFIERS VERSUS LOCATORS: PARALLEL INSIGHTS INTO LIBRARY AND INTERNET SYSTEMS

This section examines the roles of identifiers and locators in object identification, focusing on the differences between library and internet systems. Starting with an analysis of the International Standard Book Number (ISBN) and Digital Object Identifier (DOI), this discussion explores how these persistent identifiers operate outside the typical Internet namespace, drawing insights from digital libraries and publishing. The section then shifts to U-systems, including the URL, URI, URN, and IP namespace, to compare identity models across these varied contexts. A summary of distinct approaches to identifiers is presented in Table 1.

A. INTERNATIONAL STANDARD BOOK NUMBER

This section explores important registry key systems that publish persistent identifiers outside the Internet namespace to provide insight into information management and resource identification for digital and non-digital resources. The examination of registry key systems such as the International Standard Book Number (ISBN) and Digital Object Identifier (DOI) [88] in Section III-B are useful examples of the challenges they solve in the transformation of various media types.

ISBN is a separate model from the publishing industry to create a unique identifier independent of the media type, e.g., hardback, paper, and digital. The purpose of ISBN is to associate content with a unique identifier [44]. The International ISBN Agency manages the central registration of ISBNs, ensuring that each allocated ISBN is unique and corresponds accurately to its intended media. Importantly, the ISBN space is internally structured, facilitating the unique identification of the media.

When multiple copies and media types exist, identifying specific instances without a unique identifier can be challenging. For example, if two books have the same title, author, and publisher but different publication dates and editions, it may be difficult to discern which book is being referred to. The ISBN solves the distinguishing problem of assigning a unique key to a book. While numerous copies and media types may exist, the ISBN serves as a unique identifier for each book edition. However, the ISBN does not provide information on how to locate or retrieve a specific media instance. To find the media and its type, a separate catalogue search system or database containing a mapping function is required to determine the availability and retrieval options. In addition, to resolve the ISBN, one must use its unique identifier as an input to retrieve publishing information related to the identified entity [88].

Once an ISBN is assigned to a book or monograph, it identifies the exact entity to be referred to. This identifier remains static, and should the content be revised, a new ISBN should be issued. This contrasts sharply with the characteristics of a Uniform Resource Locator (URL) [11] on the World Wide Web (WWW). Unlike an ISBN, a URL does not specify the exact content retrieved but points to the location where it can be found. Moreover, the content associated with the URL is dynamic and can change over time. Content may also change locations, which poses a challenge for assigning persistent identifiers, as discussed in Section III-B.

B. DIGITAL OBJECT IDENTIFIER

The Digital Object Identifier (DOI) system assigns persistent identifiers to content entities, whether physical, digital, or abstract [88]. Although both ISBN and DOIs offer persistent identifiers for content entities, the DOI system was developed as an extension of the ISBN system to address challenges in the digital landscape. The handle system influenced the DOI system [124], created by the Corporation for National Research Initiatives (CNRI) as a framework for managing digital information. The handle system includes a global naming service that stores persistent identifiers, called handles, for digital resources and resolves those handles into the information needed to locate, access, and utilize the

resources. The handle remains a persistent identifier, whereas its associated values can change to reflect the current state of the identified entity.

Libraries and publishing communities were among the first to adopt Digital Object Architecture (DOA) [63], [64] because they recognized the need to uniquely and unambiguously identify content entities rather than reference them by location, which is an issue with existing identifiers such as URL. Although some URLs are stable, they lack version control mechanisms to track changes to specific versions that users might be interested in. Furthermore, when digital objects identified by URLs move between locations over time, these URLs can become invalid, causing additional challenges in updating the citation indices. This problem of maintaining a consistent identity with the associated metadata, exacerbated by the potential instability of URLs, is especially relevant for libraries responsible for digital preservation [48].

The Handle System was developed outside the DNS system, enabling the creation of location-agnostic qualification strings, thus forming a separate namespace [34]. However, extending the DNS as a general-purpose name service encountered two main challenges. First, DNS operates on a zone-level management structure that lacks administrative capabilities at the individual name level. This presents a considerable challenge regarding the fine-grained control and governance of digital identifiers, especially for entities such as publishers who require individual control over a multitude of digital objects. Second, administrative controls cater to network administrators, making them suitable for domain administration but less so for general-purpose administration [124], [125].

The DOIs are governed and administered by the International Digital Object Identifier Foundation (IDF), which publishers use. The DOI system builds on the Handle System, an integral component of the Digital Object Architecture (DOA) [48]. Like the Handle System, DOIs were initially mapped outside the DNS, requiring specialized client-side software to resolve handles [10]. However, as publishers and service providers were responsible for making DOIs resolvable to digital entities, the namespace was eventually integrated with the DNS using a URL scheme. By mapping DOIs to DNS, users can access and retrieve information through a common web browser without requiring client-side software, making the DOI system more accessible to a broader audience.

In summary, the identity representation of the ISBN and DOI is a persistent, unique identifier for a specific instance of a manifestation. All instances within the same identifier are considered complete and precise clones. The ISBN and DOI serve as identifiers that distinguish a member of a particular class (e.g., a book) from others. This system is location-independent, meaning that the location of an entity is not essential for identification. In this context, ISBN and DOI systems rely on a categorization-based identity that is exclusionary in nature because content entities without an ISBN or DOI are not considered part of the specified class.

C. URI, URL, AND URN

In networked systems, a resource identifier serves as a compact surrogate for a resource, enabling identification, retrieval, or both functions. Resource identifiers are represented as textual strings of characters that must be resolvable to yield the associated resource [62]. In the context of the World Wide Web (WWW), a Uniform Resource Identifier (URI) [10] was introduced to define a method for identifying digital object instances, such as documents, images, videos, web pages, or services that exist anywhere. Similar to ISO Object Identifiers (OIDs), URIs were primarily intended to serve as pure identifiers rather than retrieve, dereference, or locate a resource [2], [8], [10]. In addition to ISBN and DOI, a URI is considered a location-independent identifier, reinforcing the concept of a web object identity as a generic identifier [9].

URIs provide resource identification via a naming scheme or other resource attributes. Generic URI syntax specification does not imply the inclusion of names and addresses from various namespaces to be mapped in the URI strings [11]. URIs are general constructs in which the initial ‘‘scheme’’ part of the URI determines the structure and semantics of the remainder of the URI string. The following example demonstrates an abstract identifier consisting of a sequence of components referred to as the scheme, followed by a (‘‘:’’) and then a string (scheme-specific-part) [10].

scheme : scheme-specific-part

The URI syntax does not impose a specific structure or set of rules for the scheme-specific part of the URI. URIs, although relatively niche on the Internet, are not widely used. Their limited traction might be attributed to their amorphic and insubstantive nature, offering less immediate utility than URLs’ tangible benefits. Conversely, the URL, a subclass of the URI, has been widely adopted as it appeared easier to use. The URL identifies a resource by representing its primary access mechanism [74], thus establishing it as a functional form of identity that is location-dependent. The URL syntax enforces a specific structure to represent hierarchical relationships within a namespace. The following HTTP [40] URL example consists of a sequence of components referred to as the scheme, authority, path, query, and fragment.

https://example.com/articles/technology?view=summary

In this example, the scheme defines the Hypertext Transfer Protocol Secure (HTTPS) protocol [104], which is used to access the resource. The authority includes the domain name ‘example.com,’ which resolves the identifier to a specific host. The subsequent path, such as ‘/articles/technology,’ represents an internal path within the local filestore of the identified server. Finally, the query component pro-

TABLE 1. Identity models and comparisons.

Approach	Focus	Identifier Type	Advantages	Disadvantages
Location-Independent Identification	Describes 'what' the identity. Identity by categorization or classification. Identifiers that are exlusionary in nature.	Unique and Persistent identifiers (URI, URN DOI, ISBN)	Persistent identity, independent of location, effective for managing content across multiple instances	May not be suitable for resolving dynamic content, can be complex to implement and maintain, may provide multiple answers, including misleading ones, due to the vast amount of information associated with locators.
Location-Dependent Identification	Definition of 'where' the entity is. An Algorithmic-Based approach.	Locators with Specific Server-Side Instructions (URL)	Simple and efficient for one-to-one mapping of host to resource..	Not effective for content in multiple locations, can be vulnerable to changes in location. Requires CDN and DNS support for resolving URL to multiple locators.
Hybrid-Identification	Complementation of 'what' and 'where' the entity is.	IP Addresses, unique identifiers and locators	Duality expressed in the IP namespace is complementation between the two identity models.	Overloaded semantics, significant privacy concerns due to persistent identity tied to IP, potential for tracking and profiling.

vides additional information to the server. In this case, 'view=summary' indicates a request for a summary view of the technology article. In the example provided, the URL builds on the DNS System. The DNS name is contextualized as a reference point, while the rest of the URL have semantic significance and are subject to server-side translation. In the context of URLs, a domain name reliably identifies the designated host. However, changes in resource location pose challenges. The DNS has solutions for this, with Canonical Name (CNAME) records mapping an alias to a true domain name and Delegation Name (DNAME) records redirecting an entire domain subtree to another domain. These features ensure stability at the domain level despite changes in resource locations. DNS also incorporates advanced mechanisms such as Naming Authority Pointer (NAPTR) records. Rather than merely redirecting, a key role of NAPTR records is to phrase new queries through name transformation. Section IV-C provides more details on these DNS techniques.

In contrast to the challenges associated with the instability of URLs, Uniform Resource Names (URNs) are defined as a subclass of URIs that provide a persistent identifier for resources, independent of their location and access method [109], [123]. The resolution process for URNs is more complex than that of URLs. With URLs, the access mechanism is subsequently used to dereference the locator, retrieving a representation of the associated resource, such as a document [10], [109]. In contrast, URN resolution involves mapping a URN to one or more URLs, which still requires dereferencing the mapped locators to one or more resource representations. The key distinction in this identity system is persistence [109].

As a conceptual example, when a user interacts with a URN, the browser returns a set of candidate URLs, selecting one based on the location or access method to retrieve the resource while remaining entirely transparent to the user of the system [2]. However, similar to the challenges with URLs, if the locators change, such as the DNS name, the system must refresh and update the list to ensure that the URN remains resolvable and continues to yield the desired resource. The URN example identifier includes the scheme, namespace identifier (NID), and namespace-specific string (NSS):

urn:namespace-identifier:namespace-specific-string

In this example, the scheme identifies the type as URN. The URN includes a NID maintained by the Internet Assigned Numbers Authority (IANA) [126]. Examples of namespace identifiers include "NBN" for National Bibliography [46], 'ISBN' for International Standard Book Number [47] and 'ISSN' for International Serial Standard Number [108]. The NSS contains an opaque flat string within the NID that provides the format rules, such as a unique bibliographic identifier from the NBN.

There are several reasons that URNs have not been widely adopted. One reason is that retrofitting the Internet to include the complex mapping of URNs to dynamic locators, which was introduced relatively late in the development of the Internet, created maintenance challenges regarding how URLs would be updated and refreshed as resources changed locations or cloned [22]. Additionally, URNs attempt to perform too many roles at once without a clear focus on one purpose, such as providing a persistent URL, location independence, a resolution system, or a pure identifier [2].

D. INTERNET PROTOCOL ADDRESS

The previous sections discussing ISBN, DOI, and URI have revealed two distinct approaches to identity. The first approach involves assigning identifiers, such as ISBN or DOI, to each instance of a content entity that is an exact clone. In this model, identity is determined by the entity's membership within a specific class, unlike other existing classes. The second approach is an algorithmic-based method for identifying the location where the content entity can be found. In this model, identity is location-dependent. Aspects of identity, unique location and unique categorization are essential for defining an object. These two models of identity, 'what' and 'where,' complement each other in providing a useful namespace. The IP address [99]

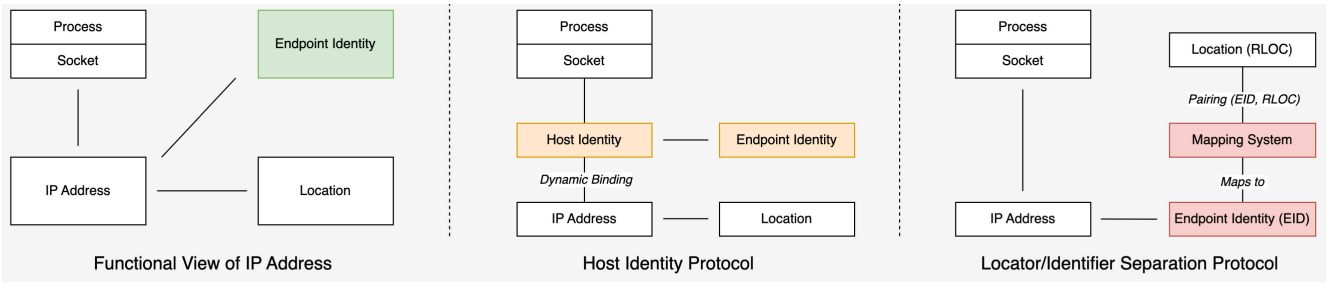


FIGURE 3. Role of an IP address: Example implementations for identity and location separation.

can be considered the ultimate complementation of these two identity models. However, IP addresses are heavily overloaded semantic because they attempt to serve multiple purposes simultaneously [16], [77].

Fundamentally, an IP address [99] is used to identify the point of attachment of a host to a network [117]. Unlike prior address spaces such as DECnet, where all interfaces share a common identifier to represent the host [97], an IP address does not equate to a host. A deliberate decision was made in IP to adopt a more functional view of the address and address numbering. The function of this identifier is to indicate how the network interface of a host is reached. For example, when the forwarding system matches a packet's destination address, it directs a packet to the network interface associated with the host [110]. As traditional networking views assume a fixed model, the identifier is spatially and temporally unique and identical to its locator [103].

Consequently, an IP address not only identifies the host's interface but also serves as a means to access it [37], [117]. The dual role of IP addressing, serving as an identifier and locator, is inherently advantageous but also introduces the potential for confusion. This duality has been widely embraced, largely owing to the static nature of hosts, which permits a level of permanence in address assignment.

Mobility was never considered an attribute that required attention until the realization that all endpoints - devices, services, and content might need to be able to frequently change network attachment [70], [127]. Including mobility as a network capability involves two approaches when an endpoint changes location. The first approach consists of updating the network's forwarding information to reflect the new network location, which maintains service continuity but burdens the routing system. Alternatively, the address of the attachment point could be adjusted to reflect its new location. This approach reduces the impact on the routing system but may cause service disruptions during location changes.

Although IP addresses are heavily overloaded semantic, as they attempt to serve multiple purposes simultaneously [16], [77], deliberate efforts have been made to develop various techniques for disambiguating location and identity to address mobility challenges. These approaches, extensively covered in the literatures [66], [101], [122], [130], and [137], share three key components in their design principles: a stable identifier for a mobile endpoint, a locator representing the IP address of the current location, and a mapping function to associate the identifier with the locator [130]. Refer to Figure 3 for a representation of these principles.

Within the context of architectural support for mobility, there are two main techniques: handling host relocation, which is transparent to the network using a level of indirection in the routing architecture, and transport-level approaches, which ensure that sessions are sustained in the event of an endpoint IP address change. Although numerous methods have been proposed, the following examples illustrate using the IP address as a network namespace.

Mobile IP [92], [93], [96], the Home Address (HA) is elevated from its role as an interface identifier to a constant endpoint identifier, independent of the host's point of attachment within the network domain [121]. The Care-of Address (CoA) serves as a temporary locator, changing as the endpoint moves between the network domains. The home agent maintains a binding between the HA and CoA, which the architecture utilizes when an endpoint transitions between the locations. The home agent is responsible for managing the mobility within the network, primarily by establishing tunnels [72], [95]. This facilitates seamless packet delivery to the relocated endpoint [94]. Importantly, this added layer of indirection localizes the implications of a location change, affecting only the devices engaged in communication with the mobile entity. This approach avoids burdening the entire routing system with an incremental load, thereby promoting efficiency and continuity in mobile networking environments.

The Host Identity Protocol (HIP) [80], [86], described in Figure 3, creates a new namespace between the IP layer and transport protocols. In this namespace, the host identifier is a public cryptographic key that allows hosts to authenticate to their peer hosts directly using their host identity [85]. The HIPs identity namespace replaces the traditional role of IP addresses in binding communication sessions, providing a stable connection if the endpoint's IP address changes [85]. The role of the IP address in the model is for routing purposes, creating a location/identity separation in the architecture.

The Locator/ID Separation Protocol (LISP) [38], described in Figure 3, creates a disjoint IP namespace where the Endpoint Identifier (EID) is decoupled from the locator

(RLOC) and is stored in its mapping system. LISP provides topological aggregation, leveraging a dynamic locator approach for IP routing, similar to the earlier effort of Pseudo-IP (PIP) [41]. LISP employs UDP encapsulation for its delivery mechanism, with the outer IP header containing the destination Routing Locator (RLOC). Based on this RLOC, the network core forwards packets to the destination's LISP-speaking gateway at the edge. The edge gateway is responsible for de-encapsulating the packet and subsequently forwarding it to the endpoint interface by utilizing the EID to determine the original destination.

Despite various proposals for separating endpoint identity from location, there are known disadvantages. For example, there is the added cost of administrative overhead, additional infrastructure, and the need for extra levels of indirection through several encapsulation techniques, each with inherent challenges in the Internet's routing architecture [71]. Another concern is privacy, as IP addresses visible in IP packet headers serve as identifiers and can be associated with a client's identity. A skilled attacker in the path between the client and destination can potentially infer the client's identity and activity on the network through passive monitoring [39]. To address this privacy issue, efforts have been made to create temporal identities by generating randomized identifiers, particularly in IPV6 [29], [82].

In a recent effort to address the mass consumer market to ensure privacy protection is more accessible to Internet users, Apple's iCloud Private Relay was designed to safeguard user privacy online. This service protects user privacy by concealing their online activities from network intermediaries and passive observers. The principle of a Private Relay is that IP addresses identifying users should be separated from the names of websites users access [57]. As described in Figure 4, the architecture consists of two tiers: ingress and egress. The responsibility of the ingress relay is for clients to establish a connection in which the original IP address is visible; however, the website names requested by the client are encrypted. Proxying traffic through the ingress relay shields the client's IP address from the egress relay, third-party intermediaries and destination server [111]. The technique employed by an ingress relay hides the client's IP address while exposing sufficient information for the egress relay to understand the client's location for address allocation purposes. The ingress relay converts the client's IP address into a 'geohash,' representing the user's location coordinates. Consequently, as the client communicates with the egress relay, the client's IP address is not forwarded. Refer to Figure 5 for a visual illustration of this sequence.

The egress relay performs three primary functions. Firstly, it receives the 'geohash' from private relay clients, then consults its database to identify and assign the nearest corresponding IP address to the client. Second, it decrypts the website name, requested by the client. Finally, it completes the connection between the client and the requested website. The egress relay is not privy to the user client's original IP address. The egress relay only receives location information

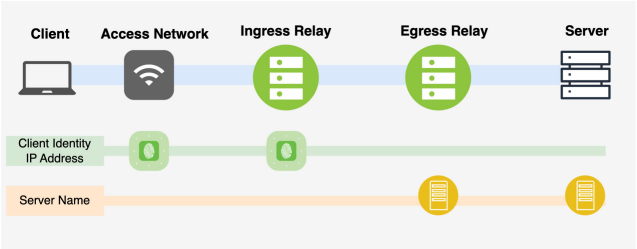


FIGURE 4. Apple relay: Anonymizing client's identity.

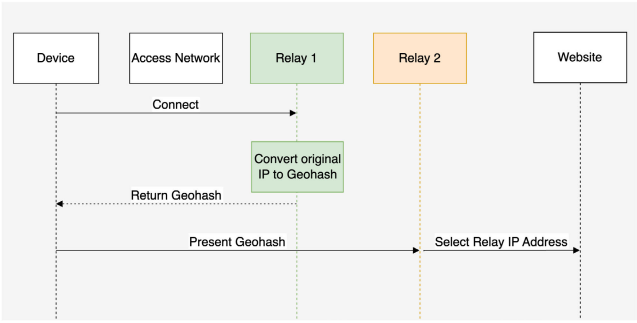


FIGURE 5. Ingress and egress relay: IP address identity, and location [57].

to assign the client the appropriate locator, which maps to the region from which the request originated.

Because the egress relay has no knowledge of the original IP address of the user, the Relay IP addresses rotate over time and between sessions, preventing their use as stable identifiers for the user. In addition, the iCloud Private Relay utilizes QUIC [60] to connect to the ingress relay. The tunnel to the egress employs a secure proxying protocol using the HTTP/3 [12], [90] proposed by the Multiplexed Application Substrate over QUIC Encryption (MASQUE) IETF working group [81]. Using QUIC as the transport protocol, MASQUE provides a secure connection that can combine multiple connections within a single proxy connection [111]. As part of this architecture, QUIC introduces a unique approach for sustaining transport-level connections, which is particularly useful in mobility scenarios for roaming clients. Unlike previous transport-level approaches [61], [121], QUIC provides a semantic distinction between the identity of a connection to an endpoint and the current IP address and port numbers, similar to HIP [1] and Serval [87]. QUICs connection endpoint identifier allows sessions to survive if the IP addresses and ports change [1]. For instance, if a client switches from a cellular to a Wi-Fi network, the QUIC connection ID allows the session to be recognized as an ongoing stream, irrespective of changes in the source IP address or UDP port numbers [1]. In summary, the fundamental changes brought about by mobility exert pressure on the IP namespace. To complement location and identity, distinct roles are assigned to IP addresses, requiring additional mapping systems to create disjoint namespaces, thus decoupling identity from location. In essence, names become isomorphic to IP addresses, and IP serves not only as a locator but also as an identity.

IV. SEARCH VERSUS IDENTIFICATION: COMPARING DIRECTORY AND DNS APPROACHES

This section reviews the transformation of HOSTS.TXT into a distributed database and provides an overview of the initial namespaces that are critical to the early stages of Internet development, such as HOSTS.TXT and X.500. The focus is on a thorough exploration of the theme of search versus identification, offering a comparison and contrast between the X.500 and the DNS. The section then examines the evolution of the DNS in lookup design, including the adoption of successive generations of resource records that meet evolving business needs for flexibility, currency, and speed.

A. HOSTS.TXT

In the early development of ARPAnet [50], the HOSTS.TXT file served as a directory to provide a human mnemonic for identifying resources, i.e., the printer [30], [65], [67]. Each line in the host table file contains a human-readable label, a network address, and protocol-specific information supported by the resource [46]. An identical copy of this host table exists in each host. The Network Information Center (NIC) at the Stanford Research Institute (SRI) was responsible for managing the registrations and distribution of the HOSTS.TXT file. As each host was registered to the network or changes were made to label mappings, the NIC would maintain that the host table is current and distribute it to every host on the network.

However, as ARPAnet grew in popularity, the increased frequency of host table updates and its distribution became burdensome and prone to failure. Consequently, the following challenges surfaced when using the HOSTS.TXT file: First, the host table approach assumes that changes occur infrequently and that the environment is constrained to a small number of hosts. The increased frequency of host table updates results from an increase in the number of hosts growing on the network. The increased frequency of updates posed a challenge for maintaining HOSTS.TXT so that network hosts could receive the latest version in a timely manner. Subsequently, the central server where HOSTS.TXT resided was overwhelmed by the number of download attempts, exacerbated by the HOSTS.TXT growing in size [25]. Second, the flat architecture of the HOSTS.TXT namespace introduces a significant challenge. Registrants who did not have the most up-to-date version of the host file could potentially attempt to register an already assigned name. Therefore, the registration process necessitates rigorous checks to maintain the uniqueness of entries in the host file [21].

While the NIC could uniquely assign addresses from the address space, it had no controls to prevent an administrator from erroneously entering a host with a conflicting name. Moreover, host table specifications were established to provide names with a discernible structure and promote ease of recall. The specifications include format rules, such as limiting names to no more than 24 characters composed of case-insensitive letters (A-Z), digits (0-9), and hyphens (‘‘-’’). Additionally, names were prohibited from starting or ending with a hyphen and containing spaces or any other special characters [49]. In summary, the growing success of ARPAnet illuminates the inefficiencies and limitations of the host table file approach, highlighting its potential points of failure. The need for a more efficient and timely system became apparent, shifting the focus toward developing an on-demand query resolution system using a distributed data model. This new direction aimed to replace the static distribution of a host file with a dynamic model, facilitating timely updates and improved overall efficiency. This represented a critical step in the search for a scalable, manageable, and globally available system that can handle the increasing demands of a rapidly expanding digital network [73].

B. X.500

The Open Systems Interconnection (OSI) reference model was standardized by the International Organization for Standardization (ISO) to create a vendor-neutral network architecture. The primary objective of this initiative was to devise an abstract network architecture that promotes inter-operation rather than fostering dependency on a particular vendor’s implementation.

Thereafter, the X.500 Directory Service was developed as an integral part of the OSI’s effort to manage network resources effectively. To address this issue, the X.500 directory service was created and standardized in ISO/IEC 9594 [102]. The X.500 directory is a logical database containing information about objects that model the physical world. The directory service manages names and their associated attributes and provides access to read and retrieve information in the database [98]. An important distinction of this directory system is that the returned answers are independent of the identity or location of the client that initiated the query [106]. Therefore, similar to a telephone directory, the same answer is provided irrespective of the querier.

Although X.500 shares similarities with a telephone directory, key differences and limitations exist when applying these models in the digital space. For instance, a telephone directory can only return a telephone number when the complete name and address of the provider are given [105]. Otherwise, the telephone directory cannot facilitate the query without providing these detailed attributes to disambiguate across all entries. Additionally, physical telephone directories are published no more than once a year, resulting in incorrect and outdated information that cannot be refreshed until the next publication. By contrast, X.500 addressed these challenges through its global namespace design, allowing a hierarchical search to facilitate incomplete queries based on its structured information model. If additional information is required, the attributes can be updated promptly to ensure the presence of distinguishing factors. For example, If two entries have the same first and last names, the supplementary information can be updated [98].

The implementation of the X.500 directory is based on the Directory Information Base (DIB) and Directory Information Tree (DIT). DIB is a structured information model composed of information that classifies objects called entries. Each entry consisted of one or more attributes. In addition, each attribute type contains one or more values that define the attribute characteristics [106]. DIT defines the relationships between entries to allow for a hierarchical search. The Distinguished Name (DN) uniquely identifies each entry in the Directory Information Tree (DIT) based on its location within the DIT hierarchy and the associated attributes, which are expressed through the namespace assigned to the object [83].

Each attribute associated with the entry is called a Relative Distinguished Name (RDN). To find a given entry's distinguished name (DN), the search starts at the tree root and traverses top-down until the entry is found. The DN concatenates each entry's relative distinguished names (RDNs) traversed [106]. For example:

$$\begin{aligned} \textbf{countryName} &= \text{Australia} \\ \textbf{organization} &= \text{UTS} \\ \textbf{organizationalUnit} &= \text{Faculty of Engineering \& IT} \\ \textbf{commonName} &= \text{Andrew Martin} \end{aligned}$$

As described in the above example, the upper levels of the hierarchy are based on the country, organization, and organizational unit. This approach was designed to describe the geographic location of an object. Country, organizational, and organizational units are helpful for hierarchical search, allowing users to locate candidate objects to choose from [105]. In contrast, other lookup systems, such as the DNS hierarchy, are based on naming conventions using a sequence of labels to construct an FQDN to identify objects that are not necessarily location-based. Additionally, the DNS is not a searchable database and only returns an answer based on the exact query provided. The fundamental difference between these approaches lies in the distinction between search and identification, as depicted in Figure 6. In a directory system, the emphasis is on strict search capabilities that enable users to perform targeted queries and retrieve specific information. Conversely, the DNS operates on a system of exact matches, where all queries and corresponding responses are exact [128].

In summary, the X.500 directory service has many attribute sets to describe its resources and users, which requires coordination to manage multiple points of authority at different levels of the hierarchy. Despite documented enhancements, deployments, and case studies [76], [119], [132], the complexity of the Directory Access Protocol (DAP) limits its widespread adoption. However, it is important to note that the development of a Lightweight Directory Access Protocol (LDAP), a simplified version derived from DAP, resulted in significant advancements and widespread adoption. With its lightweight and streamlined approach, LDAP has been successful in various systems, including the Microsoft ecosystem [51], [69].

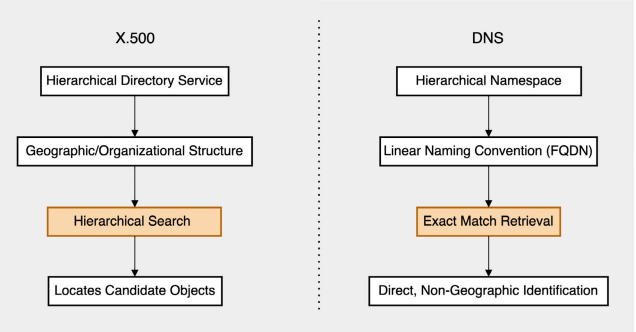


FIGURE 6. X.500 and DNS: Search versus identification.

C. THE DOMAIN NAME SYSTEM

The Internet employs multiple identities, each of which has distinct roles. As discussed in Section III-D, IP addresses illustrate a duality by functioning as identifiers for naming resources and services while also acting as locators essential for packet forwarding within the network. The DNS identity within URLs, detailed in Section III-C, maintains the identity of the service, requiring no modification even when the associated IP address changes. The domain name introduces an abstraction layer over the IP addresses. This allows URL identifiers to point to services without directly specifying their present location on the Internet [37].

The DNS can be described as method-independent [32], [33]. In essence, it assigns names to hosts regardless of the specific type of server or service in use. It functions as a naming system for naming computers or services, similar to how individuals are named in a telephone network, highlighting the notion of endpoint identity. The primary motivations of the DNS was to establish human-friendly identifiers for protocol addresses, while its hierarchical structure was designed to ensure scalability. The DNS established itself of critical importance to the Internet early on in its development owing to several important properties:

- The **Decentralized nature** of the distributed database was used to populate the DNS namespace, eliminating the need for a centralized organizational infrastructure to operate the entire namespace. Each entity operating a delegated part of the namespace can do this autonomously. This decentralization allows the DNS to scale without creating critical bottlenecks, operating with limited coordination with others, which aligns precisely with the broader operation of the Internet itself.
- The **DNS resolution function** operates deterministically, using an exact match between the queried names and DNS names. There is no intended scope within the DNS for ambiguity in the name resolution function, rendering it highly suitable for automation in a deterministic computation environment. This is important because it can support a single unique naming

space across the entire network where the same query generates the same response regardless of the location or identity of the querier. This determinism is the most important reason the DNS is critical to the Internet.

- **Resilience and Efficiency** in name resolution are the defining features of the DNS. The system avoids relying on any single point of failure. Caching results for a predetermined period using the TTL field reduces the load on the DNS resolver, thereby enhancing the efficiency of the system.
- The **Syntax** of DNS names involves concatenating individual labels to construct a DNS name and using a limited set of characters. This design permits the inclusion of natural, human-readable words within DNS labels, although it is constrained to only support ASCII characters in its standard format [32], [33].

The emergence of the DNS represented a significantly simpler form when compared to X.500 in Section IV-B; however, it drew insights from the latter. It also incorporates elements from the antecedents of the DECnet naming system that used a two-level hierarchy. As described in section II-A, each cluster in DECnet was assigned an area name, permitting the allocation of unique names within that specific area. An identical name could be reused in different areas due to its namespace 'AREA::NAME' hierarchy. In contrast to the two-level hierarchy, the DNS design sought to move beyond these constraints. Recognizing the limitations of a two-level hierarchy, DNS is deliberately designed to accommodate an arbitrary number of hierarchical levels. Each node in the hierarchical structure has an associated label. A unique domain name associated with a node is formed by assembling all labels from a specific node at the root of the tree. [79], [133]

In the predecessor HOSTS.TXT [78] approach, the host table file operated on a just-in-case principle, providing all possible host mappings as needed. The maintenance of the HOSTS.TXT file was challenging, particularly when hosts were added, removed, or had their IP addresses changed. This process required every file instance to be updated and distributed to each host, a procedure discussed in more detail in Section IV-A. DNS pioneered a just-in-time query protocol, enabling efficient retrieval of specific records when necessary. Users can query a particular resource and obtain the corresponding answer instead of retrieving all the mappings. This approach assumes a robust and highly available network that enables specific database record queries on demand. One such specific record is the Address (A) Record [33], which created a synthesis between the directory and protocol-level address where the namespace gained greater ascendancy. Addressing the deficiencies of HOSTS.TXT, the A record ensures that when a new host is added or an IP address changes, only the relevant record needs updating, eliminating the need to modify the entire host mapping file. As the Internet's complexity increased, so did the requirements for DNS identifiers. While the A record, which is fundamental to the system, continued to play its central role, it was complemented by successive generations of DNS identifiers. These new iterations introduced additional abstractions and enhanced functionality, meeting the evolving needs of the Internet landscape and, importantly, serving the demands of developing an identifier system. Among these needs is the necessity to accommodate changes in the location of resources while ensuring the permanence of names. With these evolving requirements and DNS advancements, the following subsections will explore three main themes: first, the shift in control of a name in its entirety through outsourcing an indirection to a third party; second, the introduction of a persistent identifier that allows publishers greater control over the resolution process; and lastly, DNS's role as a rendezvous control point, fostering service-level connectivity based on intent.

1) OUTSOURCE CONTROL: CANONICAL NAME RECORDS

As Internet outsourcing expanded, the necessity to relocate resources without altering their names became increasingly apparent. There grew a need for entities to map individualized names into the infrastructure managed by specialized service providers. Although delegating control of a DNS zone to a third party was possible, transferring control of the name proved challenging. This constituted a significant limitation in maintaining a name while letting a third party run the corresponding service.

To address the challenge of shifting control of a name to a third party, a Canonical Name (CNAME) was utilized [33], [35]. With a CNAME record, a name can point to another namespace without official delegation of the zone. This was instrumental in mapping names to a third-party infrastructure. Consider specialized service providers that host numerous entities. As described in Figure 7, these entities could utilize the CNAME record to establish an alias in the service provider's domain by outsourcing an indirection. This strategy enables entities to preserve their individual identities. Simultaneously, the naming system redirects queries to the resources managed by the hosting provider, which contributes to efficient query resolution. Importantly, this process operates in a manner that remains invisible to the end user.

However, the use of CNAMEs introduces unique considerations that have been the focus of industry discussion. Significantly, a CNAME alias controls the name in its entirety. This introduces the issue of handling the apex of a delegation, as CNAMEs cannot reside at the apex; they must be situated at least one level down. Moreover, it is imperative to distinguish that CNAMEs are affiliated with a distinct name, not a hierarchy. In contrast, the DNAME record [107] executes a broader role by redirecting an entire sub-tree of the DNS hierarchy to another domain. Effectively, a DNAME assimilates a name with all its descendants and maps it onto a disparate point within the namespace.

2) PUBLISHER CONTROL: POINTER RECORDS

Building on our earlier discussion of DNS operations, this section delves into the challenges related to the control

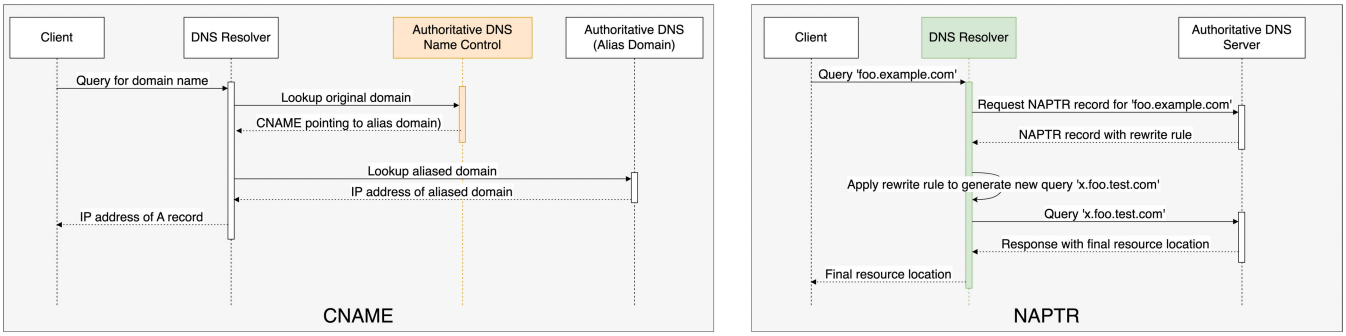


FIGURE 7. DNS lookup process: CNAME and NAPTR records.

of the resolution path, which is an aspect intrinsic to DNS as a lookup service. These challenges stem from the widespread caching of DNS information. When a DNS name propagates, its mapping achieves a certain level of persistence, which can complicate efforts to modify or customize the associated responses. Disseminating a DNS name involves a commitment to resolvability; however, this is not total or indefinite. Publishers retain control via the record's 'time to live' fields, which can regulate how long a specific DNS record remains cached.

As described in Section III-C, a key factor in maintaining a URL's stability is the enduring existence of its associated DNS name, which is a critical consideration when the resource changes locations. In this context, the functionality of the Naming Authority Pointer (NAPTR) [75], a distinct DNS record type, is apparent. The NAPTR record enables dynamic translation into designated target names. It establishes a separate layer that maintains independence between naming and the resolution system, along with its associated protocols and services. This ensures that changes in one system do not directly impact the other [28]. NAPTR records provide a pivotal mechanism to facilitate redirection or transformation, thus reinforcing the permanence of URLs. For example, in Figure 7, if a domain holder intends to relocate a resource from 'foo.example.com,' the NAPTR records serve to redirect incoming requests. Rather than directly resolving 'foo.example.com,' the DNS instructs the resolver to phrase a new query, such as 'x.foo.test.com.' This method ensures the integrity of the original URL, facilitating seamless redirection to the new resource location.

This NAPTR process involves the DNS server returning a regular expression pattern and rewrite rule to the resolver. The resolver then applies this pattern to the original query name and, following the rewrite rule, generates a completely new query. It is crucial to note that this translation lies within the control of the publisher and not the user. This arrangement allows the publisher to relocate content while ensuring a persistent identity for the domain name, thereby transforming the domain resolution process from a semi-permanent to a dynamic state. This technique provides considerable flexibility in terms of rewriting rules.

The Straight-forward-Naming Authority Pointer (S-NAPTR) was introduced as a simplified alternative to the NAPTR, which utilizes regular expressions for name transformations based on specific patterns. By contrast, S-NAPTR focuses on service discovery by working in conjunction with SRV records [45] to provide explicit service details [84]. NAPTR also differs in approach compared to CNAME and DNAME. While CNAME translates one name into another in an iterative process, and DNAME maps all names under one delegation point to another, NAPTR provides a more nuanced approach. The NAPTR offers an adaptable class of names that adheres to a specified pattern and maps them to target names. Although not identical to the original, these target names must conform to the provided pattern. This is in contrast to DNS records such as DNAME, where the query labels to the left of the DNAME point are fixed. In contrast, the NAPTR allows these labels to undergo transformation according to a defined set of rules.

Despite their potential, NAPTR records have not seen widespread adoption owing to their complexity and the perception of unnecessary indirection. Many users find it simpler to customize the namespace directly instead of complicating the resolution process. In the identity system space, the publisher bears the burden, not the user. Systems that impose significant costs or burdens on the user are likely to fail, given the availability of free alternatives that transfer the entire burden onto the publisher. This principle is why systems such as ISBN and DOI, which could have imposed costs on users, continue to exist because they do not impose additional name resolution burdens on the user.

3) RENDEZVOUS AS INTENT: SERVICE RECORDS

The advent of SVCB records [115] in DNS has marked a significant shift from previous DNS record types. Initially, service names were bound to domain names, meaning that all services sharing a logical name were grouped under a common domain name, offering limited flexibility. Services might share a common domain name for various human-use reasons, but there may be a need to disambiguate services and direct them differently, invisible to end users.

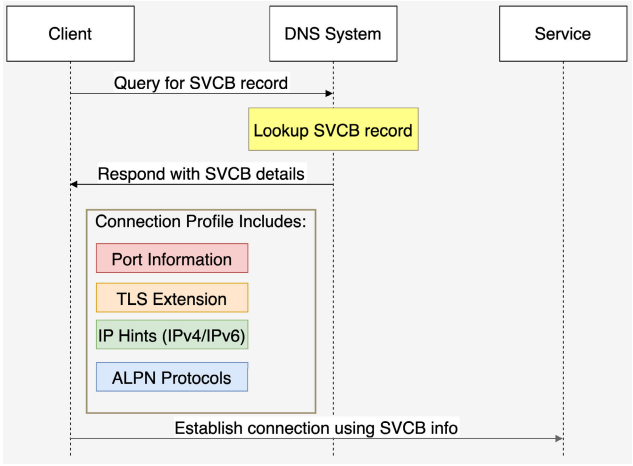


FIGURE 8. Service binding record.

The primary goal of the SVCB record is to allow a client to gain all necessary information to establish a connection to a service with a single DNS query. Moreover, SVCB records provide a DNS aliasing functionality, which is especially useful at the apex of a domain. This functionality allows for the separation of service identity from location, eliminating the need for additional CNAME records and reducing the round trips required for name resolution.

Although SVCB and SRV [45] records exhibit shared characteristics, significant differences exist. An SRV record is non-extensible and has a fixed number of fields. Any alterations to these fields require the creation of a new record type. By contrast, SVCB records can contain any number of key-value pairs, allowing for extensibility with new keys and values. Moreover, SRV records require an extra resolution step because they do not include elements comparable to the IPv4 or IPv6 hints found in the SVCB records. As described in Figure 8, these hints allow clients to initiate connections more rapidly by reducing the number of necessary DNS queries.

In the context of SVCB records, service binding requires encoding the service name, which is prepended in front of the domain name. This follows the Attrleaf naming pattern [26], [27]. For example, _service._port.example.com. With the introduction of SVCB records, DNS transitioned from being a simple locator to acting as a rendezvous broker, enabling more precise, user-intention-driven service discovery. Instead of translating a name to an IP address, DNS interprets the user's intent to determine the specific service they wish to access. This service is defined by a tuple of parameters such as port, protocol, and label, providing a precise description of the service point.

In this evolved model, when a query is made, DNS considers the user's location and the locations of all the potential service points. It then returns a connection profile with detailed information about the requested service, including aspects such as Application-Level Protocol Negotiation (ALPN) [55] and the necessary TLS parameters [42] for the subsequent connection [138].

In essence, the DNS evolved into a rendezvous broker, offering more than a simple IP address. It now provides a means of accessing specific services, thus significantly expanding its role. It is no longer just objects or locations anymore; it is about guiding users to the services they seek based on their intentions. This concept is an abstraction beyond the basic model, making DNS an arbitrator of service access within a namespace.

V. EVOLUTIONARY PRESSURES AND FUTURE PERSPECTIVES

Originally, the DNS modeled the telephone directory, a static system published periodically with invariant lookup keys. This telephone directory was non-customizable and uniform for all users, permitting anyone to use the same lookup key and retrieve identical information. This approach has held appealing properties of a digital identity system. One could share a key, and others could access the same database to retrieve identical results, making it possible to refer to entities via their keys. However, with the evolution of the Internet, the concept of identity systems also has evolved. The use of identifiers, especially within the DNS, began to shift. Instead of treating identifiers as constant indices in an invariant database, there was a growing need to customize these identifiers according to the needs of individual users. In the currently evolving DNS, queries are also based on the intention to generate a more comprehensive set of information, outlining the specific steps required to fulfill that intention. These intention-oriented queries consider a user's location and motivation, resulting in responses tailored to individual users. The extent to which intentions can be parameterized within these queries remains an active area of exploration.

Evolutionary pressures are driving the transformation of the DNS from a simple, efficient single lookup system into a more flexible computational resource that tailors answers to queries. In this context, query names no longer function merely as static lookup keys in a local database; instead, they can be regarded as microcode instructions, with the server operating as a computational engine. Furthermore, evolutionary pressures within the DNS, encompassing issues of uniqueness, time, performance, and evolving service models, are reshaping the definition of an identifier within a namespace. From a computing perspective, this creates a shift in which identifiers' behavior increasingly aligns with roles, services and clients' identities, rather than simply distinguishing one resource from another.

The first two subsections examine these changes in detail. First, Metadata and Query Qualification will explore increasingly sophisticated metadata usage to inform DNS queries. Second, as a Computational Resource, DNS observes query names as microcode instructions, a fundamental shift in how the DNS is utilized. These strategies, born from evolutionary pressures, carry distinct implications and properties, influencing various aspects of the DNS.

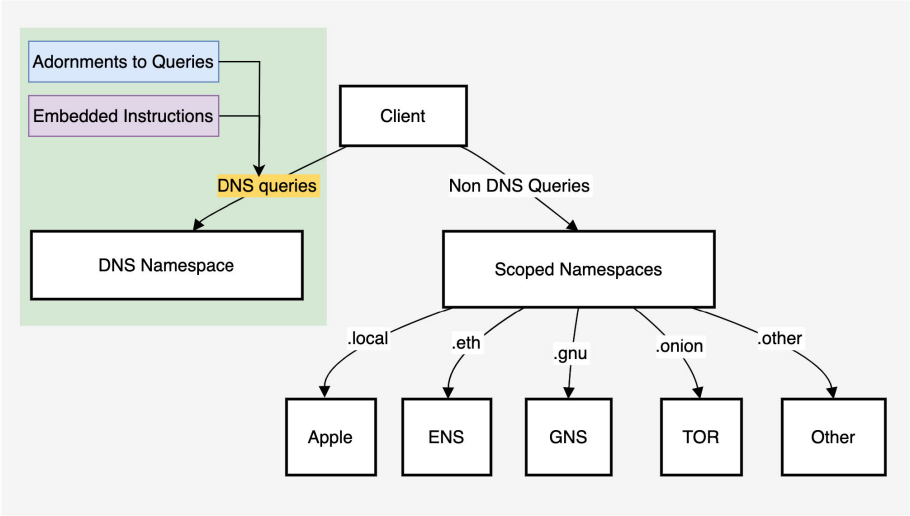


FIGURE 9. Evolutionary pressues and expanding namespaces.

Beyond these transformations within the DNS, the broader Internet landscape has been evolving outside the conventional DNS realm, as per Figure 9. These developments have led to the emergence of alternative identifier systems that can cater to specific needs and scenarios. Following the DNS-centric discussion, the scope expands to explore alternative identifier systems in their respective subsections. These include Apple's usage of the '.local' domain, TOR's '.onion' service, Ethereum's '.eth' approach, and the alternate GNU Name System. Each of these systems offers a unique perspective in addressing the Internet namespace challenges outside the DNS, providing insights into the potential evolutionary directions of digital identifier systems. Please refer to Table 2 for a summary of the subsections below.

A. METADATA AND QUERY QUALIFICATION

In conventional DNS operations, a client's identity typically remains undisclosed. However, with the advent of Extension Mechanisms for DNS (EDNS0) [118] and the inclusion of the Client Subnet (ECS), [23], a client's IP address prefix is exposed and utilized as a part of the retrieval process. A DNS query goes beyond simply requesting a name in an evolved model. It now encompasses the parameters dictated by EDNS0 extensions and the client's subnet (ECS) capability. This process necessitates a form of information trade, where the client provides its assumed identity, encapsulated in its subnet address, to receive an optimally proximate resource based on the server's estimation. Consequently, the previously invariant DNS response is now tailored to these additional query attributes. This progression introduces a distinct tension as the traditionally anonymous DNS unveils more information about the client. However, this transition also brings its own set of advantages. The precision of programmatic DNS responses can be enhanced by exposing the aspects previously concealed by the DNS, as illustrated in Figure 10.

With the advent of DNS over HTTPS (DoH), introducing query adornments could further customize the query by including various HTTP attributes. The objective is not to overload the label space with customization code but to integrate this customization into the query as adornments that can refine or limit the server's responses, ensuring each response is uniquely customized. This shift in the operational paradigm of the DNS suggests a dichotomy. On the one hand, the namespace is viewed as static, providing invariant responses. On the other hand, the DNS query is decorated with metadata that determines the identifier behavior. The server operates as a computational engine, and the DNS response is relative and customized to each query. This transformation underscores the tension between preserving client anonymity and the drive towards enhanced customization, a key challenge in the DNS and a possible evolutionary direction.

B. DNS AS COMPUTATIONAL RESOURCE

A proposal known as Oblivious DNS (DNS) [113] was introduced, although it has not been adopted by DoH or DNS over TLS (DOT) [52]. In the original DNS model, the query name undergoes encryption, yielding a new query name encoded in Base64. This encoded query name is directed not to the upper-level tree but to a specific domain name: **'obliviousdns.net.'**. This process involves a query string encrypted from the initial query. When the query reaches the oDNS server, acting as an authoritative server, it decrypts the encoded query name and retrieves the complete query name without prior knowledge of the query initiator. Essentially, the recursive resolver poses the question, not the original client. The authoritative server then switches its roles and becomes a stub resolver. It fully resolves the requested record, encrypts the result using the same key, sends it back to the recursive resolver, and subsequently delivers the answer to the client.

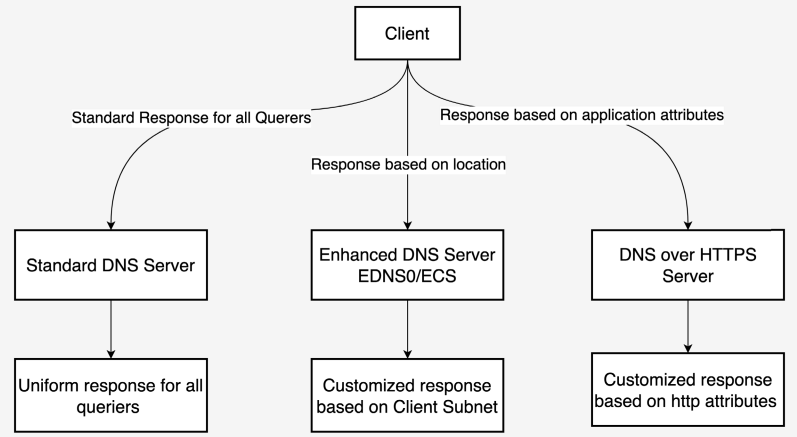


FIGURE 10. Evolution in customized responses.

The primary focus in this example is not the identity-hiding mechanism of DNS but rather the embedded information within the query name. The server assigned to that query name now performs computations on behalf of the client. This situation marks a departure from the traditional DNS lookup. It now involves the query name posing a computational task, which the server executes.

To extend this concept, a DNS server can dynamically construct the response from the context established in the query name. The query name could contain instructions that prompt a specific reaction from the server, effectively transforming the DNS server into a programmatic, systematic entity. This concept could steer the future direction of DNS. This technique allows names to become more elastic, representing a static lookup and a computational task. Consider the DNS as a microcode language, where the query effectively acts as a program for the server. In this scenario, the identifier namespace is not just a dictionary or directory but evolves into a distributed computational platform. From an evolutionary standpoint, shifting the dynamism of the namespace to convert DNS into a computational vehicle is a plausible future direction worth exploring.

C. MULTICAST DNS AND APPLES AND THE '.LOCAL' NAMESPACE

With Zero Configuration Networking (Zeroconf), Apple aimed to make technology more user-friendly by automating the discovery and communication process between devices and eliminating manual configuration. This vision was actualized through Bonjour, Apple's proprietary implementation of Zeroconf [20], [58].

Central to Bonjour's operation is the '.local' pseudo-domain, a namespace intentionally designed to streamline local networking environments. This namespace, primarily employed within smaller networks, is deliberately designed with restricted visibility to foster seamless device-to-device communication. Multicast DNS (mDNS), a protocol adept at providing name resolution services in localized networks where conventional DNS would be inefficient, was utilized to achieve this [18]. The mDNS executes multicast queries with a time-to-live (TTL) value of 1, effectively confining network traffic within its originating subnet [20]. Within this namespace, user-friendly devices such as printers can broadcast their capabilities, enhancing their discoverability by other network clients.

Unlike public DNS domains, which ensure global uniqueness, '.local' domains are designed to be link-local and not globally unique [59]. Hostnames within the '.local' domain are managed cooperatively by mDNS responders present within the local network, preserving the local naming scope and allowing the reuse of the same name in different contexts [7], [20]. In the event of a name conflict within the local network, Bonjour's design ensures automatic reassignment to a new, unique name while maintaining the integrity and consistency of network references.

However, this approach presents the Internet not as a single coherent namespace but as a plethora of local namespaces. These are intentionally limited and inaccessible through standard DNS queries in the public namespace. This unique strategy for creating application-specific namespaces brings forth challenges in distinguishing between the global namespace of DNS and these local domains. The '.local' domain is considered a special-use domain, not intended for use on the public Internet [19]. If this domain were to be delegated into the public space, it would disrupt the intentional constraints that Apple has implemented. While Apple's implementation with the '.local' domain is a noteworthy example, it is important to highlight that this strategy of localized, scoped visibility is part of a broader trend and an evolutionary pressure pushing towards a need for systems that limit the visibility of names to specific, defined sub-ranges rather than making them universally visible, as is the case with traditional DNS.

D. TOR NETWORK AND THE '.ONION' NAMESPACE

The motivation for creating The Onion Router (TOR) network was to establish an Internet environment that safeguards user privacy. The TOR architecture employs

TABLE 2. Application domain of summarized evolutionary responses.

Evolutionary Response	Namespace	Application Domain	Usecase
Metadata Enhanced DNS Queries	DNS	Content Steering, Internet Infrastructure	Optimize content delivery.
DNS as Computation Resource	DNS	Dynamic Content Delivery, Internet Infrastructure	Compute the answer, rather than the location of the answer.
Apple mDNS	.local	Local Network Services	Discovery and communication on local networks.
TOR Network	.onion	Privacy and Secure Communication	Anonymous access to TOR services
Blockchain-based Naming	.eth, .bit, .crypto, etc.	Blockchain, Decentralized Web	Provide readable names to wallets and websites.
GNS Name System	.gnu	Censorship Resistant Name Service	Alternate name system peer-to-peer network for storing and resolving names.

multilayered encryption to anonymize the end user. Each layer of encryption ensures that the user's profile remains confidential and cannot be exposed [31].

The TOR network uses public DNS infrastructure to handle regular DNS queries. However, it employs a technique to protect the identities of authoritative servers responsible for resolving popular domains on the public Internet and the identities of clients accessing the network. A name within the '.onion' namespace is derived from the public key of the asymmetric key pair [31]. When an onion service is set up, it automatically generates a private key and a corresponding public key. The public key is then hashed and truncated. The resulting hash is encoded using Base32, which yields a character string. This string is then suffixed with ''.onion' to create the fully qualified domain name of the onion service, indicating that it is accessible only over the Tor network.

To enhance the recognizability of an onion address, vanity domains provide a means of generating key pairs until the hash of the public key aligns with the desired string, thereby making the domain name reflective of the desired content [31]. It is important to note that the '.onion' domain does not rely on traditional DNS for resolution. Instead, queries for the '.onion' namespace are handled by the TOR protocol, which maps names to their corresponding network addresses within the TOR network.

The TOR was not originally intended to extend or augment the DNS. The primary goal was to respond to the pressures of providing anonymous communications. Similarly to Apple's '.local' namespace method, the '.onion' namespace is a special-use Top-Level Domain (TLD) that serves a distinct purpose [4]. It is deliberately designed to avoid resolution through the traditional DNS infrastructure. This deliberate separation ensures that the hidden TOR services remain concealed within the TOR network and are not exposed to the global public DNS namespace. Consequently, the Special-Use Domain Name Registry maintains a registry for these types of domain names reserved for special purposes, ensuring they are not utilized for general purposes.

E. ETHEREUM NAME SERVICE AND THE '.ETH' NAMESPACE

Enforcement of uniqueness is a key aspect of online identities and transactions. This is typically maintained via centralized systems, where names are recorded on a ledger, ensuring that another cannot claim it once a name is used. However, alternatives exist to this standard model, one of which is Blockchain-based systems. For instance, the Ethereum Name Service (ENS) leverages smart contracts to manage the decentralized registry of '.eth' domain names across the Ethereum network [135].

The ENS manages domains through three smart contracts: the Registry, Registrar, and Resolver. The ENS Registry contains three key pieces of information: the owner of the ENS domain, the Resolver associated with the domain, and the time-to-live TTL for the domain records. The Registrar contract owns and allocates domains and subdomains under pre-defined rules in its smart contract. Finally, the Resolver synthesizes the domain names into associate pointers, such as cryptocurrency addresses [100].

The ENS system ensures the uniqueness of each domain name through an implicit collision mechanism, which is activated when there is an attempt to register a claimed name. This decentralized approach mirrors the functionality of a central entity coordinating the uniqueness of domain names [136]. However, it is important to note that while the ENS operates on a decentralized framework, decisions regarding the registration, renewal and governance of top-level domains (TLDs) such as '.eth.' are centralized.

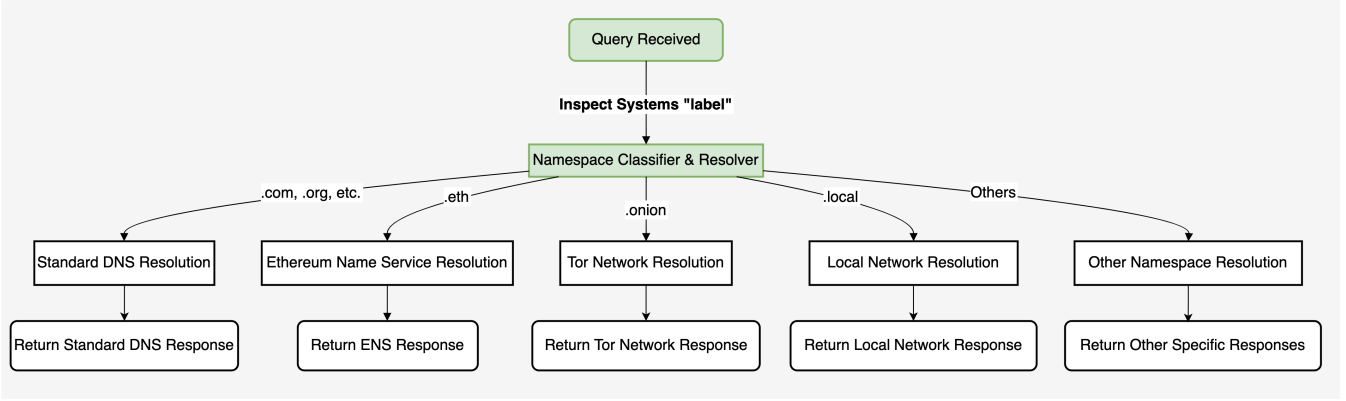


FIGURE 11. Classifier for multi namespace forwarding.

This distinctive architecture presents an intriguing set of evolutionary pressures with advantages and disadvantages. Decentralized systems, such as ENS, offer benefits such as increased user pseudonymity. However, the inherent transparency of these systems can become challenging in public communication settings, distinct from private ones, where the lack of complete anonymity can pose significant issues. Striking a balance between pseudonymity, where users have a persistent but not directly identifiable presence, and full anonymity, analogous to sending mail without a return address, often leads to societal dilemmas. Consequently, a degree of residual accountability, such as a return address for mailed letters, is often seen as a positive attribute in public systems.

F. GNU NAME SYSTEM AND THE '.GNU' NAMESPACE

GNS uses '.gnu,' a pseudo-top-level domain (TLD), to distinguish names resolved in the GNS namespace from those resolved in the DNS namespace [112]. This unique structure leads to a distinctive approach in name resolution, where, unlike hierarchical DNS, every user's zone is treated as an independent root. This results in all zones becoming equal starting points for the resolution process [129]. The resolution process in the GNS is transitive, following delegations from one zone to another based on public keys. These keys, pivotal for navigating different zones, are located using a Distributed Hash Table (DHT), which is a decentralized data structure that efficiently maps keys to values [129].

In contrast, DNS uses an iterative resolution process, which begins at a globally recognized root zone and continues through a sequence of authoritative servers from the TLD to the subdomain to resolve the IP address of the requested resource. Although both systems use delegation, their resolution methodologies are fundamentally different. Like DNS, GNS allows owners complete control over their namespaces. Additionally, users can delegate a part of their namespace, requiring any label within that domain to map to another identity. This feature mirrors DNS, easing the transition to this alternative architecture [129].

The emergence of alternative naming systems outside the traditional DNS has amplified evolutionary tensions toward the interoperability of various Internet namespaces. For instance, the '.gnu' TLD is a distinguishing label that prompts a GNS-aware resolver to transition from the traditional DNS resolution process to the GNS protocol. This mechanism allows names intended for resolution in the GNS namespace to be handled correctly, thus mitigating conflicts with DNS and other name systems.

G. INNOVATION VERSUS INTEGRATION: A DISCUSSION

The use of names within references determines how resources are accessed. A name passed between two entities should resolve to the same digital resource. This is only achievable if names are allocated from a unified namespace and the resolution of the name into network addresses is consistent and coherent. Market competition dynamics for digital presence naturally encourage all participants to be bound within this unified namespace residing in the DNS. Moreover, for the Internet to continue functioning as a public communications system, it is essential to remain within the same namespace to maintain coherence. This study outlines examples of alternate namespaces and notes the caveat that the parallel use of such alternate name systems risks fragmenting the Internet's namespace, eroding the value of a unified and cohesive communication network.

In response to support cohesion across diverse namespaces, one approach involves augmenting name resolvers with classifier functionality for various name suffixes such as '.eth,' '.bit,' '.onion,' '.gnu,' and '.local.'. As described in Figure 11, this method introduces a distinguisher label to specify the namespace that is used. From an architectural point of view, this approach may provide a potential avenue toward interoperability among multiple namespaces, necessitating minimal changes to the existing global DNS infrastructure [54]. However, the challenge lies in determining who decides which distinguisher labels are used by each of these namespaces. This will only be an effective approach with a coherent and universally accepted method for organizing such distinguishing labels.

VI. CONCLUSION

The evolutionary pressures that shaped the alternative namespace have not strayed far from the root of the DNS and remain fundamentally mappable to the DNS; the key challenge lies in preserving cohesiveness.

The foundation of trust in Internet interactions primarily stems from confidence in the naming system, a necessity made more acute by the indeterminism of the routing and addressing system, influenced in the latter case by IPv4 address exhaustion. In this context, the trust placed in Internet transactions relies on the exchange of credentials between the engaging parties. This crucial process enables the receiving party to verify that the responses they receive are from an entity that controls the domain name of the accessed service.

Navigating the evolution of Internet identifiers is intricate, as it is critical to accommodate various aspects of technical innovation in a gradual and incremental manner while simultaneously preserving the coherence of the identifier space. These may appear to be conflicting goals, as they often stand in opposition.

REFERENCES

[1] (Jul. 2017). *A Look at Quic Use*. [Online]. Available: https://www.potaroo.net/ispcol/2022-07/quic.html
[2] (2007). *About URIs*. [Online]. Available: https://www.loc.gov/standards/uri/about.html
[3] A. Adya, W. J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J. R. Douceur, J. Howell, J. R. Lorch, M. Theimer, and R. P. Wattenhofer, "FARSITE: Federated, available, and reliable storage for an incompletely trusted environment," *ACM SIGOPS Operating Syst. Rev.*, vol. 36, no. SI, pp. 1–14, 2002.
[4] J. Appelbaum and A. Muffett, *The 'Onion' Special-Use Domain Name*, document RFC 7686, Oct. 2015. [Online]. Available: https://www.rfc-editor.org/info/rfc7686
[5] A. Babakian, P. Monclus, R. Braun, and J. Lipman, "A retrospective on workload identifiers: From data center to cloud-native networks," *IEEE Access*, vol. 10, pp. 105518–105527, 2022.
[6] T. C. Bartee, *ISDN, DECnet, and SNA Communications*. USA: H. W. Sams, 1989, p. 414.
[7] M. Bartosh and R. Faas, *Essential Mac OS X Panther Server Administration: Integrating Mac OS X Server Into Heterogeneous Networks*. USA: O'Reilly Media, 2009.
[8] T. Berners-Lee, *Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as Used in the World-Wide Web*, document RFC 1630, Internet Requests for Comments, Jun. 1994.
[9] T. Berners-Lee. (Aug. 1996). *The World Wide Web: Past, Present and Future*. [Online]. Available: https://www.w3.org/People/Berners-Lee/1996/ppf.html
[10] T. Berners-Lee, R. T. Fielding, and L. Masinter, *Uniform Resource Identifier (URI): Generic Syntax*, document STD 66, RFC3986, Internet Requests for Comments, RFC Editor, Jan. 2005. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3986.txt. http://www.rfc-editor.org/rfc/rfc3986.txt
[11] T. Berners-Lee, L. Masinter, and M. McCahill, *Uniform Resource Locators (URL)*, document RFC 1738, Internet Requests for Comments, RFC Editor, Dec. 1994. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1738.txt
[12] M. Bishop, *HTTP/3*, document RFC 9114, Jun. 2022. [Online]. Available: https://www.rfc-editor.org/info/rfc9114
[13] M. Bowman, S. K. Debray, and L. L. Peterson, "Reasoning about naming systems," *ACM Trans. Program. Lang. Syst.*, vol. 15, no. 5, pp. 795–825, 1993.
[14] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, I. Stoica, and S. Shenker, "ROFL: Routing on flat labels," in *Proc. Conf. Appl., Technol., Architectures, Protocols Comput. Commun.*, 2006, pp. 363–374.
[15] M. Calder, A. Flavel, E. Katz-Bassett, R. Mahajan, and J. Padhye, "Analyzing the performance of an anycast CDN," in *Proc. Internet Meas. Conf.*, 2015, pp. 531–537.
[16] B. E. Carpenter and B. E. Carpenter. (Jun. 2007). *General Identifier-Locator Mapping Considerations*. Internet Eng. Task Force, Internet-Draft. [Online]. Available: https://datatracker.ietf.org/doc/draft-carpenter-idloc-map-cons/01/
[17] L. M. Chan, S. S. Intner, and J. Weihs, *Guide To the Library of Congress Classification*. USA: ABC-CLIO, 2016.
[18] S. Cheshire and M. Krochmal, *Multicast DNS*, document RFC 6762, Feb. 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc6762
[19] S. Cheshire and M. Krochmal, *Special-Use Domain Names*, document RFC 6761, Feb. 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc6761
[20] S. Cheshire and D. Steinberg, *Zero Configuration Networking: The Definitive Guide*. U.K.: O'Reilly Media, 2006.
[21] D. E. Comer and L. L. Peterson, "Issues in using DARPA domain names for computer mail," in *Proc. 9th Symp. Data Commun.*, 1985, pp. 158–164.
[22] *The Relationship Between URNs, Handles, and Purls*, Library of Congress, Washington, DC, USA, 1997.
[23] C. Contavalli, W. van der Gaast, D. C. Lawrence, and W. A. Kumari, *Client Subnet in DNS Queries*, document RFC 7871, May 2016. [Online]. Available: https://www.rfc-editor.org/info/rfc7871
[24] G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems: Concepts and Design*, 2nd ed. Boston, MA, USA: Addison-Wesley, 2001, p. 779.
[25] National Research Council, *Signposts in Cyberspace: The Domain Name System and Internet Navigation*. USA: National Academies Press, 2005.
[26] D. Crocker, *DNS Attrleaf Changes: Fixing Specifications That Use Underscored Node Names*, document RFC 8553, Mar. 2019. [Online]. Available: https://www.rfc-editor.org/info/rfc8553
[27] D. Crocker, *Scoped Interpretation of DNS Resource Records Through 'Underscored' Naming of Attribute Leaves*, document RFC 8552, Mar. 2019. [Online]. Available: https://www.rfc-editor.org/info/rfc8552
[28] D. R. Daniel and M. H. Mealling, *Resolution of Uniform Resource Identifiers Using the Domain Name System*, document RFC 2168, Jun. 1997. [Online]. Available: https://www.rfc-editor.org/info/rfc2168
[29] D. S. E. Deering and B. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, document RFC 8200, Jul. 2017. [Online]. Available: https://www.rfc-editor.org/info/rfc8200
[30] L. Deutsch, *Host Names On-Line*, document RFC 606, Internet Requests for Comments, RFC Editor, Dec. 1973.
[31] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proc. 13th USENIX Secur. Symp. (SSYM)*, Aug. 2004, pp. 303–320.
[32] *Domain Names—Concepts and Facilities*, document RFC 1034, Nov. 1987. [Online]. Available: https://www.rfc-editor.org/info/rfc1034
[33] *Domain Names—Implementation and Specification*, document RFC 1035, Nov. 1987. [Online]. Available: https://www.rfc-editor.org/info/rfc1035
[34] A. Durand, "Digital object architecture and the handle system," ICANN| Office CTO, Los Angeles, CA, USA, 2019. Accessed: Dec. 7, 2023. [Online]. Available: https://www.icann.org/en/system/files/files/octo-002-14oct19-en.pdf
[35] R. Elz and R. Bush, *Clarifications to the DNS Specification*, document RFC 2181, Jul. 1997. [Online]. Available: https://www.rfc-editor.org/info/rfc2181
[36] M. P. Evans and S. Furnell, "The resource locator service: Fixing a flaw in the web," *Comput. Netw.*, vol. 37, nos. 3–4, pp. 307–330, 2001.
[37] P. Fältström and G. Huston. (Dec. 2004). *A Survey of Internet Identities*. Internet Engineering Task Force, Internet-Draft. [Online]. Available: https://datatracker.ietf.org/doc/draft-iab-identities/02/
[38] D. Farinacci, V. Fuller, D. Meyer, D. Lewis, and A. Cabellos-Aparicio, *The Locator/ID Separation Protocol (LISP)*, document RFC 9300, Oct. 2022. [Online]. Available: https://www.rfc-editor.org/info/rfc9300
[39] S. Farrell and H. Tschofenig, *Pervasive Monitoring is an Attack*, document RFC 7258, May 2014. [Online]. Available: https://www.rfc-editor.org/info/rfc7258
[40] R. T. Fielding, J. Gettys, J. C. Mogul, H. F. Nielsen, L. Masinter, P. J. Leach, and T. Berners-Lee, *Hypertext Transfer Protocol—HTTP/1.1*, document RFC 2616, Internet Requests for Comments, RFC Editor, Jun. 1999. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2616.txt

[41] P. Francis, *Pip Near-Term Architecture*, document RFC 1621, May 1994. [Online]. Available: https://www.rfc-editor.org/info/rfc1621

[42] S. Friedl, A. Popov, A. Langley, and S. Emile, *Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension*, document RFC 7301, Jul. 2014. [Online]. Available: https://www.rfc-editor.org/info/rfc7301

[43] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing," *Commun. ACM*, vol. 42, no. 2, pp. 39–41, 1999.

[44] S. Griffiths, "ISBN: A history," *Inf. Standards Quart.*, vol. 27, nos. 2–3, pp. 46–48, 2015.

[45] A. Gulbrandsen and D. L. Esibov, *A DNS RR for Specifying the Location of Services (DNS SRV)*, document RFC 2782, Feb. 2000. [Online]. Available: https://www.rfc-editor.org/info/rfc2782

[46] J. Hakala, *Using National Bibliography Numbers As Uniform Resource Names*, document RFC 8458, Internet Requests for Comments, RFC Editor, Oct. 2018.

[47] J. Hakala and H. Walravens, *Using International Standard Book Numbers As Uniform Resource Names*, document RFC 3187, Internet Requests for Comments, RFC Editor, Oct. 2001.

[48] J. Hakala, "Persistent identifiers—An overview," Nat. Library Finland, Helsinki, Finland, 2010.

[49] K. Harrenstien, M. Stahl, and E. Feinler, *DoD Internet Host Table Specification*, document RFC 952, Oct. 1985.

[50] M. Hauben, *History of Arpanet*, vol. 17. Porto, Portugal: L'Instituto Superior de Engenharia do Porto, 2007, pp. 1–20.

[51] T. Howes, M. Smith, and G. S. Good, *Understanding and Deploying LDAP Directory Services*. Reading, MA, USA: Addison-Wesley, 2003.

[52] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. E. Hoffman, *Specification for DNS Over Transport Layer Security (TLS)*, document RFC 7858, May 2016. [Online]. Available: https://www.rfc-editor.org/info/rfc7858

[53] G. Huston, "DNS evolution: Innovation or fragmentation?" 2022. Accessed: Nov. 27, 2023. [Online]. Available: https://circleid.com/posts/20221001-dns-evolutioninnovation- or-fragmentation

[54] G. Huston, P. Koch, A. Durand, and W. A. Kumari. (Sep. 2016). *Problem Statement for the Reservation of Special-Use Domain Names Using RFC6761*. Internet Engineering Task Force, Internet-Draft. [Online]. Available: https://datatracker.ietf.org/doc/draft-adpkja-dnsop-special-names-problem/06/

[55] A. Hutton, J. Uberti, and M. Thomson, *The ALPN HTTP Header Field*, document RFC 7639, Aug. 2015. [Online]. Available: https://www.rfc-editor.org/info/rfc7639

[56] *IAB Technical Comment on the Unique DNS Root*, document RFC 2826, Internet Requests for Comments, RFC Editor, May 2000.

[57] (Dec. 2021). *iCloud Private Relay Overview*. [Online]. Available: https://www.apple.com/privacy/docs/iCloudPrivateRelayOverviewDec2021.PDF

[58] Apple Inc. (2013). *Net Services Programming Guide*. Accessed: Jul. 24, 2023. [Online]. Available: https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/NetServices/Introduction.html

[59] Apple Inc. (2013). *Resolving DNS Domain Names*. Accessed: Jul. 24, 2023. [Online]. Available: https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/NetServices/Articles/domainnames.html

[60] J. Iyengar and M. Thomson, *QUIC: A UDP-Based Multiplexed and Secure Transport*, document RFC 9000, May 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9000

[61] J. Iyengar, C. Raiciu, S. Barre, M. J. Handley, and A. Ford, *Architectural Guidelines for Multipath TCP Development*, document RFC 6182, Mar. 2011. [Online]. Available: https://www.rfc-editor.org/info/rfc6182

[62] G. Janée, "Resource identifier," in *Encyclopedia of Database Systems*. Santa Barbara, CA, USA: Institute for Computational Earth System Science, Univ. of California at Santa Barbara, 2009.

[63] R. Kahn and R. Wilensky, "A framework for distributed digital object services," *Int. J. Digit. Libraries*, vol. 6, no. 2, pp. 115–123, Apr. 2006.

[64] R. Kahn and R. Wilensky, "A framework for distributed digital object services," *Int. J. Digit. Libraries*, vol. 6, no. 2, pp. 115–123, 2006.

[65] P. Karp, *Standardization of Host Mnemonics*, document RFC 226, Internet Requests for Comments, RFC Editor, Sep. 1971.

[66] M. Komu, M. Sethi, and N. Beijar, "A survey of identifier–locator split addressing architectures," *Comput. Sci. Rev.*, vol. 17, pp. 25–42, Aug. 2015.

[67] M. Kudlick, *Host Names On-line*, document RFC 608, Internet Requests for Comments, RFC Editor, Jan. 1974.

[68] R. Kumbhar, *Library Classification Trends in the 21st Century*. Amsterdam, The Netherlands: Elsevier, 2011.

[69] G. Lavender and M. Wahl, "Internet directory services using the lightweight directory access protocol," in *Practical Handbook Internet Comput.*, Sep. 2004, doi: 10.1201/9780203507223.pt4.

[70] E. Lear and R. Droms. (2003). *What's In A Name: Thoughts From the NSRG*. Internet Engineering Task Force, Internet-Draft. [Online]. Available: https://datatracker.ietf.org/doc/draft-irtf-nsrg-report/10/

[71] T. Li, *Recommendation for a Routing Architecture*, document RFC 6115, Feb. 2011. [Online]. Available: https://www.rfc-editor.org/info/rfc6115

[72] T. Li, D. Farinacci, S. P. Hanks, D. Meyer, and P. S. Traina, *Generic Routing Encapsulation (GRE)*, document RFC 2784, Mar. 2000. [Online]. Available: https://www.rfc-editor.org/info/rfc2784

[73] C. Liu and P. Albitz, *DNS Bind*. USA: O'Reilly Media, 2006.

[74] M. Mealling and R. Denenberg, *Report From the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations*, document RFC 3305, Internet Requests for Comments, RFC Editor, Aug. 2002.

[75] M. H. Mealling and D. R. Daniel, *The Naming Authority Pointer (NAPTR) DNS Resource Record*, document RFC 2915, Sep. 2000. [Online]. Available: https://www.rfc-editor.org/info/rfc2915

[76] N. Meulemans, "A yellow pages service based on X.500," *Comput. Netw. ISDN Syst.*, vol. 28, no. 14, pp. 1939–1946, Nov. 1996.

[77] D. Meyer, L. Zhang, and K. Fall, *Report From the IAB Workshop on Routing and Addressing*, document RFC 4984, Internet Requests for Comments, RFC Editor, Sep. 2007. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4984.txt

[78] D. Mills, *Internet Name Domains*, document RFC 799, Internet Requests for Comments, RFC Editor, Sep. 1981.

[79] P. Mockapetris and K. J. Dunlap, "Development of the domain name system," in *Proc. Symp. Commun. Architectures Protocols*, 1988, pp. 123–133.

[80] R. Moskowitz, P. Jokela, T. Henderson, and P. Nikander, *Host Identity Protocol*, document RFC 5201, Apr. 2008. [Online]. Available: https://www.rfc-editor.org/info/rfc5201

[81] (May 2020). *Multiplexed Application Substrate Over Quic Encryption*. [Online]. Available: https://datatracker.ietf.org/doc/charter-ietf-masque/

[82] D. T. Narten, R. P. Draves, and S. Krishnan, *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, document RFC 4941, Sep. 2007. [Online]. Available: https://www.rfc-editor.org/info/rfc4941

[83] G. W. Neufeld, "Descriptive names in X.500," in *Proc. Symp. Commun. Architectures Protocols*, Aug. 1989, pp. 64–71.

[84] A. Newton and L. Daigle, *Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)*, document RFC 3958, Jan. 2005. [Online]. Available: https://www.rfc-editor.org/info/rfc3958

[85] P. Nikander, A. Gurtov, and T. R. Henderson, "Host identity protocol (HIP): Connectivity, mobility, multi-homing, security, and privacy over IPv4 and IPv6 networks," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 2, pp. 186–204, 2nd Quart., 2010.

[86] P. Nikander and R. Moskowitz. (Dec. 2009). *Host Identity Protocol Architecture*. Internet Engineering Task Force, Internet-Draft. [Online]. Available: https://datatracker.ietf.org/doc/draft-moskowitz-rfc4423-bis/00/

[87] E. Nordström, D. Shue, P. Gopalan, R. Kiefer, M. Arye, S. Y. Ko, J. Rexford, and M. J. Freedman, "Serval: An end-host stack for service-centric networking," in *Proc. 9th USENIX Symp. Networked Syst. Des. Implement. (NSDI)*, 2012, pp. 85–98.

[88] N. Paskin, "Digital object identifier (DOIő) system," in *Encyclopedia of Library and Information Sciences*, vol. 3. U.K.: Taylor & Francis, 2010, pp. 1586–1592.

[89] S. Paul, J. Pan, and R. Jain, "A survey of naming systems: Classification and analysis of the current schemes using a new naming reference model," Citeseer, WUSTL Tech. Rep., 2009.

[90] T. Pauly, E. Rosenberg, and D. Schinazi. (Mar. 2023). *QUIC-Aware Proxying Using HTTP*. Internet Engineering Task Force, Internet-Draft. [Online]. Available: https://datatracker.ietf.org/doc/draft-pauly-masque-quic-proxy/06/

[91] G. Peng, "CDN: Content distribution network," 2004, *arXiv:cs/0411069*.

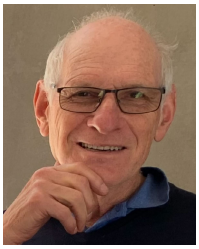
[92] C. Perkins, *IP Mobility Support for IPv4*, document RFC 3344, Internet Requests for Comments, RFC Editor, Aug. 2002. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3344.txt

[93] C. Perkins, D. Johnson, and J. Arkko, *Mobility Support in IPv6*, document RFC 6275, Internet Requests for Comments, RFC Editor, Jul. 2011. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6275.txt

[94] C. E. Perkins, "Mobile IP," *IEEE Commun. Mag.*, vol. 35, no. 5, pp. 84–99, May 1997.
[95] C. E. Perkins, *IP Encapsulation Within IP*, document RFC 2003, Oct. 1996. [Online]. Available: https://www.rfc-editor.org/info/rfc2003
[96] C. E. Perkins, *IP Mobility Support*, document RFC 2002, Oct. 1996. [Online]. Available: https://www.rfc-editor.org/info/rfc2002
[97] R. Perlman, *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*. Reading, MA, USA: Addison-Wesley, 2000.
[98] D. M. Piscitello and A. L. Chapin, *Open Systems Networking: TCP/IP and OSI*. Reading, MA, USA: Addison-Wesley, 1993.
[99] J. Postel, *Internet Protocol*, document STD 5, Internet Requests for Comments, RFC Editor, Sep. 1981. [Online]. Available: http://www.rfc-editor.org/rfc/rfc791.txt
[100] B. Rajendran, G. Palaniappan, R. Dijesh, and S. D. Sudarsan, "A universal domain name resolution service—Need and challenges–Study on blockchain based naming services," in *Proc. IEEE Region 10 Symp. (TENSYMP)*, Jul. 2022, pp. 1–6.
[101] W. Ramirez, X. Masip-Bruin, M. Yannuzzi, R. Serral-Gracia, A. Martinez, and M. S. Siddiqui, "A survey and taxonomy of ID/Locator split architectures," *Comput. Netw.*, vol. 60, pp. 13–33, Feb. 2014.
[102] *X. 509| ISO/IEC 9594-8*, ITUT Rec Inf. Technology-Open Syst. Interconnection, The Directory: Authentication Framework, 1993.
[103] Y. Rekhter, J. Crowcroft, and B. E. Carpenter, *IPv4 Address Behaviour Today*, document RFC 2101, Feb. 1997. [Online]. Available: https://www.rfc-editor.org/info/rfc2101
[104] E. Rescorla, *HTTP Over TLS*, document RFC 2818, Internet Requests for Comments, RFC Editor, May 2000. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2818.txt
[105] J. K. Reynolds, S. Heker, and C. Weider, *Technical Overview of Directory Services Using the X.500 Protocol*, document RFC 1309, Mar. 1992. [Online]. Available: https://www.rfc-editor.org/info/rfc1309
[106] M. T. Rose, *The Open Book: A Practical Perspective on OSI*. Upper Saddle River, NJ, USA: Prentice-Hall, 1990.
[107] S. Rose and W. Wijngaards, *DNAME Redirection in the DNS*, document RFC 6672, Jun. 2012. [Online]. Available: https://www.rfc-editor.org/info/rfc6672
[108] S. Rozenfeld, *Using the ISSN (international Serial Standard Number) As URN (uniform Resource Names) Within an ISSN-URN Namespace*, document RFC 3044, Internet Requests for Comments, RFC Editor, Jan. 2001.
[109] P. Saint-Andre and J. Klensin, *Uniform Resource Names (URNs)*, document RFC 8141, Internet Requests for Comments, RFC Editor, Apr. 2017.
[110] J. H. Saltzer, *On the Naming and Binding of Network Destinations*, document RFC 1498, Aug. 1993. [Online]. Available: https://www.rfc-editor.org/info/rfc1498
[111] P. Sattler, J. Aulbach, J. Zirngibl, and G. Carle, "Towards a tectonic traffic shift: Investigating Apple's new relay network," in *Proc. 22nd ACM Internet Meas. Conf.*, Oct. 2022, pp. 449–457.
[112] M. Schanzenbach, C. Grothoff, and B. Fix, "The GNU name system," RFC Editor, Tech. Rep. RFC 9498, p. 74, Nov. 2023. [Online]. Available: https://www.rfc-editor.org/info/rfc9498
[113] P. Schmitt, A. Edmundson, A. Mankin, and N. Feamster, "Oblivious DNS: Practical privacy for DNS queries: Published in PoPETS 2019," in *Proc. Appl. Netw. Res. Workshop*, Jul. 2019, pp. 17–19.
[114] K. Schomp, O. Bhardwaj, E. Kurdoglu, M. Muhaimen, and R. K. Sitaraman, "Akamai DNS: Providing authoritative answers to the World's queries," in *Proc. Annu. Conf. ACM Special Interest Group Data Commun. Appl., Technol., Architectures, Protocols Comput. Commun.*, Jul. 2020, pp. 465–478.
[115] B. M. Schwartz, M. Bishop, and E. Nygren. (Mar. 2023). *Service Binding and Parameter Specification via the DNS (DNS SVCB and HTTPS RRs)*. Internet Engineering Task Force, Internet-Draft. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-dnsop-svcb-https/12/
[116] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.
[117] J. Shoch, "Inter-network naming, addressing, and routing," in *Proc. IEEE COMPCON Fall*, 1978, pp. 72–79.
[118] J. da Silva Damas, M. Graff, and P. A. Vixie, *Extension Mechanisms for DNS (EDNS(0))*, document RFC 6891, Apr. 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc6891
[119] D. L. Silver, J. W. Hong, and M. A. Bauer, "X.500 directory schema management," in *Proc. IEEE 10th Int. Conf. Data Eng.*, Feb. 1994, pp. 393–400.
[120] H. A. Simon, "The architecture of complexity," *Proc. Amer. Philos. Soc.*, vol. 106, no. 6, pp. 467–482, 1962.
[121] A. C. Snoeren and H. Balakrishnan, "An end-to-end approach to host mobility," in *Proc. 6th Annu. Int. Conf. Mobile Comput. Netw.*, Aug. 2000, pp. 155–166.
[122] A. C. Snoeren, H. Balakrishnan, and M. F. Kaashoek, "Reconsidering Internet mobility," in *Proc. 8th Workshop Hot Topics Operating Syst.*, 2001, pp. 41–46.
[123] K. Sollins, *Architectural Principles of Uniform Resource Name Resolution*, document RFC 2276, Internet Requests for Comments, RFC Editor, Jan. 1998.
[124] S. Sun, L. Lannom, and B. Boesch, *Handle System Overview*, document RFC 3650, Internet Requests for Comments, RFC Editor, Nov. 2003.
[125] S. Sun, S. Reilly, and L. Lannom, *Handle System Namespace and Service Definition*, document RFC 3651, Internet Requests for Comments, RFC Editor, Nov. 2003.
[126] (2023). *Uniform Resource Names (URN) Namespaces*. [Online]. Available: https://www.iana.org/assignments/urn-namespaces/urn-namespaces.xhtml
[127] A. Venkataramani, J. F. Kurose, D. Raychaudhuri, K. Nagaraja, M. Mao, and S. Banerjee, "MobilityFirst: A mobility-centric and trustworthy Internet architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 74–80, 2014.
[128] P. Vixie, "What DNS is not," *Commun. ACM*, vol. 52, no. 12, pp. 43–47, Nov. 2009.
[129] M. Wachs, M. Schanzenbach, and C. Grothoff, "A censorship-resistant, privacy-enhancing and fully decentralized name system," in *Proc. 13th Int. Conf. Cryptol. Netw. Security (CANS)*, col. 13. Crete, Greece: Springer, Oct. 2014, pp. 127–142.
[130] R. Wakikawa, Z. Zhu, and L. Zhang, *A Survey of Mobility Support in the Internet*, document RFC 6301, Jul. 2011. [Online]. Available: https://www.rfc-editor.org/info/rfc6301
[131] B. Walker, G. Popek, R. English, C. Kline, and G. Thiel, "The LOCUS distributed operating system," *ACM SIGOPS Operating Syst. Rev.*, vol. 17, no. 5, pp. 49–70, Dec. 1983.
[132] C. Weider and R. Wright, "A survey of advanced usages of X.500," RFC Editor, Request Comments, Tech. Rep. RFC 1491, p. 18, Jul. 1993. [Online]. Available: https://www.rfc-editor.org/info/rfc1491
[133] *Naming and Name Management Systems: A Survey of the State of the Art*, Dept. Comput. Sci., Univ. Western Ontario, London, U.K., 1989.
[134] W. A. Wiegand, "The 'Amherst method': The origins of the Dewey decimal classification scheme," *Libraries Culture*, vol. 33, no. 2, pp. 175–194, 1998.
[135] P. Xia, H. Wang, Z. Yu, X. Liu, X. Luo, G. Xu, and G. Tyson, "Challenges in decentralized name management: The case of ENS," in *Proc. 22nd ACM Internet Meas. Conf.*, Oct. 2022, pp. 65–82.
[136] P. Xia, H. Wang, Z. Yu, X. Liu, X. Luo, and G. Xu, "Ethereum name service: The good, the bad, and the ugly," 2021, *arXiv:2104.05185*.
[137] L. Zhang, A. J. Mankin, J. W. Stewart III, D. T. Narten, and D. M. Crawford. (Oct. 1999). *Separating Identifiers and Locators in Addresses: An Analysis of the GSE Proposal for IPv6*. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-ipngwg-esd-analysis/05/
[138] J. Zirngibl, P. Sattler, and G. Carle, "A first look at SVCB and HTTPS DNS resource records in the wild," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroS&PW)*, 2023, pp. 470–474.

ANDREW BABAKIAN is currently pursuing the Ph.D. degree in engineering with the University of Technology Sydney, Australia. He has held numerous technical leadership positions driving innovation efforts in software-defined networking and identity technologies for cloud-native architectures. His current research interests include naming systems and the evolution of Internet identifiers.

GEOFF HUSTON received the A.M., B.SC., and M.Sc. degrees. He was an Internet Researcher and an ISP Systems Architect with National Telco and a network operator at various times. He is currently the Chief Scientist with Asia-Pacific Network Information Centre (APNIC), the Regional Internet Registry that serves the Asia–Pacific Region. He has been closely involved with the technical evolution of the internet for many years, particularly within Australia, where he was responsible for building the internet within Australian Academic and Research Sector, in 1990. He is the author of several internet-related books. He was a member of the Internet Architecture Board, from 1999 to 2005. He served on the Board of Trustees of the Internet Society, from 1992 to 2001, and chairing several IETF Working Groups.



ROBIN BRAUN (Life Senior Member, IEEE) received the B.Sc. degree (Hons.) from Brighton University, Brighton, U.K., in 1980, and the M.Sc. (Eng.) and Ph.D. degrees from the University of Cape Town, Cape Town, South Africa, in 1982 and 1986, respectively. In 1986, he started his academic career at the University of Cape Town. In 1998, he moved to the University of Technology Sydney, Australia, where he occupied the Chair of Telecommunications Engineering. Prior to moving to academia, he spent ten years in industry, mostly with Philips and Plessey, where he worked on the design of precision electronic distance measuring equipment. He is currently an Emeritus Professor with the School of Electrical and Data Engineering. His recent work has been in network protocols and the management of complex next-generation networks. He is very active in software-defined networks. He is a Founder Member of Australia and New Zealand Software Defined Networking (ANZSDN).

JUSTIN LIPMAN (Senior Member, IEEE) received the Ph.D. degree in telecommunications engineering from the University of Wollongong, Australia, in 2004. He is currently an Industry Associate Professor with the University of Technology Sydney (UTS) and a Visiting Associate Professor with the Graduate School of Engineering, Hokkaido University. He is also the Director of Research Translation with the Faculty of Engineering and IT and the Director of the RF Communications Technologies (RFCT) Laboratory, where he leads industry engagement in RF technologies, cybersecurity, privacy-preserving technologies, the Internet of Things, and tactile internet. He serves as a Committee Member in Standards Australia contributing to the International IoT Standards and Digital Twins. Previously, he was the Deputy Chief Scientist of the Food Agility Cooperative Research Center. Prior to joining UTS, over a 12 year period, he held several senior management and technical leadership roles at Intel and Alcatel, driving research and innovation, product development, architecture, and IP generation. His research interests include enabling ''things'' to be adaptive, connected, distributed, ubiquitous, and secure.