SVTA Configuration Interface

IETF/CDNi Metadata Model Extension:
draft-power-cdni-cache-control-metadata-01
July 2023 (IETF 117)
• RFC 8006 defines an initial set of CDNI GenericMetadata objects that allow a dCDN to serve content requests on behalf of a uCDN

• Additional CDNI GenericMetadata objects are defined that allow control over:
  • Internal and external cache policy
  • Caching of negative responses (e.g., 503)
  • Serving of stale cached resources if an uCDN is unavailable
  • Customization of cache key for more advanced production use cases
  • Selective bypass of cached resources

• draft-power-cdni-cache-control-metadata-00 (presented at IETF-116) proposed extensions to give Content Providers and uCDNs more fine-grained control over dCDN caching

• draft-power-cdni-cache-control-metadata-01 (presented at IETF-117) addresses feedback from initial draft and provides clarifying examples.
Key Changes from Original Draft:

- No longer indicates that RFC8006 is being extended. Only new GenericMetadata objects are being proposed.
- Added missing RFC2119 boilerplate
- general: the "Values" definition that appears in some of the properties can just be part of the "Type" definition
- Clarified definitions of properties that can contain either an enumeration or an integer value representing a number of seconds.
- Added references to external documents for Processing Stages and Metadata Expression Language.
- Clarified definition of NegativeCachePolicy to indicate that it is an alternative to using Processing Stages for simple use cases.
- Added a complex example using CachePolicy in conjunction with Processing Stages to illustrate how multiple cache policies can be used with different HTTP status codes from the origin.
- Cleaned up specification of default values.
- Clarified documentation of use of wildcard 4xx, 5xx values.
- Clarified use of RFC2119 terms.
**MI.CachePolicy**

- Allows the internal and external cache policy of resources to be configured
- Supports cache TTL values in seconds and directives: as-is, no-cache, no-store
- Properties ‘force-internal’ and ‘force-external’ allow the cache policy to override Cache-Control provided by the Origin, if required
- Example:

```
{
  "generic-metadata-type": "MI.CachePolicy",
  "generic-metadata-value": {
    "internal": "5",
    "external": "no-cache",
    "force-internal": "true",
    ...
  }
}
```
**MI.NegativeCachePolicy**

- Allows cache policy for ‘negative’ responses to be configured
- Supports setting cache policy for an array of ‘negative’ response codes
- Example:

```
{
    "generic-metadata-type": "MI.NegativeCachePolicy",
    "generic-metadata-value": {
        "error-codes": ["404", "503", "504"],
        "cache-policy": {
            "internal": "5",
            "force-internal": "true"
        }
    }
}
```
**MI.StaleContentCachePolicy**

- Allows the cache policy of 'stale' resources to be configured
- Supports stale-while-revalidating, stale-if-error, and failed-refresh-ttl
- Example:

```json
{
    "generic-metadata-type": "MI.StaleContentCachePolicy",
    "generic-metadata-value": {
        "stale-while-revalidating": "true",
        "stale-if-error": ["5xx"],
        "failed-refresh-ttl": "5"
    }
}
```
**MI.ComputedCacheKey**

- Allows advanced control over setting the cache key for resources
- Supports constructing a cache key using attributes from a request or from an origin response.
- Uses the proposed CDNI Metadata Expression Language (MEL) to dynamically construct cache keys
- Example:

  ```json
  {
    "generic-metadata-type": "MI.ComputedCacheKey",
    "generic-metadata-value": {
      "expression": "req.h.X-Cache-Key"
    }
  }
  ```

- Note: CDNI Metadata Expression Language (MEL) will be presented in a draft planned for IETF-117. Specification currently under review within the SVTA.
**MI.CacheBypassPolicy**

- Allows a client request to bypass cache
- For example, allow a request to bypass cache while testing
- Existing cached resources are not evicted by this directive
- Example:

```json
# Match on path or request header value
{
  "generic-metadata-type": "MI.CacheBypassPolicy",
  "generic-metadata-value": {
    "bypass-cache": "true"
  }
}
```
Conclusion

Based on the contents of this presentation, Can the CDNI working group accept this document as a Working Group Draft?