

CDNI FCI Analysis

CDNI Working Group

IETF 89 London

March 2014

presented by

Daryl Malas (D.Malas@cablelabs.com)

on behalf of

Matt Caulfield (mcaulfie@cisco.com)

Background

Two major competing proposals for the CDNI
Footprint and Capabilities Interface (FCI)

Volunteered at Vancouver meeting to produce a
detailed analysis comparing the proposals

Presentation summarizes email from Feb 22

draft-seedorf-cdni-request-routing-alto-06

Primarily based on the ALTO protocol

- Each dCDN hosts an ALTO server, uCDN is ALTO client
- ALTO Network Map indicates “footprint” (PID)
- Property Map provides capabilities per footprint
- Future extensions cited for incremental updates

draft-ma-cdni-capabilities-04

Primarily based on CDNI-specific representation

- Each dCDN hosts an HTTP server, uCDN is HTTP client
- Each capability has a name, list of values, and an optional list of footprints
- Each footprint has a type, list of values, and a mode
- Incremental updates via HTTP POST with seq #s

Transport & Encoding

draft-seedorf

- HTTP transport
- JSON encoding

draft-ma

- HTTP transport
- JSON encoding

JSON chosen by CDNI WG as default encoding

Data Representation

draft-seedorf

- ALTO Network Map and Property Map
- Leverages existing work done by ALTO WG

draft-ma

- CDNI-specific syntax for footprints and capabilities
- Custom solution for CDNI

Fundamental difference and decision point for WG

Hierarchy

draft-seedorf

- Footprints have capabilities
- More intuitive in the cascaded CDN case
(footprints from many dCDNs may be concatenated)

draft-ma

- Capabilities have footprints
- Less intuitive in cascaded CDN case
(requires merge of multiple capabilities)

Cost Information

draft-seedorf

- Loosely described using ALTO Cost Maps

draft-ma

- No solution described

Importance unclear, given business rules may override

Extensibility & Versioning

draft-seedorf

- Not described

draft-ma

- Not described

Details lacking, a clear gap in both drafts

Dependencies

draft-seedorf

- Depends on multiple Internet Drafts from ALTO WG
- Leverages existing error handling, security, etc.

draft-ma

- CDNI development and consensus

Fundamental difference and decision point for WG

Capability Inheritance

draft-seedorf

- PID Property Map rules for implicit inheritance may add complexity to implementations

draft-ma

- Completely explicit capabilities (no inheritance)

Explicit capabilities is a better approach

Update Notifications

draft-seedorf

- No method described for receiving update notices

draft-ma

- Asynchronous HTTP POST from dCDN to uCDN
- May violate RESTful principles

POST is a concrete approach, but potentially flawed

Incremental Updates

draft-seedorf

- Uses ALTO Incremental Update proposal
- Relies on JSON Patch for encoding

draft-ma

- HTTP POST header indicates seq# of update
- Footprints include mode for overwrite vs append

Both approaches reasonable,
but new HTTP header may not be achievable

Conclusions

- 1) Both drafts well-written and good starting points
- 2) WG must decide if the benefits of reusing ALTO syntax and semantics outweigh the costs
 - Benefits: existing error handling, security, encoding, scale, proposed ALTO drafts for extending functionality
 - Drawbacks: dependence on ALTO WG, some inflexibility
- 3) Recommend we focus on a simple HTTP GET before attempting to solve incremental updates (if ever)