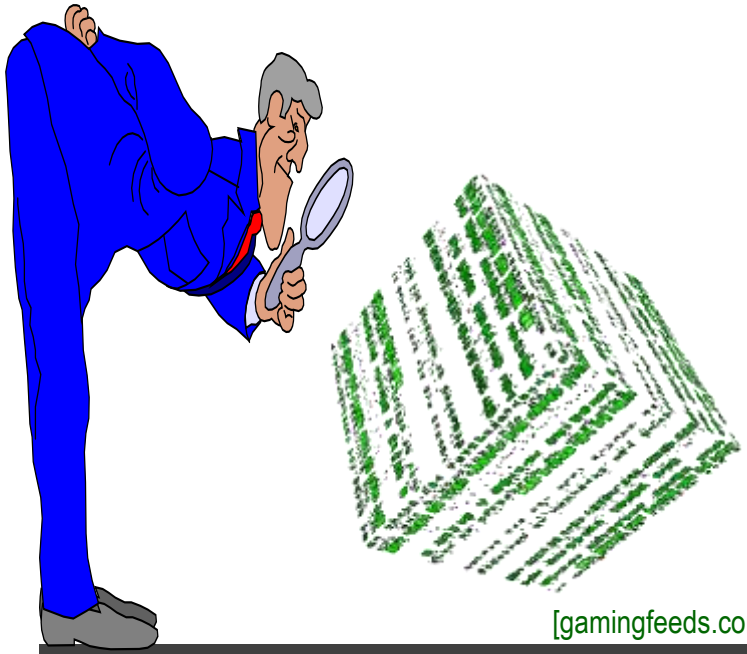
Spatio-Temporal Coverage World

Through The Looking Glass

Alex Dumitru

Jacobs University | rasdaman GmbH

dumitru@rasdaman.com

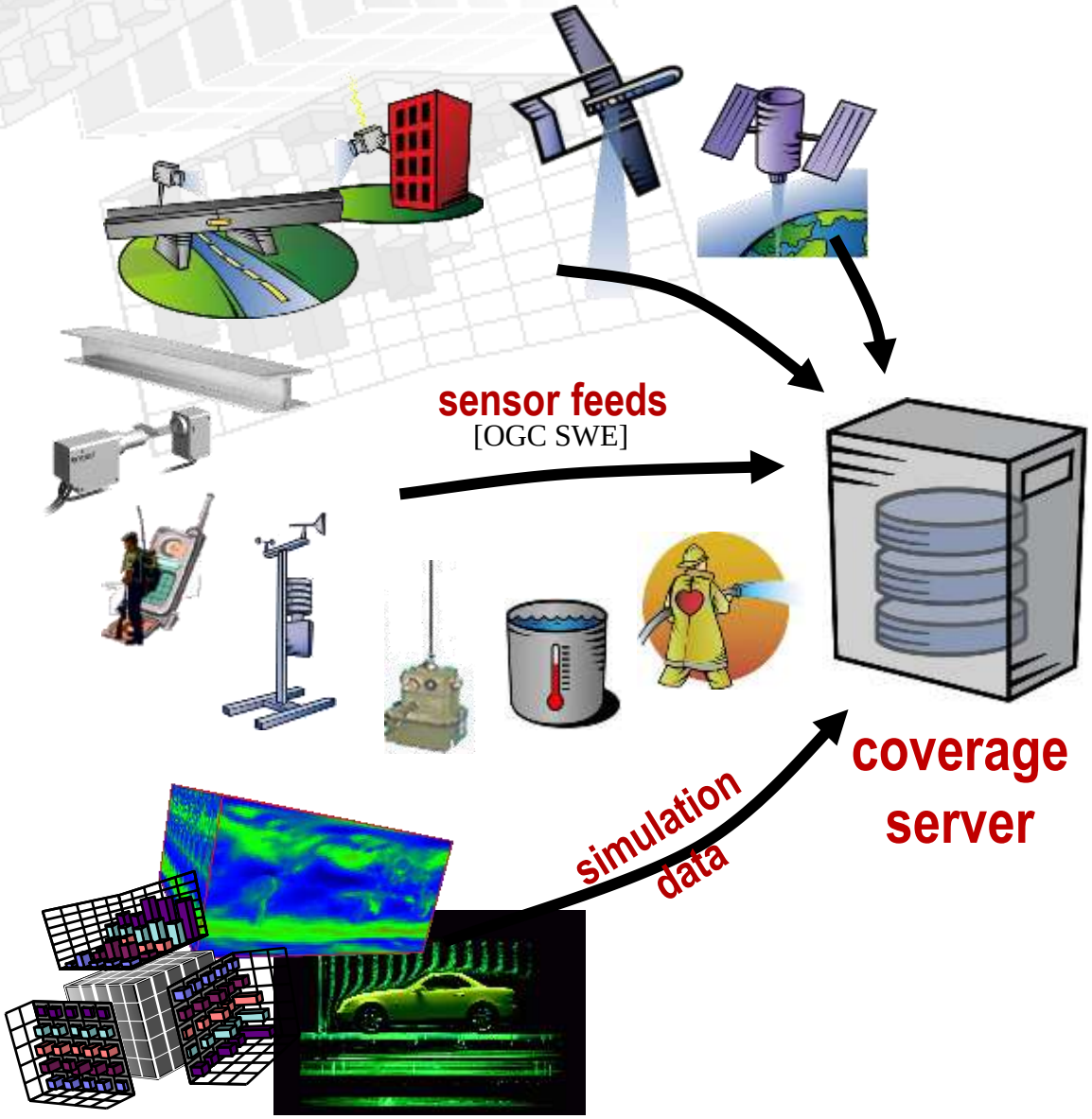


[gamingfeeds.com]

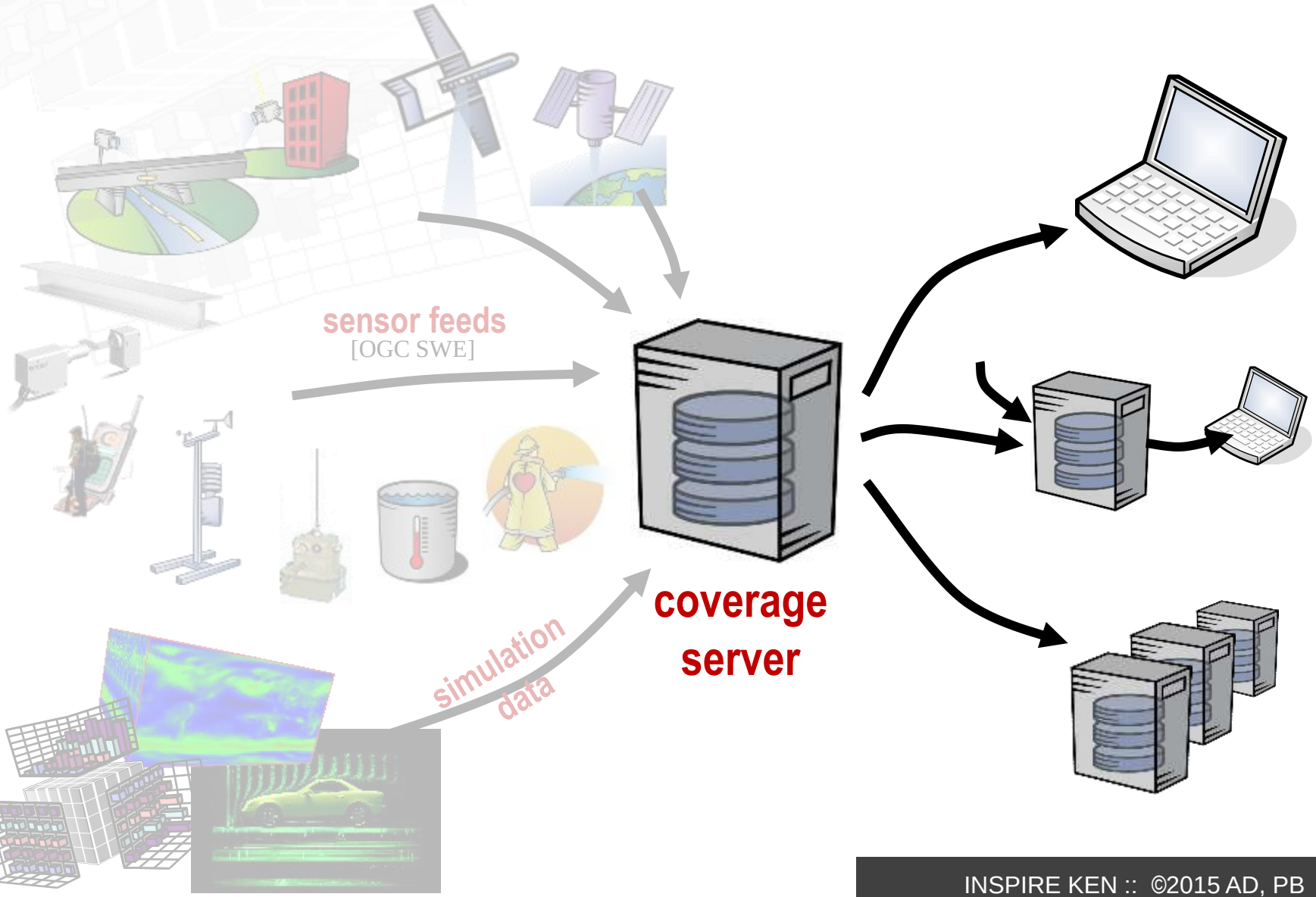
Overview

- Introduction
- Coverage data: the OGC Coverage Model
- Coverage services: the OGC WCS Suite
- Hands-on examples
- Standardization status
- Summary

Collecting Coverages

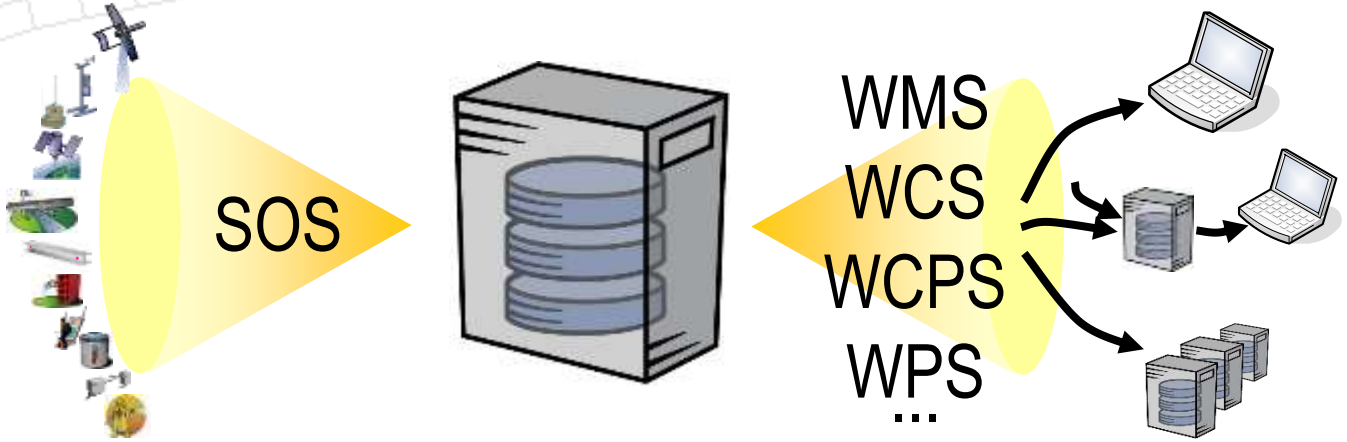


Serving Coverages



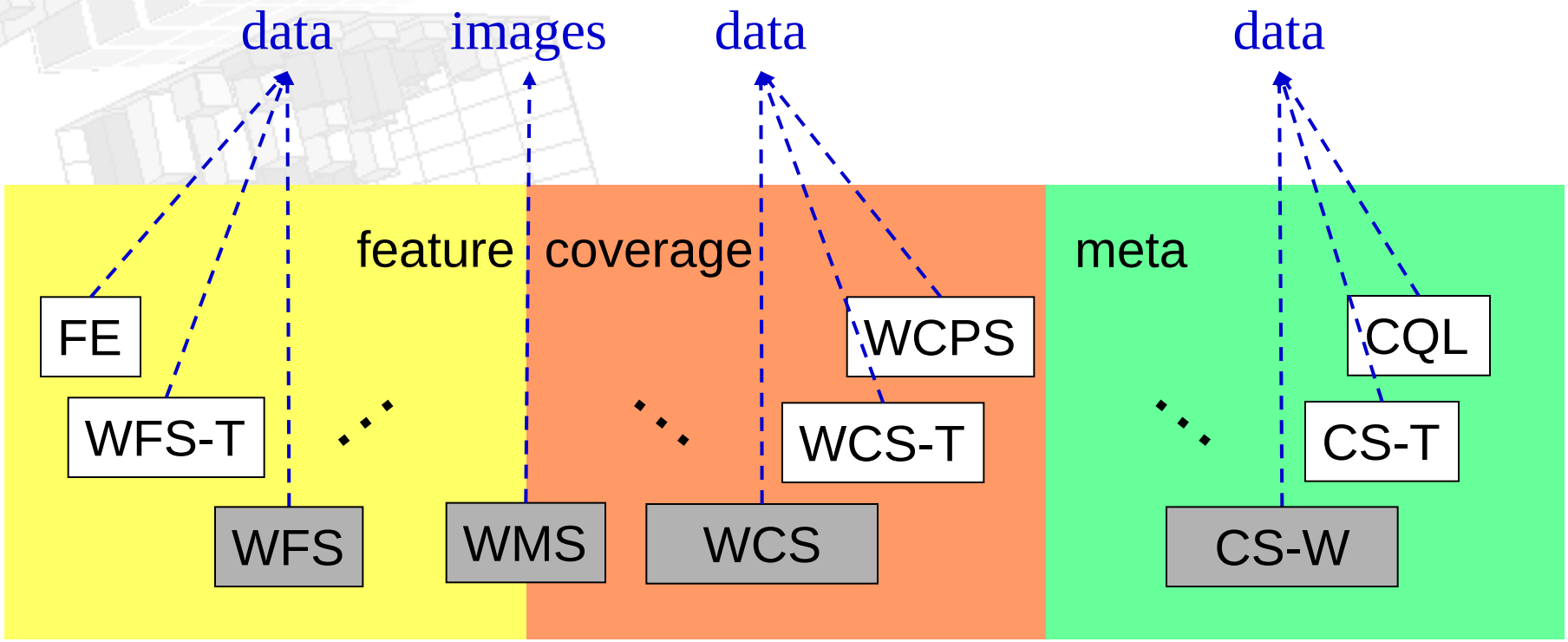
Service Orchestration

W*S: downstream
download, processing, visualization



SWE, SOS: upstream
sensor data capturing

(Part of) The OGC Standards Quilt



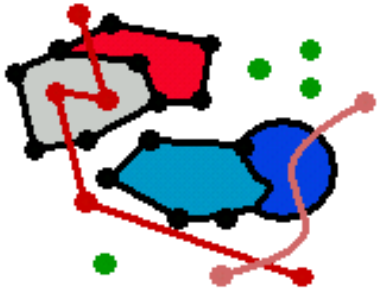
- WMS "portrays spatial data → pictures"
- WCS: "provides data + descriptions; data with original semantics, may be interpreted, extrapolated, etc.“

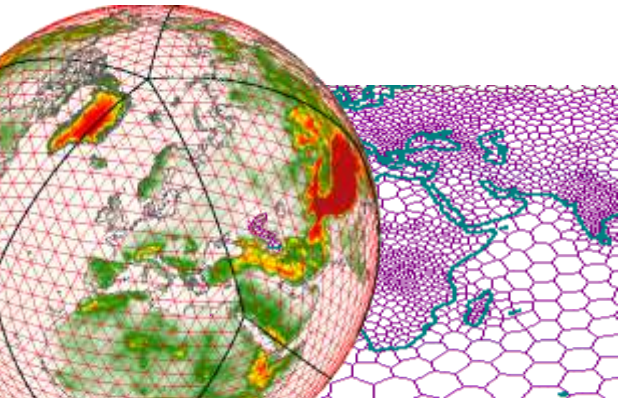
[09-110r3]

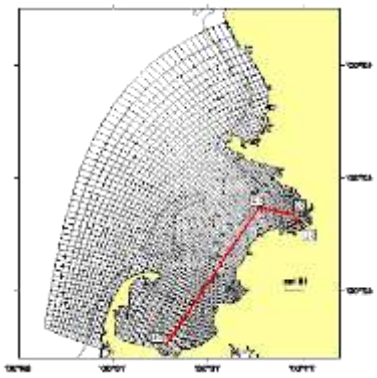
Coverages

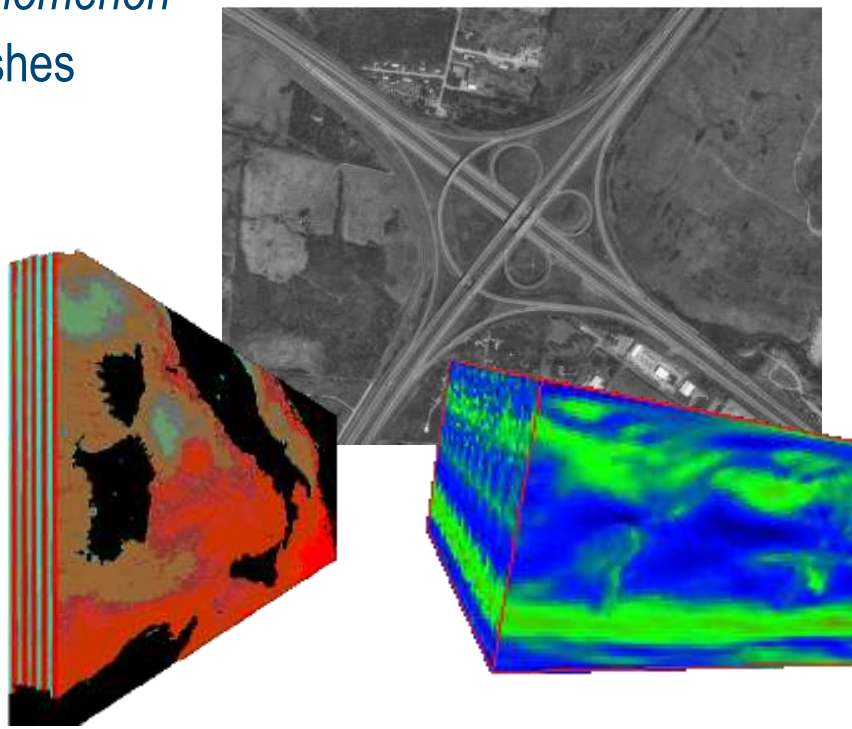
Features & Coverages

- The basis of all: geographic feature
  - = *abstraction of a real world phenomenon* [OGC, ISO]
- Special kind of feature: coverage
  - = *space-time varying multi-dimensional phenomenon*
  - = regular & irregular grids, point clouds, meshes
- Usually, Big Geo Data are coverages









OGC Coverage Types

- Spatio-temporal !

as per GML 3.2.1

«FeatureType»
Abstract Coverage

Grid Coverage

Rectified GridCoverage

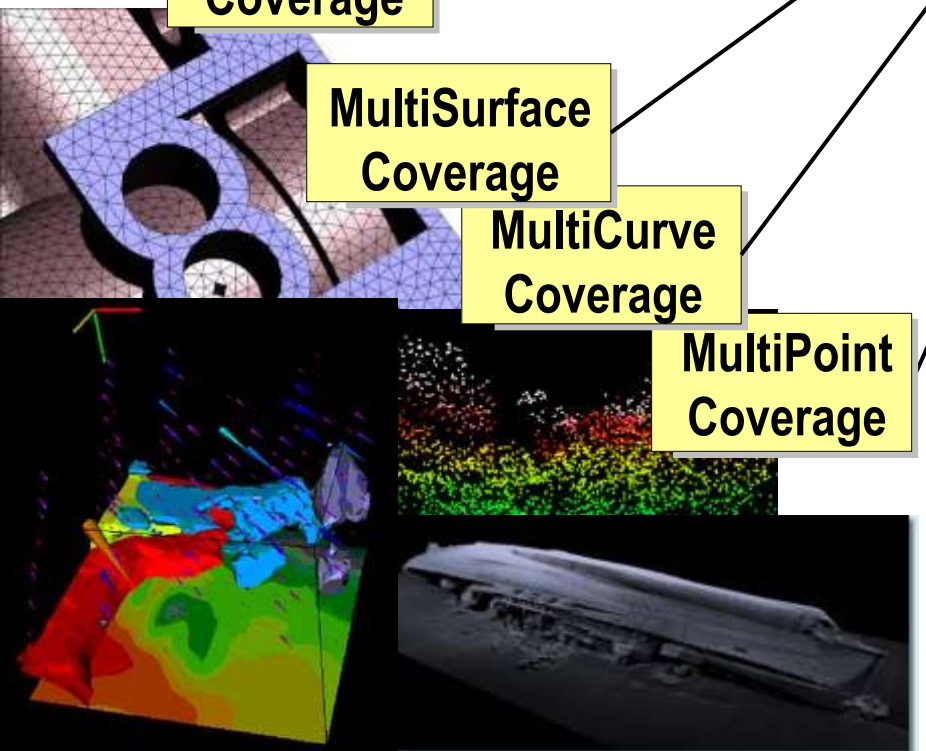
Referenceable GridCoverage

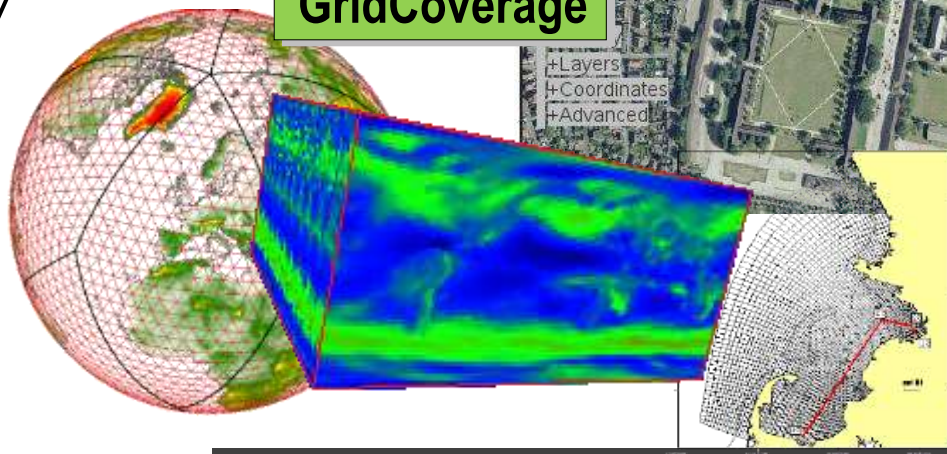
MultiSolid Coverage

MultiSurface Coverage

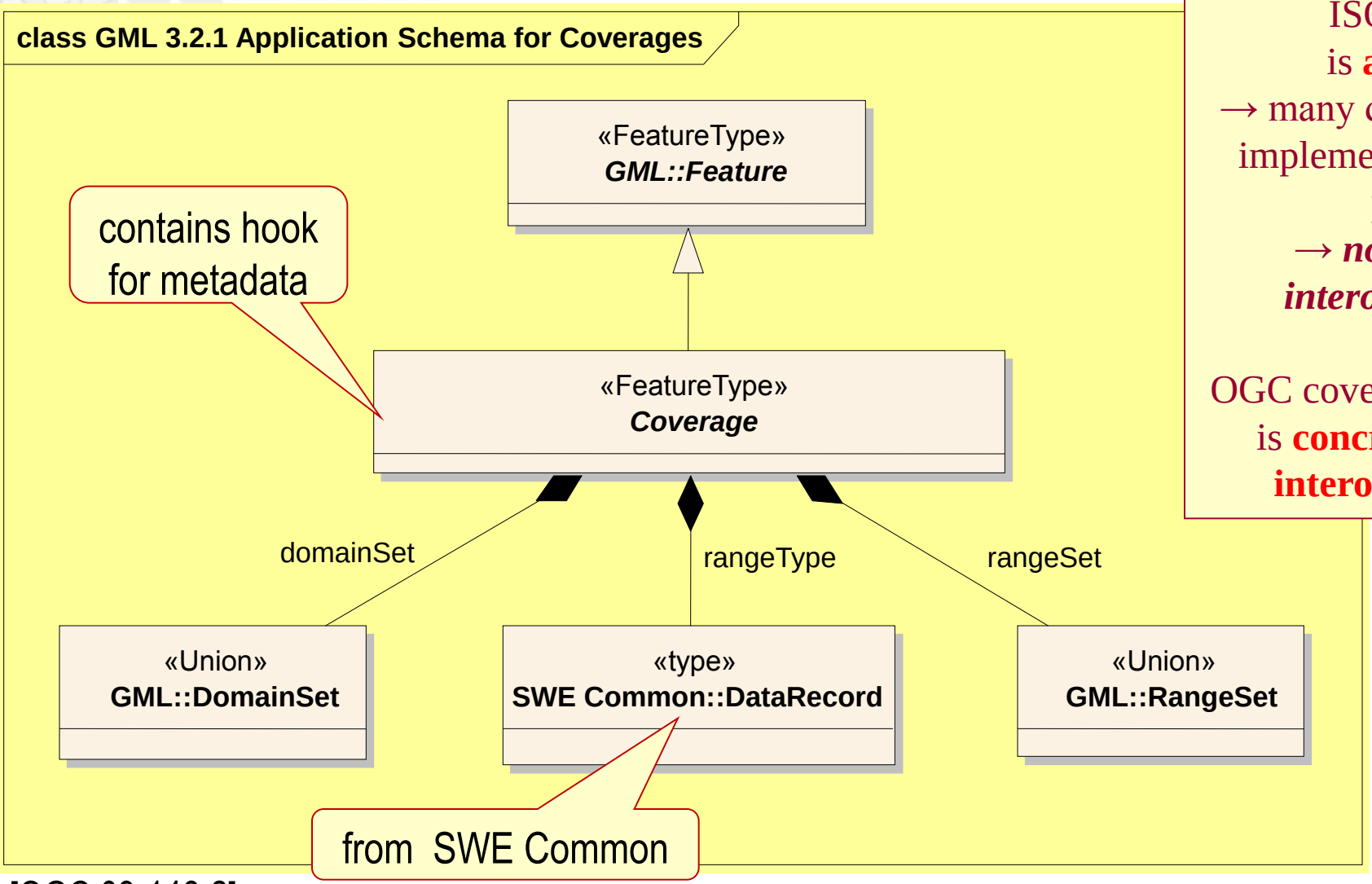
MultiCurve Coverage

MultiPoint Coverage





Coverage Definition



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

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

[OGC 09-146r2]

Sample Grid Coverage (GML)

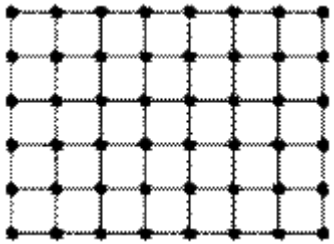
```
<?xml version="1.0" encoding="UTF-8" ?>
- <gmlcov:GridCoverage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns="http://www.opengis.net/gml/3.2" xmlns:swe="http://www.opengis.net/swe/2.0" xmlns:gmlcov="http://www.opengis.ne
  xsi:schemaLocation="http://www.opengis.net/swe/2.0 http://schemas.opengis.net/sweCommon/2.0/swe.xsd
  http://www.opengis.net/gmlcov/1.0 ../../../../gmlcov/1.0/gmlcovAll.xsd" gml:id="C0001">
 - <gml:boundedBy>
  - <gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/4326" axisLabels="Lat Long" uomLabels="deg deg" srsDimensio
      <gml:lowerCorner>1 1</gml:lowerCorner>
      <gml:upperCorner>5 3</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
 - <gml:domainSet>
  - <gml:Grid gml:id="gr0001_C0001" dimension="2">
    - <gml:limits>
      - <gml:GridEnvelope>
          <!--  This is a 5-by-3 matrix  -->
          <gml:low>1 1</gml:low>
          <gml:high>5 3</gml:high>
        </gml:GridEnvelope>
      </gml:limits>
      <gml:axisLabels>Lat Long</gml:axisLabels>
    </gml:Grid>
  </gml:domainSet>
 - <gml:rangeSet>
  - <gml:DataBlock>
      <gml:rangeParameters />
      <gml:tupleList>1 2 3 4 5 6 7 8 9 10 11 12 13 1
    </gml:DataBlock>
  </gml:rangeSet>
```

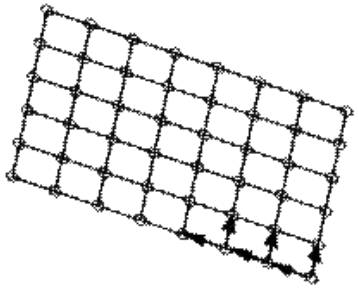
```
  - <gmlcov:rangeType>
   - <swe:DataRecord>
    - <swe:field name="singleBand">
     - <swe:Quantity definition="http://www.opengis.net/def/property/OGC
        <gml:description>Panchromatic Channel</gml:description>
        <gml:name>single band</gml:name>
        <swe:uom code="W/cm2" />
       - <swe:constraint>
        - <swe:AllowedValues>
            <swe:interval>0 255</swe:interval>
            <swe:significantFigures>3</swe:significantFigures>
          </swe:AllowedValues>
        </swe:constraint>
      </swe:Quantity>
    </swe:field>
   </swe:DataRecord>
  </gmlcov:rangeType>
</gmlcov:GridCoverage>
```

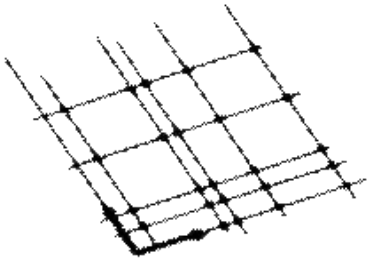
samples for each coverage type provided with specification bundle

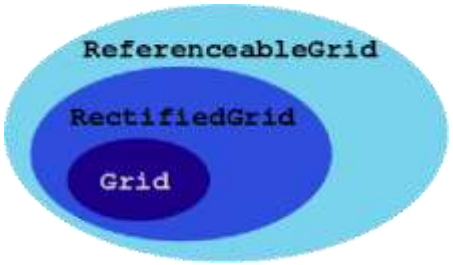
Gridded Coverage Types

- Not georeferenced, „just pixels“
  - GMLCOV::GridCoverage
- Georeferenced, possibly oblique
  - GMLCOV::RectifiedGridCoverage
- 1+ irregular axes
  - All axes irregular: GML 3.3 ReferenceableGridByVectors *
  - GMLCOV::ReferenceableGridCoverage
- 1+ axes warped
  - All axes warped: GML 3.3 *ReferenceableGridByArray* *
  - GMLCOV::ReferenceableGridCoverage









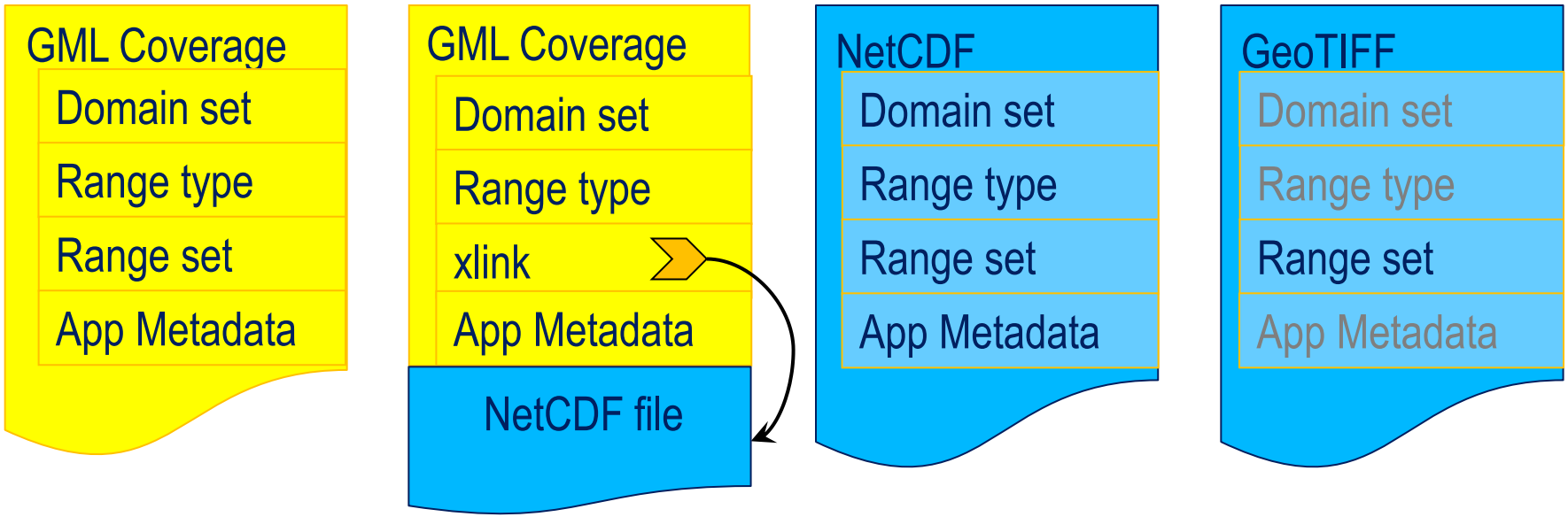
Mix, eg, with sat image timeseries

[Campalani 2013]

*) to be unified in GMLCOV 1.1

Coverage Encoding

- Pure GML: complete coverage, in GML
- Special Format: other suitable file format (ex: MIME type “image/tiff”)
- Multipart-Mixed: multipart MIME, type “multipart/mixed”



```
Content-Type: Multipart/Related; boundary=wcs;
      start="GML-Part"
      type="text/xml"

--wcs
Content-type: text/xml
Content-ID: GML-Part

<?xml version="1.0" …>
<gmlcov:RectifiedGridCoverage …>
<gml:domainSet>…</gml:domainSet>
 <gml:rangeSet>
  <gml:File>
   <gml:rangeParameters xlink:href="grey.tif"
    xlink:role="http://www.opengis.net/spec/WCS_coverage-encoding_geotiff/1.0/"
    xlink:arcrole="fileReference"/>
   <gml:fileReference>grey.tif</gml:fileReference>
   <gml:fileStructure/>
   <gml:mimeType>image/tiff</gml:mimeType>
  </gml:File>
 </gml:rangeSet>
 <gmlcov:rangeType>…</gmlcov:rangeType>
</gmlcov:RectifiedGridCoverage>
--wcs
Content-Type: image/tiff
Content-Description: coverage data
Content-Transfer-Encoding: binary
Content-ID: grey.tif
Content-Disposition: INLINE

...binary TIFF data...
--wcs--
```

How to request a particular encoding from a server? See later!

Sample Mixed Encoding: TIFF

- Multipart/related MIME
- Part 1: GML
  - Without pixels
- Part 2: TIFF
  - Can be extracted with any ol' MIME tool around
- Consistency in metadata required
  - Otherwise bug

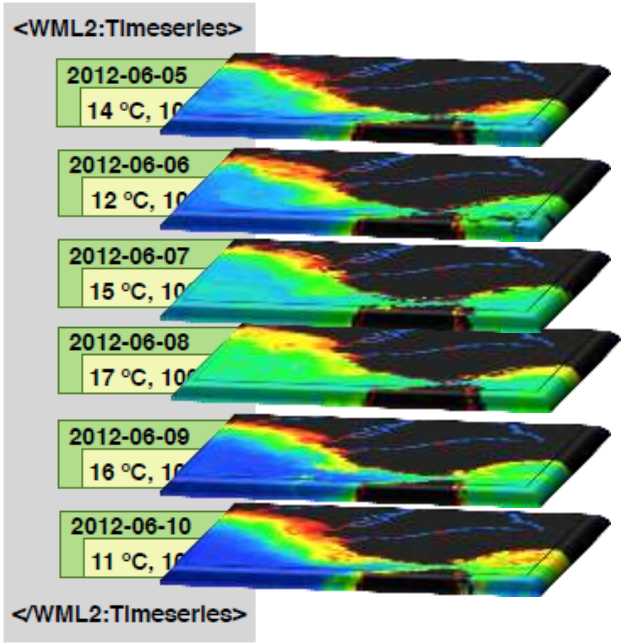
Adding Metadata To Coverages

- 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
- Inverse possible, too: metadata record contains (or links) coverage

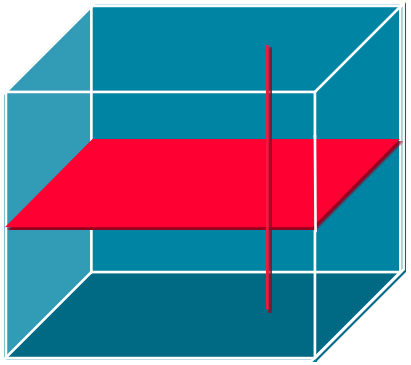
```
−<wcseo:RectifiedDataset
 gml:id="MER_FRS_1PNPDE20060822_092058_000001972050_00308_23408_0077_uint16_r
 xsi:schemaLocation="http://www.opengis.net/wcseo/1.0 http://schemas.opengis.net/wcs
 /1.0/wcsEOAll.xsd">
 +<gml:boundedBy></gml:boundedBy>
 +<gml:domainSet></gml:domainSet>
 +<gml:rangeSet></gml:rangeSet>
 +<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>
```

Inset: WaterML 2.0 Time Handling

- WaterML 2.0: timeseries = time slices
  - layout hardwired

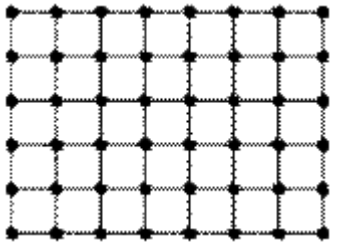


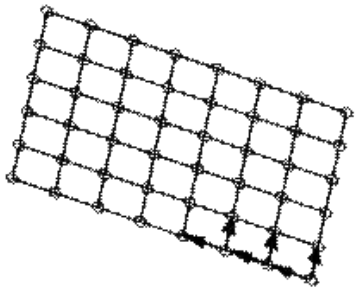
- OGC Coverages: time just another axis
  - Implementation can choose efficient layout

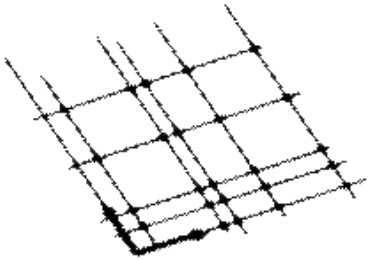


CIS 1.1 – New coverage model

- More flexible model, no set-in-stone coverage types
- On-the-fly typing of coverages based on axis definitions
- Several axis types:
  - Regular axis
  - Irregular axis
  - Distorted axis
  - Algorithm axis
- Coverage partitioning



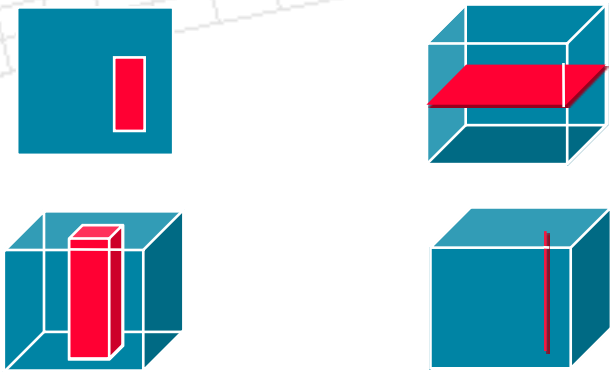




WCS Core

Web Coverage Service (WCS)

- WCS Core: Simple & efficient access to n-D coverages & subsets
  - Format conversion on the fly
  - subset = trim | slice



- WCS Extensions: for optional functionality facets
- WCS Application Profiles: domain-oriented bundling
  - EO, MetOcean, Sensors, ...

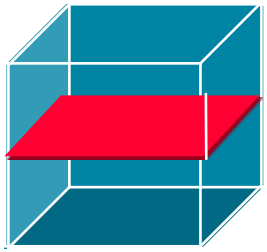
WCS Core *GetCoverage*

- Download a coverage (or a subset thereof), values guaranteed unchanged
- Ex: „*download coverage c001*“

  http://www.acme.com/wcs ? SERVICE=WCS & VERSION=2.0
  & REQUEST=GetCoverage & COVERAGEID=c001

- Ex: „*coverage c001, lat/long cutout, time slice t=2009-11-06T23:20:52*”

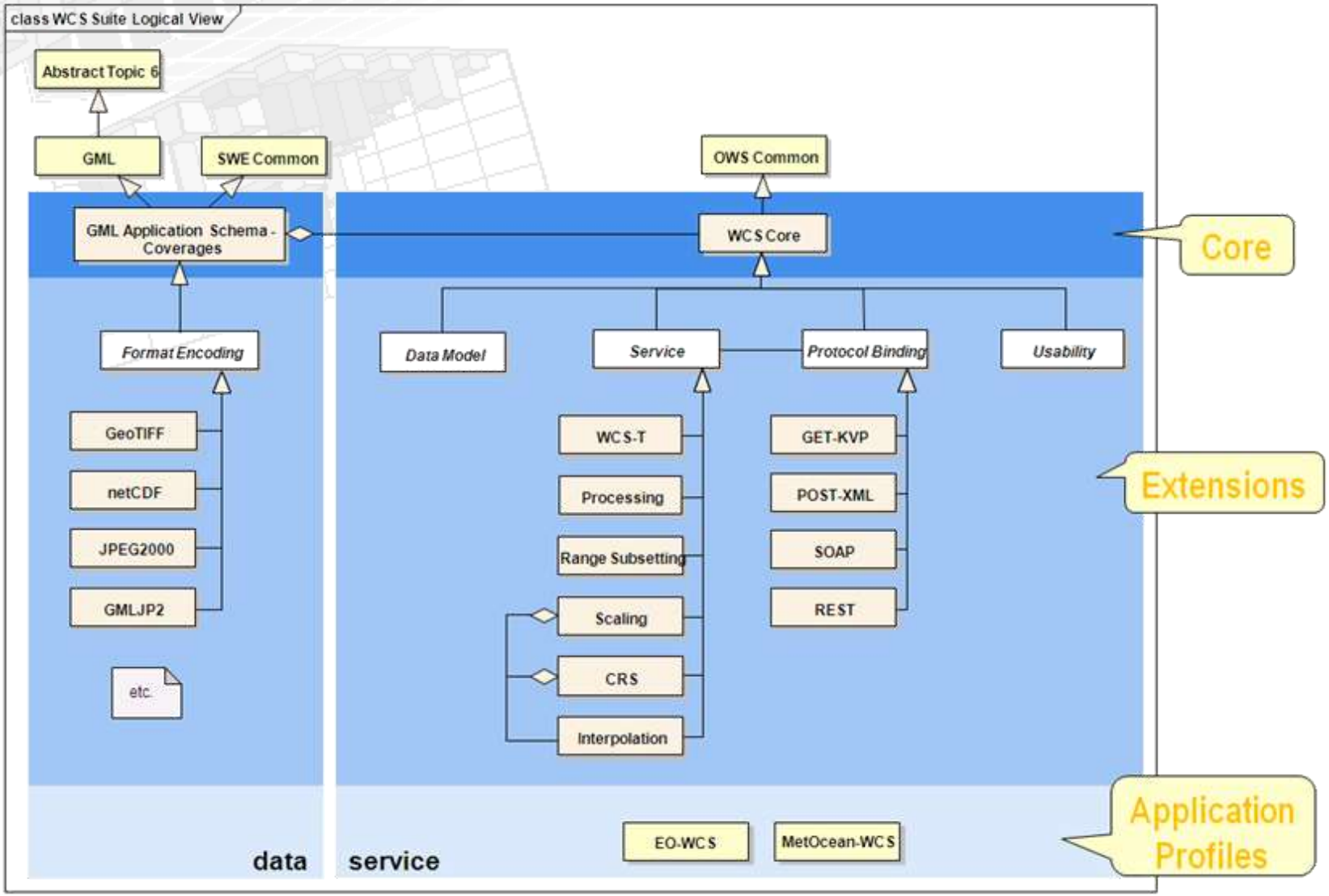
  http://www.acme.com/wcs ? SERVICE=WCS & VERSION=2.0
  & REQUEST=GetCoverage & COVERAGEID=c001
  & SUBSET=Long(100,120) & SUBSET=Lat(50,60)
  & SUBSET=time("2009-11-06T23:20:52")



- Ex: “*coverage c001, in GeoTIFF*”

  http://www.acme.com/wcs ? SERVICE=WCS & VERSION=2.0
  & REQUEST=GetCoverage & COVERAGEID=c001 & FORMAT=“image/tiff“

WCS Suite Big Picture



WCS Extension – Range Subsetting [OGC 12-039]

- Extract range components
  - „bands“, „variables“
  - Extension to *GetCoverage* request
- Ex: http://www.acme.com/wcs ? SERVICE=WCS & VERSION=2.0
  & REQUEST=GetCoverage & COVERAGEID=c001
  & RANGESUBSET=red

  - or: …& RANGESUBSET=nir,red,green &…
  - or: …& RANGESUBSET=green,red,blue &…
  - or: …& RANGESUBSET=nir:green &…
  - or: …& RANGESUBSET=band01,band03:band05,band19:band21 &…

Requesting Scaled Coverages

- By way of example:
- …& ScaleByFactor=2.0 &…
- …& ScaleAxesByFactor=lat(2.0),long(2.0), time(1.0) &…
- …& ScaleToExtent=lat(10,20),long(20,30) &…

2.0 = scale DOWN by factor of 2

WCS Extension – Transaction [OGC 13-057]

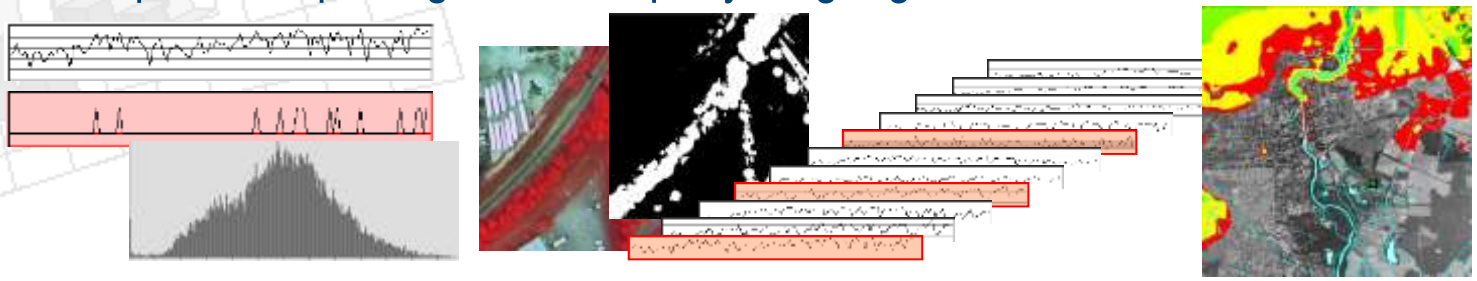
- = WCS-T: Modify coverage offerings on a server via Web
  - New requests:
    InsertCoverage + *DeleteCoverage* + *UpdateCoverage* (partial replacement)
- Core design goal: *GetCoverage* → *InsertCoverage*
- Ex:

```
http://www.acme.com/wcs
    ? SERVICE=WCS & VERSION = 2.0
    & REQUEST=InsertCoverage
    & COVERAGEREF=http://bcme.com/archive/hurricane.nc
    & USEID=new
```

Web coverage service – Hands-on examples

WCS Extension – Processing [OGC 13-057]

- WCS wrapper for OGC Web Coverage Processing Service (WCPS)
  - high-level spatio-temporal geo raster query language



- "From MODIS scenes M1, M2, M3: difference between red & nir, as TIFF"
  - …but only those where nir exceeds 127 somewhere

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

→ $(\text{tiff}_A, \text{tiff}_C)$

...and now: Integration!

- WCPS 2.0 (draft): coverage expressions + XQuery
- Ex1: „*difference of red, nir bands for all coverages on Austria*“

```
for $c in doc("http://acme.com/wcs")//coverage
where
    some( $c.nir > 127 ) and metadata/@region = "Austria"
return
    encode( $c.red - $c.nir, "image/tiff" )
```

- Ex2: „*name & location of coverages showing some phenomenon*“

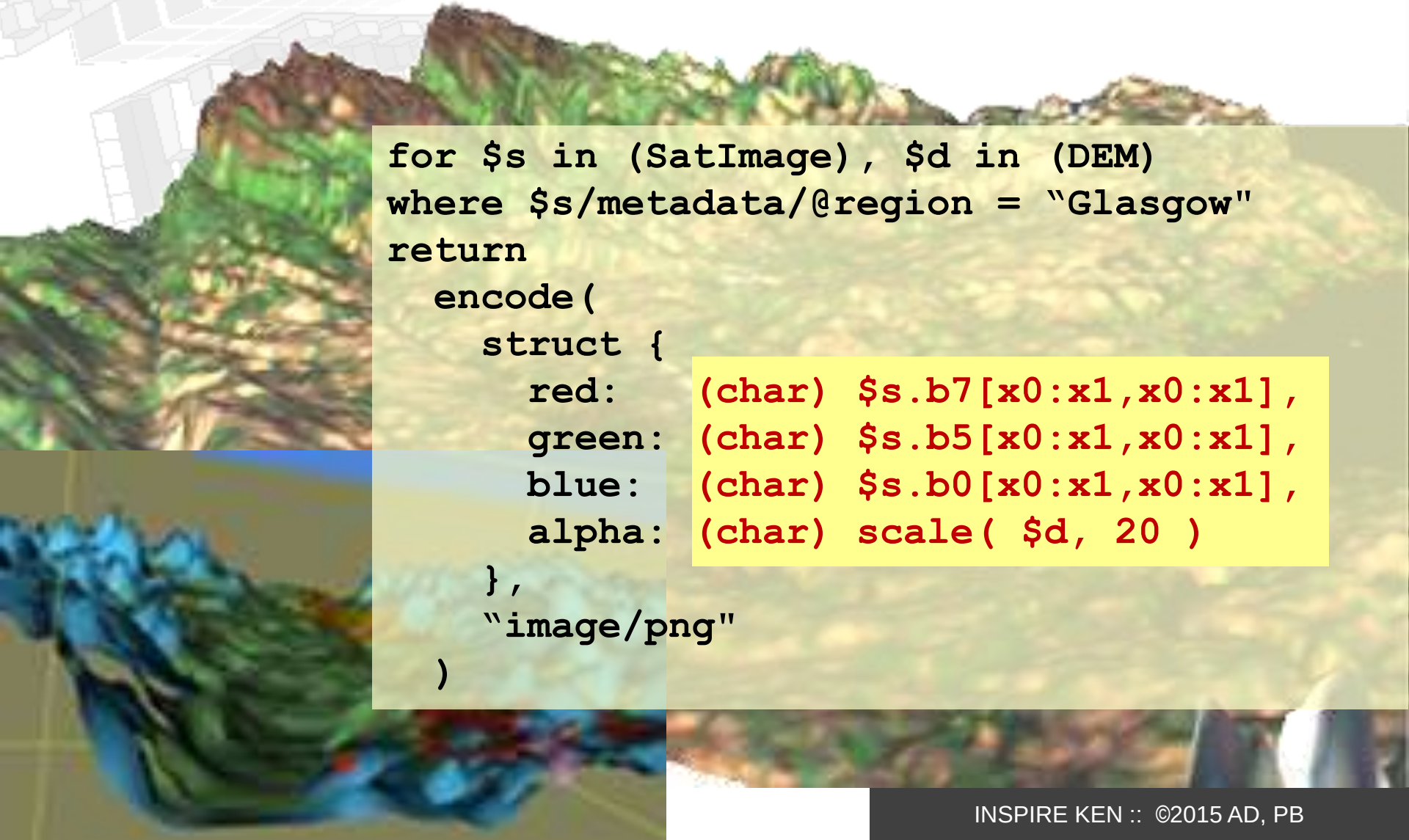
```
for $c in doc("WCPS")//coverage/[ some( $c.nir - $c.red > 0 )]
return
    <id> { $c/@id } </id>
    <area> { $c/boundedBy } </area>
```

- Implementation: federation of BaseX + rasdaman

[Jacobs University, Athena Research]

Visualization-as-a-Query

[JacobsU, Fraunhofer; data courtesy BGS, ESA]



```
for $s in (SatImage), $d in (DEM)
where $s/metadata/@region = "Glasgow"
return
  encode(
    struct {
      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 )
    },
    "image/png"
  )
```

Web coverage processing service – Hands-on examples

Why "Semantic" Interoperability?

- Formal semantics for language allows machine-machine communication, no human intervention required
  - Clients (other services?) can compose requests
- Ex:
  - Client: "Let's see, which server can handle reprojection / exponentials / ... ?"
  - In a cloud: "hm, this subexpression I better pass on to node X"
  - "Evaluating this request will take an estimated 3.5min, over 500 objects match."
  - "Sorry, this request's complexity exceeds your CPU quota "
- WPS offers syntactic interoperability, WCPS semantic interoperability
  - Under work: WCPS as an „Application Profile“ of WPS → combine generality of WPS (like asynchronous processing) with specificity of WCPS query language

Semantic Interoperability: WPS vs WCPS

- WCPS: semantics in query → machine understandable

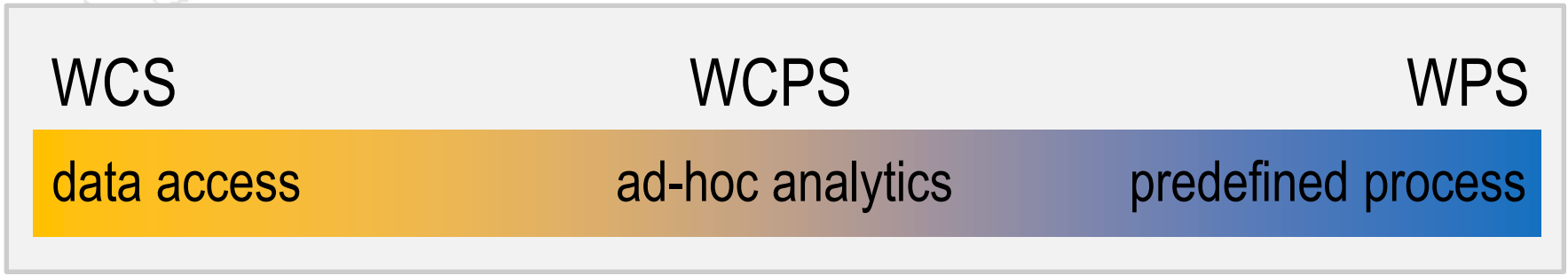
```
for $c in ( M1, M2, M3 )
return encode abs( $c.red - $c.nir ), "hdf" )
```

- 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
```

Inset: OGC Big Data Coverage Service Portfolio

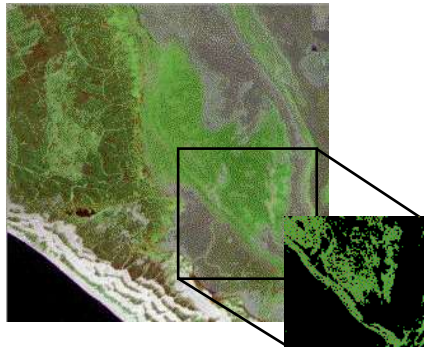
- OGC standards cover full range
  from data-intensive to processing-intensive „Big Data“ coverage services

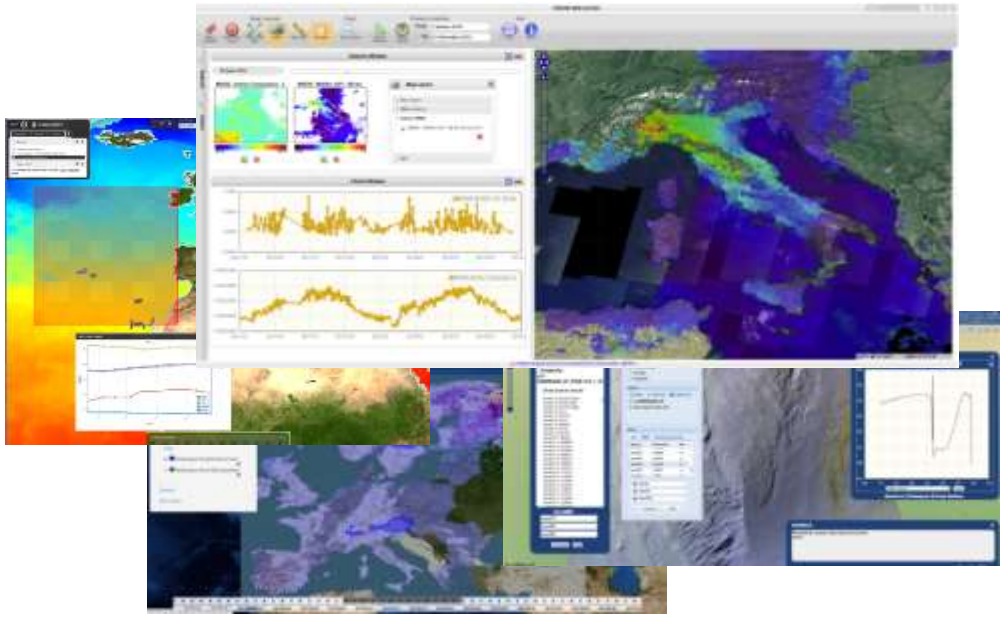


Implementation

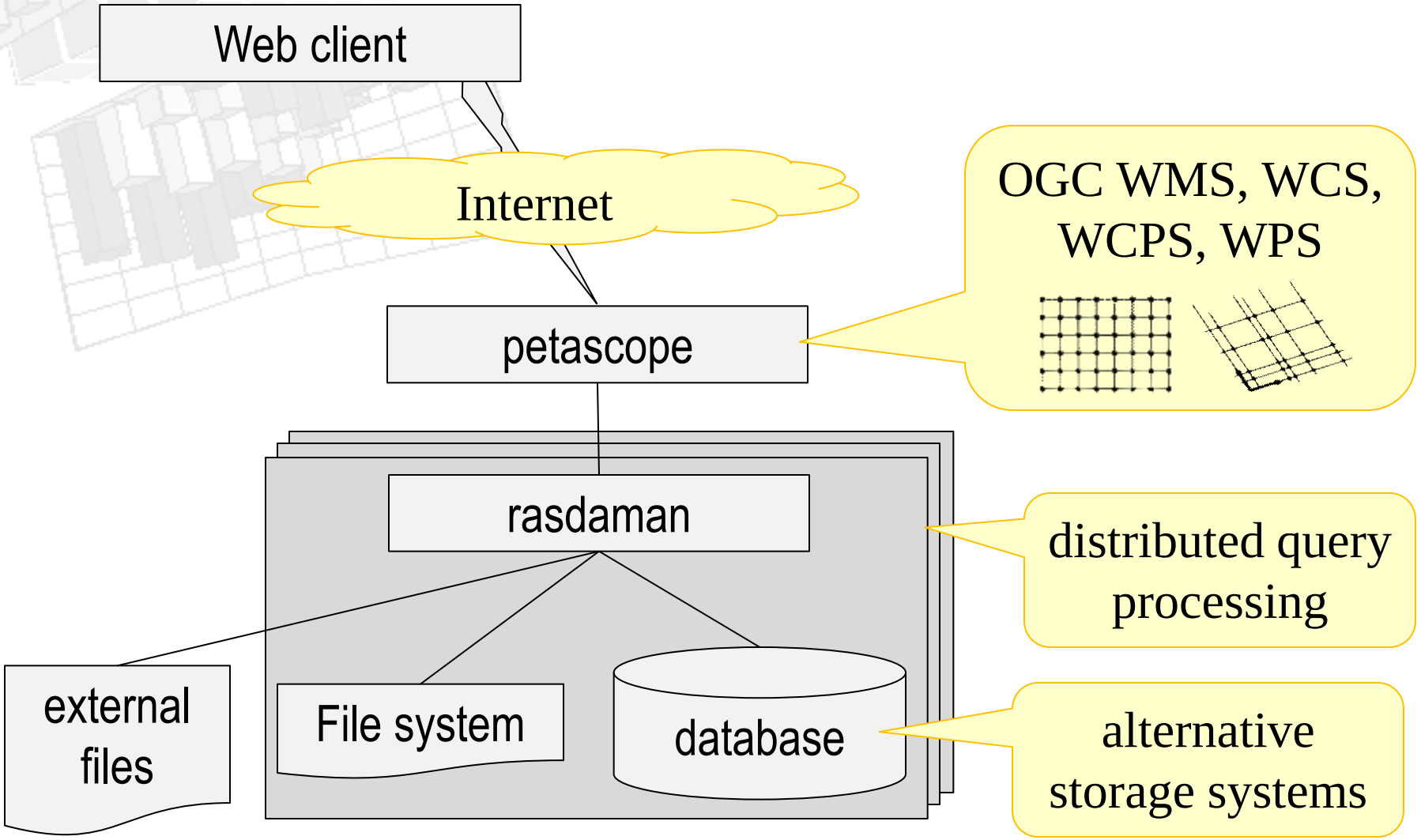
The rasdaman Array DBMS

- „raster data manager“: SQL + n-D arrays
  - Scalable parallel “tile streaming” architecture
- In operational use
  - Supports R, QGIS, OpenLayers, MapServer, GeoServer, EOxServer, THREDDS, ncWMS, Pyxis, ERDAS, ArcGIS, ...
  - OGC WCS Core Reference Implementation



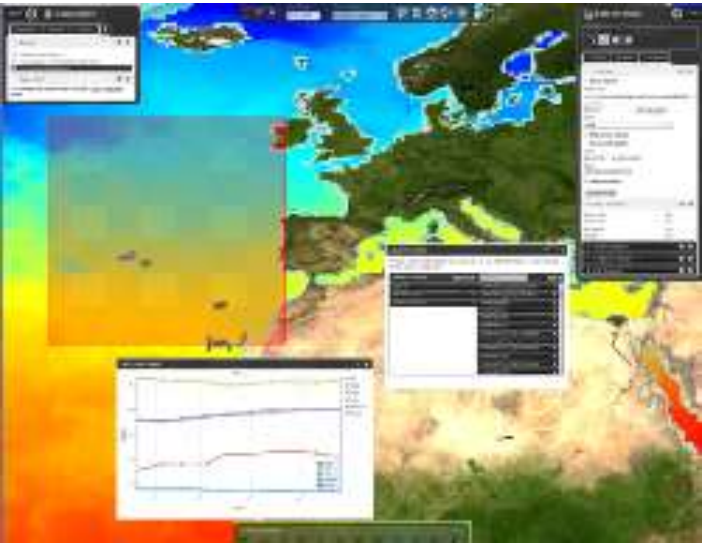


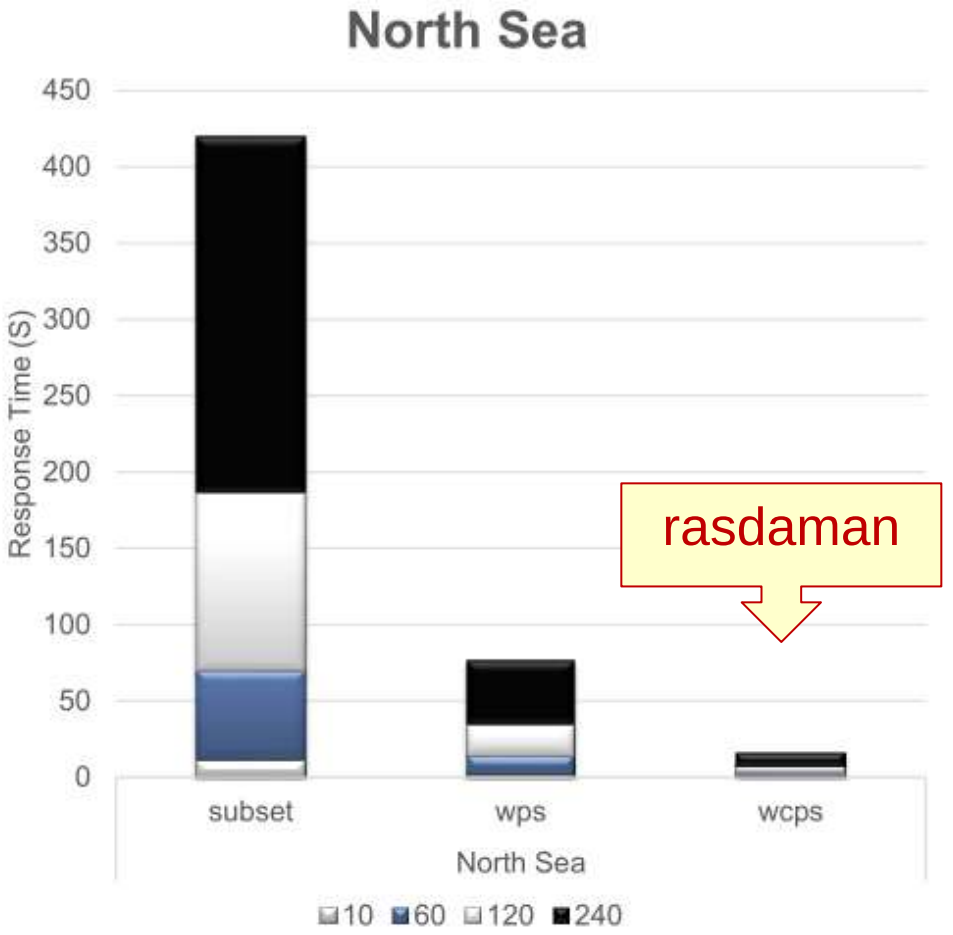
Parallel Geo Service Architecture



Use Case: Plymouth Marine Laboratory

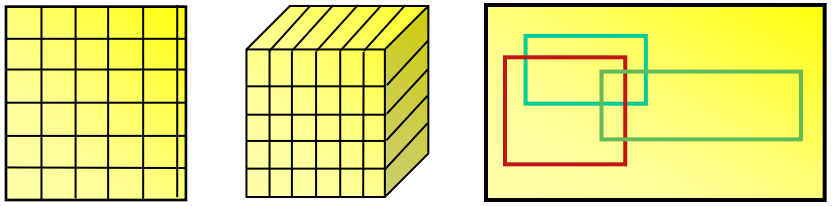
- "Avg chlorophyll concentration for given area & time period, from x/y/t cube"
  - 10, 60,120, 240 days
- Conclusions:
  - „we must minimise data transfer as well as [client] processing"
  - "standards such as WCPS provide the greatest benefit"



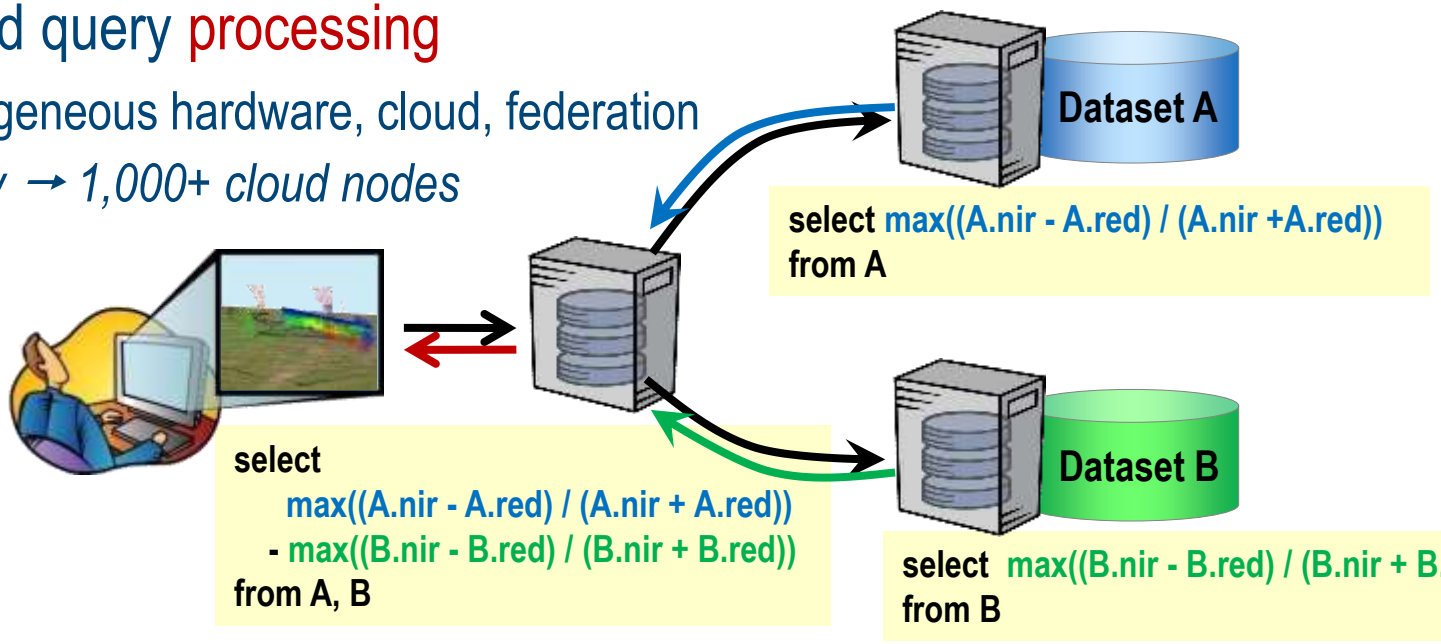


Scalability

- Adaptive data partitioning & distribution
  - storage layout language
  - *130+ TB datacubes*



- Distributed query processing
  - Heterogeneous hardware, cloud, federation
  - *1 query → 1,000+ cloud nodes*



More Standardization

WCS Adoption

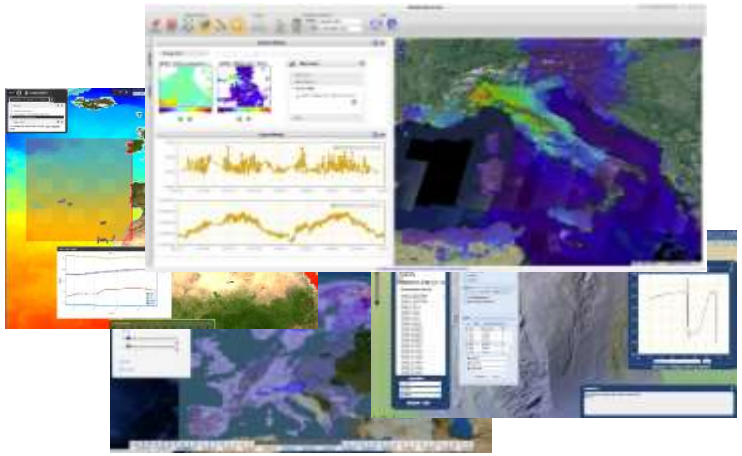
- Broad implementation support, known:
  - rasdaman, GDAL, MapServer, GeoServer, EOxServer
  - Pyxis, ArcGIS
- Feasibility proven in large-scale deployments
  - 130+ TB per single database
  - 1 query → 1,000+ cloud nodes
- Standards in adoption beyond OGC ↳

Current Standardization Status

- Done in OGC:
  - WCS suite as per initial plans (cf. Big Picture)
  - WCS-T in adoption vote
  - ETSs all with OGC, pending massaging into CITE TEAM
- Under work in OGC:
  - SensorML coverages
  - Irregular („referenceable“) grids
  - Extensions for nested / grouped coverages, polygon subsetting, ...
  - WCPS 2.0
  - MetOcean-WCS
- ISO about to adopt WCS
  - OGC Coverage Information Model → ISO 19123-2
  - Brush up 19123 → ISO 19123-1
  - Then WCS Core + POST/XML
  - Then look at further extensions
- W3C Spatial Data on the Web WG
  - coverages

Wrap-Up

- Coverage = spatio-temporal sensor / image / simulation / statistics data
  - regular & irregular grids, point clouds, general meshes
  - OGC Coverage Implementation Model [09-146r2] = concise, interoperable, format-independent
  - Common = can be used by all OGC services (WMS, WFS, WCS, WPS, SOS, ...)
- OGC Web Coverage Service (WCS) suite for direct access to n-D coverage data
  - Tailorable: from simple access (WCS Core) to complex processing (WCPS)
- Interactive demo site: http://standards.rasdaman.com



Relevant Links

- Wikipedia primers:
  - Coverages
  - Web Coverage Service
  - Web Coverage Processing Service
- OGC:
  - coverages info page: http://external.opengeospatial.org/twiki_public/CoveragesDWG/WebHome
  - Authoritative standards source: http://www.opengeospatial.org/standards/wcs
- Coverage service standards online demo: http://standards.rasdaman.com
- The rasdaman Array Database System
- The EarthServer initiative

Document Overview

www.opengeospatial.org/standards

- Coverage data structure:
  - **09-146r2** **Coverage Implementation Model** (was: GML Application Schema for Coverages)
  - 12-100 GeoTIFF Coverage Encoding Extension
  - 12-108 JPEG2000 Coverage Encoding Extension
  - 11-010 NetCDF Encoding Format Extension
- Coverage service:
  - **09-110r4** **WCS 2.0 Core**
  - 12-101 WCS - GeoTIFF Coverage Encoding Extension
  - 09-147r1 WCS KVP protocol extension
  - 09-148 WCS XML/POST protocol extension
  - 09-149 WCS XML/SOAP protocol extension
  - 08-059 WCS - ProcessCoverages Extension
  - 12-040 WCS – Range Subsetting Extension
  - 12-039 WCS - Scaling Extension
  - 11-053 WCS - CRS Extension
  - 12-049 WCS - Interpolation Extension
  - 13-057 WCS - Transaction Extension
- Other:
  - 08-068r2 Web Coverage Processing Service (WCPS) Language
  - 10-140 WCS 2.0 - Earth Observation Application Profile
  - 11-135 Name Type Specification for CRSs