Content Delivery Networks Interconnection
Internet-Draft
Intended status: Standards Track
Expires: 29 August 2025

A. Warshavsky
Qwilt
G. Bichot
Broadpeak
G. Goldstein
Lumen Technologies

25 February 2025

CDNI Private Features Metadata
draft-warshavsky-private-features-metadata-02

Abstract

This specification defines a mechanism for downstream content delivery networks (dCDNs) to define private extensions to the metadata model that are mutually agreed upon between participating upstream content delivery networks (uCDNs) and dCDNs.

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 29 August 2025.

Copyright Notice

Table of Contents

1. Introduction

There is a natural tension between standardization and differentiation - all CDNs and Open Caching systems are not identical, and not all content providers use the same methodologies for controlling access to their resources.

While [RFC8006] provides the GenericMetadata object as the basis for extensibility, creating a formal structure for private features as a distinct MI object conveys the explicit understanding of the usage context. This document details a private features mechanism that allows for custom configuration metadata to be easily added, enabling rapid development of new features, and allowing companies to mutually agree on a feature that may not be of general interest to the industry.

Through the CDNI Footprint & Capabilities Interface [RFC8008], dCDNs advertise their capabilities to uCDNs. This capabilities advertisement allows a dCDN to declare the private features that it supports.

Examples of private features include:

* Custom authentication schemes

* Definitions of custom scripts to be invoked on the CDN during request processing

2. Requirements

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].

3. MI.PrivateFeatureList

The dCDN MAY gather a certain number of private features (i.e., not [yet] adopted in a standard or considered marginal) that it MAY want to expose to the content provider and/or the uCDN. Although private, the announcement, selection, and configuration of this private feature can be done through the CDNI interfaces.

MI.PrivateFeatureList is a GenericMetadata configuration object used as a base generic object that permits the control of private features.

Property: features

* Description: The list of feature configuration objects.

* Type: Array of MI.PrivateFeature objects

* Mandatory-to-Specify: Yes

3.1. MI.PrivateFeature

MI.PrivateFeature is a subobject of MI.PrivateFeatureList that defines a single private feature and contains the following properties:

Property: feature-oid

* Description: The feature organization ID is the owner/organization that has specified the feature, typically the name of the company or organization defining the private feature.

* Type: String

* Mandatory-to-Specify: Yes

Property: feature-type

* Description: Indicates the type/name of the private feature configuration object, driving the definition of feature-value.

* Type: String

* Mandatory-to-Specify: Yes

Property: feature-value

* Description: The feature configuration object.

* Type: Object. Format and object type is defined by the value of the feature-type property above.

* Mandatory-to-Specify: Yes

Note that the private features exposed by the dCDN can be advertised through a dedicated FCI object.

The following example illustrates the Broadpeak S4Streaming feature:

```
{
  "generic-metadata-type": "MI.PrivateFeatureList",
  "generic-metadata-value": {
    "features": [
      {
        "feature-oid": "Broadpeak",
        "feature-type": "S4Streaming",
        "feature-value": {
          "footprint": {
            "footprint-type": "ipv4cidr",
            "footprint-value": [
              "192.0.2.0/24",
              "198.51.100.0/24"
            ]
          },
          "activation": "ON",
          "mode": "transparent",
          "policy": "bandwidth-max"
        }
      }
    ]
  }
}
```

Figure 1

4. Named Private Features

Named private features provides an alternative for specifying private features inside the MI.PrivateFeatureList object, allowing them to be specified individually rather than as part of a group under the host match and path match inheritance/override hierarchy.A named private feature corresponding to a feature inside an MI.PrivateFeatureList object in the same generic metadata array, takes precedence and overrides the latter.

4.1. MI.NamedPrivateFeatureType

MI.NamedPrivateFeatureType is a structured string corresponding to the MI.PrivateFeature object by encoding its two identifier properties feature-oid and feature-type into a string that can be incorporated into the extended definition of MI_payloadtype.

Property:

* Description: A string in the form of MI.PrivateFeature.<org id>.<feature id>where org-id consists of alphanumeric characters, and feature-id MAY also contain dots for additional internal namespacing within the feature name.

* Type: String, of pattern: : ^MI\.PrivateFeature\.[a-z0-9A-Z]+(?:\.[a-z0-9A-Z]+){1,}$

4.2. MI.NamedPrivateFeatureValue

The actual structure of the MI.NamedPrivateFeatureValue object is the one defined in the property feature-value of the object MI.PrivateFeature.

* Type: object

This following example illustrates the transformation from an MI.PrivateFeature object residing inside an MI.PrivateFeatureList, into an MI.GenericMetadata object residing in a metadata array of a host or a path match.

```
{
  "generic-metadata-type": "MI.PrivateFeatureList",
  "generic-metadata-value": {
    "features": [
      {
        "feature-oid": "ACME",
        "feature-type": "Hammer",
        "feature-value": {
          "enable": true
        }
      }
    ]
  }
}
```

Figure 2

```
{
  "generic-metadata-type": "MI.PrivateFeature.ACME.Hammer",
  "generic-metadata-value": {
    "enable": true
  }
}
```

Figure 3

5. FCI.PrivateFeatures

While the MI objects introduced in this document allow a uCDN to configure private features with any required configuration metadata, FCI.PrivateFeatures enables a dCDN to advertise the set of private features that it supports.

Property: features

* Description: The list of supported private features.

* Type: Array of nested objects of FCI.PrivateFeature

Example:

```
{
  "capabilities": [
    {
      "capability-type": "FCI.PrivateFeatures",
      "capability-value": {
        "features": [
          {
            "feature-oid": "Broadpeak",
            "feature-type": "S4Streaming"
                 }
        ]
      }
    }
  ]
}
```

Figure 4

5.1. FCI.PrivateFeature

This subobject identifies a specific private feature in the FCI capabilities advertisement, and contains the following properties:

Property: feature-oid

* Description: The owner/organization that has specified the feature.

* Type: String

* Mandatory-to-Specify: Yes

Property: feature-type

* Description: Indicates the type/name of the private feature configuration object.

* Type: String

* Mandatory-to-Specify: Yes

6. Security Considerations

The FCI and MI objects defined in the this document are transferred via the interfaces defined in CDNI [RFC8006] which describes how to secure these interfaces by protecting integrity and confidentiality while ensuring the authenticity of the dCDN and uCDN.

7. IANA Considerations

7.1. CDNI Payload Types

This document requests the registration of the following entries under the "CDNI Payload Types" registry hosted by IANA:

Payload Type	Specification
MI.PrivateFeatureList	RFCthis
MI.PrivateFeature	RFCthis
MI.NamedPrivateFeatureType	RFCthis
MI.NamedPrivateFeatureValue	RFCthis
FCI.PrivateFeatures	RFCthis
FCI.PrivateFeature	RFCthis

Table 1: CDNI Payload Types

8. Acknowledgements

The authors would like to express their gratitude to the members of the Streaming Video Technology Alliance [SVTA] Open Caching Working Group for their contributions and guidance.

Particulary the following people contribute in one or other way to the content of this draft:

* Chris Lemmons - Comcast
* Pankaj Chaudhari - Disney Streaming Services
* Will Power - Lumen
* Rajeev RK - picoNETS
* Shmuel Asafi - Qwilt
* Yoav Gressel - Qwilt
* Nir Sopher - Qwilt

* Eric Klein - Sirius XM

* Alfonso Siloniz - Telefonica

* Ben Rosenblum - Vecima

9. Normative References

[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>.

[RFC8006] Niven-Jenkins, B., Murray, R., Caulfield, M., and K. Ma, "Content Delivery Network Interconnection (CDNI) Metadata", RFC 8006, DOI 10.17487/RFC8006, December 2016, <https://www.rfc-editor.org/info/rfc8006>.

[RFC8008] Seedorf, J., Peterson, J., Previdi, S., van Brandenburg, R., and K. Ma, "Content Delivery Network Interconnection (CDNI) Request Routing: Footprint and Capabilities Semantics", RFC 8008, DOI 10.17487/RFC8008, December 2016, <https://www.rfc-editor.org/info/rfc8008>.

10. Informative References

[SVTA] SVTA, "Streaming Video Technology Alliance Home Page", <https://www.svta.org>.

Authors' Addresses

Arnon Warshavsky
Qwilt
Israel
Email: arnon@qwilt.com

Guillaume Bichot
Broadpeak
France
Email: guillaume.bichot@broadpeak.tv

Glenn Goldstein
Lumen Technologies
United States of America
Email: glenng1215@gmail.com