



SVA Configuration Interface

IETF/CDNi Metadata Model
Extensions

July 2021

Configuration Interface: Mission

- Fill the gaps between systems focused on content distribution configuration interfaces.
- Identify areas where the existing IETF CDNi metadata model and interfaces may be useful and where practical needs may dictate alternate solutions and formats.
- Goals:
 - Simplify the management of CDN and Open Caches for content providers.
 - Provide benefits for participating CDNs and Open Caches within the ecosystem by having one configuration format and API to cover most use cases.

Isn't CDNi Metadata Sufficient As-Is?

Quick Answer: Almost, but not quite.

- The CDNi Metadata (MI) and Control (CI) Interfaces were designed for the limited scope use cases related interchanges between upstream CDNs and downstream CDNs.
- As we look at the wider set of use cases involving Content Providers managing multiple CDNs, along with use cases in the Open Caching ecosystem, we see gaps.

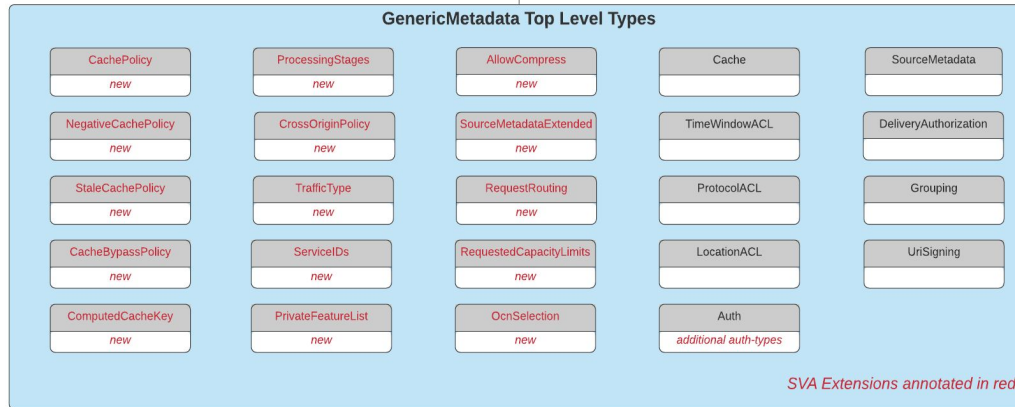
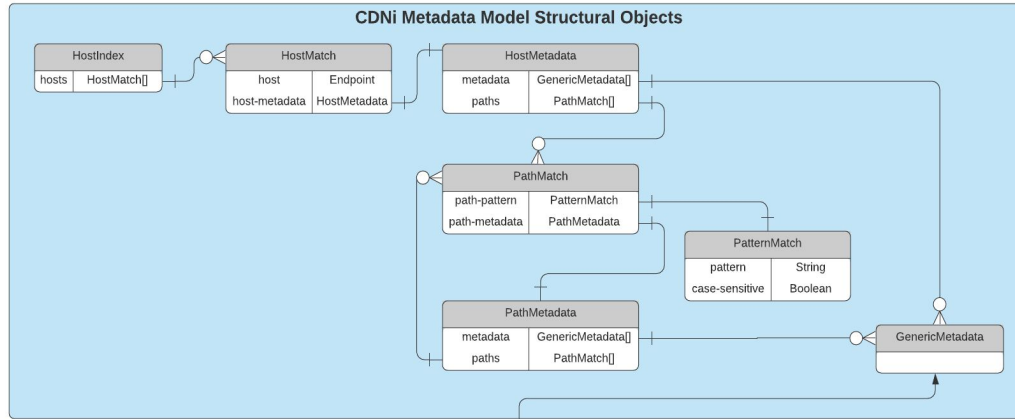
What CDNi Metadata Provides	Gaps - What's Missing
Simple Metadata Object Model	Metadata Object Model meeting more complex requirements of CDN and OC industries
Interfaces for retrieving metadata, and triggering metadata repositioning, invalidation and purging	Simple push-style metadata publishing (POST)
Interfaces to check status or cancel trigger requests	Advanced configuration publishing capabilities required by Content Providers (publishing, versioning, deployment)

Metadata Model Extensions

The following requirements and proposed extensions are documented in detail in the *SVA Configuration Interface Part 2* specification and will be submitted to the IETF as extensions to RFC-8006

- ❖ Enhanced Source Definitions:
 - ❖ Origin Load Balancing, Failover
 - ❖ Origin Authentication Methods
 - ❖ Cache Control Policies & Computed Keys
 - ❖ Dynamic CORS Headers
 - ❖ Traffic Type Metadata
 - ❖ ServiceID and Property ID Metadata
 - ❖ SVA Open Caching Configuration Metadata
 - ❖ Private Features (for extensibility)
- ❖ Processing Stages:
 - ❖ Match on Request/Response elements (Headers, URIs, HTTP Status)
 - ❖ Request URI Rewrites
 - ❖ HTTP Header Modifications
 - ❖ HTTP Status Modifications
 - ❖ Synthetic Responses
 - ❖ An expression language for matching rules & synthesis of dynamic values

CDNi Metadata Object Model

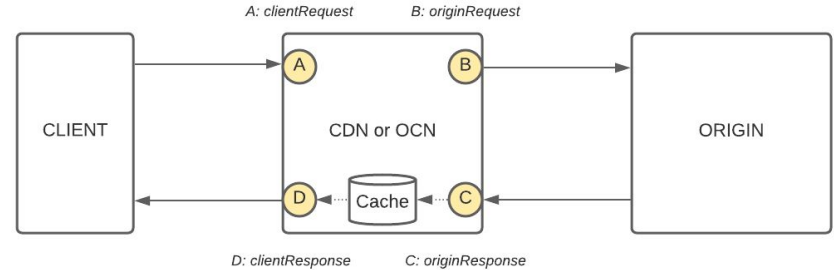


CDNi Metadata Model Extension Example

Processing Stages

Allows for specification of metadata rules to be applied at a specific stage in the pipeline. Typical stage-specific processing includes:

- Application of metadata (such as cache policies or TTLs)
- Request Transformations (Header modifications, URI re-writes)
- Response Transformations (Header modifications, status code overrides)
- Generating Synthetic Responses



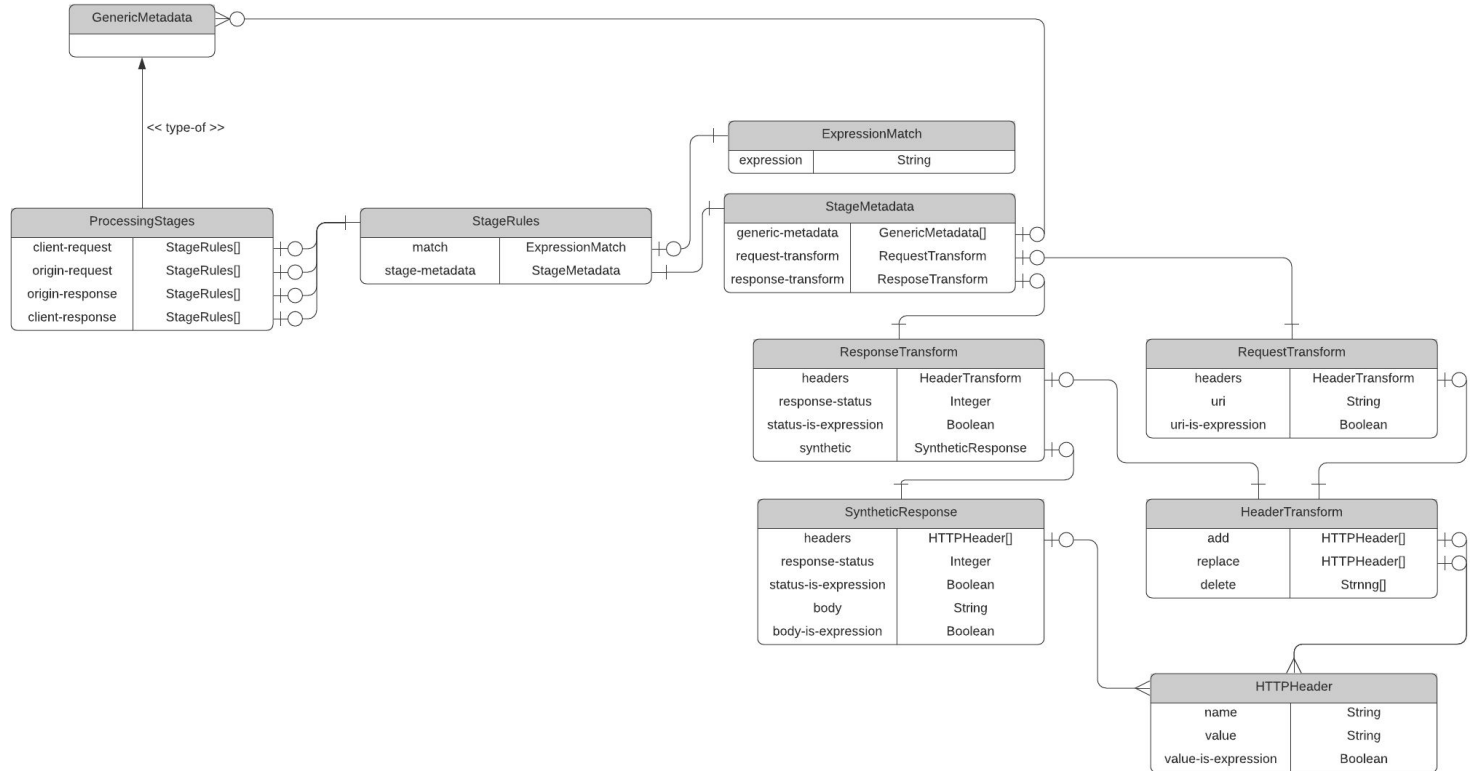
clientRequest - Rules run on the client request prior to further processing.

originRequest - Rules run prior to making a request to the origin.

originResponse - Rules run after response is received from the origin and before being placed in cache.

clientResponse - Rules run prior to sending the response to the client. If response is from cache, rules are applied to the response retrieved from cache prior to sending to the client.

Stage Processing Object Model



Capabilities Interface

- The proposed CDNi Metadata Model extensions are optional, with dCDNs able to advertise their support via the Footprint & Capabilities Interface (FCI).
- Any extension that is embodied as a new GenericMetadata object can be advertised as supported via the CDNi standard FCI.Metadata object.
- Some proposed extensions entail many features, and it is quite possible that a dCDN may support some (but not all) of these features.
- To allow for more fine-grained advertisement of feature support, additional FCI objects will be defined containing feature flags that are specific to each extended GenericMetadata object.

Extending the Metadata Interface

In addition to extending the CDNi Metadata model, the SVA is also working on APIs that extend the interface:

- Extend the basic metadata retrieval interface defined in RFC-8006 with metadata publishing capabilities to allow a uCDN to publish and delete metadata on a dCDN.
- Add capabilities to publish and reference sets of named GenericMetadata Objects (extending the current HREF concept).
- Provide CDN configuration life cycle management capabilities such as publishing, versioning, staged deployments, profiles, and templates.
- Provide configuration for additional aspects of CDN operation such as provisioning of Certificates and configuration of log delivery endpoints.