Hypermedia Web API for enhanced Heterogeneous Missions Accessibility

Y. Coene, Spacebel s.a.

Frascati, June 30, 2015
Outline

• Architecture trends
  - REST
  - Hypermedia API
• Aspects of Hypermedia API
  - REST: main characteristics
  - Asynchronous operations
  - Other considerations
  - Hosted processing
  - API metadata technology
  - Identity Mgt
• Lightweight metadata model
• Conclusion
## Comparison of 3 Distributed Architecture Styles

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Object-oriented</th>
<th>Resource-oriented</th>
<th>Service-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Granularity</td>
<td>Object instances</td>
<td>Resource instances</td>
<td>Service instances</td>
</tr>
<tr>
<td>Main Focus</td>
<td>Marshalling parameter values</td>
<td>Request addressing (usually URLs)</td>
<td>Creation of request payloads</td>
</tr>
<tr>
<td>Addressing / Request routing</td>
<td>Routed to unique object instance</td>
<td>Unique address per resource</td>
<td>One endpoint address per service</td>
</tr>
<tr>
<td>Are replies cacheable?</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Application interface</td>
<td>Specific to this object / class - description is middleware specific (e.g. IDL)</td>
<td>Generic to the request mechanism (e.g. HTTP verbs)</td>
<td>Specific to this service - description is protocol specific (e.g. WSDL)</td>
</tr>
<tr>
<td>Payload / data format description</td>
<td>Yes – usually middleware specific (e.g. IDL)</td>
<td>No – nothing directly linked to address / URL</td>
<td>Yes – part of service description (e.g. XML Schema in WSDL)</td>
</tr>
</tbody>
</table>
• REST
• JSON API
- **JSON**
  - OGC 15-053, Implementing JSON/GeoJSON in an OGC Standard (J. Maso) – 35 recommendations to OGC.
  - OGC 15-033, RESTful API and JSON Encoding WaterML, Best Practice
  - OGC 14-113, OGC JSON Position Statement (Carl Reed).
  - OGC 14-009r1, Engineering Report, "OGC Testbed-10 Rules for JSON and GeoJSON Adoption: Focus on OWS-Context"
  - Testbed-11/12: define rules for JSON and GeoJSON extensions to OGC Web Services encodings.
• OASIS
  - OData JSON Format Version 4.0
Architectural Trends - OASIS

- OASIS CAMP 1.1
  - RESTful
  - JSON representation
• Support for JSON-LD markup.

JSON-LD 1.0
A JSON-based Serialization for Linked Data
W3C Recommendation 16 January 2014
This version:
http://www.w3.org/TR/2014/WD-json-ld-20140116/
Latest published version:
http://www.w3.org/TR/2015/REC-json-ld-20150113/
Previous version:
http://www.w3.org/TR/2013/PR-json-ld-20131105/
Editors:
Marty Spero, Digital Bazaar
Group Kibibi, Kilobyte Associates
Markus Lanthaler, Grad University of Technology
Authors:
Marty Spero, Digital Bazaar
Laura Lastow, Digital Bazaar
Group Kibibi, Kilobyte Associates
Markus Lanthaler, Grad University of Technology
Nikola Lanthaler
Please refer to the errata for this document, which may be
This document is also available in this non-normative for
The English version of this specification is the only norm
Copyright © 2014 W3C® (MIT, ERCIM, Keio, Beihang). All Rights

JSON-LD is the newest and simplest markup format: It lets you embed a block of
JSON data inside a script tag anywhere in the HTML. Since the data does not
have to be interleaved with the user-visible text, it’s much easier to express nested
data items (say, the Country of a PostalAddress of a MusicVenue of an Event). Also,
Google can read JSON-LD data even when it is dynamically injected into the page’s
contents, such as by Javascript code or embedded “widgets”.

Google is in the process of adding JSON-LD support to more markup-powered
features. So far, JSON-LD is supported for all Knowledge Graph features, sitelink
search boxes, and Event Rich Snippets; Google recommends the use of JSON-LD for
these features. For the remaining Rich Snippets types and breadcrumbs, Google
recommends the use of microdata or RDFa.
Outline

- Architecture trends
  - REST
  - Hypermedia API
- Aspects of Hypermedia API
  - REST: main characteristics
  - Asynchronous operations
  - Other considerations
  - Hosted processing
  - API metadata technology
  - Identity Mgt
- Lightweight metadata model – JSON-LD
- Conclusion
REST Maturity Model

- Maturity Model (Richardson)

- Service provides link to other resources
- "verbs", e.g. CRUD
- Web services
- "resources"
- Web services with XML Payload (RPC, OGC, SOAP)
Main characteristics

- Statelessness constraint
  - Stateless client-server interaction

- Uniform interface constraint
  - (a) Identification of resources
    - At application domain level
    - Identified by URL
  - (b) Manipulation of resources through representations
    - Representation represents state of resource
  - (c) Self-descriptive messages with well-defined semantics (GET, POST, PUT, DELETE, ...)
  - (d) Hypermedia constraints (HATEOAS)
• RESTful HMA Model definition started in HMA-S with "Ordering". For complete model, refer to OGC 13-042 "RESTful interface for ordering".
(a) Resource identification

- Current RESTful HMA Model
  - Partial coverage or "resources"
    - OGC 13-042 "RESTful interface for ordering"
    - OGC 14-012r1, RESTful encoding of Sensor Planning Service for Earth Observation Satellite Tasking.
    - OGC 13-043, Download service for EO Products, Version 1.0, 31/01/2014.

- Future work:
  - Complete coverage of all resources
(b) Representations

- **Current model**
  - Different Media Types per specification:
    - Atom
    - XML
    - OSDD

- **Future work:**
  - Separate "representations" from rest of model to facilitate extension with additional media types.
  - Same media types across specifications: e.g. Atom, JSON-LD
  - Allow representation choice by client or server:
    - Via content-negotiation (Header Parameter)
    - Via resource extension (Path Parameter, e.g. ".atom")
(b) Representations

- Preference for media types supporting hypermedia:
  - XML and JSON do not include such support. Require Xlink, Atom, HAL etc. extensions.

- Trade-off:
  - API to use different media type to cope with different semantics (not every XML is equal)
  - Extend media type with "profile=" (RFC 6906)
  - Use generic media type allowing to express different semantics.
    - Complies with REST "uniform interface" requirement
    - JSON-LD : has semantic expressivity of RDF
(d) Hypermedia constraints

- Hypermedia as the Engine of Application State
  - Simultaneous presentation of information and controls (*actionable*)
    - Defines valid "state transitions"
    - Permitted actions within responses
  - Standardised Link Relations `<link rel="…">` (RFC 5988)
    - ‘search’: Two step search
    - ‘enclosure’: Product download
    - ‘order’: Product order page or ordering capabilities (OGC 06-141) or OSDD (OGC 13-042)
      - type="text/html"
      - type="application/vnd.ogc.oseo.capabilities.response_xml" (TBC).
      - type="application/opensearchdescription+xml"
  - Hypermedia formats: Atom, JSON-LD
(d) Hypermedia constraints

- JSON-HAL (Hypertext application Language)
  - https://tools.ietf.org/html/draft-kelly-json-hal-06

- JSON-Schema
  - http://json-schema.org

```json
{
  "links": [
    {
      "rel": "self",
      "href": "http://fedeo.esa.int/series/A3A_IN_OF/datasets/A3A_IN_OCNPDE20120407_102338_000000173113_00257_52557_6356.N1"
    },
    {
      "rel": "alternate",
      "href": "http://fedeo.esa.int/series/A3A_IN_OF/datasets/A3A_IN_OCNPDE20120407_182338_000000173113_00257_52557_6356.N1.atom",
      "mediaType": "application/atom+xml"
    },
    {
      "rel": "enclosure",
      "href": "https://cs-virtual-archive4.esa.int/supersites/A3A_IN_OCNPDE20120407_182338_000000173113_00257_52557_6356.N1",
      "mediaType": "application/binary"
    }
  ]
}
```
(d) Hypermedia constraints

- JSON-LD approach proposed in OGC 15-053:
  - Key name corresponding to Atom link relation "rel"
  - Optionally consider Hydra vocabulary

```json
{
  "@id": "http://fedeo.esa.int/series/ASA_IM_0?/datasets/ASA_IM_0CNPDE20120407_182038_000000173113_00257_52857_6356.N1",
  "@type": "data:DataSet",
  "atom:enclosure": {
    "@id": "https://eo-virtual-archive4.esa.int/supersites/ASA_IM_0CNPDE20120407_182038_000000173113_00257_52857_6356.N1",
    "@type": "atom:link",
    "atom:length": 193003137,
    "atom:title": "File in the N1 [ENVISAT] format",
    "atom:type": "application/binary"
  },
  "dc:identifier": "ASA_IM_0CNPDE20120407_182038_000000173113_00257_52857_6356.N1",
  "dc:term:spatial": "MULTIPOLYGON((-120.992484 49.133766,-121.987525 49.24517,-122.270375 48.105365,-121.312357 47.993146),
        "cop:instrument": "ASAR",
        "cop:orbitDirection": "DESCENDING",
        "cop:orbitNumber": "52357",
        "cop:parentIdentifier": {
          "@id": "http://fedeo.esa.int/series/ASA_IM_0?"
        },
        "cop:platform": "ENVISAT",
        "ical:dtend": "2012-04-07T18:20:55.043Z",
        "ical:dtstart": "2012-04-07T18:20:38.310Z"
    
```
Asynchronous operations

- Consistent model for asynchronous REST operations
  - Model already applied in OGC 13-043 Download Service for EO Products (HMA-S)
  - HTTP codes: 202 and 303.
  - Extensions to OGC 13-043:
    - Allow different more resource representations, including "hypermedia".

<table>
<thead>
<tr>
<th>Response Name</th>
<th>HTTP Status</th>
<th>Response Type</th>
<th>MIME Type</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct Download</td>
<td>200 – Ok</td>
<td>Binary</td>
<td>application/octet-stream</td>
<td>Specific to the product file format</td>
</tr>
<tr>
<td>Metalink Download</td>
<td>200 – Ok</td>
<td>XML</td>
<td>application/metalink+xml</td>
<td>Metalink.xsd</td>
</tr>
<tr>
<td>Partial Content</td>
<td>206 – Partial Accepted</td>
<td>Binary</td>
<td>application/octet-stream</td>
<td></td>
</tr>
<tr>
<td>Accepted Download</td>
<td>202 – Accepted</td>
<td>XML</td>
<td>text/xml</td>
<td>geo.xsd</td>
</tr>
<tr>
<td>Forwarded Download</td>
<td>303 – See Other</td>
<td>ASCII</td>
<td>text/html</td>
<td>See http specifications</td>
</tr>
</tbody>
</table>

Table 8-6: GetProduct operation responses.
Asynchronous operations

Future work:

- Additional resource representations
  Atom, JSON-LD (content negotiation)
- Exploit resource constraints: e.g. allow "Cancel"

HTTP/1.1 202 Accepted
Location: /queue/621252
GET /queue/621252 HTTP/1.1

HTTP/1.1 200 OK
Content-Type: text/xml;

<?xml version="1.0" encoding="UTF-8"?>
<dseo:ProductDownloadResponse xmlns:dseo="http://www.opengis.net/spec/DSEO/1.0/schema/ProductDownloadResponse"
xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/spec/DSEO/1.0/schema/ProductDownloadResponse">
  <atom:link rel="http://www.opengis.net/spec/DSEO/1.0/Cancel" method="Delete" href="/queue/621252"/>
  <dseo:ResponseCode>IN_PROGRESS</dseo:ResponseCode>
  <dseo:RetryAfter>30</dseo:RetryAfter>
</dseo:ProductDownloadResponse>
Other Design considerations

- API Design Issues
  - "N+1 select problem"
  - Reference Expansion (a.k.a. Entity expansion, Link expansion).
    E.g. support "fields" selector as in Google APIs
  - Paging
OGC 14-065: WPS 2.0

iii. Preface

In contrast to the prior version, WPS 2.0 provides a core conceptual model that may be used to specify a WPS in different architectures such as REST or SOAP.

If OGC baseline and related specifications should further progress towards REST-oriented interfaces, the development of a REST-oriented WPS interface standard should be considered.

Future work:

- REST Model Extension
- JSON-LD Encoding
Hosted Processing

WPS 2.0 Core Specification

- WPS 2.0 Conceptual Model (OGC 14-065)
- WPS 2.0 Common WPS Service Model (OGC 14-065)
- Dismiss Extension (OGC 14-065)
- WPS 2.0 asynchronou... (OGC 13-070)
- WPS 2.0 Transactional Discussion Paper (OGC 13-071)

- WPS 2.0 HTTP/GET KVP Binding Extension (OGC 14-065)
- JSON-LD Encoding Extension
  - To be defined ...
- SWECommon Encoding Extension
  - To be defined ...
- WPS 2.0 RESTful Model Extension
  - To be defined ...
Hosted Processing

- WPS RESTful Model to cover:
  - Asynchronous use case
  - Posting/Publication of "processing modules"? (i.e. "Transactional" capabilities)
- Relation with chaining/orchestration?
  - Facilitated with REST homogeneous interface and "actionable" links?
  - Description of processes and input/output? OWSContext (OGC 12-084r2)
  - OASIS TOSCA (Topology and Orchestration Specification for Cloud Applications), ....
Hosted Processing

- Convergence with Platform-as-a-Service (PaaS) specs
  - Cloud API "standards"
    - OASIS CAMP (Cloud Application Mgt for Platforms),
    - OGF OCCI (Open Cloud Computing Interface),
  - Commercial Cloud API
  - Container technology (de-facto standards)
    - Docker, Rocket, ...
    - Open Container Project (OCP) - http://www.opencontainers.org/
API Metadata Technology

- Make available online API documentation with examples.
  - Swagger (http://swagger.io/)
  - RAML (http://raml.org/), Blueprint (https://apiblueprint.org/),
  - W3C Hydra (https://www.w3.org/community/hydra/)
  - https://developers.google.com/discovery/v1/reference/apis
Discovery Document

Contents
- Methods
- Resource Representations

The Discovery Document describes the surface for a particular version of an API. The Discovery Document includes API-level properties, such as a description, resource schemas, authentication scopes, and methods.

Methods
The Discovery document focuses on the RESTful method of invocation. The discovery.apis.getRest method retrieves the REST-based Discovery API.

getRest
Retrieve the description of a particular version of an API.

list
Retrieve the list of APIs supported at this endpoint.

Resource Representations

```json
{
    "kind": "DiscoveryRestDescription",
    "description": "v1",
    "id": "string",
    "name": "string"
}
```
• Left: Hydra vocabulary used in RDF Schema definitions for "properties" and "methods".
• Right: RDF Schema (OBEOS) example.

Listing 2. Definition of the comments property (excerpt)
A RESTful API should be stateless. This means that request authentication should not depend on cookies or sessions. Instead, each request should come with some sort authentication credentials.

By always using SSL, the authentication credentials can be simplified to a randomly generated access token that is delivered in the user name field of HTTP Basic Auth. The great thing about this is that it's completely browser explorable - the browser will just popup a prompt asking for credentials if it receives a 401 Unauthorized status code from the server.

However, this token-over-basic-auth method of authentication is only acceptable in cases where it's practical to have the user copy a token from an administration interface to the API consumer environment. In cases where this isn't possible, OAuth 2 should be used to provide secure token transfer to a third party. OAuth 2 uses Bearer tokens & also depends on SSL for its underlying transport encryption.

**Future work:**
- Integrate OAuth 2.0/SSL (RFC6750 "Bearer Token Usage") use case in OGC 07-118r9.

Outline

- Architecture trends
  - REST
  - Hypermedia API
- Aspects of Hypermedia API
  - REST: main characteristics
  - Asynchronous operations
  - Other considerations
  - Hosted processing
  - API metadata technology
  - Identity Mgt
- Lightweight metadata model
- Conclusion
Lightweight EOP Data Model

- Encoding of product metadata (OGC 10-157r4) in JSON-LD
  - Simplified – less mandatory fields
  - Flatter structure
    - More than generic automatic EOP O&M (XML) to JSON-LD transformation.
    - Support "fields" selector as in Google APIs
  - Predicates taken from OGC 10-157r4 and to be registered at OGC NA.
  - Exploiting well-known vocabularies (namespaces) where useful including Dublin Core, W3C DCAT, Schema.org, GeoJSON, etc.
  - HATEOAS "rel" for next actions (in header or in JSON-LD).
    - Is beyond pure metadata. E.g. link to order the product, to search the collection, to download the product, to get the ordering options etc.
«MULTIPOLYGON (((-120.99 49.13, -121.98 49.24, 122.27 48.10, -121.31 47.99, ...)))»

«DESCENDING»

«2012-04-07T18:20:38»

«2012-04-07T18:20:55»

«ASA_IM__0CNPDE20120407_182038_000000173113_00257_52857_6356.N1»

Lightweight EOP Data Model

HMA-AWG | Hypermedia Web API | Frascati | 30 June 2015
Lightweight EOP Data Model

• Model in W3C JSON-LD

```json
{
  "@context": {
    "do": "http://purl.org/dc/elements/1.1/",
    "dcat": "http://www.w3.org/ns/dcat#",
    "dcterms": "http://purl.org/dc/terms/",
    "eop": "http://www.opengis.net/eop/2.1/",
    "ical": "http://www.w3.org/2002/12/ical#",
    "cdf": "http://www.w3.org/1999/02/22-cdf-syntax-ns#",
    "sdo": "http://www.w3.org/2000/01/XMLSchema#",
    "xsd": "http://www.w3.org/2001/XMLSchema"
  },
  "@id": "http://fedeo.esa.int/series/ASA_TM_0P/datasets/ASA_TM_CONEDE20120407_182038_000000173113_00257_52857_6556.N1",
  "@type": "dcat:Dataset",
  "dcterms:spatial": "MULTIPOLYGON ((-120.982484 49.135766, -121.987525 49.24517, -122.270375 48.105385, -121.332357 47.998146, -120.982484 49.135766))",
  "eop:instrument": "ENSAT",
  "eop:orbitDirection": "DESCENDING",
  "eop:orbitNumber": "52857",
  "eop:parentIdentifier": {
    "@id": "http://fedeo.esa.int/series/ASA_TM_0P"
  },
  "eop:platform": "ENVIROAT",
  "eop:satelliteIdentifier": "I36",
  "eop:wrsLatitudeGrid": "2619",
  "eop:wrsLongitudeGrid": "257",
  "ical:dtend": "2012-04-07T18:20:05.400Z",
  "ical:dtstart": "2012-04-07T18:20:38.120Z"
}
```
Lightweight EOP Data Model

- With semantic annotations: e.g. "eop:instrument"

```json
{
  "$schema": {
    "dc": "http://purl.org/dc/elements/1.1/",
    "dcat": "http://www.w3.org/ns/dcat#",
    "dcterm": "http://purl.org/dc/terms/",
    "eop": "http://www.opengis.net/eop/2.1/",
    "ical": "http://www.w3.org/2002/12/iCal/iCal#",
    "rdfs": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
    "skos": "http://www.w3.org/2004/02/skos/core#",
    "xsd": "http://www.w3.org/2001/XMLSchema#"
  },
  "id": "http://fedco.eea.int/series/ASA_IM_GS/datasets/ASA_IM_GCFE20120407_162038_000000173113_00257_52857_6356.NL",
  "type": "GeoDataSet",
  "dc:identifier": "ASA_IM_GCFE20120407_162038_000000173113_00257_52857_6356.NL",
  "dcterms:spatial": "MULTIPOLYGON((-120.892484 49.133766, -121.887526 49.24517, -122.270375 48.105365, -121.312357 47.988146, -120
  "eop:instrument": {
    "$id": "http://gcmdservices.gsfc.nasa.gov/kms/concept/912c3308-23bc-4e12-k"
  },
  "eop:orbitDirection": "DESCENDING",
  "eop:orbitNumber": "52857",
  "eop:parentIdentifier": {
    "$id": "http://fedco.eea.int/series/ASA_IM_GS"
  },
  "eop:platform": "ENVI", "eop:swathIdentifier": "ISS",
  "eop:wrsLatitudeGrid": "2619",
  "eop:wrsLongitudeGrid": "257",
  "ical:dtend": "2012-04-07T18:20:38.04Z",
  "ical:dtstart": "2012-04-07T18:20:38.810Z"
}
```
Lightweight EOP Data Model - choices

- Several alternatives for vocabulary, e.g. polygon:
  - `<dcterms:spatial>` (WKT)
  - GeoJSON + GeoJSON-LD (geojson.org/vocab)
  - GeoSPARQL (WKT)
  - GeoJSON v WKT?

```json
{ "type": "Polygon",
  "coordinates": [
    [[100.0, 0.0], [101.0, 0.0], [100.0, 1.0], [101.0, 0.0], [100.0, 0.0]]
}
```
Lightweight EOP Data Model - choices

- GeoSPARQL geometries

```xml
<sf:Polygon rdf:about="http://example.org/ApplicationSchema#AEkactGeom">
  <geo:asWKT rdf:datatype="http://www.opengis.net/ont/geosparql#wktLiteral">
    <![CDATA[
      Polygon((-83.6 34.1, -83.2 34.1, -83.2 34.5, -83.6 34.5, -83.6 34.1))
    ]]> 
  </geo:asWKT>
</sf:Polygon>
```
• OGC 13-026r5 "EO Extension of OpenSearch"
  – Currently Atom response type with
    • ISO 19139 <atom:link>s or payload.
    • OGC 10-157r4 <atom:link>s or payload.
  – Support for JSON-LD response type:
    • "Feed" and "entry" level metadata to be mapped to JSON-LD.
    • Consider "paging" options:
      – Paging with rel="..." in HTTP header, or HTTP body (RFC 5988)?
    • Consider HATEOAS atom:link rel:"..." equivalent in JSON-LD
      – rel="search"
      – rel="enclosure"
      – rel="order", etc...
    • Consider definition of relations with namespace: e.g. "eo:order",
      "eo:..." for missing "operations" beyond CRUD or custom MIME types
      matching "resource" model.
Future work:

- JSON-LD representation for:
  - OGC 10-157r4 (Product Metadata)
  - OGC 11-035r1 / ISO19139 / INSPIRE (Collection Metadata)
    - See https://github.com/adiwg/mdBook/blob/master/mdjson_schemas/README.md
  - OGC 13-026r5 (OpenSearch Responses)
Conclusion

• Trends
  – Web Hypermedia API
  – JSON(-LD) encoding

• Model applicable in HMA Context
  – Partly addressed in HMA-S
  – Presented several topics for additional / future work