OAPI-Coverages

Peter Baumann

October, 2021

The world's leading and comprehensive community of experts making location information:

- Findable
- Accessible
- Interoperable
- Reusable
Wait...”Coverages”?

- Coverage = n-D "field"
  - regular & irregular grids, point clouds, meshes
- OGC Coverage Implementation Schema (CIS) 1.1
  - Identical to ISO 19123-2, EU INSPIRE coverages
  - Incompatible: CoverageJSON
- Basic structure of a coverage:
Coverages – in GML, JSON, RDF, ...

```xml
<domainSet>
  <generalGrid srslName="http://www.opengis.net/def/crs/OGC/0/Index2D">
    <axisLabels>
      <regularAx>
        <irregular>
          <regularAx>
            <irregular>
              0<=>100
            </irregular>
            201<=>201
          </irregular>
          <irregular>
            <regularAx>
              0<=>100
            </regularAx>
            201<=>201
          </irregular>
        </irregularAx>
      </axisLabels>
      <generalGrid>
        <domainSet type="CoverageByDomainAndRangeType">
          <rangeSet type="RangeSetType">
            <dataBlock type="VDataBlockType">
              <values>1,2,3,4,5,6,7,8,9</values>
            </dataBlock>
          </rangeSet>
          <rangeType>
            <field definition="ogcType:unsignedInt" uom="10^0"/>
          </rangeType>
        </domainSet>
      </generalGrid>
    </axisLabels>
  </generalGrid>
</domainSet>
```

Interoperability: can translate all
Coverage Service Model, Classic

- Many services can support coverages
  - Web Coverage Service, WMS, WPS, SOS, ...

- Modular WCS suite
  - from simple subsetting to complex analytics
• Member of the OAPI-* family:
  • being accessible from an API landing page for a particular dataset,
  • enabling the API to be described and documented using OpenAPI,
  • defining conformance classes specific to coverages,
  • providing access to geospatial data as a coverage.

• Primary focus on (YAML +) HTML + JSON

• Status: draft, under active development
  • Spec: https://github.com/opengeospatial/ogcapi-coverages

• Several implementations
  • Gnosis, INRS, EOX, Spatialys, rasdaman, 52° North, Pangaea Innovation, CREA F, CubeWerx, CRIM, ...
OAPI-Coverages: Access Structure

- List collections
  - Description of {coverageId}
    - All but range set, coverage metadata
    - The coverage itself
      - Optional query parameters:
        - Subset
        - Range subset
        - Scaling
        - Bbox
        - Coverage parts:

{datasetAPI}/collections
{datasetAPI}/collections/{coverageId}
{datasetAPI}/collections/{coverageId}/coverage
{datasetAPI}/collections/{coverageId}/coverage?subset=Lat(40:50),Lon(10:20)
{datasetAPI}/collections/{coverageId}/coverage?range-subset=B02,B03,B04
{datasetAPI}/collections/{coverageId}/coverage?scale-factor=2
{datasetAPI}/collections/{coverageId}/coverage?bbox=10,40,20,50
{datasetAPI}/collections/{collectionId}/coverage/domainset
{datasetAPI}/collections/{collectionId}/coverage/rangetype
{datasetAPI}/collections/{collectionId}/coverage/rangeset
name: subset
in: query
description: |

The `subset` parameter allows for subsettings against a given coverage axis with the following characteristics (using an Extended Backus Naur Form (EBNF) fragment):

```
SubsetSpec: "subset"=axisName(intervalOrPoint)[,axisName(intervalOrPoint)]*
axisName: \{text\}
intervalOrPoint: interval | point
interval: low : high
low: point | *
high: point | *
point: \{number\} | \"{text}\"
Where:
\" = double quote = ASCII code 0x42,
\{number\} is an integer or floating-point number, and
\{text\} is some general ASCII text (such as a time and date notation in ISO 8601).
```

required: false
schema:
type: string
• Modular OGC API Workflows
  • Ecere, Université Laval, INRS, EOX, Spatialys, rasdaman, 52° North, Pangaea Innovation, CREA, CubeWerx, CRIM
• 2020-2021
• funding: NRCan
• Impact:
  • OAPI-Coverages
  • OAPI-Processes
WCS vs OAPI-Coverages: High-Level Comparison

- Space/time bbox subsetting ✓ ✓
- Format encoding ✓ ✓
- Range subsetting ✓ ✓
- Reprojection ✓ ✓
- Scaling ✓ ✓
- Interpolation ✓ ×
- Datacube analytics (WCPS) ✓ ✓
- Transaction = insert/update/delete ✓ ×
Subsetting as example for functionality comparison

**WCS:**
```
?subset=Long(100,120)&subset=Lat(50,60)&subset=time("2009-11-06T23:20:52")
```

**OAPI:**
```
?bbox=160.6,-55.95,-170,-25.89
?subset=Lat(40:50),Lon(10:20),time("2009-11-06T23:20:52")
```

Generic n-D subsetting as conformance class in OAPI-Common
Follow links, carefully understand link relation types!
- “this path is not fixed and may not exist (follow the link)"
- Ex: `{datasetAPI}/collections/{collectionId}/coverage/domainset` vs external reference via link

Expect the unexpected → „resource sniffing“
- WCS Capabilities: list only coverages
- OAPI `{datasetAPI}/collections` : list data available, “some of which may support being accessed as a coverage”

If in doubt, check user guide (to come)
- https://github.com/opengeospatial/ogcapi-coverages/tree/master/users-guide

If JSON, use OGC Coverage Implementation Schema: CIS JSON
- CoverageJSON, W3C Coverages incompatible
WCPS Datacube Analytics

- OGC Web Coverage Processing Service = geo datacube analytics language
  - Semantic interoperability
  - automatic server-side optimization + parallelization + distribution

- Protocol independent:
  - WCS: http://acme.com/wcs?SERVICE=WCS&VERSION=2.0.1&REQUEST=ProcessCoverages&QUERY=for $c in (ERA5-cube) return max( $c.temperature )
  - OAPI: http://acme.com/oapi/wcps?q=for $c in (ERA5-cube) return max( $c.temperature )

- Proven on multi-PB EarthServer.xyz datacube federation

basis of ISO 19123-3
**Summary**

- **OAPI-Coverages**: OpenAPI support for coverage services
  - Based on Coverage Implementation Schema (CIS) standard
  - YAML-centric

- **Evolution, not revolution of coverage services**
  - Code can reuse WCS functionality → preserve investment
  - Subtle changes → caveat developer

- **To be done:**
  - Start test suite
  - 35 open issues + 7 todos
  - User guide
  - Start adoption process

- **Enhancement to OGC’s service ecosystem**