Network Working Group X. Lee
Internet-Draft B. Peng
Intended status: Standards Track J. Lee
Expires: 8 May 2026Institute of Computing Technology, Chinese Academy of Sciences
4 November 2025

DNS-based Resolution of Heterogeneous Identifiers (Hi-DNS)
draft-lee-dns-hidns-00

Abstract

Formally, the Domain Name System (DNS) has the capability to resolve any registered name to its associated data. However, current heterogeneous identifiers (such as DOI, Handle, Ecode, etc.) rely on isolated and incompatible resolution systems, leading to fragmented client implementations and failing to fully leverage the existing Internet infrastructure. This specification defines an operational practice that achieves DNS-based heterogeneous identifier resolution by mapping identifier values to dedicated DNS namespaces under the authority of their respective identifier systems. This mapping follows standardized rules compatible with the Internationalized Domain Names in Applications (IDNA) framework, allowing identifiers to be converted into domain names that can be resolved through standard DNS queries (for A, AAAA, or CNAME records). This approach establishes a unified semantic scope for identifier resolution, providing clients with a single, consistent access interface. Its primary advantage lies in integrating diverse identifier systems into the robust, scalable, and globally deployed DNS infrastructure, thereby simplifying client development and improving overall resolution efficiency.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 8 May 2026.

Table of Contents

1. Introduction

1.1. Problem Statement

In modern data infrastructures, entities such as subjects, platforms, and data products are typically identified by different types of heterogeneous identification systems, such as unified social credit codes, ID numbers, mobile phone numbers, Ecode, Handle, DOI, and so on. These identification systems operate independently, employing different naming rules, resolution protocols, and management authorities, making cross-system and cross-domain identity authentication, resource addressing, and service discovery highly complex. Currently, there is a lack of a unified, open, and interoperable resolution mechanism, which forces clients to integrate multiple proprietary resolution libraries, increasing development and maintenance costs. Moreover, existing resolution systems have not fully leveraged the widely deployed and highly reliable Domain Name System (DNS) infrastructure on the internet, leading to resource redundancy and inefficiency.

To achieve the goal of 'one-time identification, global resolution,' it is urgent to build a public resolution framework capable of integrating various heterogeneous identifiers. This framework should be highly scalable, strongly compatible, and seamlessly integrated into the existing internet architecture.

1.2. Brief Overview For Hi-DNS

This paper proposes a technical framework called Hi-DNS (DNS-based Resolution of Heterogeneous Identifiers), designed to use the existing DNS system as a global resolution hub to enable unified querying and locating of heterogeneous identifiers in various formats. Hi-DNS does not alter the semantics or format of the original identifiers. Instead, it uses a standardized mapping mechanism to convert any heterogeneous identifier into a DNS-compliant domain name format and establishes corresponding mapping records to resolution service endpoints within a dedicated DNS namespace.

The design principles of this solution fully draw on and are compatible with the core concepts and processing flow of [RFC3490] (Internationalizing Domain Names in Applications, IDNA). Similar to

IDNA, which converts Unicode strings into ASCII Compatible Encoding (ACE) labels to fit DNS, Hi-DNS transforms structured heterogeneous identifiers through a four-step process: "structured segmentation — label internationalization — type binding — domain name construction," generating standardized DNS-resolvable domain names. This process ensures an unambiguous conversion from the original identifier to the DNS query name and supports reverse mapping, thereby achieving semantic equivalence between identifiers and domain names.

Through this mechanism, Hi-DNS provides a unified semantic scope and a single access interface for heterogeneous identifiers, allowing any client supporting standard DNS queries to transparently resolve various identifiers. This significantly simplifies application development and enhances system interoperability and resolution efficiency.

2. Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] and [RFC8174].

3. Terminology

* Code Point: The integer value corresponding to a character in the Unicode character set, for example, U+0041 corresponds to the character ’A’.

* LDH Code Point: An ASCII code point that belongs to letters (az, AZ), digits (09), or hyphens (’-’), which is the character set allowed in traditional DNS domain name labels.

* Label: A subsegment of a domain name, separated by dots (’.’). For example, in the domain name www.example.com, www, example, and com are all labels.

* Internationalized Label: A label that contains non-LDH characters (such as Chinese, Arabic, or other Unicode characters), for example, .

* ACE Label (ASCII Compatible Encoding Label): An ASCII string generated by encoding an internationalized label using the Punycode algorithm ([RFC3492]), prefixed with xn--, for example, xn--ceshi-5qf is the ACE representation of .

* International Domain Name (IDN): A domain name composed of one or more internationalized labels, for example, .在.

* U-label: The user-readable original Unicode string form of a label, that is, the internationalized label before encoding.

* A-label: Synonymous with ACE Label, referring to the ASCII format label after Punycode encoding.

* Preparation: The standardization and preprocessing procedure performed on an input string according to [RFC3491], including Unicode normalization (NFC), case mapping, illegal character filtering, etc.

* Punycode: An encoding algorithm defined in [RFC3492], used to convert Unicode strings into ASCII strings containing only LDH characters to allow IDN compatibility in DNS.

* Hi-registration: The process of registering a heterogeneous identifier in the Hi-DNS system, including structured segmentation, internationalization processing, domain name construction, and DNS record binding.

* Hi-query: The process where a client submits the original heterogeneous identifier, executes Hi-domain conversion via a local proxy, initiates a DNS query, and ultimately obtains the resolution result.

4. Requirement and Application

4.1. Basic Requirements

The design of Hi-DNS must meet the following core requirements:

* Unified Access: Provide a single, standardized DNS interface that supports resolution requests for all types of heterogeneous identifiers.

* Transparency: Users or applications only need to submit the original identifier string without needing to understand its source system or underlying resolution mechanism.

* Compatibility: Fully compatible with existing DNS protocols ([RFC1034], [RFC1035]), IPv6, DNSSEC, and IDNA ([RFC3490], [RFC5890][RFC5894]) standards.

* Extensibility: The namespace design should support the addition of new identifier types and segmentation rules in the future, ensuring the system’s long-term adaptability.

* Security: Ensure the integrity and authenticity of resolution data through DNSSEC, preventing cache poisoning and man-in-the-middle attacks; the registration process must verify the identity of the entity and the ownership of the identifier.

4.2. Application Scenarios

The application scenarios supported by Hi-DNS include but are not limited to the following:

* Cross-Domain Identity Authentication: By entering the unified social credit code, Hi-DNS returns its affiliated trusted authentication node, enabling cross-institution identity verification.

* Data Product Discovery: In a data network or trusted data space, entering a data product’s unique identifier allows Hi-DNS to resolve its registration platform, metadata interface, and access policy.

* IoT Device Service Discovery: IoT devices register their device IDs through Hi-DNS, allowing other systems to discover their service endpoints and achieve automated integration.

* Privacy-Preserving Collaborative Computing: Participants use encrypted or hashed identifiers to query Hi-DNS, enabling anonymized yet trusted addressing and collaboration.

5. Hi-registration

Hi-registration is the complete process of registering a heterogeneous identifier into the Hi-DNS system, managed uniformly by regional/industry function nodes or global function nodes to ensure authoritative and consistent resolution.

5.1. Segmentation and Rule Binding

In the Hi-DNS framework, a segmentation rule is part of the registered metadata and is explicitly specified and bound by the identifier provider for its specific heterogeneous identifier during registration. This rule defines how to parse the original, structured identifier string (U-label) into a sequence of semantic sub-labels.

The input is the original identifier string (U-label). During the registration process, for the same type of identifier, the identifier provider can declare a unique, predefined segmentation method that guides the system on how to parse the internal structure of this identifier. If no declaration is made, the default segmentation method r-by-auto must be used. This declared segmentation method generates a corresponding rule reference, which becomes the permanent metadata for that identifier in the .viv.cn namespace.

Classification Type	Segmentation Method	Example Input → Sub-labels	Generated rule-ref	Segmentation Method Description
Manual	No segmentation	abc123→[abc123]	r-flat	No segmentation.
Manual	By dot (.)	a.b.c→[a, 2eb, 2ec]	r-by-dot	Split by dot (.).
Manual	By hyphen (-)	a-b-c→[a, b, c]	r-by-hyphen	Split by hyphen (-).
Manual	By slash (/)	a/b/c→[a, 2fb, 2fc]	r-by-slash	Split by slash (/).
Manual	By specified symbol	a@b@c→[a, b, c]	r-by-symbol.40	Split by the specified symbol, recording in UTF-8.
Manual	First character differs	a@b.c.d→[a, b, c, d]	r-by-head.40-2e	Find the first @ and split, then split the remainder from front to back by .
Manual	Last character differs	a.b.c@d→[a, b, c, d]	r-by-tail.40-2e	Find the last @ and split, then split the remainder from front to back by .
Manual	By specified	a.b/c@d→[a, b, c, d]	R-by-order.2e-	Split in the

	order		2f-40	specified order; each rule represents one split; record the delimiter in UTF-8.
Manual	Fixed length (e.g., 4)	abcd1234→[abcd,1234]	r-custom.len4-6	Split into fixed-length segments according to the specified order.
Manual	Regex extraction	Using regex to match segments	r-custom.regex-v1	Extract using regular expressions.
Automatic	Cut by non-DNS-legal ASCII codes	user.name@host→[user, name, @host]→[user, 2fname, 40host]	r-by-auto	Split by non-DNS-legal ASCII codes; except the first label, the prefix of each subsequent label is encoded as UTF-8 of the delimiter.

Table 1

5.2. Subdomain Label Internationalization

Each subdomain label must independently undergo IDNA-style internationalization processing to ensure that no non-ASCII characters appear and that all are converted into DNS-compliant representations:

Convert to lowercase;

Perform Unicode normalization;

If the label contains only LDH characters, leave it unchanged. Otherwise, encode it into an ACE label (xn--) using the Punycode algorithm, referring to [RFC3490] and [RFC3492];

Verify that each A-label does not exceed 63 bytes in length. The output is a sequence of internationalized subdomain labels [intl-seg, ..., intl-seg].

5.3. Identifier Type and Identifier Provider Definition

Specify the type of naming system to which the identifier belongs (e.g., ecode, handle, doi). This field is used as part of a DNS domain name and must be a DNS-compliant label, case-insensitive. For unregistered types, a temporary name such as custom-<name> can be used.

The identifier provider can be an organization or an individual (if there is no affiliated organization). It is recommended to use an abbreviation. This field is also used as part of a DNS domain name and must be a DNS-compliant label, case-insensitive.

Identifier Type	Provider	Definition
handle	cnri	Handle-type identifier, managed by China National Research Institute (CNRI China branch).
doi	gs1	DOI-type identifier, registered by GS1 organization.
oid	nrf	OID-type identifier, managed by the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO).
ecode	cic	Ecode-type identifier, managed by the China Institute of Information and Communications Technology (CIC).

Table 2

5.4. Overall Domain Structure

Concatenate the outputs of the previous steps in order, and append the dedicated root domain suffix viv.cn to form the complete query domain name:

<intl-seg1>.<intl-seg2>.<...>.<type>.<provider>.viv.cn

5.5. Registration and Storage

The registration process must include the following steps:

* Submit Application: The entity submits the original identifier, segmentation method, encoding strategy, type, and proof of identity.

* Pre-Validation: Verify the identifier format, check code, entity legitimacy, validity of segmentation logic, and compliance of rule-ref with type.

* Domain Name Mapping of Heterogeneous Identifiers: Use the processes in sections 5.15.4 to generate a complete domain name.

* Bind DNS Records: Create resource records in the authoritative DNS server:

Record Type	Processing Method
TXT record (recommended)	Store the resolver service interface URL, such as "https://resolver.cnrs.cn/ecode" (https://resolver.cnrs.cn/ecode").
SRV record (optional)	Directly point to the service endpoint (IP and Port).
NAPTR record (advanced)	Support multi-protocol routing.

Table 3

* Register in the Database: Store the registration information (original identifier, sub-label sequence, complete domain name, resolver URL, registering entity, etc.) in the identifier information database.

* Rule Filing: If using r-custom.* rules, the semantics and segmentation logic must be filed with the registration center.

6. Hi-query

Hi-query is the standard process for clients to resolve heterogeneous identifiers and must includes the following steps:

User Input: The user or system submits the original identifier string (U-label).

Local Hi Domain Mapping: The client proxy performs the exactly same procedure as the registration end to generate the query domain name.

Initiate DNS Query: Send a standard DNS query (TXT or SRV type) to the local DNS resolver, which recursively queries the authoritative .viv.cn server.

Obtain Resolution Results: Process according to the type of returned records:

Record Type	Processing Method
TXT Record (Recommended)	Obtain the resolution service URL, e.g., https://resolver.cnrs.cn/ecode. The client sends an HTTP(S) request to this URL, carrying the original identifier, to retrieve detailed metadata such as location, status, access policy, and permissions.
SRV Record	Directly connect to the specified service endpoint (IP + Port). Suitable for low-latency scenarios where direct access is required.
NAPTR Record	Resolve using priority and regular expression matching to select the appropriate protocol and service (advanced routing).

Table 4

Result usage: completing identity authentication, service invocation, or resource location.

7. Implications for Typical Applications Using Hi-DNS

7.1. Support for Heterogeneous Identifiers as Domain Names

Any heterogeneous identifier can be converted into a Hi-domain and used as a standard DNS name for resolution, realizing the concept of "identifier as service entry" and breaking the traditional DNS limitation of supporting only LDH characters.

7.2. Protocol Transparency

Application systems only need to integrate the Hi-DNS resolution proxy module to achieve cross-system identifier resolution, without modifying existing application-layer protocols or interfaces.

7.3. Compatibility with Existing IDN Applications

In IDN-aware environments (such as modern browsers), semantic identifiers can be displayed via U-labels to enhance user experience; in IDN-unaware environments, it automatically falls back to A-labels to ensure backward compatibility.

7.4. Support for Multi-Level Resolution Architecture

Using a two-layer architecture of "DNS query for the entry point and HTTP(S) for metadata retrieval," it leverages the efficient addressing capabilities of DNS while retaining the flexibility and rich semantics of HTTP, aligning with modern service discovery patterns.

7.5. Promotion of Data Element Circulation

In scenarios such as trusted data spaces and the data internet, Hi-DNS enables "one identifier, globally resolvable," supporting cross-organization data product discovery and service invocation, providing a unified resolution entry for data assets, devices, and subjects, and reducing system integration complexity.

8. Domain Name Server Technical Requirements

8.1. Authoritative Server

The authoritative server is deployed at the full-domain or regional/ industry function nodes and is responsible for managing the resolution records under .viv.cn. It must support:

* Receiving bulk registration requests via API;

* Handling ACE labels that may exceed 63 characters in length (e.g., xn--ceshi-5qf);
* Optional support for wildcard (*) records for bulk subdomain management.

8.2. Cache Server

The cache server must recognize the .viv.cn domain namespace and support long A-label caching. It is recommended to set a short TTL (e.g., 300 seconds) to accommodate dynamic changes in identifiers.

8.3. DNSSEC Support

DNSSEC must be enabled, and all resource records must be digitally signed to ensure the authenticity and integrity of resolution results, preventing cache poisoning and man-in-the-middle attacks.

8.4. Logging and Auditing

Record all registration, query, and update operations, including fields such as time, operator, and target domain, to meet the security audit requirements outlined in Chapter 7 of the main standard.

9. Root Domain Governance and Management Requirements

9.1. Root Domain Management

.viv.cn, as the unified root domain of Hi-DNS, should be centrally managed by the national data infrastructure authority or its authorized agency, which will formulate naming standards, registration policies, and security requirements.

9.2. Hierarchical Delegation Mechanism

Support subdomain delegation based on type or organization:

* For example, handle.viv.cn is delegated to the operator of the Handle system;
* custom.cn.com.sgcc.viv.cn is delegated for independent management by the State Grid, achieving hierarchical governance.

9.3. Stability and Compliance

Use multi-node Anycast deployment to ensure high availability; comply with the Cybersecurity Law, Data Security Law, and other regulations, and provide audit interfaces to support supervision.

10. Security Considerations

The security design of Hi-DNS should cover the entire process of identifier registration, resolution, transmission, and management. Only entities that have been authenticated may register the identifiers they manage, preventing impersonation, and the ownership relationship between the identifier and the entity should be verified at the time of registration. It is recommended to use DNS over HTTPS (DoH) or DNS over TLS (DoT) to encrypt query traffic, preventing eavesdropping or tampering by intermediaries. To prevent abuse, rate limiting should be applied to high-frequency queries, and anomaly detection mechanisms should be deployed to avoid scanning attacks or resource exhaustion. All registration, query, and update operations must be fully logged, including information such as time, operating entity, and target identifier, in order to meet the security audit requirements outlined in Chapter 7 of the main standard. For custom rules (such as r-custom.*), their semantics and segmentation logic must be filed with the registration center to prevent malicious constructions that could cause resolution ambiguities or confusion attacks. Sensitive identifiers such as ID numbers and mobile phone numbers may be hashed or encrypted before registration to prevent exposure in plaintext, supporting high-security scenarios such as privacy computing. Additionally, DNSSEC must be enabled, with all resource records under .viv.cn digitally signed to ensure the authenticity and integrity of resolution results, defending against attacks such as cache poisoning.

11. IANA Considerations

This document defines no new protocol parameters that require registration by IANA. The use of existing DNS record types (TXT, SRV, NAPTR) and the IDNA-compatible encoding (Punycode) relies on mechanisms already standardized and managed by IANA. The domain suffix .viv.cn is used as an example deployment namespace and does not require IANA action.

12. Acknowledgements

The authors would like to thank the World Internet Conference (WIC) for providing valuable platforms for technical exchange and discussion, which contributed to the refinement of ideas in this document.

13. References

13.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3490] Faltstrom, P., Hoffman, P., and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 3490, DOI 10.17487/RFC3490, March 2003, <https://www.rfc-editor.org/info/rfc3490>.

[RFC3491] Hoffman, P. and M. Blanchet, "Nameprep: A Stringprep Profile for Internationalized Domain Names (IDN)", RFC 3491, DOI 10.17487/RFC3491, March 2003, <https://www.rfc-editor.org/info/rfc3491>.

[RFC3492] Costello, A., "Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)", RFC 3492, DOI 10.17487/RFC3492, March 2003, <https://www.rfc-editor.org/info/rfc3492>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

13.2. Informative References

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, <https://www.rfc-editor.org/info/rfc5890>.

[RFC5894] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale", RFC 5894, DOI 10.17487/RFC5894, August 2010, <https://www.rfc-editor.org/info/rfc5894>.

Authors' Addresses

Xiaodong Lee
Institute of Computing Technology, Chinese Academy of Sciences

Email: XL@ict.ac.cn

Botao Peng
Institute of Computing Technology, Chinese Academy of Sciences
Email: pengbotao@ict.ac.cn

Jingwen Lee
Institute of Computing Technology, Chinese Academy of Sciences
Email: lijingwen24s@ict.ac.cn