CDNI CI/T v2
IETF 123

Alan Arolovitch, Jay Robertson

Madrid, July 21, 2025

RFC 8007bis Status

- **Published v16 on February 25th 2025:**
  - https://datatracker.ietf.org/doc/html/draft-ietf-cdni-ci-triggers-rfc8007bis-16
- **Responses to Kevin’s review from July 5, 2025**
- **Plans for v17**
- **Review! Review!**

Kevin’s Review
Requirements around core features support

- **IANA Registry management: RFC vs. Spec required?**

Manage trigger specs, trigger subjects and URL types via RFC, and trigger extensions and objectlist types via spec to permit 3rd party specifications

- **What support is mandatory for trigger actions (“purge”, “invalidate”, “preposition”), trigger specs (“urls”, “ccid” etc.), trigger subjects (“content”, metadata”), object types and trigger extensions?**

Allow for lowest common denominator implementation to be compliant - any one of trigger actions, any one of trigger actions and “urls” trigger specs.

dCDN MAY advertise only one trigger subject (e.g. “content”) or advertise separate CI/Tv2 end points per subject.

Trigger extensions are all optional for implementation.

Kevin’s Review

- **Trigger access control and multi-tenancy. Questions about the requirement to associate a trigger with metadata**

Section 2.4 puts forward two validation requirements:

  - the objects referred by a trigger must align with existing metadata (e.g., for cache objects with video.example.com URL to be prepositioned, dCDN must be in possession of MI metadata for this domain)
  - this metadata should belong to the requesting uCDN, uCDN A cannot preposition video.example.com content objects, if the metadata for this domain was provided by uCDN B

- **How is the trigger collection URI identified? Is a particular URI scheme imposed by the specification?**

The trigger collection resource is the top-level resource in each CI/T v2 interface instance, unique for each uCDN. The root URI discovery is part of an interface bootstrap process, which is outside of the specification scope.
The interface follows the REST URI discoverability principle (also known as HATEOAS), so uCDNs can discovery the URIs dynamically.

Kevin’s Review
Deleting triggers

- **Does deletion of non-terminal (“pending” or “active”) triggers fail?**

Deletion is cancellation + removal of the resource (Section 3.5). In general, dCDN MAY cancel non-terminal triggers ("pending" or "active").

- **How does dCDN signal deletion failure?**

Yes, dCDN MAY reject deletion (or cancellation) requests with “501 Not Implemented”. We will recommend “409 Conflict” rather than “501 Not Implemented” for cases when the trigger state doesn't allow cancellation.

We are using 409 Conflict to do this for trigger modification (HTTP POST) (which is what trigger cancellation uses), need to update the HTTP DELETE response semantics.

- **Are trigger deletion requests idempotent?**

Yes. Will update 3.5 to say that the processing of triggers in "cancelling" state incurs no processing on the dCDN side.

Kevin’s Review
Cascading triggers

- **Completion and cancellation of cascaded triggers**

Will add clarification that in case of trigger cascading, trigger processing and cancellation do not end until all cascaded trigger requests reported back

- **"If dCDN receives a trigger that it itself originated, dCDN MAY process the trigger as required." Isn’t this in violation of the ban on “diamond configuration”?**

This is a bit of corner case, but imagine a public CDN delegating down to a chain of operator CDNs, where the last in the chain can use the originating uCDN as an overflow CDN.

In this case, uCDN may not use the same metadata objects for itself, as it does in the role of dCDN, so processing the trigger *may* make sense.

Kevin’s Review
CDN PIDs

- **Shouldn’t we make CDN PID opaque? Why use an obsolete AS-based definition?**

This definition is inherited from section 4.6 of RFC8007.

We fully agree that the use of ASN as CDN PID is outdated. There could be multiple CDNs in the same AS, some CDNs have no AS number while others have many.

Moreover, it doesn't look like CDN loops are a real concern at this time.

This is why we made this ID optional. Moreover, we need a workable CDN identifier for other purposes, e.g., defining the interface authentication and authorization.

We *could* make it opaque to allow it to be overloaded by future spec. The danger is that the good identifier we come up with is a composite object, e.g., a registry address and identifier.

So this is why we proposed to make this attribute optional.

- **Clarifications on Section 3.7.1: Error propagation "[..] where each error MAY carry its own CDN identifier"**

There could be one error with dCDN-C PID, and a separate error from dCDN-B.
OR dCDN-B could present their errors as well as the errors received from dCDN-C with dCDN-B PID.

Kevin’s Review
Location Policy

- **Use cases for LocationPolicy (section 4.1.3.3.1): “or availability of contracts" - I'm not sure what this means, or how it relates to "prepositioning"?**

It's a legacy text. Will replace with an example about selective prepositioning of renditions in a geography based on the languages prevalent in the area

- **[The LocationPolicy is on] per cache location basis, a non-match means don't execute the trigger. I think that could be made clearer. Everything else in the doc so far is CDN specific, but this is cache specific, and it took me a few reads to readjust my frame of reference. The deny is cache specific, based on the cache location.**

One of us thinks that the Location Policy extension violates the CDNI principle of separation of concerns on a CDN boundary, however we decided to retain it with appropriate clarifications. In the future an alternative extension may be introduced, that provides for trigger geo-awareness by binding it to MI Footprint objects.

Kevin’s Review

- **How is Urgency different from setting the state to "active" as described in 3.2? i see the distinction between priority and urgency, but it seems there is already a mechanism for signaling urgency?**

This is fair, we could drop the Urgency setting, if we could support the same functionality via the core CI/T framework.

- **Urgent creates a lot of corner/deadlock cases as noted here, as do prerequisites. I feel like we need something more formal (e.g., a decision tree) to prove that we handle all the cases?**

Makes sense, will prepare

- **We only list the "specs" (noting that the "objects" list is optional)? so if a single piece of content had issue, the whole spec is failed?**

This is correct. We have PrepositionPolicy in the pipeline, to be submitted together with additional trigger extensions, that would allow partial trigger failure.

The reasoning is that while we have a way to signal back what part of preposition succeeded, it would be hard to do the same in a generic way to cover purge and invalidation.

Kevin’s Review

- **It feels cumbersome to mix the list of all triggers with the list of valid filter representations. did we consider making this just return the filter representations and having a mandatory to implement "all" triggers representation, so that we can keep the concerns separated?**

We shall change the top-level object to hold only trigger collections (“all”, “all-triggers” if supported, “coll-state” array, “coll-label” array) and configuration attributes "cdn-id" and "staleresourcetime".

This object will be renamed to TriggerRegistry ot TriggerIndex (to align with MI HostIndex).

- **uCDN MUST provide configuration metadata objects" - Does this mean that the uCDN must guarantee that the dCDN already has the metadata (e.g., by prepositioning the metadata), or just that the metadata has to be available for retrieval by the dCDN (which it could theoretically do on demand)?**

We should be more flexible and allow for dCDN retrieving the metadata or uCDN providing them in time for the trigger processing.

Can rewrite as "When processing a trigger that uses this URL type, dCDN MUST be able to match the URLs with metadata objects provided by uCDN."

Kevin’s Review
FCI CIT Endpoint Capability Object

- **Is the trigger-endpoint-uri the same as the trigger collections URI?**

Correct, it's a top-level URI for specific interface instance. dCDN could advertise one trigger-endpoint-uri for content trigger subject interface and another one for metadata interface.

- **This doesn't explain what the valid "versions" are? In section 1, it talks about the new version being "v2", but from the example, it just shows an integer "2". This needs to be clearer what the format is, and where it's defined.**

Will correct. We need to add an IANA registry for CI/T interface version values.

- **A missing or empty "trigger-subject" list means that all trigger subjects are supported by the endpoint" - Does this mean all subjects at the time of this writing? or all subjects in the future? I don't think it can be the latter. The latter would make all implementations immediately non-compliant when a spec for a new subject is published?**

Such new spec would have a different version (e.g. “v2.1”). We should clarify this versioning issue and remove "2.1" from the examples.

- **Are URIs for all subjects (and/or version and subject pairs) mandatory to advertise?**

All combinations of subiect and version (there are four at this point) are supported, and what is not advertised is not supported. Shall state this explicitly.

Kevin’s Review
FCI Trigger Scope Capability Object

- **The "trigger-scope-capability" object matches the "FCI.CITScope" capability type' - I'm not sure what the "trigger-scope-capability" is? Should this just say: 'The capability type is "FCI.CITScope"'?**

The confusing language is due to the confusing object structure. Will rewrite the example and the object format as follows, in line with section 5.1

```
{
    "capabilities": [
      {
        "capability-type": "FCI.CITScope",
        "capability-value": {
          "trigger-action": "preposition",
          "trigger-subject": "content",
          "trigger-specs": [ "urls", "ccids" ],
          "trigger-extensions": [ "time-policy" ]
        },
        "footprints": { "footprint-type": "countrycode", "footprint-value": [ "us" ] }
      },
      {
        "capability-type": "FCI.CITScope",
        "capability-value": {
          "trigger-action": "invalidate",
          "trigger-subject": "content",
          "trigger-specs": [ "urls", "ccids" ]
        }
        "footprints": { "footprint-type": "countrycode", "footprint-value": [ "us" ] }
      }
   ]
}
```

Kevin’s Review
CDNI FCI CI/T Payload Types

- **Section 7.1.3.4: need to add a section for "FCI.CITPrivateUrlType"**
- **Section 7.1.3.5: need to add a section for "FCI.CITExtendedStatus"**

OK

- **Section 5.2 is the only section that references the IANA sections 7.x, as opposed to the section 4.x tables?**

If we look at the original RFC8007 as a guide, it appears that it referenced the registries in 7.x (coincidentally the same section number) instead of the tables in the document.
This seems more correct, given that a later RFC or specification could update the registries, which could include more items than what is specified by this document.

Draft v17 plans

- Incorporating Kevin’s review responses
- Add attribute "total-nodes" to Trigger resource (Section 4.1), to indicate a number of nodes that processed the trigger. We will consider uniting "total-nodes", "total-objects-count", and "total-objects-size" into an object within the trigger resource.
- Add labels to ObjectEntry (section 4.3.2.4) and ObjectList (section 4.3.2) objects
- Add versioning as a top-level attribute in the JSON-encoded ObjectList format (section 4.3.2.2)
- Modify ExecutionPolicy extension (section 4.1.3.3.3) to support prioritization of ObjectLists and/or individual ObjectEntries processing stage. This is in addition to the existing ability to prioritize selection of triggers from the pending queue.