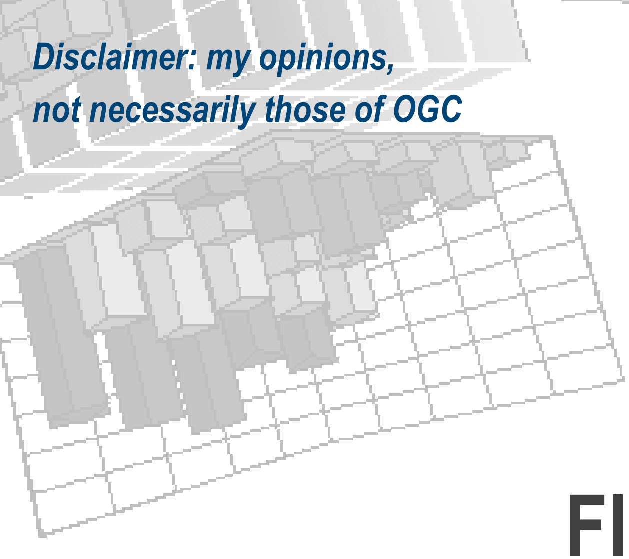
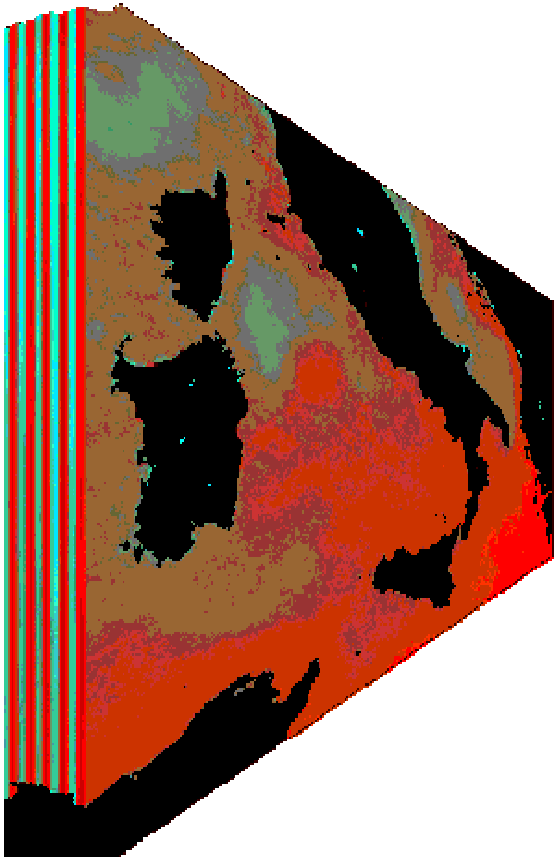


Disclaimer: my opinions,
not necessarily those of OGC

JACOBS
UNIVERSITY

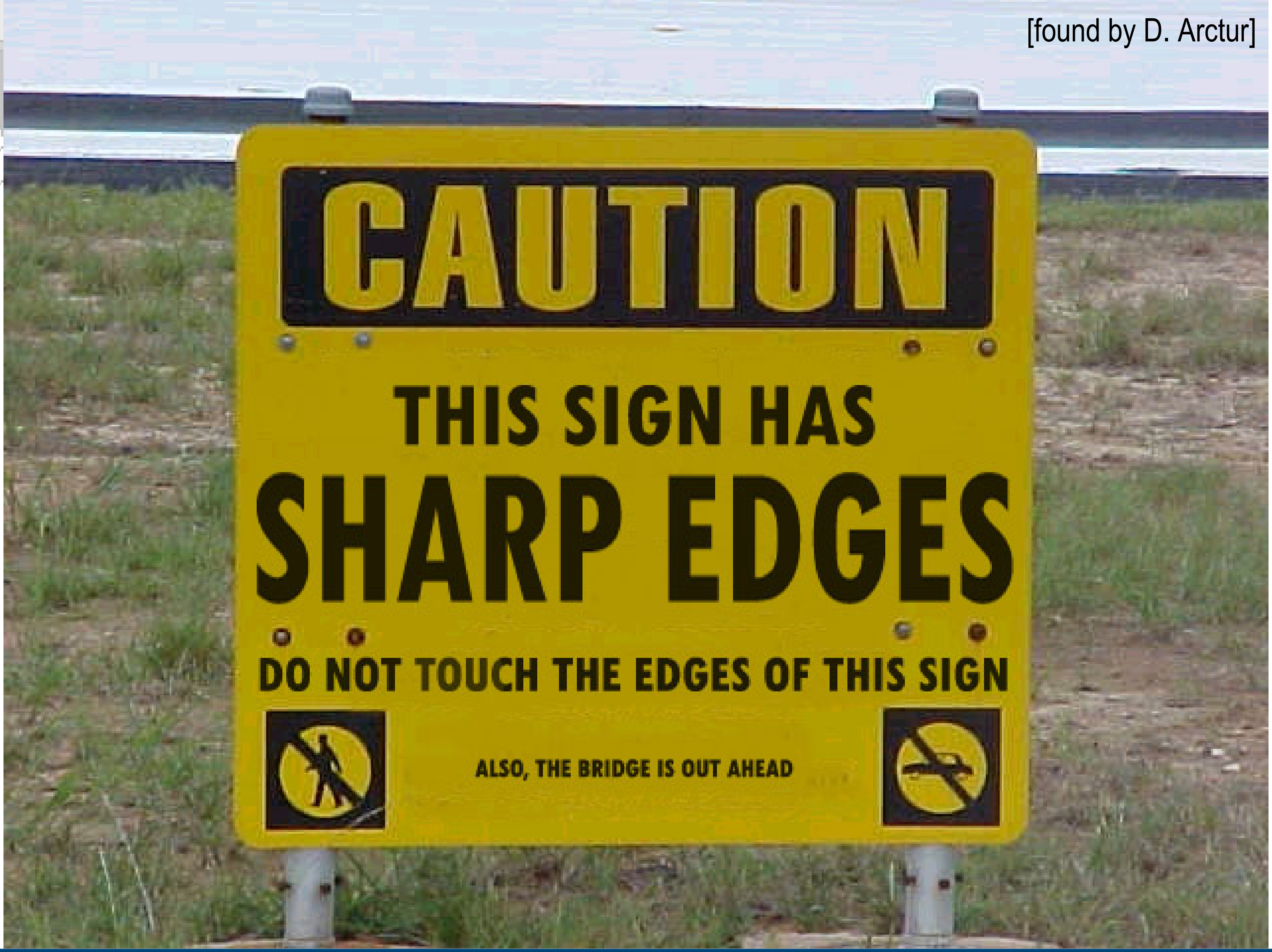
Flexible, Open, Free: The New OGC WCS 2.0 and its Reference Implementation



FOSS4G 2010, Barcelona

Peter Baumann
Jacobs University Bremen

Design By Committee?


CAUTION
THIS SIGN HAS
SHARP EDGES
DO NOT TOUCH THE EDGES OF THIS SIGN
ALSO, THE BRIDGE IS OUT AHEAD

[found by D. Arctur]

Roadmap

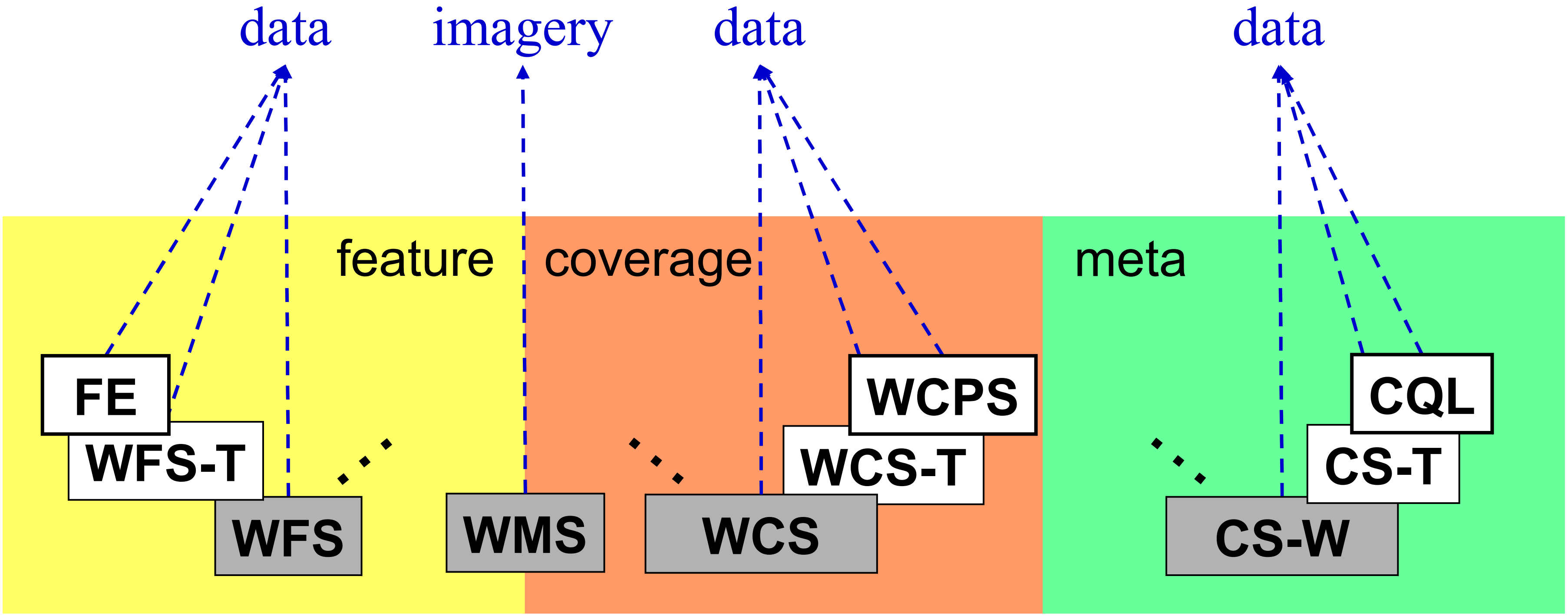
- WCS
- rasdaman
- conclusion

Geo Service Standardization

JACOBS UNIVERSITY

- Main standardization body for geo services: Open Geospatial Consortium (OGC)
  - In collaboration with ISO TC 211, OASIS, W3C, ...
  - Consensus driven
- For WCS 2.0: In-depth stakeholder discussions
  - consultation, requirements elicitation workshops, active participation by many scientific disciplines (remote sensing, atmospheric research, ocean research, astrophysics, …), industry, and governmental bodies
  - In parallel: experiments on implementation feasibility
- About the presenter (ahem, me)
  - Chairing WCS.SWG, Coverages.DWG
  - Editor of 9+ specifications, among them WCS 2.0 and WCPS 1.0

(Part of) The OGC Quilt



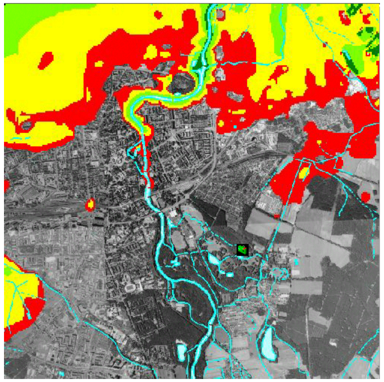
WCS 2.0 Package

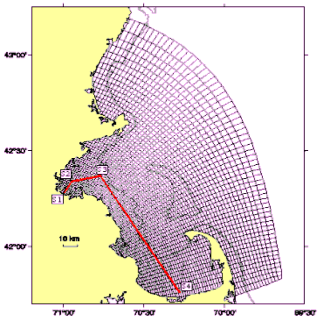
- The Data: GML Application Schema for Coverages 1.0
  - Coverage data structure, can be used separately from service (WCS)
  - Based on GML 3.2.1, harmonized with GML, SWE, WCS; coming: WPS, O&M
  - Foreseen: to be integrated into GML 4.0
- The Service: Web Coverage Service (WCS) 2.0
  - Core: simple access & subsetting service
  - Add-ons (“extensions”) for reprojection, processing, … (TBD)
  - Format encoding extensions under work for GeoTIFF, GML, NetCDF, JPEG2000

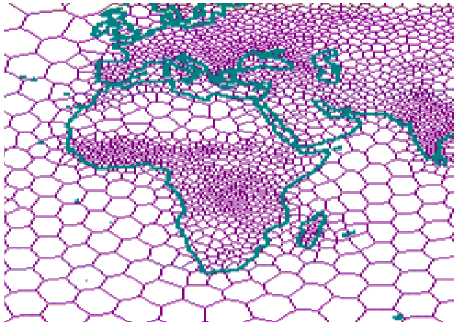
Coverage Definition

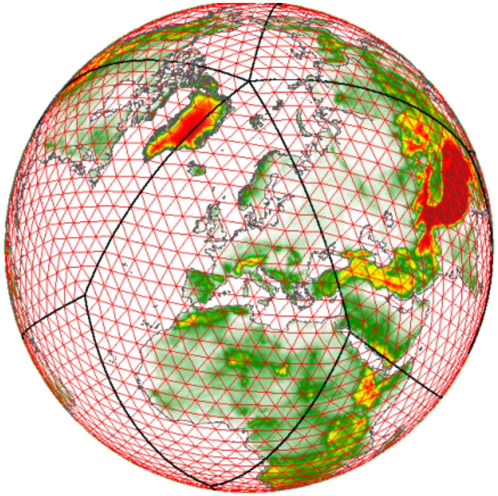
JACOBS UNIVERSITY

- Coverage = multi-dimensional spatio-temporally variable phenomenon
  - ISO 19123 (= OGC Abstract Topic 6) – *abstract, not for implementation*
  - Can model rasters, non-regular grids, curvilinear grids, TINs, meshes, ...

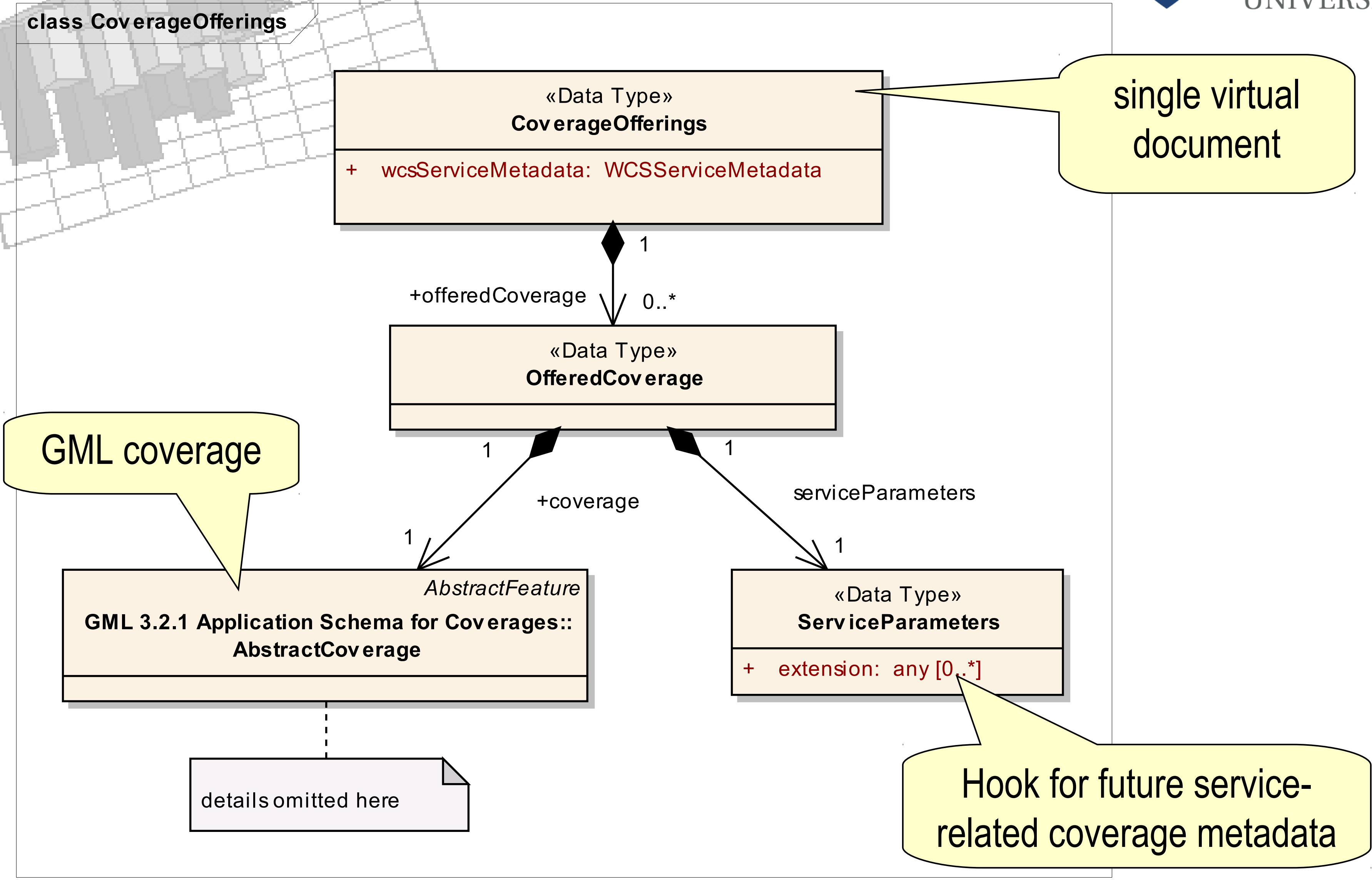








WCS 2.0 Coverage Offering



single virtual document

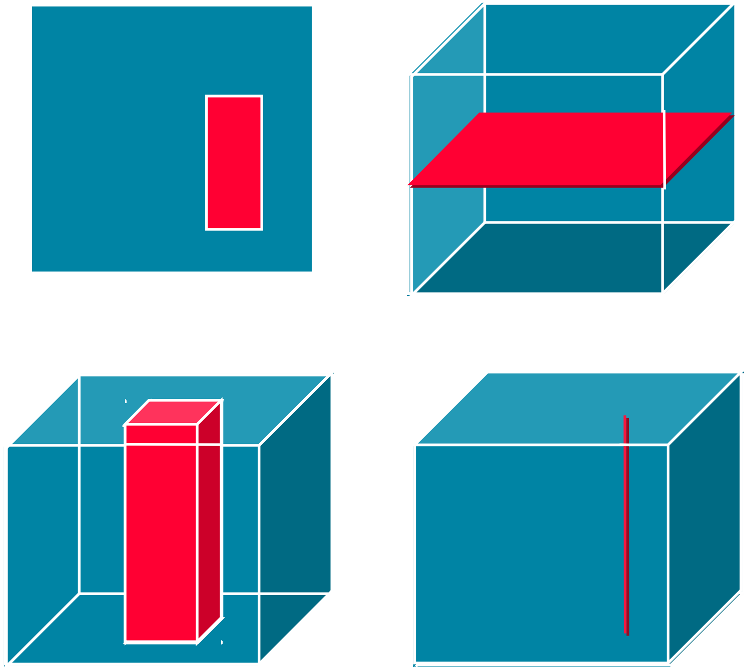
GML coverage

Hook for future service-related coverage metadata

WCS Core Functionality

In Core, simple data access (more in extension packages):

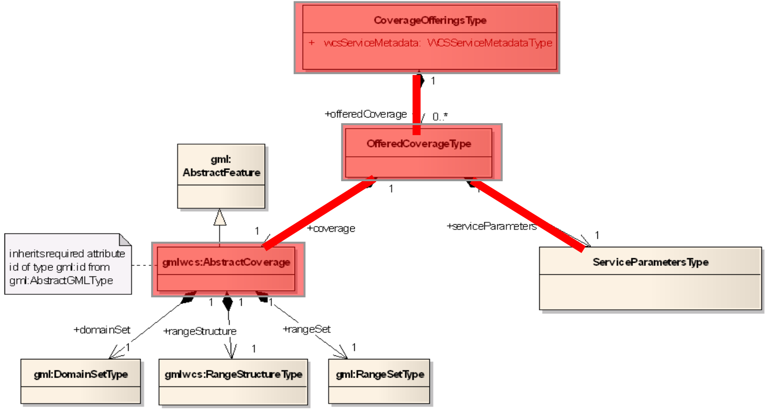
subset = trim | slice

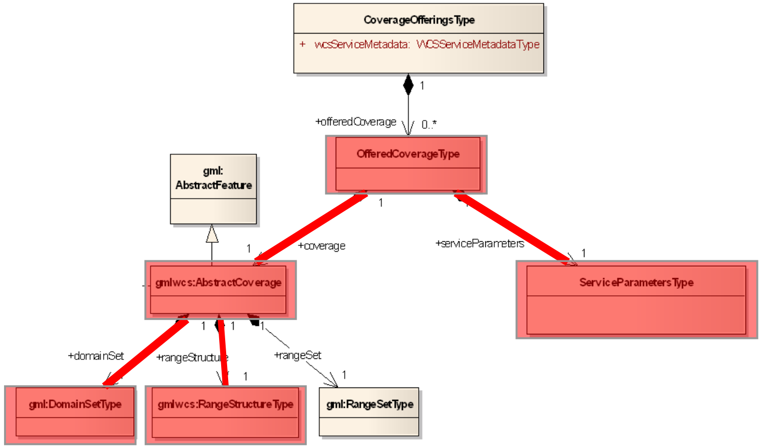


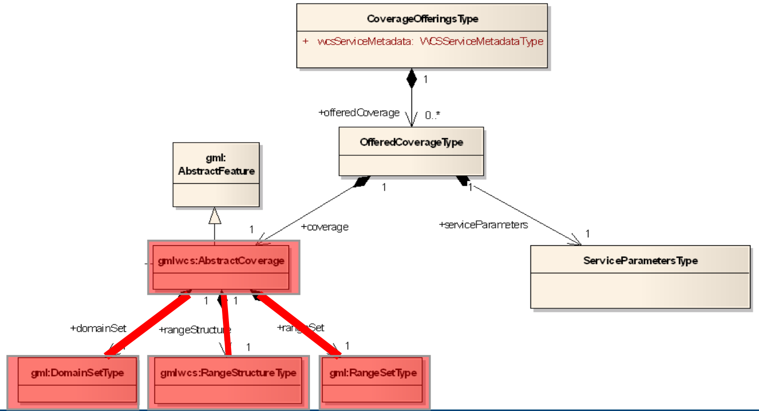
WCS Operations

- 3 request types (as before):
  - *GetCapabilities*
  - *DescribeCoverage*
  - *GetCoverage*

- concisely defined semantics: response specified by pruning coverage offerings
  - encoding can vary!







Roadmap

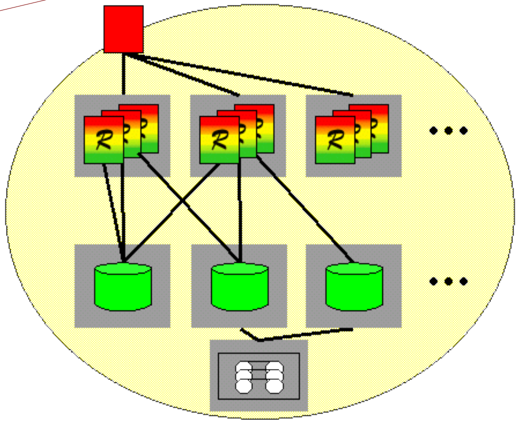
- WCS
- rasdaman
- conclusion

The *rasdaman* Raster Data Manager

JACOBS UNIVERSITY

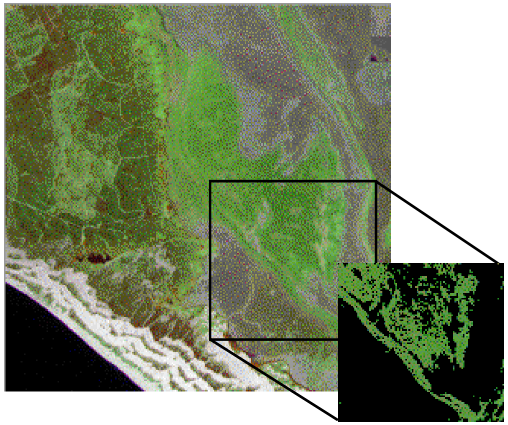
www.rasdaman.org

- „Array DBMS“ for massive n-D raster data
  - multidimensional SQL, Java, C++
  - intelligent storage & query optimization, such as HW/SW parallelization



- rasql = n-D raster expressions in SQL

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

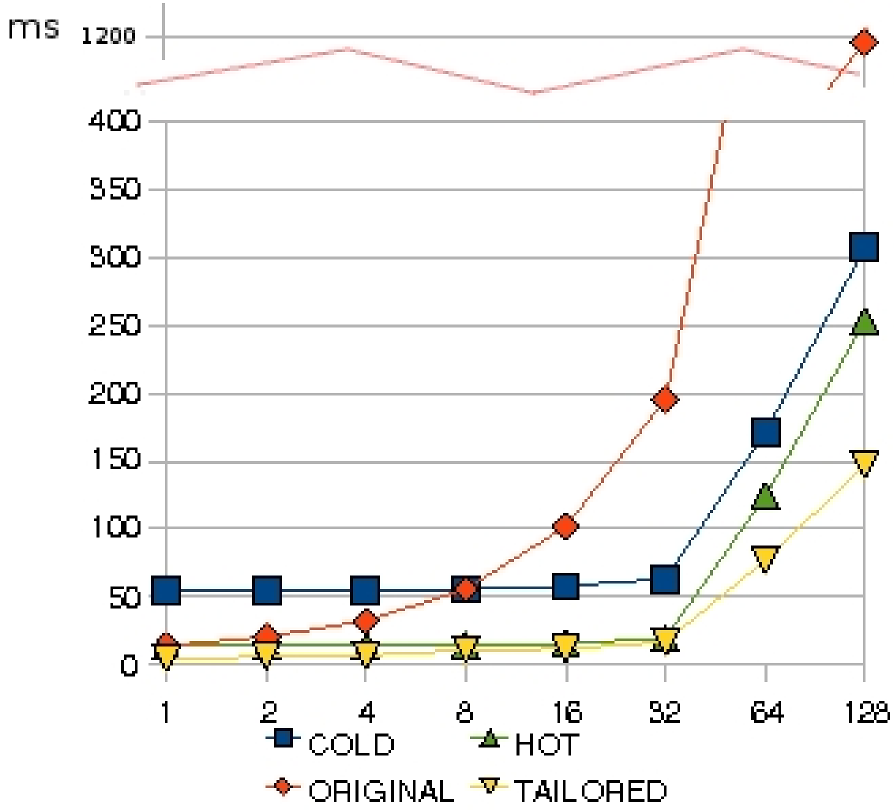


- High community impact
  - In operational use since 5+ years with dozen-TB objects
  - “most comprehensively implemented system“ (Rona Machlin, ACM PoDS, 2007)

Just One Optimization: Just-In-Time Compilation

- Observation: interpreted mode slows down
- Approach:
  - cluster suitable operations
  - compile & dynamically bind
- Benefit:
  - Speed up complex, repeated operations
- Variation:
  - compile code for GPU

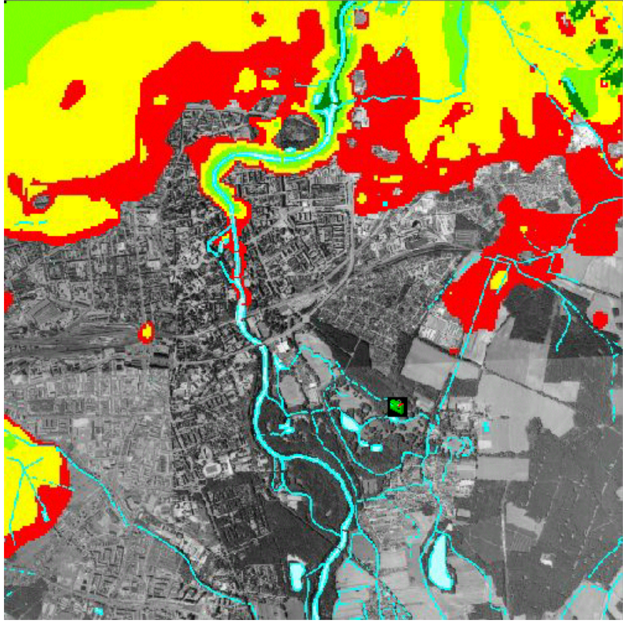
```
select x*x*...*x
from float_matrix as x
```



Times [ms] for 512^2 * n ops

Optimisation Does Pay Off!

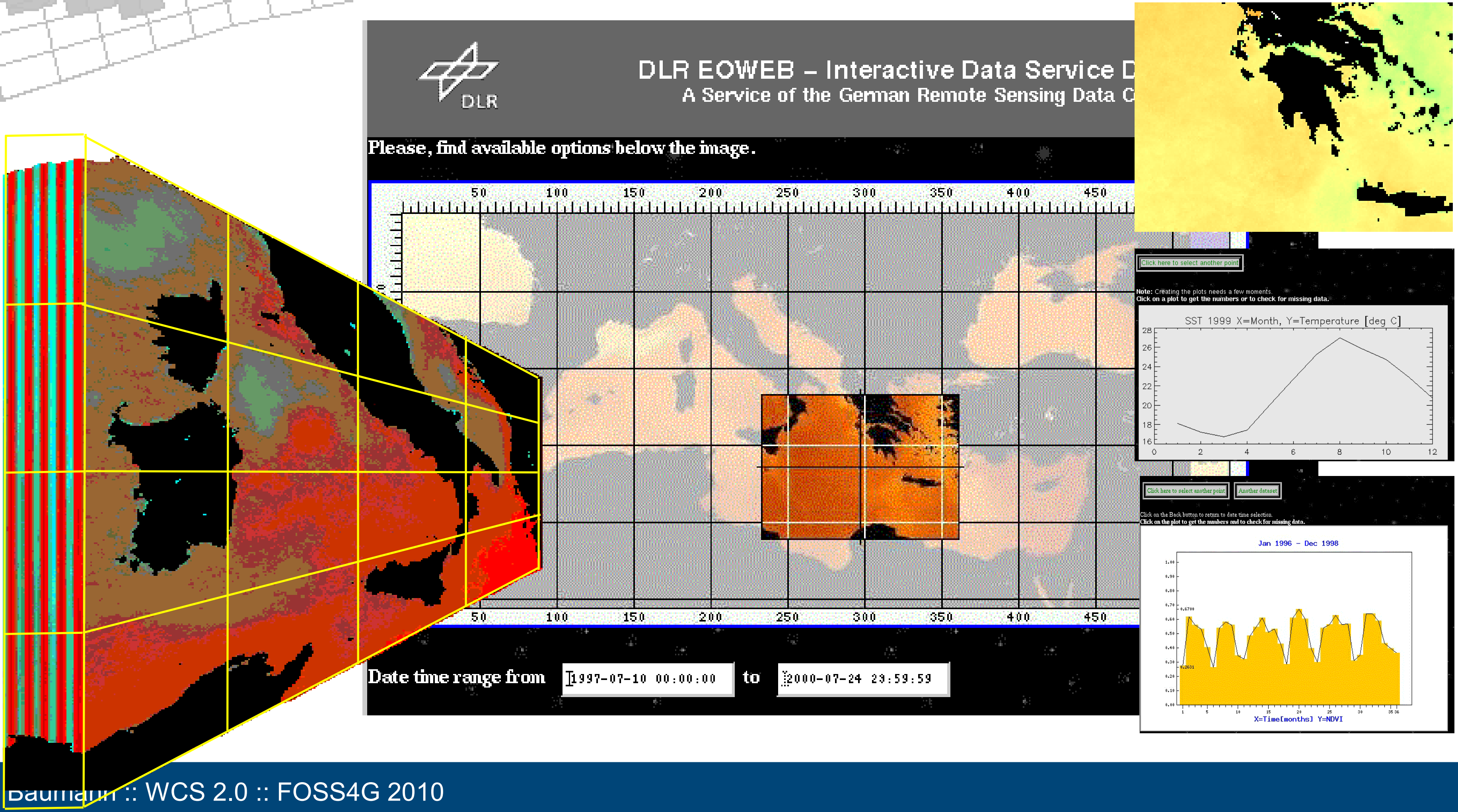
- Complex queries give more space to optimizer
- Typical OGC *Web Map Service* query:



```
select jpeg(
              scale(bild0[...],[1:300,1:300])          * { 1c, 1c, 1c}
      overlay ((scale(bild1[...],[1:300,1:300])<71.0)) * {51c, 153c, 255c }
      overlay bit(scale(bild2[...],[1:300,1:300]), 2)  * {230c, 230c, 204c}
      overlay bit(scale(bild2[...],[1:300,1:300]), 5)  * {1c, 1c, 1c}
      overlay bit(scale(bild2[...],[1:300,1:300]), 7)  * {102c, 102c, 102c}
      overlay bit(scale(bild2[...],[1:300,1:300]), 6)  * {255c, 255c, 0c}
      overlay bit(scale(bild2[...],[1:300,1:300]), 3)  * {191c, 242c, 128c}
      overlay bit(scale(bild2[...],[1:300,1:300]), 4)  * {191c, 255c, 255c}
      overlay bit(scale(bild2[...],[1:300,1:300]), 1)  * {0c, 255c, 255c}
      overlay bit(scale(bild2[...],[1:300,1:300]), 0)  * {102c, 102c, 102c}
        )
from ...
```

Sample WCS Based 3-D Service

[Diedrich et al 2001], based on rasdaman



DLR EOWEB – Interactive Data Service D

A Service of the German Remote Sensing Data C

Please, find available options below the image.

Date time range from 1997-07-10 00:00:00 to 2000-07-24 23:59:59

Note: Creating the plots needs a few moments.
Click on a plot to get the numbers or to check for missing data.

Click on the Back button to return to date time selection.
Click on the plot to get the numbers and to check for missing data.

Some Current Activities

- OSGeo incubation
- GDAL driver (accepted)
- IQL (Integrated Query Language): PostgreSQL + PostGIS + rasql = 1
- WCS for EO product distribution (ESA)
- Bridging coverage & processing standards (ESA)
- Raster QL as satellite interface (Vightel / NASA)

Roadmap

- WCS
- rasdaman
- conclusion

WCS: Call for Implementations

JACOBS UNIVERSITY

- Acceptance critically depending on function-rich, friendly, „fancy“ clients
  - Look&feel often domain dependent, no “one fits all”
- Invitation to OS developers:
  provide WCS clients to community!
- BTW, next OGC TC meetings:
  - September 20 (Toulouse)
  - November 29 (Sydney)

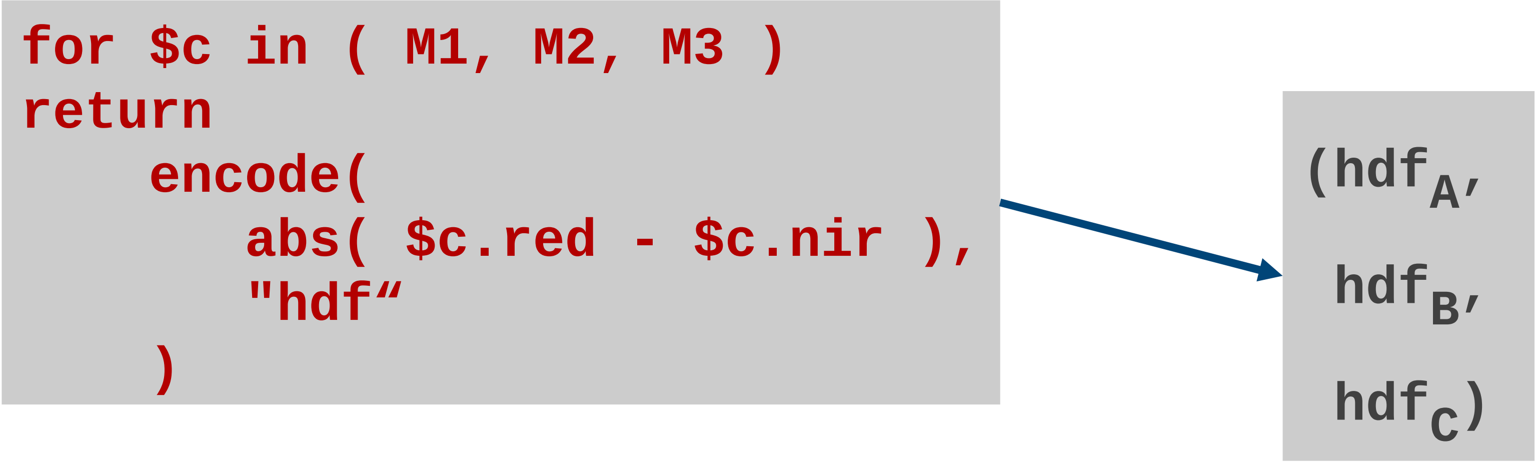
Conclusion

- OGC WCS 2.0 for open, interoperable, scalable coverage access
  - n-D rasters, irregular grids, and more
  - unified coverage model for all OGC stds
- rasdaman platform
  for reference implementation
  - Versatile, high-performance
    raster database
- Wanted: implementations
  (in particular: clients!)
  - Collaborations welcome
  - *Join developer & user community!*

WCS Processing Service (WCPS)

JACOBS UNIVERSITY

- „XQuery for coverages“: declarative, safe raster expression language
- "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“
    )
```

$(hdf_A,$

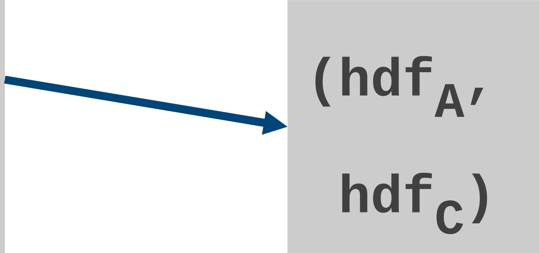
$hdf_B,$

$hdf_C)$

WCS Processing Service (WCPS)

- „XQuery for coverages“: declarative, safe raster expression language
- "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“
    )
```



$(hdf_A,$

$hdf_C)$

WCS Processing Service (WCPS)

- „XQuery for coverages“: declarative, safe raster expression language
- "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)

Raster Type Definition

-
  ```
  typedef marray
  <    unsigned char, [ 1:1024, 1:768 ]
  > XGA_Grey_Image;
  ```

-
  ```
  typedef marray
  <    struct { unsigned char red, green, blue; },
       [ *:*, *:* ]
  > RGB_Image;
  ```

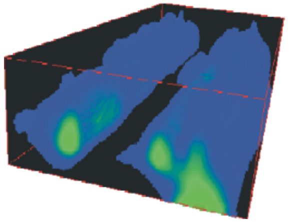
-
  ```
  typedef marray
  <    unsigned short, [ 1:1654, 1:* ]
  > G3_Fax;
  ```

-
  ```
  typedef marray
  <    struct { double vx, vy; }, [ 0:*, 0:127, 0:63, 0:16 ]
  > ECHAM_T42_Windspeed;
  ```

The rasql Query Language

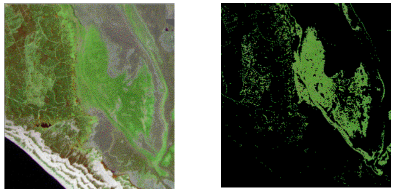
- selection & section

  -
    ```
    select c[ *:*, 100:200, *:*, 42 ]
    from   ClimateSimulations as c
    ```



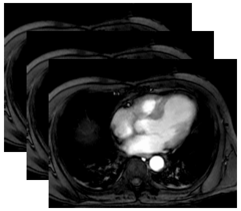
- result processing

  -
    ```
    select img * (img.green > 130)
    from   LandsatArchive as img
    ```



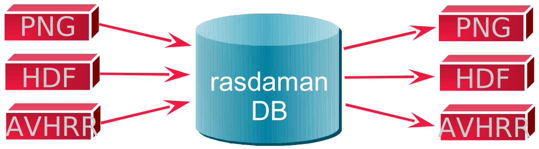
- search & aggregation

  -
    ```
    select mri
    from   MRI as img, masks as am
    where  some_cells( mri > 250 and m )
    ```



- data format conversion

  -
    ```
    select png( c[ *:*, *:*, 100, 42 ] )
    from   ClimateSimulations as c
    ```


PNG
HDF
AVHRR
rasdaman DB
PNG
HDF
AVHRR