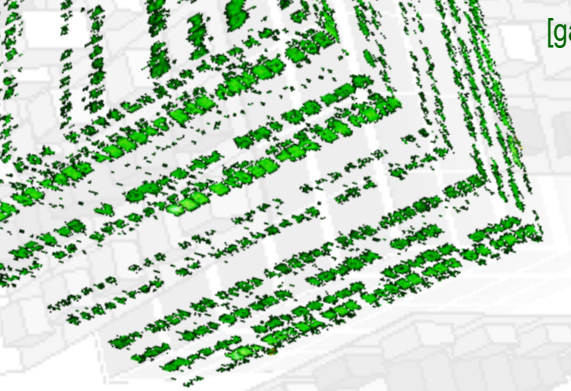


[gamingfeeds.com]

Status of the WCS 2.0 Standard

4th Workshop On The Use Of GIS/OGC Standards In Meteorology
ECMWF, Reading, 2013-mar-05

Peter Baumann
Jacobs University | rasdaman GmbH

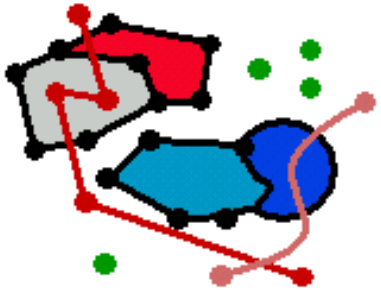
Research funded through EU *EarthServer* and ESA *DREAM*

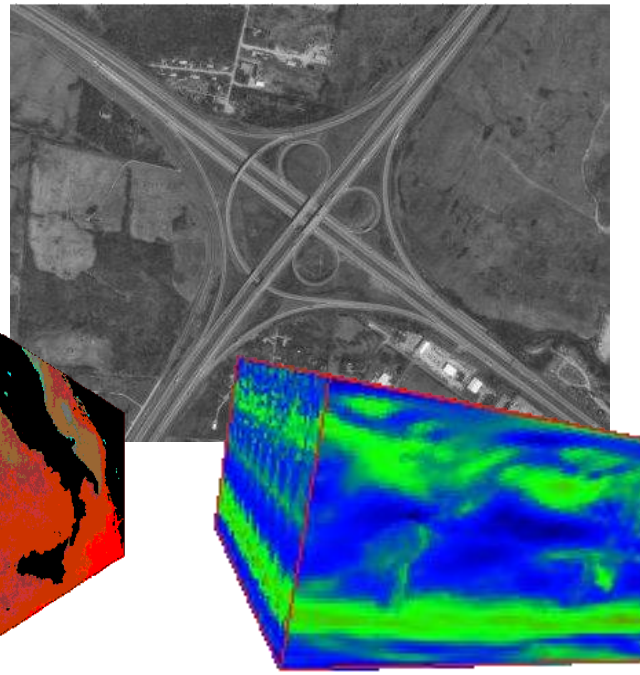
Roadmap

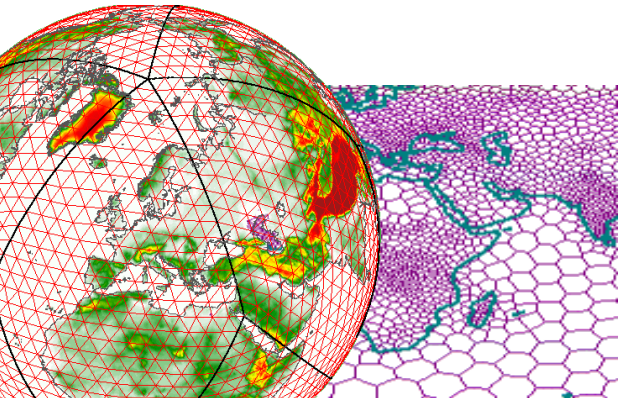
- Motivation
- Coverages: the data structure
- WCS: the data access service
- WCPS: the query language for ad-hoc processing & filtering
- WCS vs WPS vs SWE: brief comparison
- Wrap-up

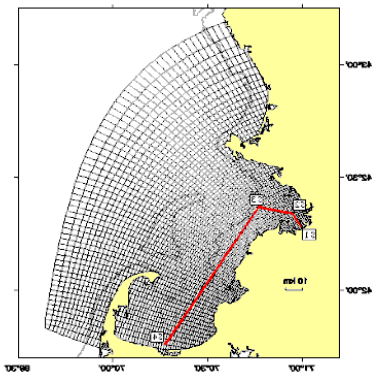
Features & Coverages

- The basis of all: geographic feature
  - = *abstraction of a real world phenomenon* [OGC, ISO]
  - associated with a location relative to Earth
- Special kind of feature: coverage
  - = *space-time varying multi-dimensional phenomenon*
  - Classic: 2-D raster image
  - *...but there is more!*
- Often Big Geo Data are coverages





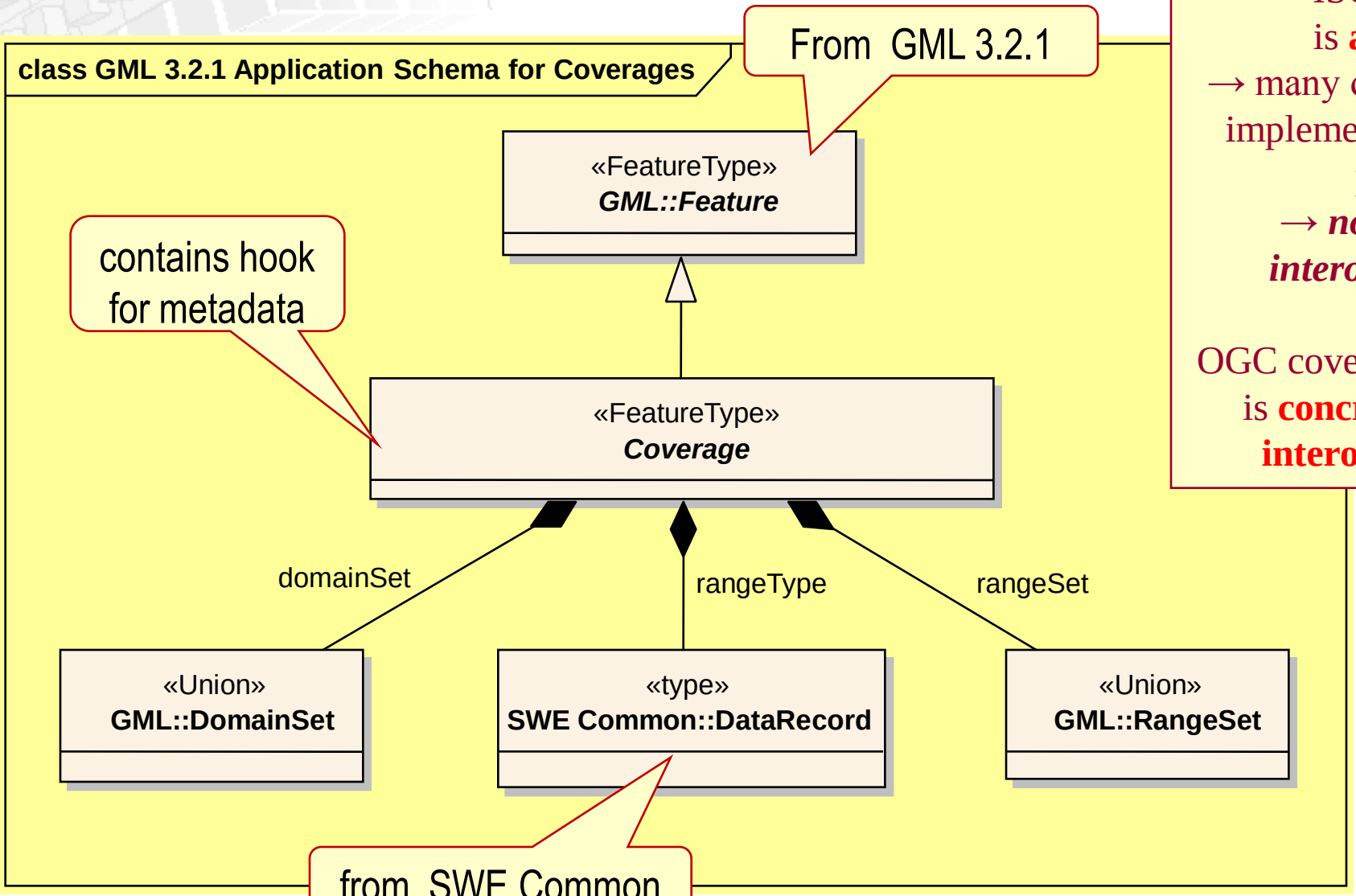




WCS Evolution

- WCS 1.0.0
  - 34p + 15p annexes
  - First attempt; limited to lat/lon + time; in places inconcise, no rigorous testing
- WCS 1.1.0
  - 56p + 60p annexes
  - More concise, but complex (65p CRS discussion!)
  - Corrigenda: WCS 1.1.1, 1.1.2
- WCS 2.0
  - 43 requirements
  - Data & service model separated
  - All coverage types, n-D, harmonized, testable, interoperable

Coverage Data Structure



ISO 19123 is **abstract** → many different implementations possible → ***not per se interoperable***

OGC coverage std is **concrete** and **interoperable**

[OGC 09-146r2]

Coverage Subtypes

all n-D !

as per GML 3.2.1

«FeatureType»
Abstract Coverage

Grid Coverage

MultiSolid Coverage

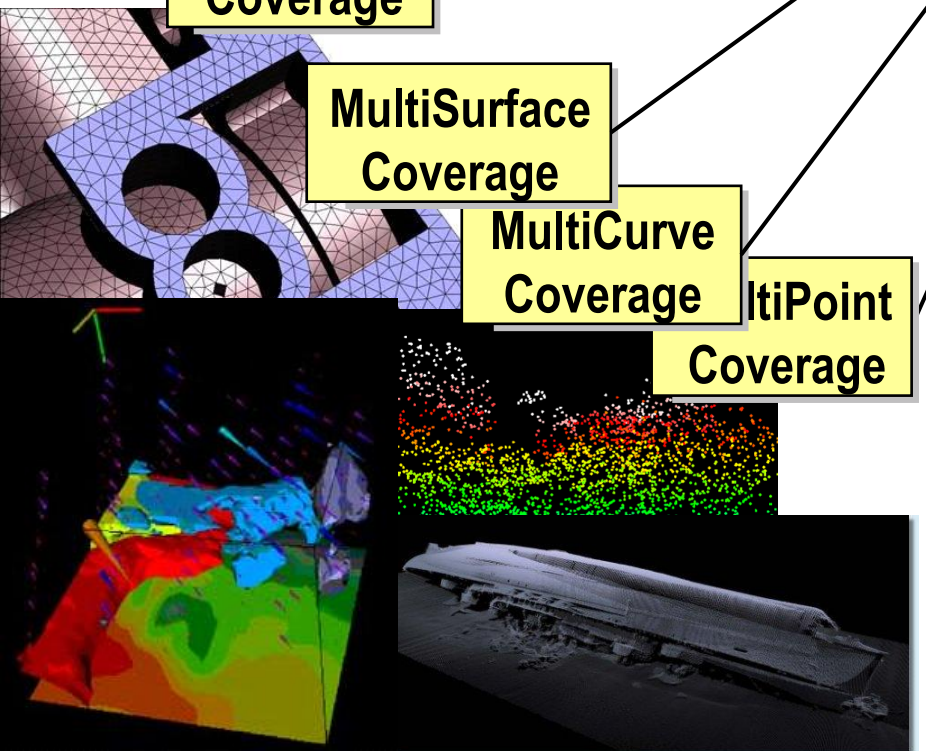
MultiSurface Coverage

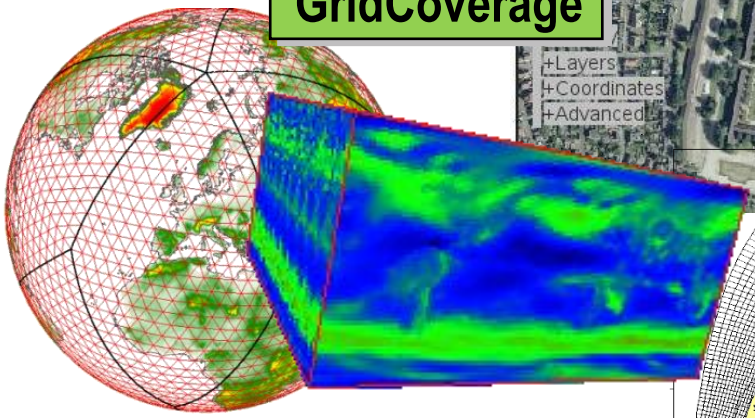
MultiCurve Coverage

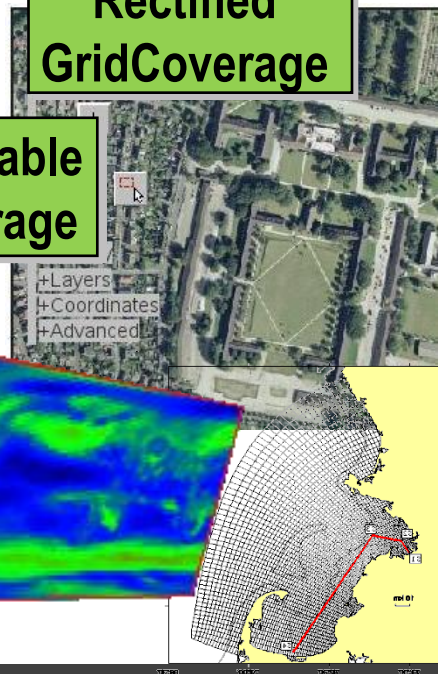
ltiPoint Coverage

Rectified GridCoverage

Referenceable GridCoverage

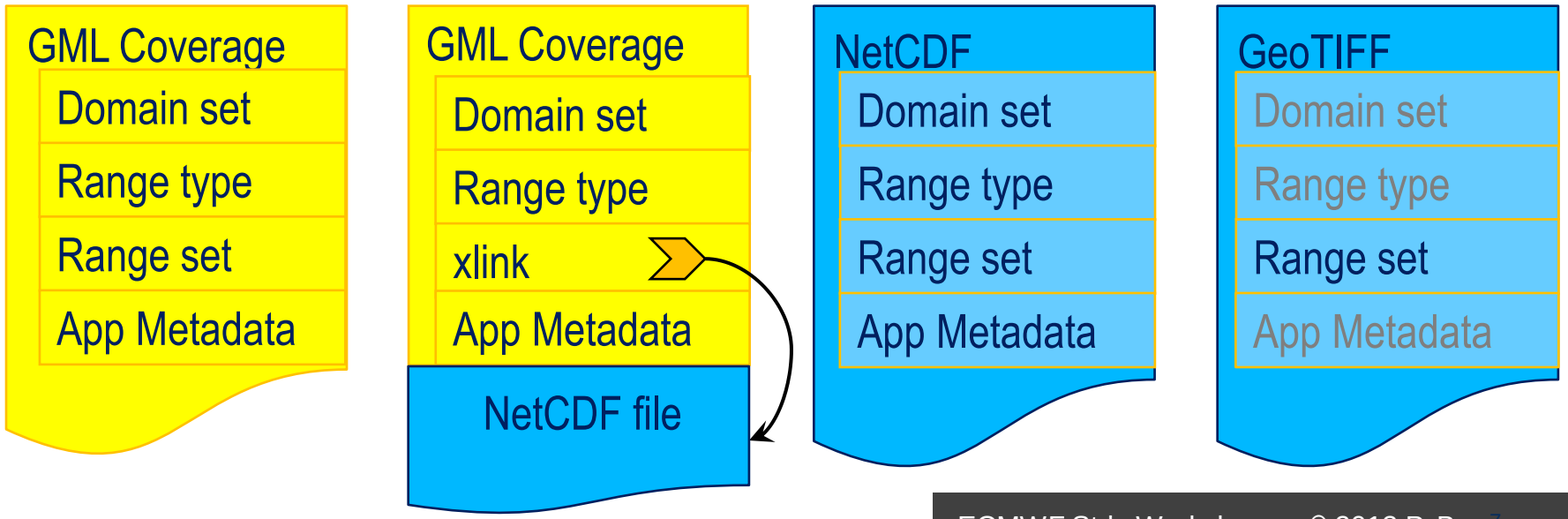






Coverage Encoding

- Pure GML: complete coverage represented by GML
- Special Format: other suitable file format (ex: MIME type “image/tiff”)
- Multipart-Mixed: multipart MIME, type “multipart/mixed”
  - Option (future): more files



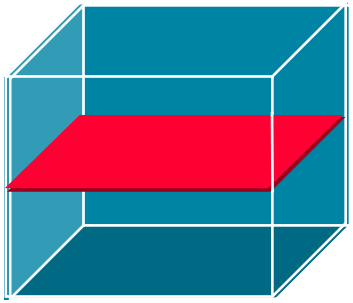
Adding Metadata To Coverages

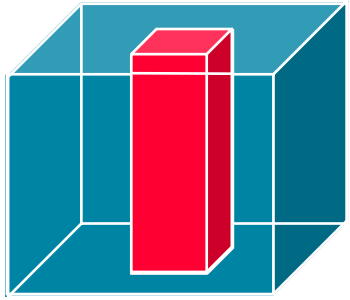
- Coverage has slot „metadata“ allowing to link in <any> kind of metadata
  - WCS will deliver this, even without knowing contents
- Ex: EO-WCS *GetCoverage* result contains EO-Metadata

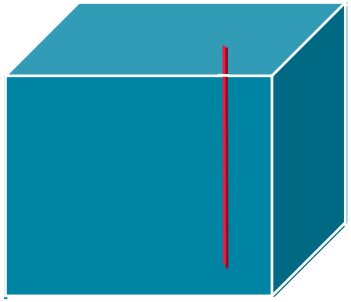
```
−<wcseo:RectifiedDataset
 gml:id="MER_FRS_1PNPDE20060822_092058_000001972050_00308_23408_0077_uint16_r
 xsi:schemaLocation="http://www.opengis.net/wcseo/1.0 http://schemas.opengis.net/wcs
 /1.0/wcsEOAll.xsd">
 +<gml:boundedBy></gml:boundedBy>
 +<gml:domainSet></gml:domainSet>
 +<gml:rangeSet></gml:rangeSet>
 +<gmlcov:rangeType></gmlcov:rangeType>
 −<gmlcov:metadata>
  −<wcseo:EOMetadata>
   −<eop:EarthObservation
    gml:id="eop_MER_FRS_1PNPDE20060822_092058_000001972050_00308_23408_007
    xsi:schemaLocation="http://www.opengis.net/opt/2.0 ../xsd/opt.xsd">
    +<om:phenomenonTime></om:phenomenonTime>
    +<om:resultTime></om:resultTime>
    +<om:procedure></om:procedure>
     <om:observedProperty xlink:href="#params1"/>
    +<om:featureOfInterest></om:featureOfInterest>
     <om:result/>
    +<eop:metaDataProperty></eop:metaDataProperty>
    </eop:EarthObservation>
   +<wcseo:lineage></wcseo:lineage>
   </wcseo:EOMetadata>
  </gmlcov:metadata>
 </wcseo:RectifiedDataset>
```

WCS Core: Simply Subsetting

- subset = trim | slice



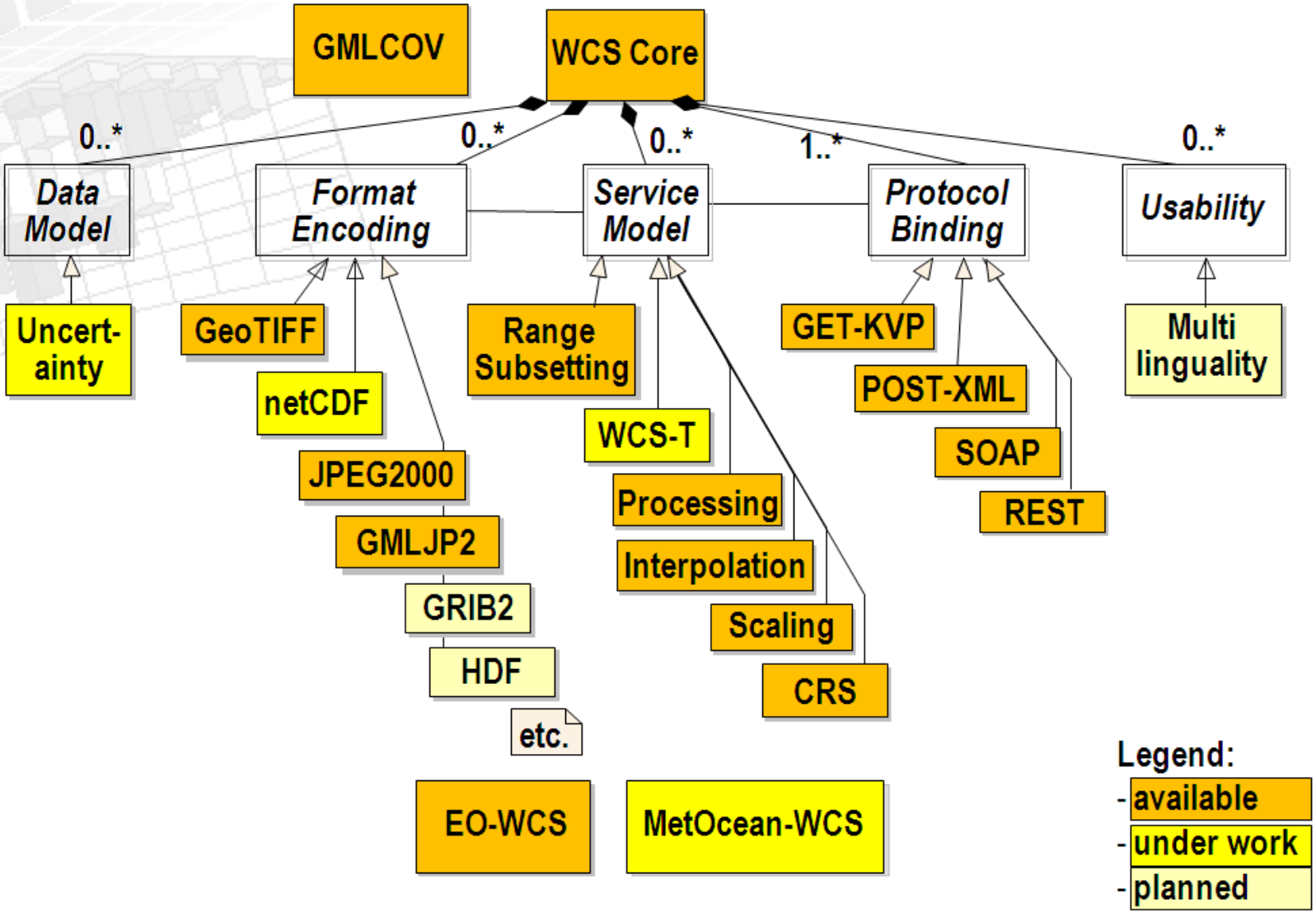




- Extensions add bespoke functionality
  - Versatile encoding, scaling, CRS, interpolation, WCPS, ...

demo on www.earthlook.org

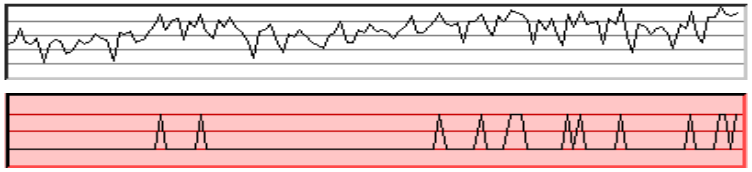
WCS: The Big Picture & Status



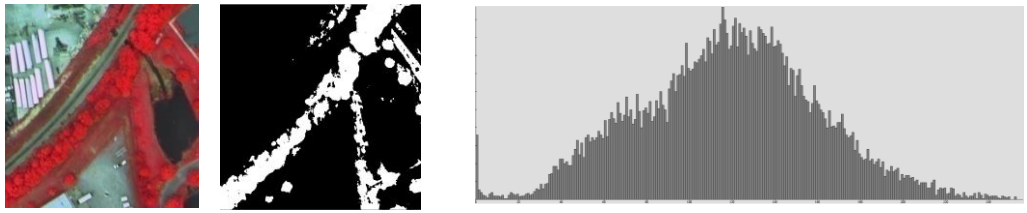
Web Coverage Processing Service

"XQuery for rasters": ad-hoc navigation, extraction, aggregation, analytics

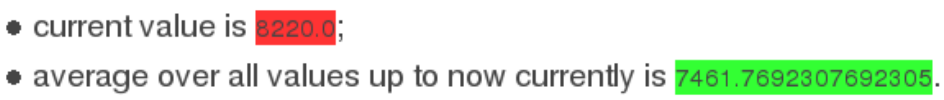
- Time series



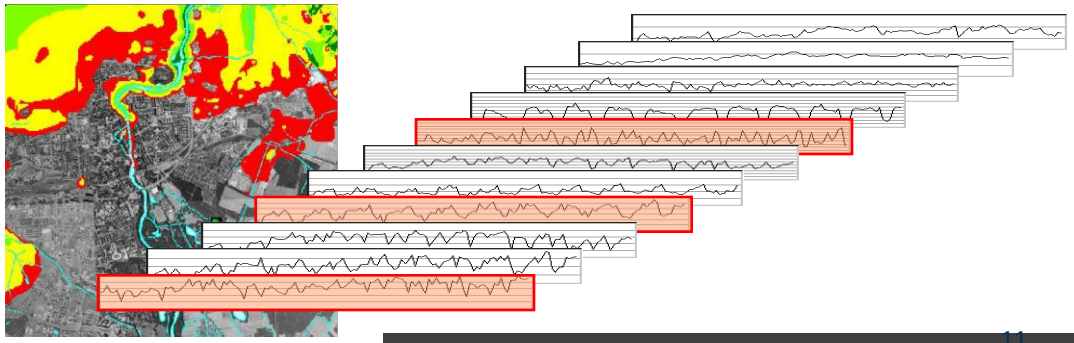
- Image processing



- Summary data



- Sensor fusion & pattern mining



WCPS By Example

- "From MODIS scenes M1, M2, and M3, the absolute of the difference between red and nir, in HDF-EOS"

```
for $c in ( M1, M2, M3 )
return
    encode(
        abs( $c.red - $c.nir ),
        "hdf"
    )
```

(hdf_A, hdf_B, hdf_C)

WCPS By Example

- "From MODIS scenes M1, M2, and M3, the absolute of the difference between red and nir, in HDF-EOS"
  - …but only those where nir exceeds 127 somewhere

```
for $c in ( M1, M2, M3 )
where
    some( $c.nir > 127 )
return
    encode(
        abs( $c.red - $c.nir ),
        "hdf"
    )
```


(hdf_A, hdf_C)

WCPS By Example

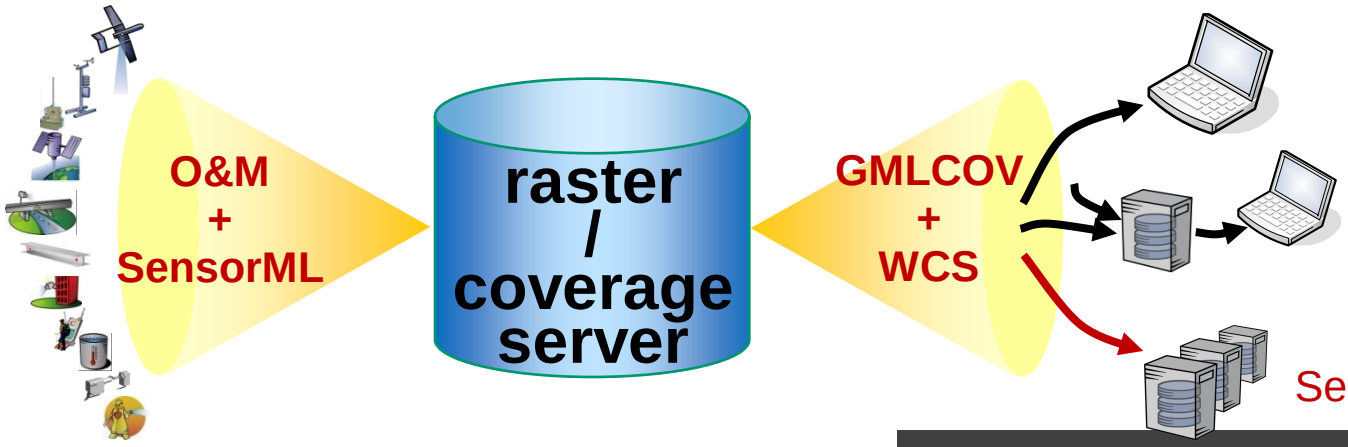
- "From MODIS scenes M1, M2, and M3, the absolute of the difference between red and nir, in HDF-EOS"
  - …but only those where nir exceeds 127 somewhere
  - …inside region R

```
for $c in ( M1, M2, M3 ),
    $r in ( R )
where
    some( $c.nir > 127 and $r )
return
    encode(
        abs( $c.red - $c.nir ),
        "hdf"
    )
```

→ (hdf_A)

OGC SWE vs WCS

- **SWE O&M and SensorML (+ friends):**
  high flexibility to accommodate all sensor types
  → **upstream** data capturing
- **GMLCOV and WCS (+WCPS):**
  one generic schema for all coverage types; generically n-D; scalable; versatile processing
  → **downstream** access & processing services



Semantic Interoperability: WPS vs WCPS

- WCPS: semantics in query → machine understandable

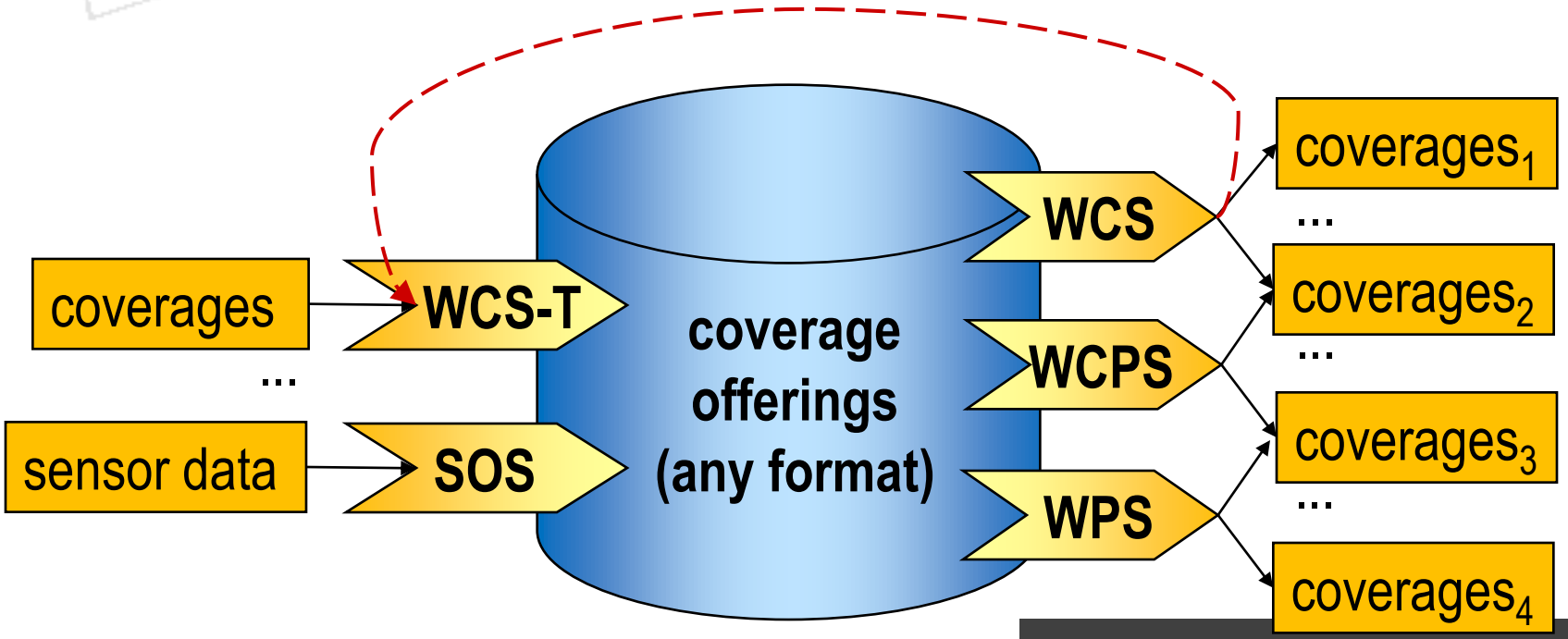
```
for $c in ( M1, M2, M3 )
return encode abs( $c.red - $c.nir ), "hdf" )
```

- WPS: semantics in human-readable text

```
<ProcessDescriptions ...>
  <ProcessDescription processVersion="2" storeSupported="true" statusSupported="false">
    <ows:Identifier>Buffer</ows:Identifier>
    <ows:Title>Create a buffer around a polygon.</ows:Title>
    <ows:Abstract>Create a buffer around a single polygon. Accepts the polygon as GML and
provides GML output for the buffered feature. </ows:Abstract>
    <ows:Metadata xlink:title="spatial" />
    <ows:Metadata xlink:title="geometry" />
    <ows:Metadata xlink:title="buffer" />
    <ows:Metadata xlink:title="GML" />
    <DataInputs>
      <Input>
        <ows:Identifier>InputPolygon</ows:Identifier>
        <ows:Title>Polygon to be buffered</ows:Title>
        <ows:Abstract>URI to a set of GML that describes the polygon.</ows:Abstract>
        <ComplexData defaultFormat="text/XML" defaultEncoding="base64" defaultSchema="http
://foo.bar/gml/3.1.0/polygon.xsd">
          <SupportedComplexData>
                                                                    1,1           Top
```

Synopsis of Coverage-Related Stds

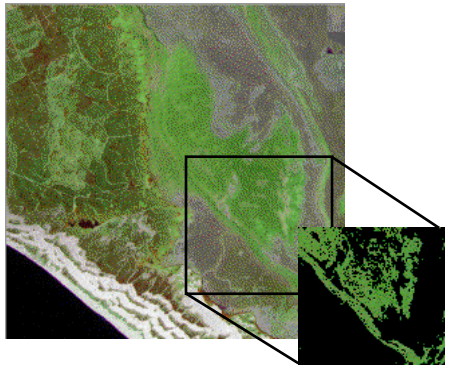
- WCS -- simple coverage access (subsetting, transforms, ...)
- WCPS -- on-demand processing & filtering by raster query language
- WPS -- on-demand processing & filtering by server code
- SOS -- sensor data acquisition



The rasdaman Raster Analytics Server

www.rasdaman.org

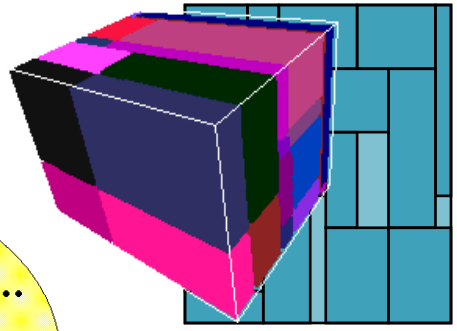
- Raster DBMS for massive n-D raster data
  - Data integration: rasters stored in standard database

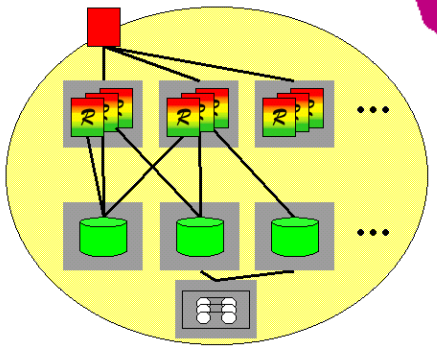


- Extending SQL with raster processing

```
select img.green[x0:x1,y0:y1] > 130
from   LandsatArchive as img
```

- Architecture: strictly tile-based
  - n-D array → set of n-D tiles
  - extensive optimization, hw/sw parallelization

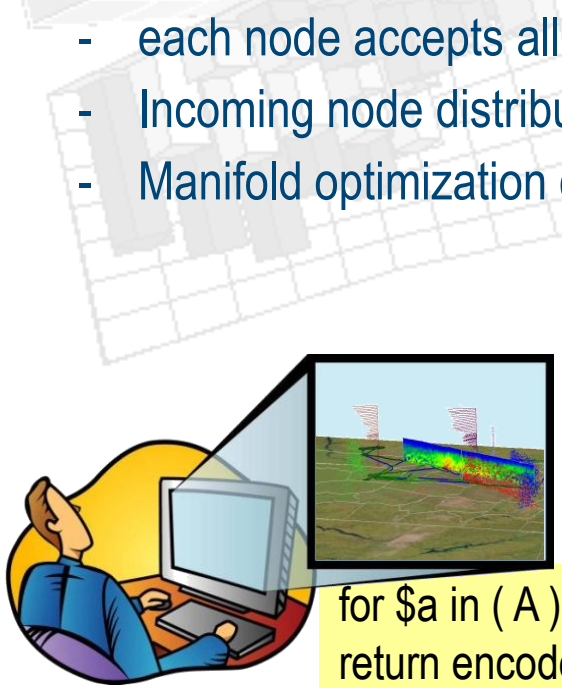


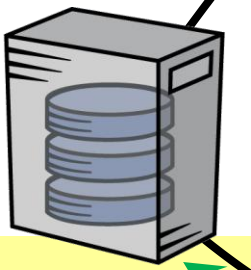


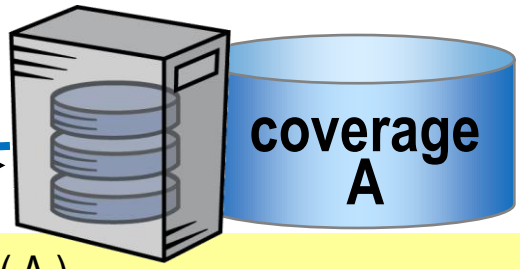
- In operational use
  - dozen-Terabyte objects
  - Analytics queries in 50 ms on laptop

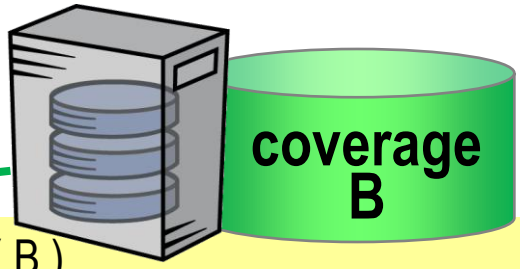
Distributed Query Processing

- WCPS peer-to-peer cloud
  - each node accepts all requests
  - Incoming node distributes query, semantics based
  - Manifold optimization criteria










```
for $a in ( A )
return encode(
    ($a.nir - $a.red) / ($a.nir + $a.red),
    "array-compressed" )
```

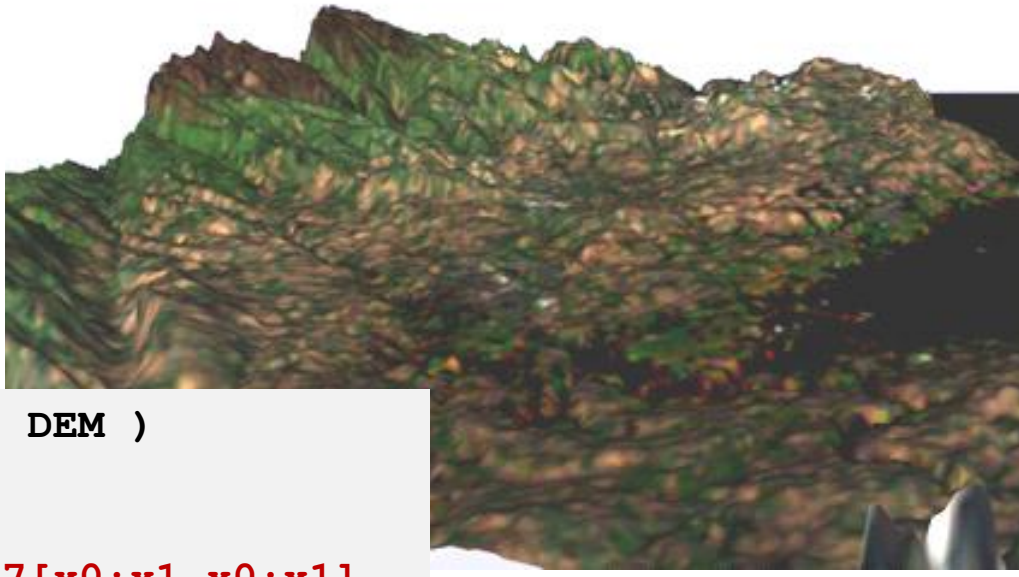
```
for $a in ( A ), $b in ( B )
return encode(
    ( ($a.nir - $a.red) / ($a.nir + $a.red)
    - ($b.nir - $b.red) / ($b.nir + $b.red)
    ), "HDF5" )
```

```
for $b in ( B )
return encode(
    ($b.nir - $b.red) / ($b.nir + $b.red),
    "array-compressed" )
```

[Owonibi 2012]

3D Clients: Experiments

- Problem: coupling DB / visualization
- Approach:
  - deliver RGBA image to X3D client, transparency as height
  - Feed directly into client GPU



```
for s in ( SatImage ), d in ( DEM )
return
    encode(
        {   red:   (char) s.b7[x0:x1,x0:x1],
            green: (char) s.b5[x0:x1,x0:x1],
            blue:  (char) s.b0[x0:x1,x0:x1],
            alpha: (char) scale( d, 20 )
        },
        "PNG"
    )
```

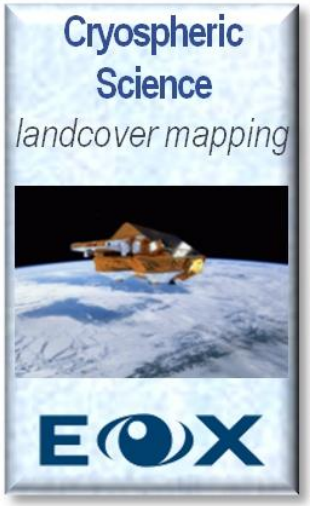
[JacobsU, Fraunhofer 2012]

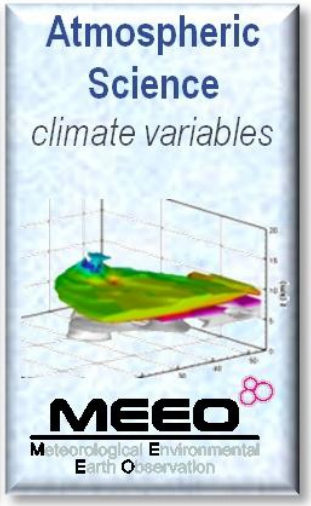
EarthServer: *Big Earth Data Analytics*

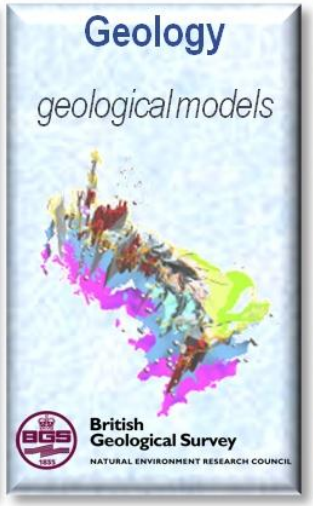
- Scalable On-Demand Processing for the Earth Sciences
  - EU funded, 3 years, 5.85 mEUR

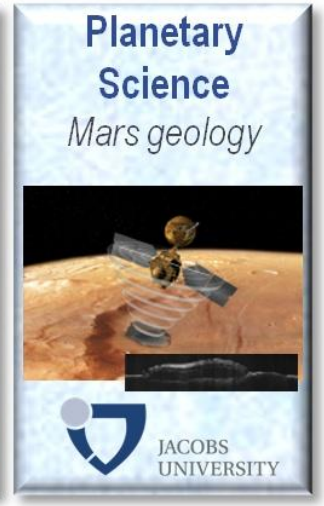
www.earthserver.eu

- 6 * 100+ TB databases for all Earth sciences + planetary science
- Platform: rasdaman Array Analytics Server
  - Distributed query processing, integrated data/metadata search, 3D clients
  - Strictly open standards: WMS + WCS + WCPS









WCS Reference Implementations

- Current version:

WCS 2.0	rasdaman	rasdaman GmbH

- Deprecated versions:

WCS 1.1	OpenGeo	GeoServer
WCS 1.0	deegree	latlon

- See http://cite.opengeospatial.org/reference

Status & Future

- GMLCOV 1.0.1 + WCS Core 2.0.1
- CITE testing established; candidate reference implementation: rasdaman
- Spatio-temporal CRS definitions established
  - CRS Name Type Specification, OGC 11-135
- Future:
  - Mixed regular / irregular axes
  - Streaming coverages
  - WCS-T
  - Coverage hierarchies, both homogeneous („mosaic“) & heterogeneous

Summary

- OGC has a stable suite of coverage standards
  - Single, coherent model → cross-domain integration

[seriouseats.com]

- The Data Model: GML Coverages
  - All GML coverage types: *nD rasters, curvilinear grids, point clouds, meshes, surfaces, ...*
  - Service-independent → coverages interchangeable between OGC services
  - Various representation schemes → efficient encoding & interchange

- The Service Suite: WCS
  - Modular : *Core, encodings, CRS, WCS-T, WCPS, EO-WCS, ...*
  - from simple access to advanced processing
  - Concisely defined interoperability
  - efficient implementation proven (rasdaman: n-D, MapServer: 2-D)