CDN Interconnect Metadata
draft-ietf-cdni-metadata-00

Ben Niven-Jenkins
David Ferguson
Grant Watson
Motivations

• Fully RESTful interface
• Provide a deterministic way for a Surrogate/Request Router to discover/obtain CDNI Metadata
  – Independent of the specific EU request that triggers the need for the CDNI Metadata
• Allow a dCDN to determine which uCDN(s) may delegate a Site
  – For a given EU request a dCDN knows which uCDN to ask for the associated CDNI Metadata
• Support both asynchronous & synchronous distribution
  – Asynchronous: dCDN can request CDNI Metadata in advance of EU request
  – Synchronous: dCDN can deterministically identify appropriate CDNI Metadata resources and request them in response to EU request
• Simple interface configuration/bootstrapping
• Fully RESTful interface
• Influenced by Atom (RFC 4287)
  – Use of “Feed” as an index into CDNI Metadata
• Leverages standard HTTP(S) for addressing & transfer of resources
• Minimal client/server coupling
  – Provides implementation flexibility in how CDNI Metadata server structures URIs used by resources
  – Responses/Resources are self-describing
    • Through the use of defined Media-Types & Relationships
• Reuses standard HTTP cache-ability semantics
  – Each object is independently addressable resource
  – Provides for “inlining” to optimize individual cacheability Vs number of RTTs to obtain all required CDNI Metadata resources
  – CDNI Metadata interface responses are cacheable Independently of the specific EU request that generated the CDNI Metadata request
Determinism

• Uses a SiteFeed as an index into the CDNI Metadata of a uCDN
  – Primary Index into CDNI Metadata is “hostname”
    • Could easily be extended to provide other indexes if required, e.g. if CDNs were to use a single hostname to aggregate/hide multiple CSPs

• Indexing on hostname & ordered list of Globs (path matches)
  – Provides a deterministic way for a Surrogate/Request Router to discover what properties apply to specific content
  – Still deterministic if paths/globs overlap

• Requests across CDNI Metadata interface do not vary depending on specific content EU requested
  – Makes responses cacheable without CDNI application knowledge
Configuration & Extensibility

• Bootstrapping
  • Only configuration required is URI of SiteFeed
    – Everything else is discoverable

• Data Model & Objects are easily extended
  • New Data Objects
    – Define new Media Types & Relationship
    – Define properties of the new Data Object
  • Extending existing Data Objects
    – Define new properties
    – Up-rev Media Type version
Data Model

List of hostnames uCDN may delegate

Default properties associated with hostname (Sources, Delivery Protocol & Parameters)

Ordered list of pattern-based path matches & specific properties for content that matches that path

Defines what constitutes a given Location

Which Locations content can be delivered from

Which Locations content can be delivered to
Object encoding

• Proposes JSON encoding
  – Easier to parse than XML
  – but could easily support an XML encoding if required

• Initial set of properties is illustrative
  – Minimal set to demonstrate the concepts
Object encoding structure

{  
    "base": JSONString,  
    "links": JSONArray,  
    "inline": JSONDictionary,  
    "property1": Value,  
    "property2": Value,  
    ...  
    "propertyN": Value  
}

• “base”:  
  – Prefix for any relative URLs in the object.  
    • Similar to XML base tag

• “links”:  
  – The relationships of this object to other addressable objects.  
    • See next slide

• “inline”:  
  – Dictionary of inlined objects  
    • Keys: URI fragments which are used to refer to inlined objects  
    • Values: The inlined object itself.

• “propertyX”:  
  – Individual properties of the object
Object encoding structure - Links

```
{
    "title": "Everywhere",
    "href":
        "http://metadata.cdni.example.com/locations/everywhere",
    "rel": "SelectionAllow",
    "type": "application/vnd.cdni.metadata.location+json"
}
```

- **“title”**: Human readable title
  - MUST be hostname in SiteFeed
- **“href”**: URI of the referenced object
- **“rel”**: Relationship between this object & the object being referenced “propertyX”:
- **“type”**: Media Type of the object being referenced