RESTCONF Protocol

draft-bierman-netconf-restconf-02
IETF 88, November 2013

Andy Bierman
Martin Bjorklund
Kent Watsen
Rex Fernando
Agenda

- RESTCONF Protocol Summary
- RESTCONF Applicability
- Ready for WG adoption?
RESTCONF Summary

• Was YANG-API, now called RESTCONF
  - old: draft-bierman-netconf-yang-api-01
  - new: draft-bierman-netconf-restconf-02
• HTTP/REST API for CRUDX operations on YANG-defined resources
• Integrated with NETCONF datastores, access control, operations, and notifications
• Simple WEB Client developer API for NM
• Draft needs early cross-area review
RESTCONF Resource Model

- Server is a collection of conceptual resources
  - **api**: API framework
  - **datastore**: NETCONF datastore
  - **data**: YANG data definition statement
    - container or list, not leaf, leaf-list, or anyxml
  - **operation**: YANG rpc statement
  - **patch**: PATCH method using YANG Patch
  - **patch-status**: YANG Patch status
  - **stream**: notification stream
Datastore and Data Resources

- Datastore is split; "config" parameter removed
  - /restconf/config: config=true data nodes
  - /restconf/operational: config=false data nodes
- Entity tags (etag) maintained by the server
  - An entity tag MUST be maintained for the running datastore resource
  - An entity tag SHOULD be maintained for the data resources in the running datastore
  - Client caches etag from GET for use in If-Match header on followup edit operation
### RESTCONF Operations

<table>
<thead>
<tr>
<th>HTTP Method</th>
<th>NETCONF Operation</th>
<th>Media Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>create</td>
<td>application/yang.data</td>
</tr>
<tr>
<td>PUT</td>
<td>replace</td>
<td>application/yang.data</td>
</tr>
<tr>
<td>PATCH</td>
<td>merge</td>
<td>application/yang.data</td>
</tr>
<tr>
<td>PATCH</td>
<td>any edit operation</td>
<td>application/yang.patch</td>
</tr>
<tr>
<td>DELETE</td>
<td>delete</td>
<td>application/yang.data</td>
</tr>
<tr>
<td>POST</td>
<td>any &lt;rpc&gt; operation</td>
<td>application/yang.operation</td>
</tr>
<tr>
<td>GET</td>
<td>&lt;get&gt;, &lt;get-config&gt;</td>
<td>application/yang.data</td>
</tr>
<tr>
<td>GET</td>
<td>&lt;create-subscription&gt;</td>
<td>text/event-stream</td>
</tr>
</tbody>
</table>

- **NETCONF**: `<config>` subtree specifies data node targets
- **RESTCONF**: request URI specifies target resource
RESTCONF POST Example

Client creates a jukebox resource

    POST /restconf/config HTTP/1.1
    Host: example.com
    Content-Type: application/yang.data+json

    { "example-jukebox:jukebox" : [null] }

Server replies with Location of new jukebox resource

    HTTP/1.1 201 Created
    Date: Mon, 23 Apr 2012 17:01:00 GMT
    Server: example-server
    Location: http://example.com/restconf/config/
              example-jukebox:jukebox
    Last-Modified: Mon, 23 Apr 2012 17:01:00 GMT
    ETag: b3a3e673be2
YANG Patch

- Structure representing an ordered edit list on a target resource
  - `patch-id, comment`: operator audit info
  - `edit`: list of user-ordered named edits
    - `operation`: edit operation
    - `target`: sub-resource or field to edit
    - `point`: sub-resource or field insertion point
    - `where`: insertion location: first, last, before, after
    - `value`: configuration data for the edit
# YANG Patch Operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>create</td>
<td>Create a new data resource if it does not already exist, or error</td>
</tr>
<tr>
<td>delete</td>
<td>Delete a data resource if it exists, or error</td>
</tr>
<tr>
<td>insert</td>
<td>Insert a new user-ordered data resource (list) or field (leaf-list)</td>
</tr>
<tr>
<td>merge</td>
<td>Merge the edit value with the target, create if it does not already exist</td>
</tr>
<tr>
<td>move</td>
<td>Move a node within a user-ordered list or leaf-list</td>
</tr>
<tr>
<td>replace</td>
<td>Replace the target with the edit value</td>
</tr>
<tr>
<td>remove</td>
<td>Delete a data resource if it exists</td>
</tr>
</tbody>
</table>
YANG Patch Status

- Structure representing the response status for a YANG Patch request
  - patch-id: operator audit info
  - global-status: choice of <ok/> or <errors>
  - edit-status: status of each edit request
    - edit: list of edit status info
      - edit-status: choice of <ok/> or <errors>
YANG Patch Example

PATCH /restconf/config/example-jukebox:jukebox/
    library/artist/Foo%20Fighters/album/Wasting%20Light HTTP/1.1
Host: example.com
Accept: application/yang.patch-status+json
Content-Type: application/yang.patch+json

{  "ietf-restconf:yang-patch" : {
    "patch-id" : "add-songs-patch",
    "edit" : [
    {
        "edit-id" : 1,
        "operation" : "create",
        "target" : "/song",
        "value" : {
            "song" : {
                "name" : "Bridge Burning",
                "location" : "/media/bridge_burning.mp3",
                "format" : "MP3",
                "length" : 288
            }
        }
    }
  }
}
HTTP/1.1 409 Conflict
Date: Mon, 23 Apr 2012 13:01:20 GMT
Server: example-server
Last-Modified: Mon, 23 Apr 2012 13:01:20 GMT
Content-Type: application/yang.patch-status+json

{ "ietf-restconf:yang-patch-status" : {
   "patch-id" : "add-songs-patch",
   "edit-status" : { "edit" : [ {
      "edit-id" : 1,
      "errors" : { "error" : [ {
         "error-type": "application",
         "error-tag": "data-exists",
         "error-path": "/example-jukebox:jukebox/library
 /artist/Foo%20Fighters/album/Wasting%20Light
 /song/Burning%20Light",
         "error-message": "Data already exists, cannot be created"
      } ] } ] } } }
Notifications

- Server creates stream resources it supports
- Client starts Server Sent Events with GET

GET /restconf/streams/stream/NETCONF/events HTTP/1.1
Host: example.com
Accept: text/event-stream
Cache-Control: no-cache
Connection: keep-alive

- RFC 5277 replay buffer supported
  - start-time and stop-time query parameters
- XPath filter supported as query parameter
RESTCONF Applicability

- Applies to Client Software Developers that prefer to use HTTP/REST tools
  - libcurl, Javascript, and JSON widely deployed for many language bindings
  - Simpler unified datastore and editing model
  - Better efficiency using JSON encoding, entity tags, HEAD method
  - Server design intentionally close to NETCONF to allow easy implementation
Ready for WG Adoption?

- Multiple independent implementations of YANG-API (or something very similar) already completed
  - positive customer feedback
    - YANG is the missing piece for REST-based NM
    - Already have WEB client tools and don't want to buy/learn new ones
    - Applications that just retrieve some server data find GET very easy to use with libcurl
Summary

● NETCONF WG should adopt RESTCONF as a work item ASAP
  – Unified NM means 1 data model, not 1 protocol

● RESTCONF needs wider review from OPS/NM and APPS area experts
  – Good starting point, but needs WG and IETF review process to finish the work