Web 3D Service (W3DS)
Status Report

Arne Schilling
University of Bonn, Germany

OGC TC Meeting 12/03/2008 Valencia

Web 3D Service (W3DS) Status Report

1. W3DS Introduction
2. New Extensions
3. OWS-6 DSS Thread

W3DS Status

Current Status: OGC Discussion Paper (OGC 05-019)

Released: February 2005
Version: 0.3.0

Editors: Udo Quadt, Thomas Kolbe

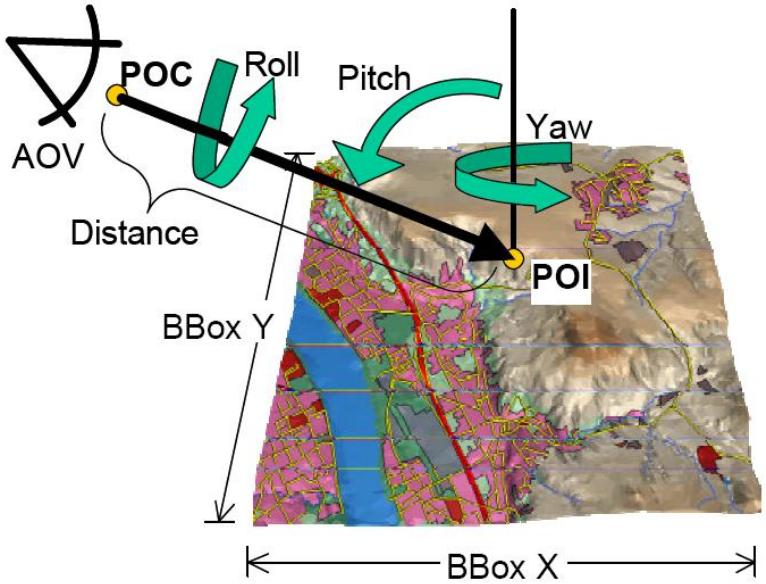
Presentations in TC Meetings:
January, 18th 2005, OGC TC meeting, New York City
November 7th 2005 OGC TC meeting, Bonn

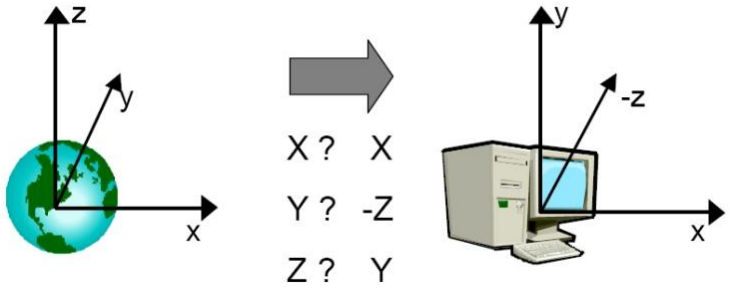
W3DS Scope

W3DS delivers 3D scenes of a selected region that can be explored interactively

What is a Scene?

1. A Scene is composed of data from one or multiple layers
2. A Scene may also contain map elements (title, compass, scale bar, legend etc.) -> “3D map”, and predefined viewpoints
3. A Scene must be provided in a CRS that can be used for visualization. NOT WGS84. Ideal: Cartesian coordinates. Large coordinates are also problematic.
4. A Scene is composed of “Display Elements” (geometries, triangles, materials, animations, lights, fog).
5. The structure of a Scene is not defined!
6. Semantics usually missing, depends on format



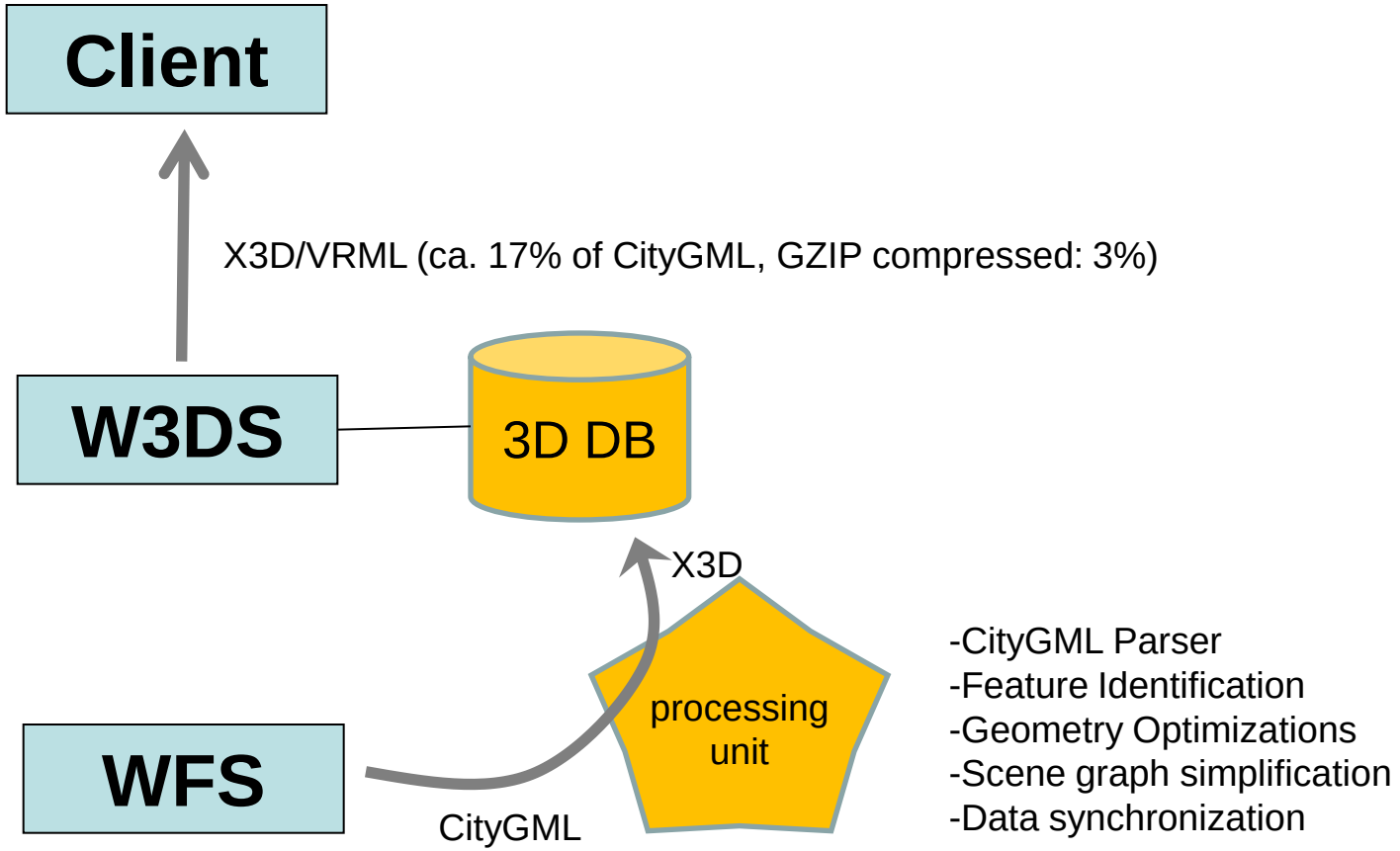


universitätbonn Geographie

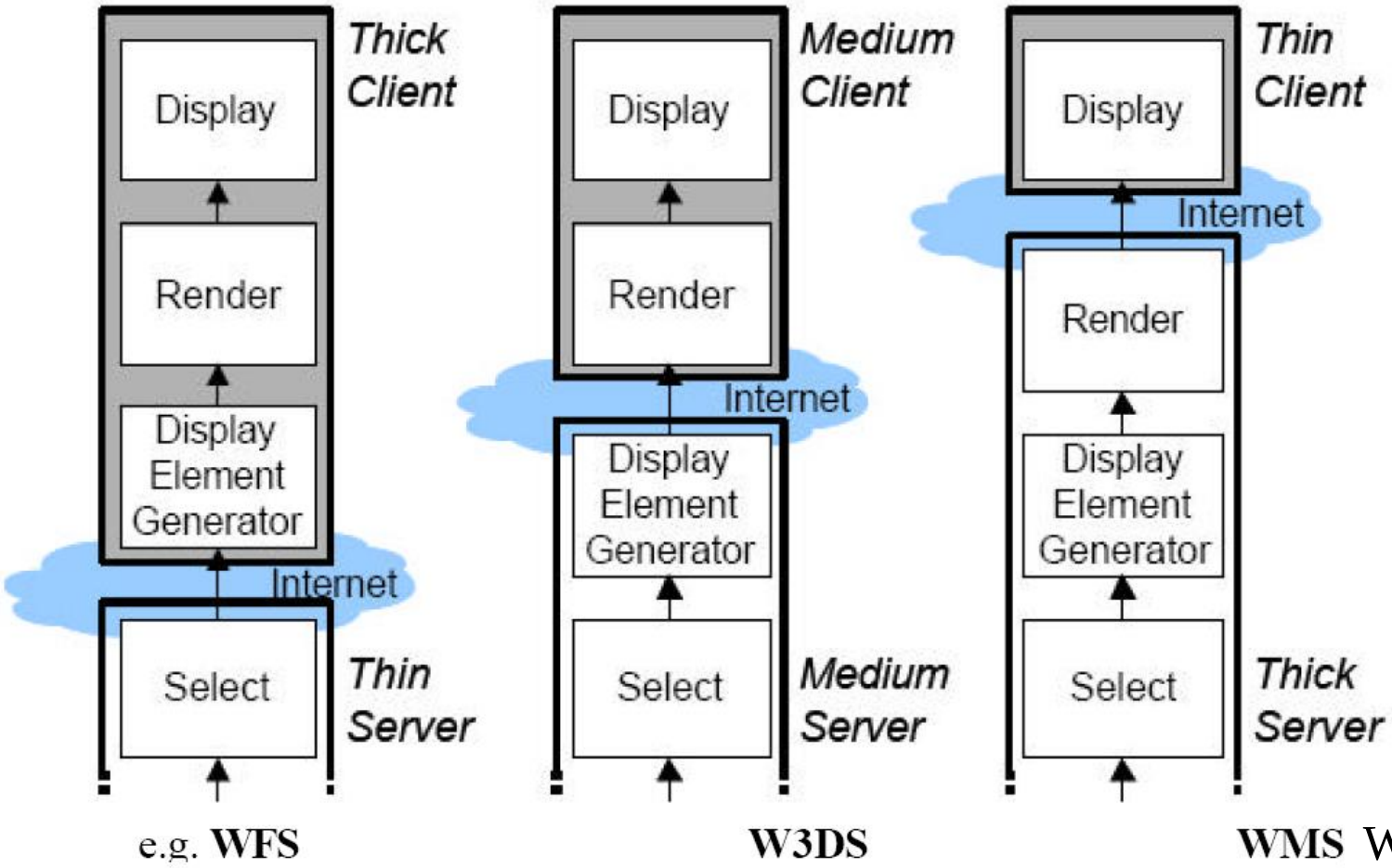
Optimization Techniques for efficient rendering

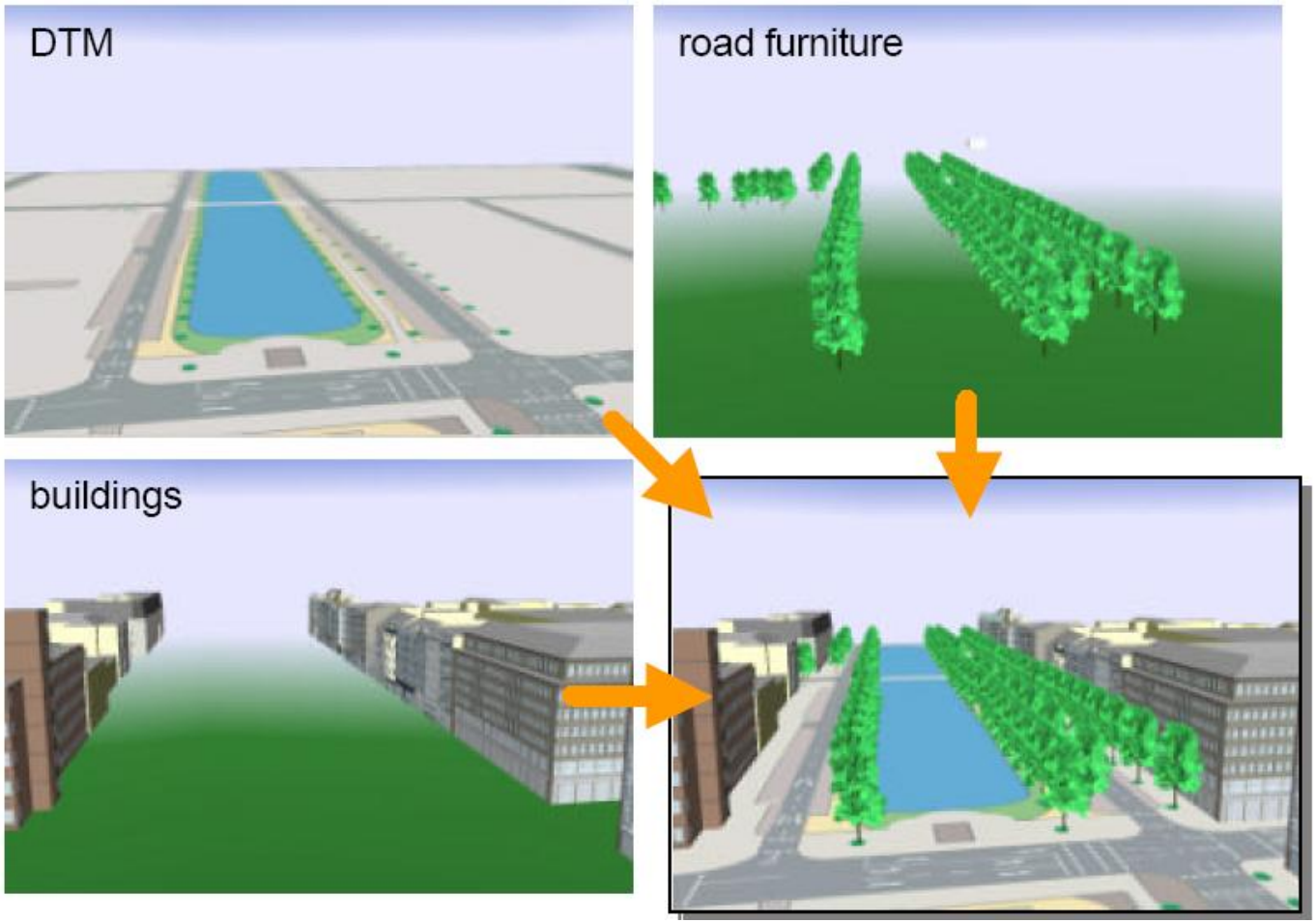
- 1) Reduce the depth of the scene graph
  - Combine nested Transform Groups, remove unnecessary Groups
- 2) Reference to existing Materials and Appearances
  - Instead of defining the same Material repeatedly for each object, re-use already defined materials -> less memory consuming
- 3) sort objects according to Material (Display Lists) **In Java3D: “compiling”**
  - Switching between materials during rendering is costly!
- 4) combine objects (3D Shapes) with same material
  - Instead of many 3D Shapes with each a geometry and a material: combine to few larger 3D Shapes. E.g. combine all roof geometries with same material into one larger geometry.
  - Structure of Scene gets lost!!!, no individual GIS objects any more!!
- 5) Create a Texture Atlas
  - Copy all textures into one large texture. Parts of the Texture Atlas are applied to the objects

Portrayal of GityGML



Portrayal Pipeline




DTM
road furniture
buildings

W3DS Operations

- GetCapabilities (mandatory)
  - -> very similar to WMS: layers, styles, bbox etc.
- GetScene (mandatory)
  - -> returns a 3D Scene / Scenegraph

W3DS Operations

- GetCapabilities (mandatory)
  - -> very similar to WMS: layers, styles, bbox etc.
- GetScene (mandatory)
  - -> returns a Scenegraph
- GetFeatureInfo (optional, Version 0.3.1)
  - -> returns attributes of selected features
- GetLayerInfo (optional, Version 0.3.1)
  - -> returns attribute names and values of selected layer

GetCapabilites Response

Scale Denominators

“The <MinScaleDenominator> and <MaxScaleDenominator> elements define the range of scales for which it is appropriate to generate a map of a Layer.”
“The scale denominator values are guidelines for clients, not firm limits.” (OGC 06-042)

<MinScaleDenominator>1e3</MinScaleDenominator>
<MaxScaleDenominator>1e6</MaxScaleDenominator>

1/6e06 – 1/12e06

1/500 – 1/3000

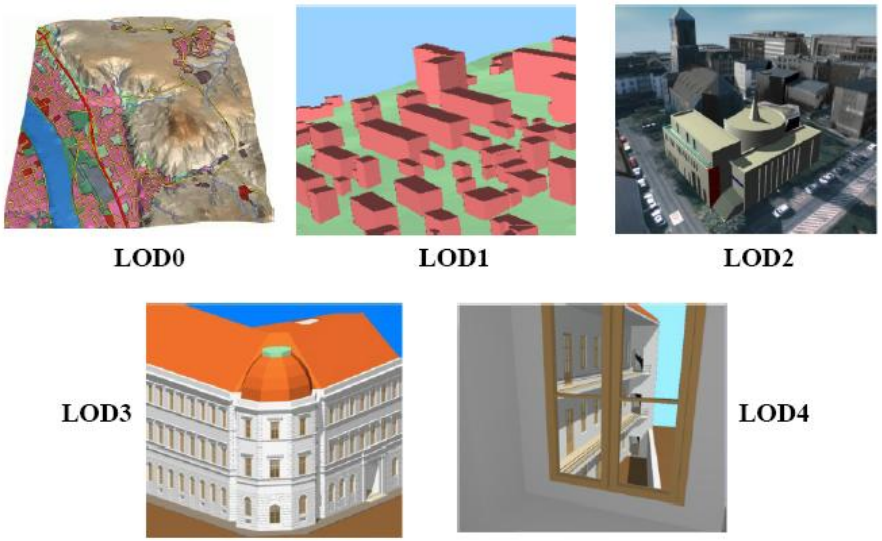
GetCapabilites Response

Levels of Detail

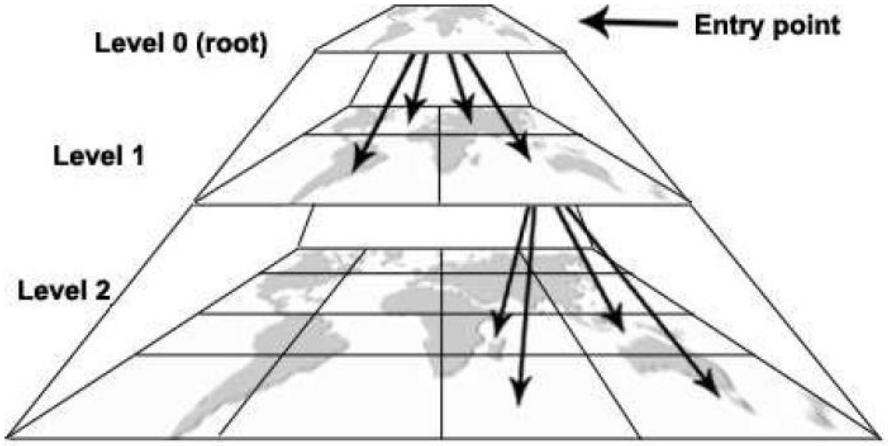
The optional <MinLevelOfDetail> and <MaxLevelOfDetail> elements describe the range of Levels of Detail that can be provided by the layer.

<MinLevelOfDetail>CityGML:1</MinLevelOfDetail>

<MaxLevelOfDetail>CityGML:4</MaxLevelOfDetail>



Extended LOD Concept



Misund, Grandlund, Kolas

Tile Level	Size	CityGML LOD	Scope
16	0,5 km	4 -indoor	Bus Stop
15	1 km	3	
14	2 km	2	Street Level
13	5 km	1	
12	10 km	0 - landscape	
11	20 km		
10	39 km		City
9	78 km		
8	156 km		
7	313 km		State
6	625 km		
5	1250 km		Country
4	2500 km		
3	45°		Continent
2	90°		
1	180°		
0	360°		Globe

GetScene Parameters

URL parameter	Required/ Optional/ Conditional	annotation
VERSION=<*version*>	R	requested version
REQUEST=GetScene	R	requested operation
SRS=namespace:identifier	R	spatial reference system
POI=<*point_of_interest*>	C	x,y,z point coordinates according to SRS
PITCH=<*pitch*>	C	angle of inclination [degree]
YAW=<*yaw*>	C	azimuth [degrees]
ROLL=<*roll*>	O	rotation around viewing vector [degree]
DISTANCE=<*distance*>	C	distance POI to POC [meter]

GetScene Parameters cont'd

POC=x, y, z	C	x,y,z coordinates of camera according to SRS
AOV=<*angle_of_view*>	C	angle of view [degree]
BBOX=*xmin,ymin,xmax,ymax*	R	2d bounding box
MINHEIGHT=<*lower_limit*>	O	displaying objects with height ≥ *lower_limit* according to SRS
MAXHEIGHT=<*upper_limit*>	**O**	displaying objects with height ≤ *upper_limit* according to SRS
LAYERS=<*layer list*>	O	comma separated list of 3D object sets
STYLES=<*style list*>	O	comma separated list of styles for each layer
FORMAT=<*format*>	R	MIME type of output
TIME=<*date_and_time*>	O	date and time
EXCEPTIONS=<*excepttype*>	O	exception format
TRANSLATE=*x,y,z*	C	translation vector that is applied to all 3D coordinates
ENVIRONMENT=on / off	O	switch on/off background elements like sky or light source
BGCOLOR=<*color*>	O	background color
BGIMAGE=<*image url*>	O	URL of background image

New GetScene Parameters in 0.3.1

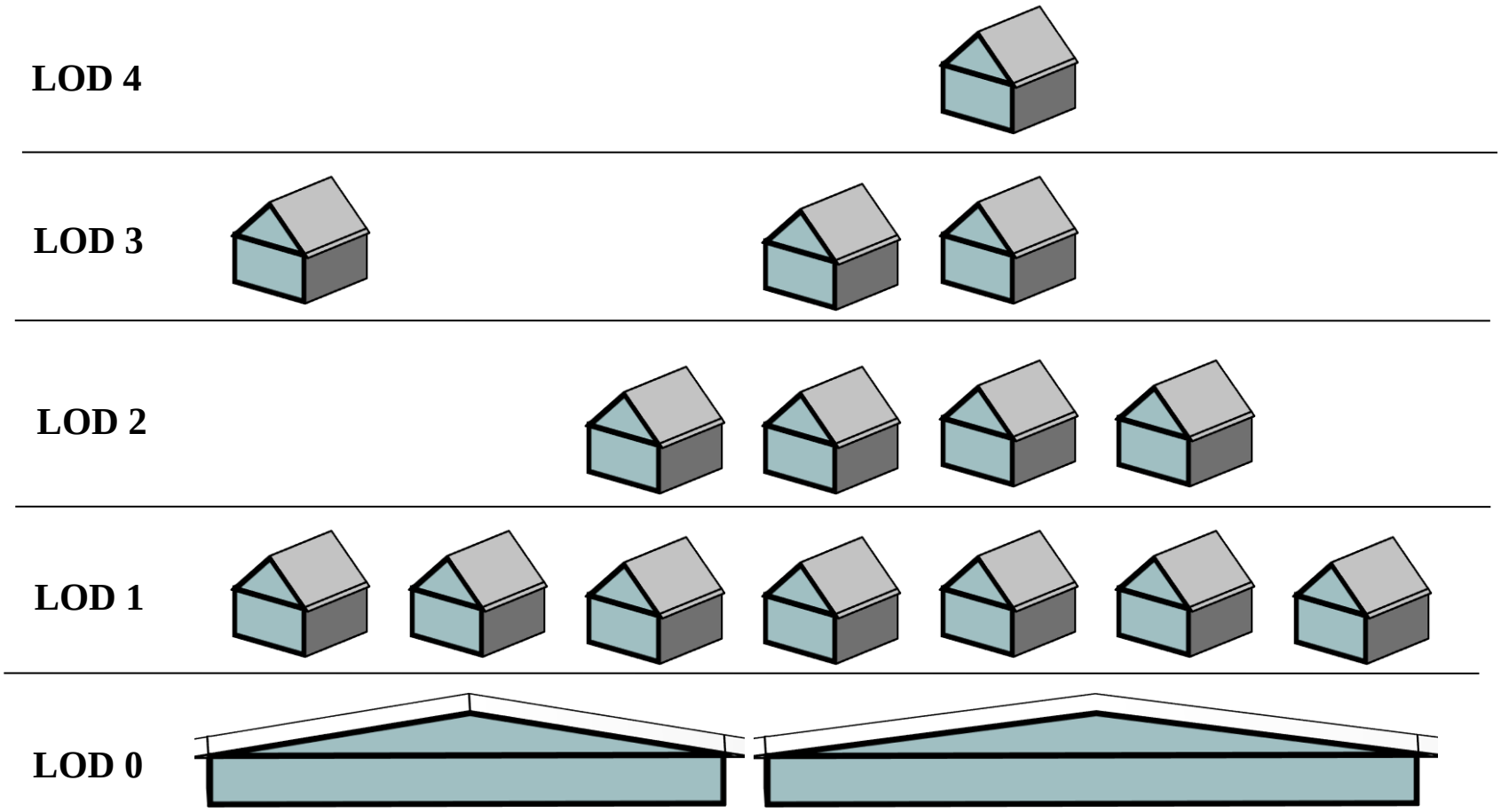
Level of Detail

- Objects of one layer may be available in multiple LODs, especially building models (block, w. roof shape, textured, detailed façade model, indoor model)
- Terrain layer may be also available in multiple resolutions (from global earth surface to very detailed local models showing ditches, dikes etc.

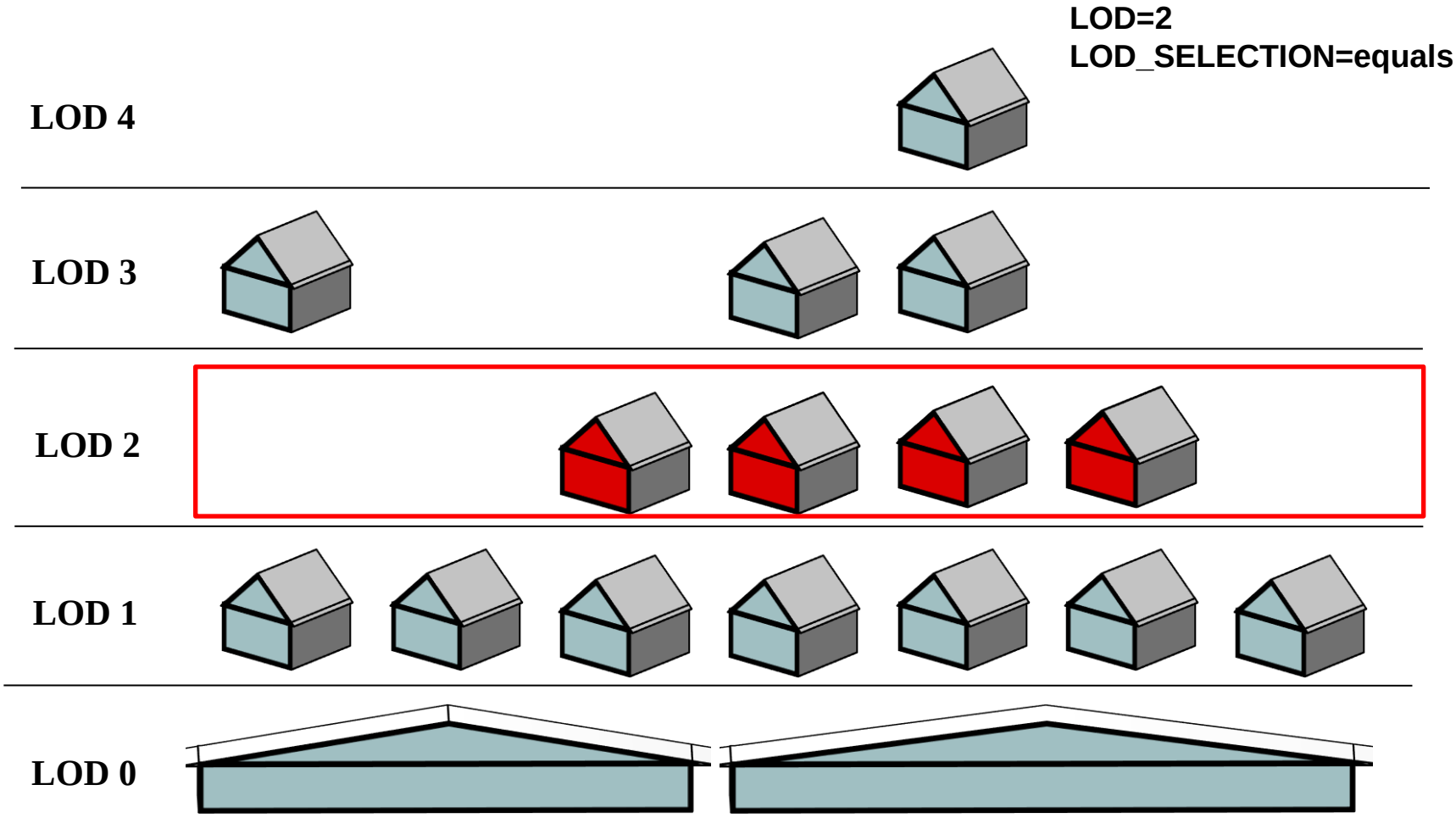
Parameters: LOD=<*string*> *(“qualifier:number”)*

LOD_SELECTION=<*string*> *(“equals”* | *“equals_or_smaller”)*

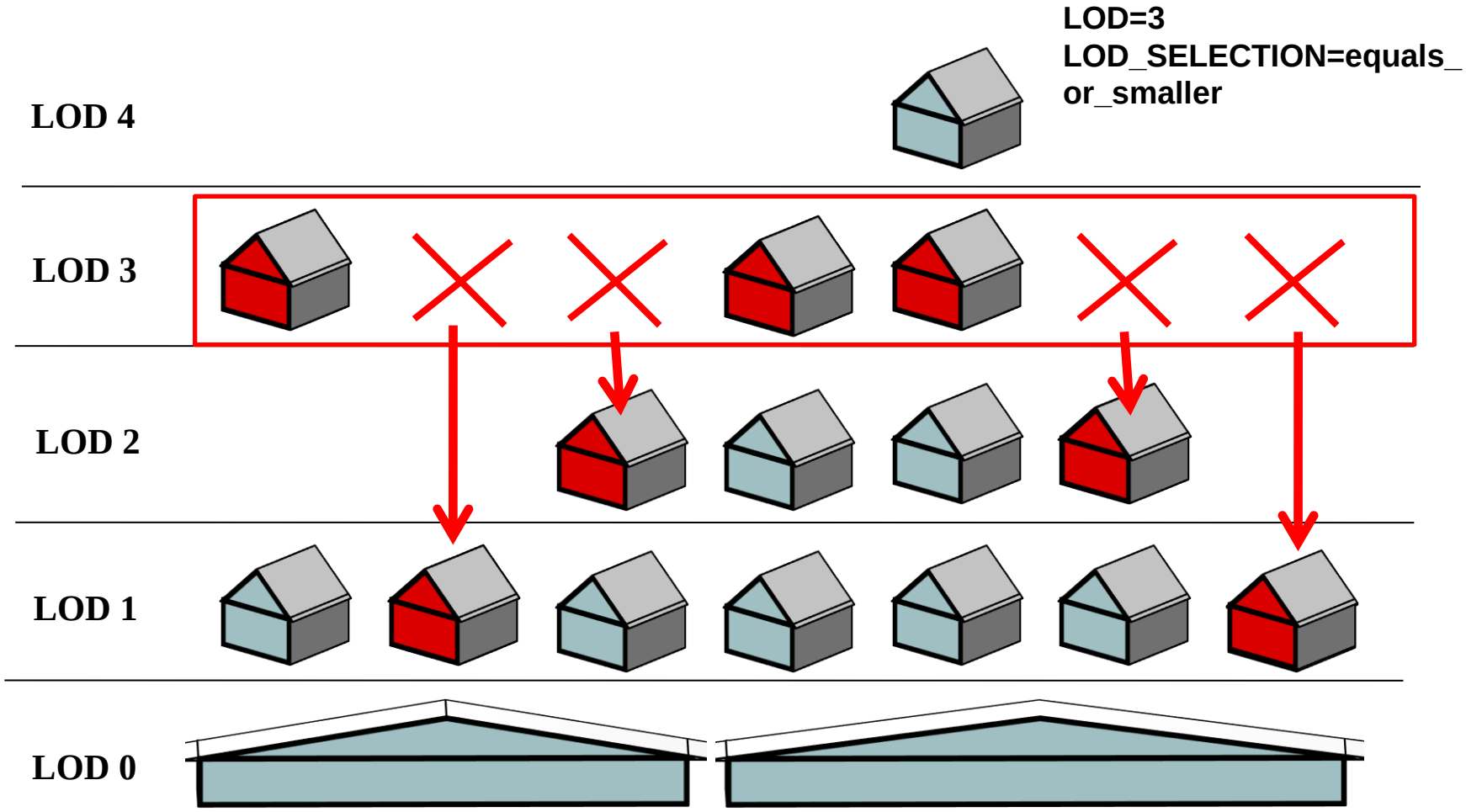
LOD Selection



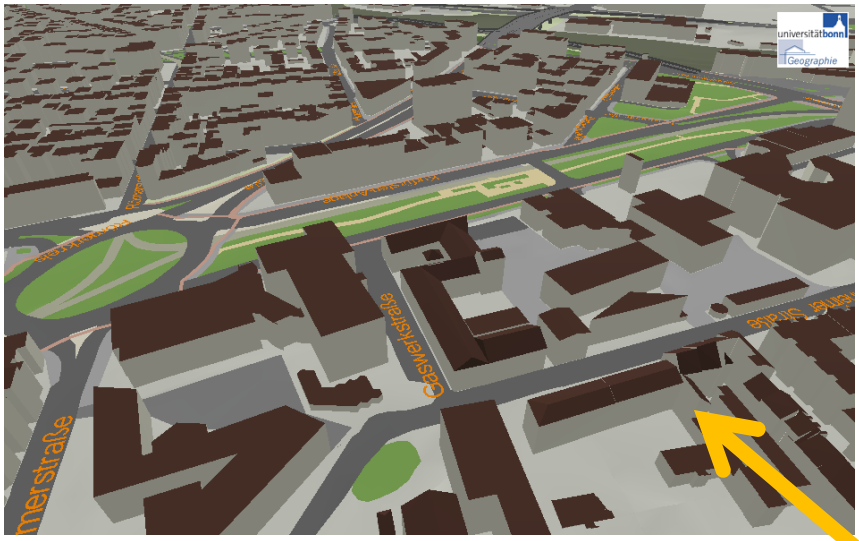
LOD Selection



LOD Selection



LOD Selection



LOD=1
LOD_SELECTION=equals

LOD=2
LOD_SELECTION=equals

LOD=2
LOD_SELECTION=equals_or_smaller

New GetScene Parameters in 0.3.1

Spatial Constraint

- Currently: a) BBOX=*xmin,ymin,xmax,ymax*
  b) MINHEIGHT=<double>, MAXHEIGHT=<double>

We need more flexibility !

- Using WKT: BOUNDS=*WKT* *(Polygon | PolyhedralSurface Z)*

Examples:

BOUNDS=Polygon((x,y, x,y,…..x,y), (x,y, x,y,…..x,y)) Polygon with Hole

BOUNDS=PolyhedralSurface Z (((x,y,z, x,y,z, x,y,z)), ((x,y,z, x,y,z, x,y,z)), ….. ((x,y,z, x,y,z, x,y,z)))

Condition: Surface must be closed!

Part of SFS 1.2 (OGC 06-103r3)

New GetScene Parameters in 0.3.1

Spatial Selection Method

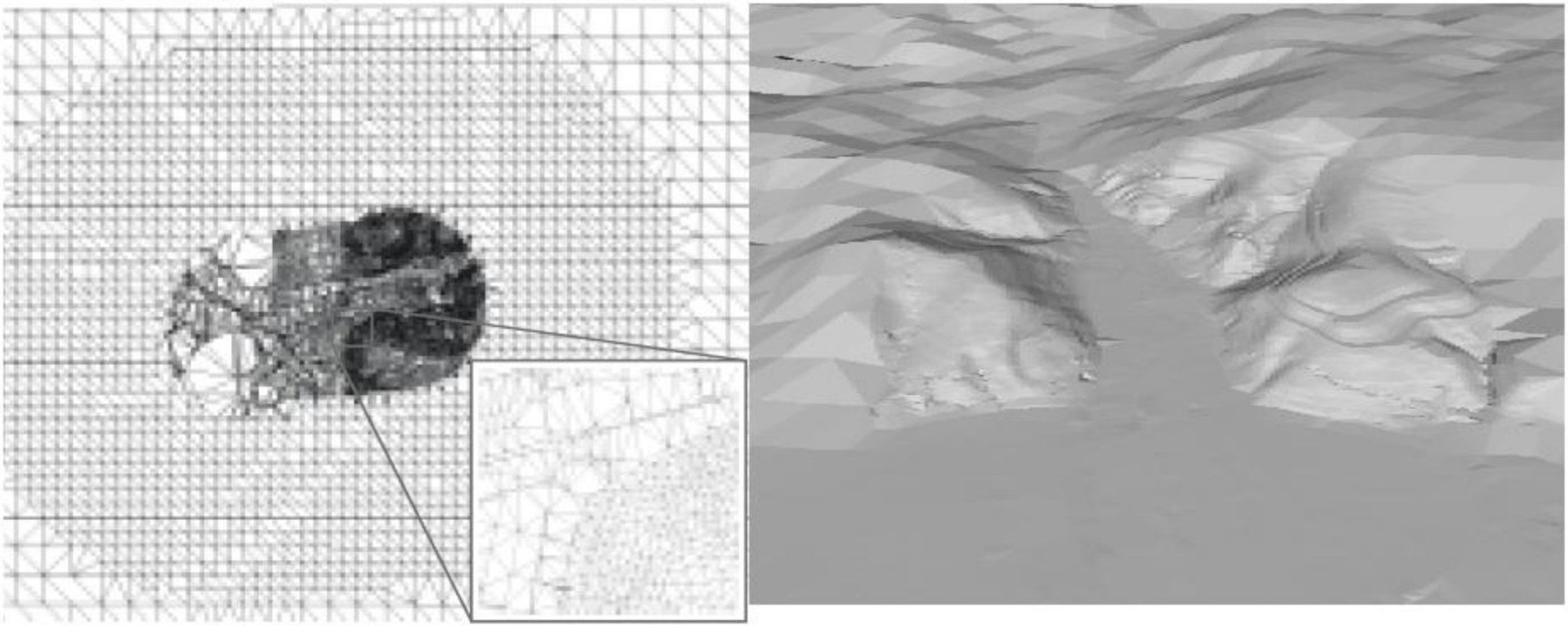
Currently: not defined! Depends on implementation

Parameter SELECTION_METHOD=<string> *(“intersection” | “by_center” | “crop”)*

Values:

- a) **intersection**: default operation, select features that touch, or intersect or are within BOUNDS
  - -> useful for static maps
- b) **by_center**: select features with center point being within BOUNDS
  - -> useful for selecting buildings displayed on a virtual globe
- c) **crop**: modify geometry of selected features (by intersection method) so that parts lying outside BOUNDS are cut away
  - -> useful for focus maps or subsets of terrain

New GetScene Parameters in 0.3.1



Focus Map: Multi-Resolution Terrain

New GetScene Parameters in 0.3.1

Styled Layer Descriptors (SLD)

- Enables user-defined Styling. Styles are defined as Styled Layer Descriptors. The Symbology Encoding (SE) needs extensions in order to style 3D objects.

One of 3 alternative parameters possible:

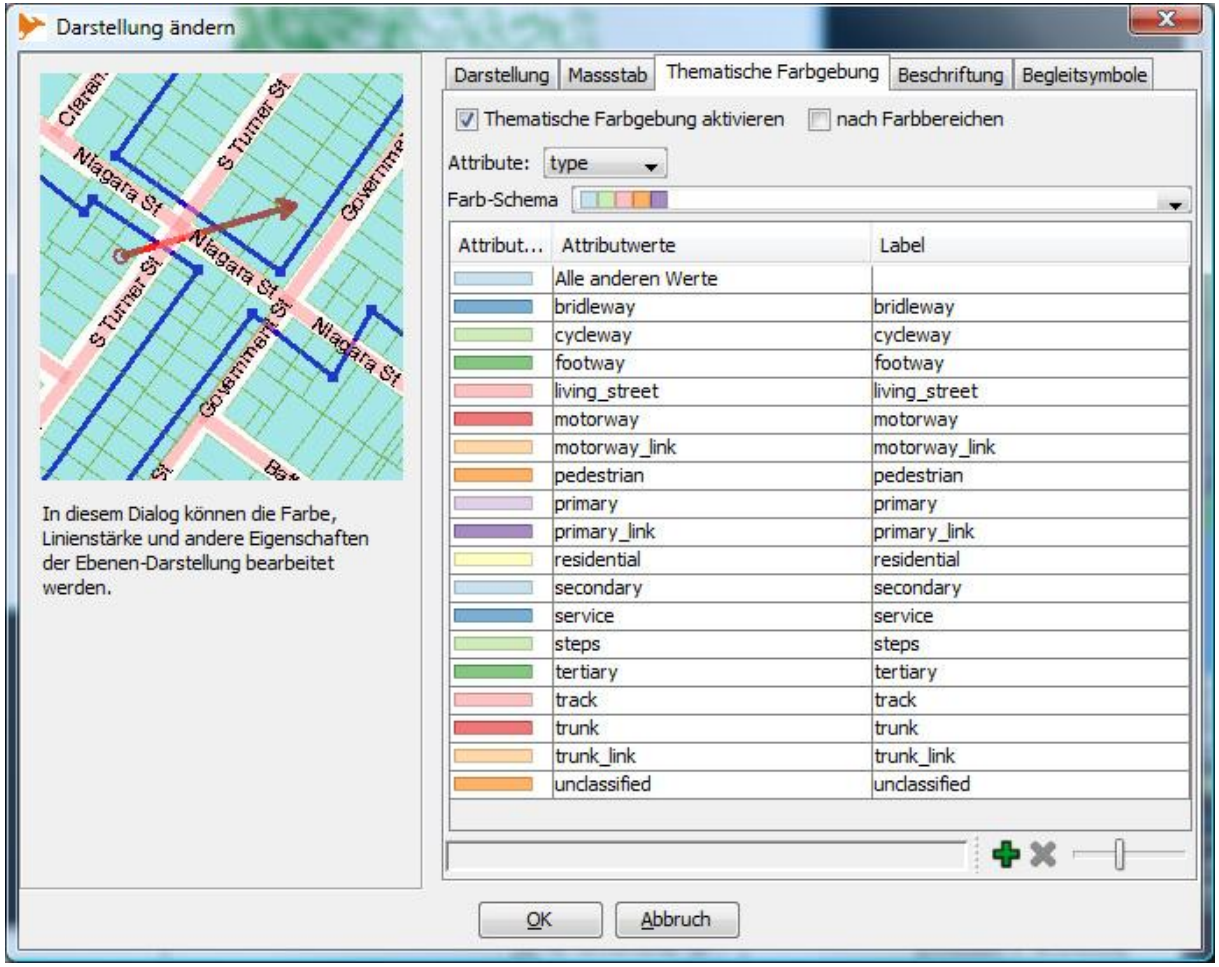
Parameters: SLD=<*string*>: *URL reference to SLD document*
SLD_BODY=<*string*>: *inline SLD Document in GET request*
StyledLayerDescriptor=<xml>: inline SLD Document in POST request

Defined in SLD profile of the WMS IS (OGC 05-078r4)

GetFeatureInfo Request

The GetFeatureInfo operation is designed to provide clients of a W3DS with more information about features within a scene that is currently displayed.

GetLayerInfo Request



Style Editor in OpenJUMP

universitätbonn Geographie

GetLayerInfo Request

The purpose of the GetLayerInfo request is to collect information on the available attribute names and the values in the attribute table of a specific layer

Table 4 — Parameters of the GetLayerInfo request

URL parameter	Required/ Optional/ Conditional	annotation
VERSION=0.3.1	M	Request version.
REQUEST=GetLayerInfo	M	Request name.
LAYER=<*layer*>	M	One Layer
COLUMNNAME=<column list>	O	Comma-separated list of one or more column to be queried.
FORMAT=output_format	M	Return format of feature information (MIME type).

GetLayerInfo Request

GetLayerInfo request:

http://www.myserver.de/W3DS?REQUEST=GetLayerInfo&VERSION=0.3.1&LAYER=Terrain&FORMAT=text/xml

GetLayerInfo response:

```
<GetLayerInfo>
    <Layer>
        <Name>Terrain</Name>
        <Attribute>
            <Name>id</Name>
        </Attribute>
        <Attribute>
            <Name>landuse</Name>
        </Attribute>
    </Layer>
</GetLayerInfo>
```

GetLayerInfo Request

GetLayerInfo request:

http://www.myserver.de/W3DS?REQUEST=GetLayerInfo&VERSION=0.3.1&LAYER=Terrain &**COLUMNNAME=landuse**&FORMAT=text/xml

GetLayerInfo response:

```
<GetLayerInfo>
    <Layer>
        <Name>Terrain</Name>
        <Attribute>
            <Name>landuse</Name>
            <Values>
                <Value>Bahn</Value>
                <Value>Baubloecke</Value>
                <Value>Gruenflaechen</Value>
                <Value>null</Value>
                <Value>Strassen</Value>
                <Value>Waldflaechen</Value>
                <Value>Wasserflaechen</Value>
            </Values>
        </Attribute>
    </Layer>
</GetLayerInfo>
```

Information Retrieval Concepts

W3DS is a Portrayal Service

- Result is optimized for efficient visualization
- no guarantee on the internal structure of the scene
- Access to attribute and meta data through additional server requests (GetFeatureInfo operation, OpenLS geocoder, directory service)
-Attribute are stored as tables, classic GIS Feature concept
-All CityGML tags can be transformed into attribute tables (including ID, address, object key, etc.)

OWS-6 Testbed DSS Thread

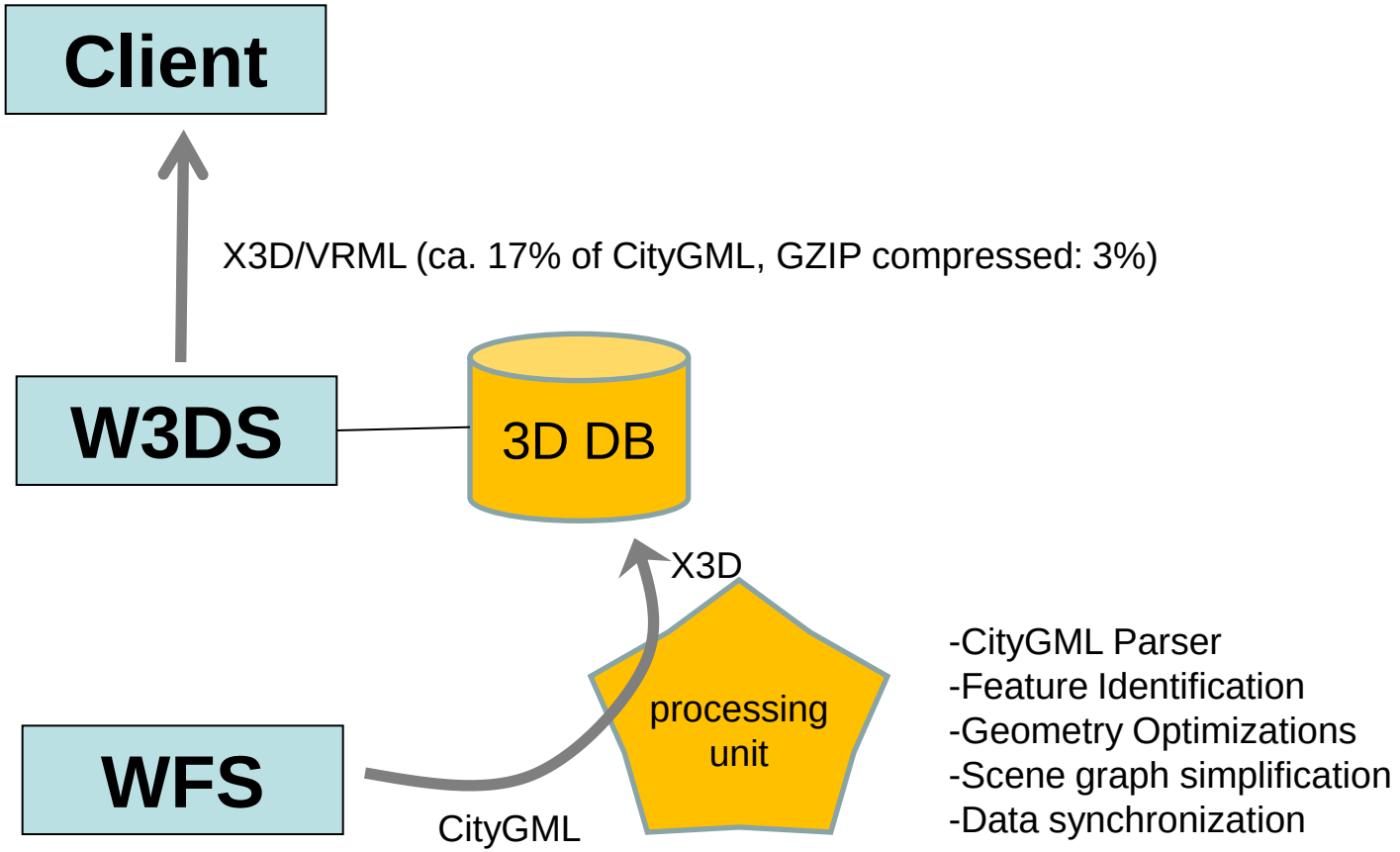
Scope

The focus for DSS in OWS-6 builds on portrayal, WMS Tiling, and integrated client work from OWS-3, OWS-4 and OWS-5, with additional work on 3D visualization and integration of the built environment and landscape. This thread will encompass these capabilities and task areas:

- ISO 19117 and OGC SLD Portrayal
- 3D Portrayal of GML with Fly-through
- Outdoor and indoor 3D route services
- WMS performance (tiling)
- Integrated Client for multiple OWS services

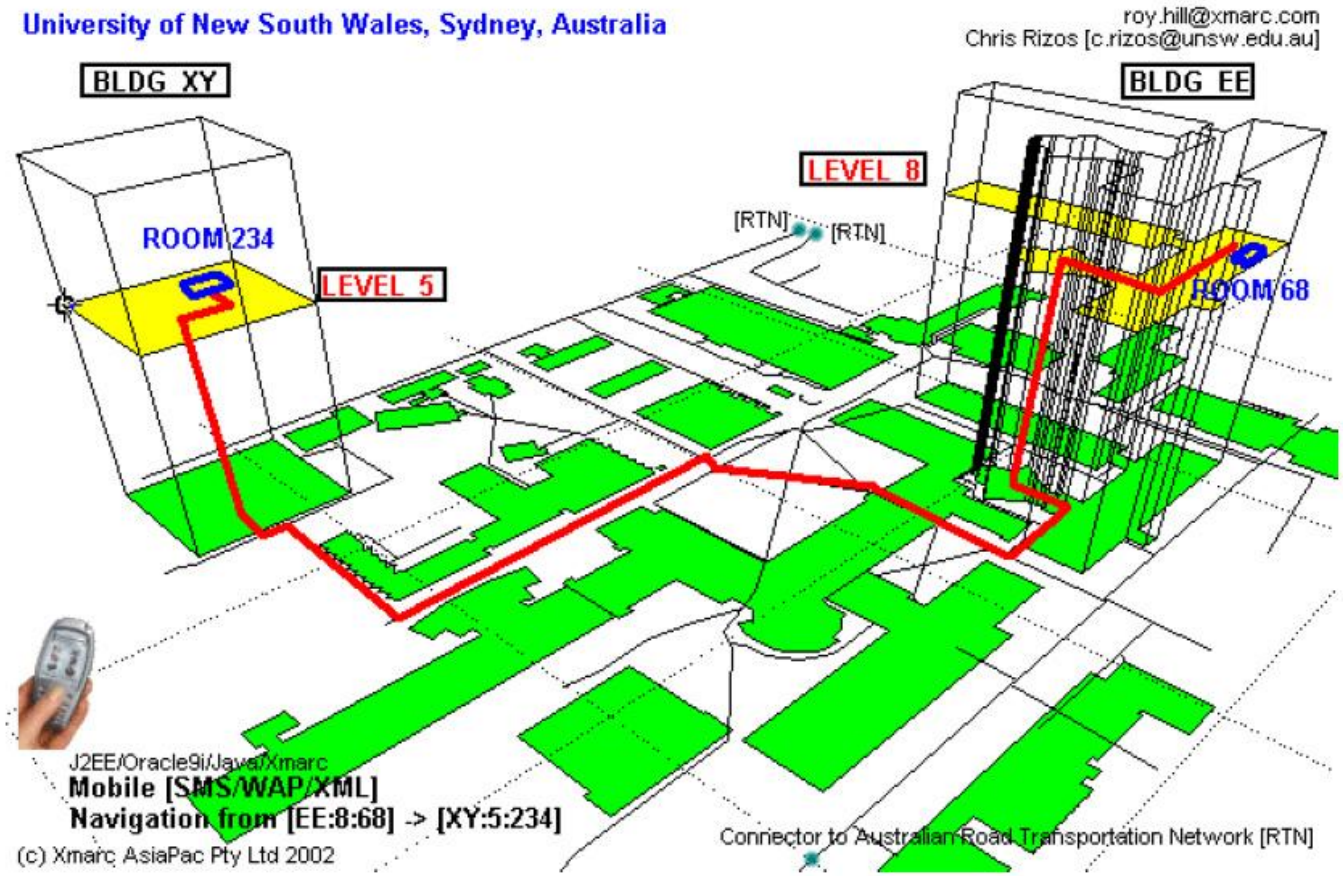
OWS-6 Testbed DSS Thread

Portrayal of CityGML with 3D Fly-through



OWS-6 Testbed DSS Thread

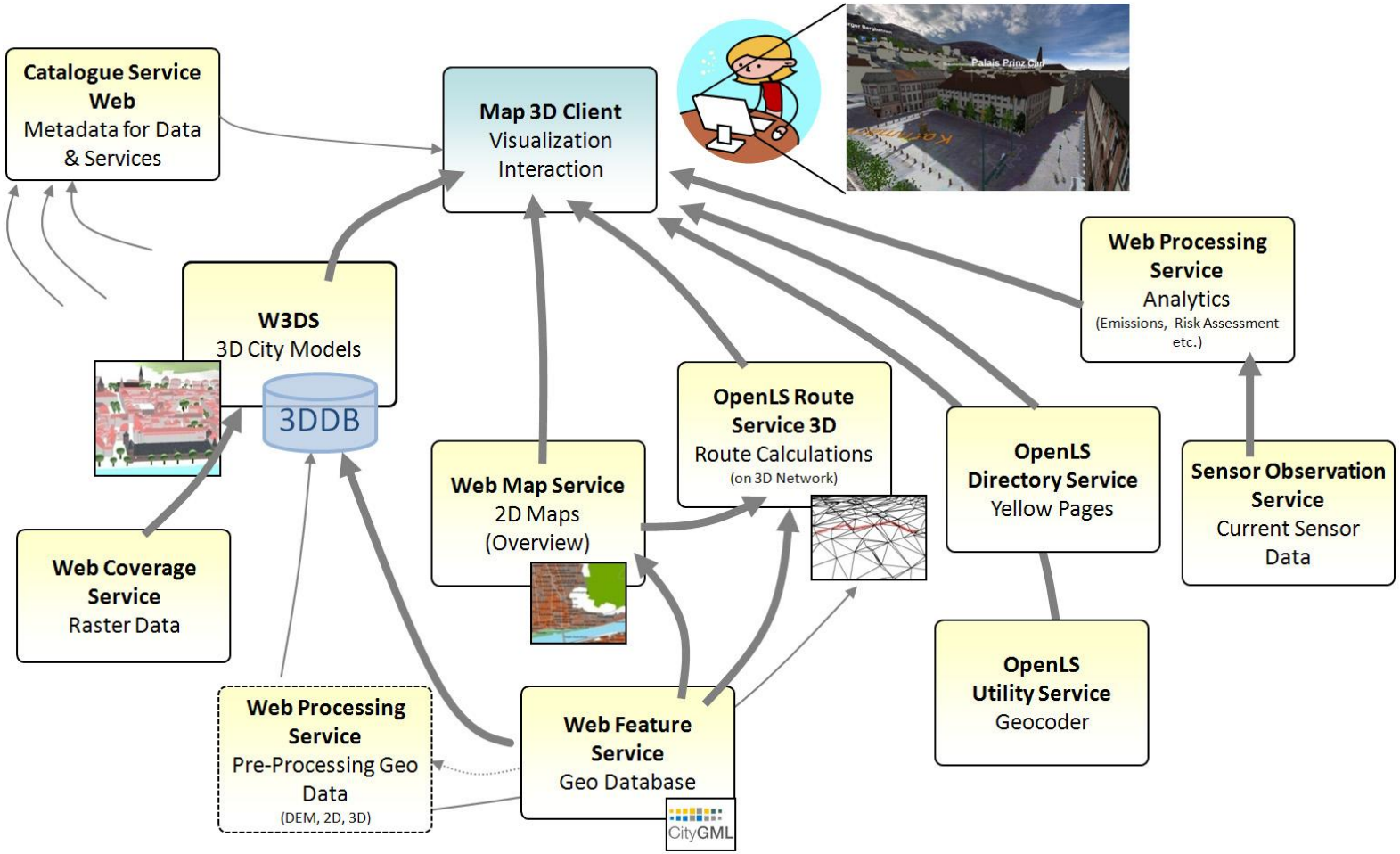
3D Indoor and Outdoor Routing



universitätbonn Geographie

OWS-6 Testbed DSS Thread

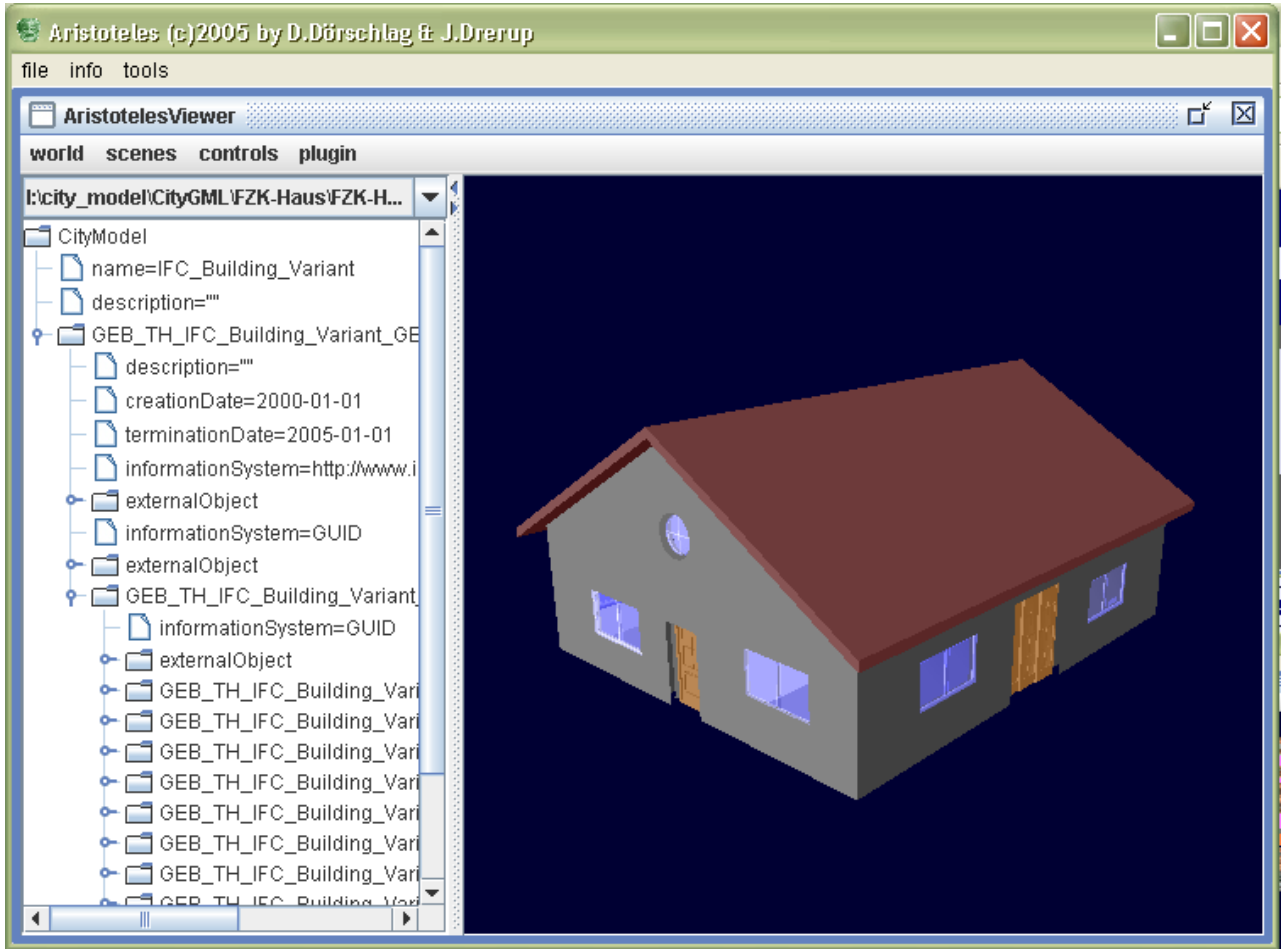
Integrated Client for multiple OWS Services: XNavigator (Uni Bonn Karto)



universitätbonn Geographie

OWS-6 Testbed DSS Thread

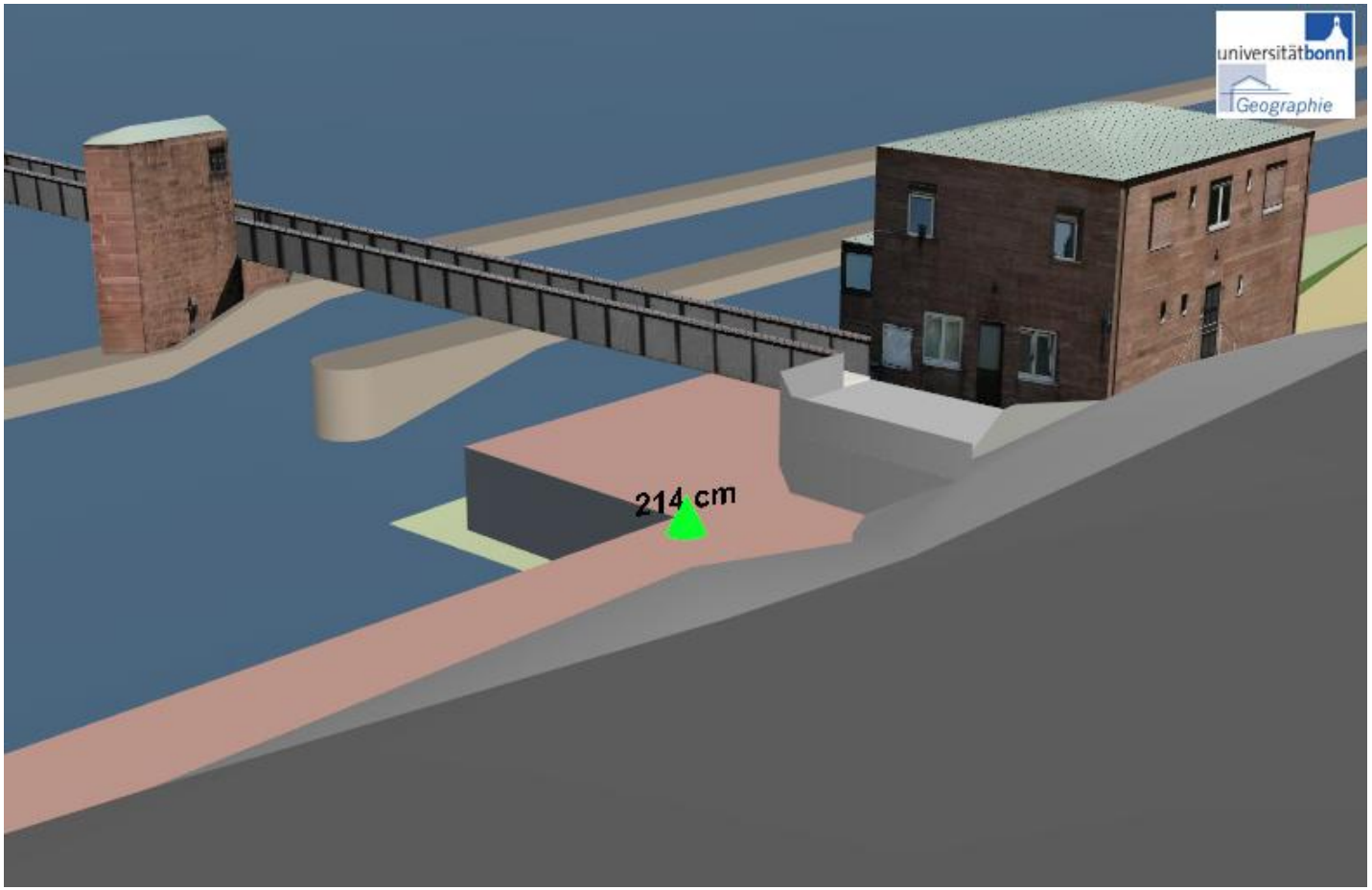
Integrated Client for multiple OWS Services: Aristoteles (Uni Bonn IGG)



universitätbonn Geographie

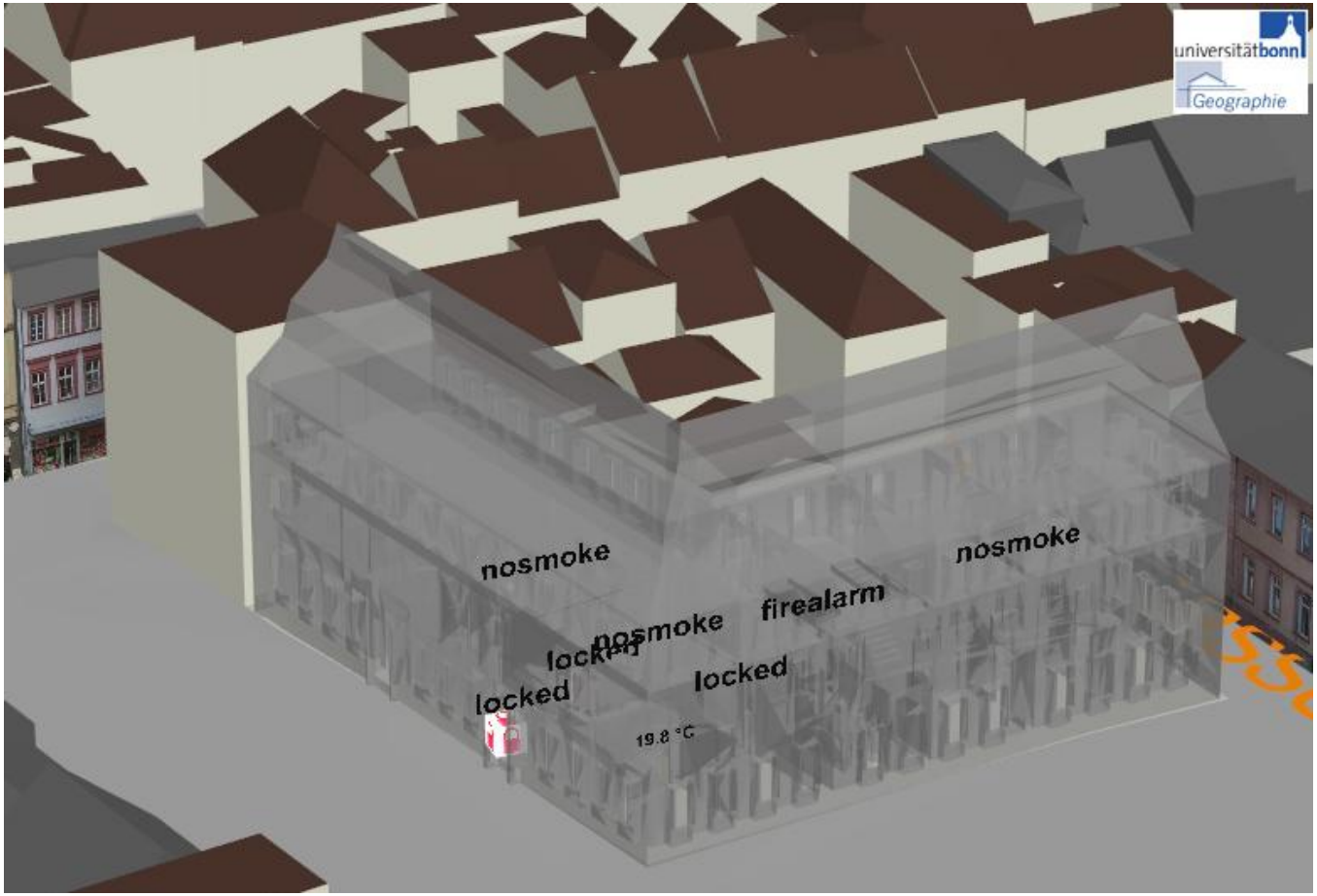
OWS-6 Testbed DSS Thread

Sensor Data Integration



OWS-6 Testbed DSS Thread

Sensor Data Integration



The End

Thank you for your Attention