ORACLE®

Oracle Spatial 11*g*

Dr. Siva Ravada

Oracle Spatial 11g Features

- 3D Support
- Spatial Web Services
- Java API
- GeoRaster
- Network Data Model
- Workspace Manager

3D Support

3-D Spatial

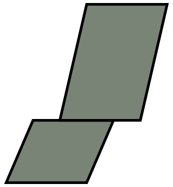
- Address growing number of 3-d applications
- GIS
  - City Planning
  - Property rights
- City Modeling (City GML)
- Business Intelligence
  - Real estate
  - Advertising
- VR, Medical Applications

3-D Support in Oracle Spatial

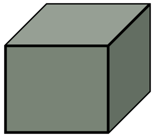
- 3D Types for points/lines/polygons/solids
- Specialized types for large volumes of 3D point data
  - Represent scenes as a set of 3-d points obtained using laser scanners etc. (point clouds)
  - surface representation using TINs
- 3D Coordinate System Support

SDO_GEOMETRY for 3D Data

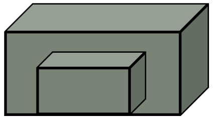
- Points
- Lines
- Simple Surfaces
  - All points of a surface lie in a 3D plane
  - A 3 point 3D polygon is the simplest surface
  - A simple surface can have any polygonal shape
- Composite surfaces
  - has one or more connected simple surfaces
  - It can be closed or open
  - The simple surfaces in a composite surface can not cross each other
  - surface of a cube is an example of a composite surface
    - Cube has six simple surfaces
    - Each simple surface is a 3D square

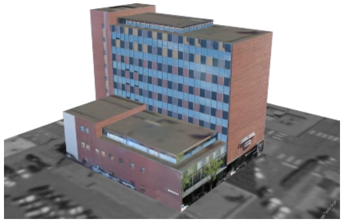


SDO_GEOMETRY for 3D Data



- Simple Solids
  - Solids are composed of closed surfaces
  - It has to have one outer surface and one or more interior surfaces
  - Cube is an example of a simple solid
  - A pyramid is another example of a simple solid
- Composite Solids
  - Consists of n simple solids as a connected solid
  - Can be represented as a simple solid with a composite surface
  - Topologically there is an equivalent simple solid, but the composite solid representation is easier
    - Example: A building composed of rooms





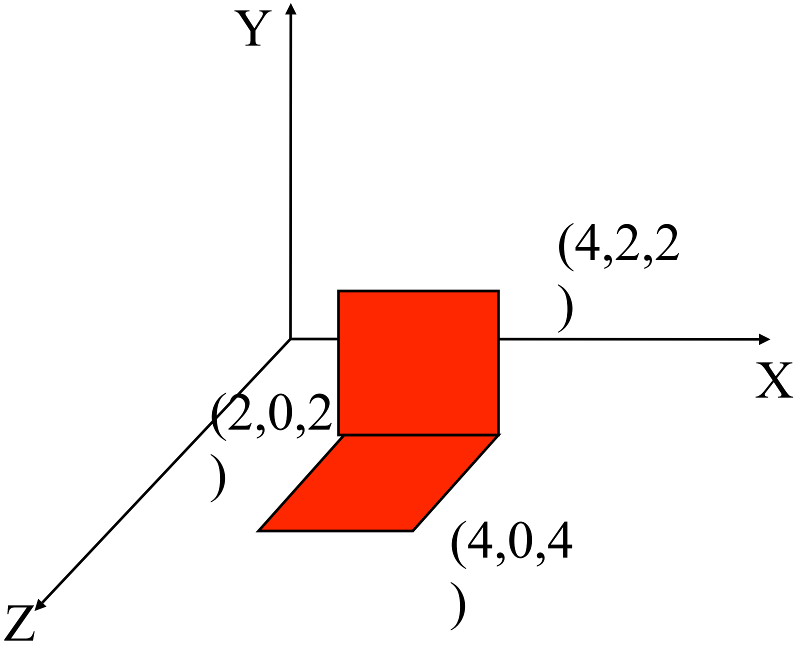
SDO_GEOMETRY for 3-D Data

- Support for multi-points, multi-lines, multi-surfaces, multi-solids
  - Multi-surface is different from composite surface
  - Multi-solid is different from composite solid
- No arcs supported
- No parametric surfaces supported
- Follows GML3.1.1, ISO 19107 Specifications

3D SDO_GEOMETRY: GTYPE, ETYPE combinations

Name	GTYPE	ETYPE	Interpretation
Point	3001	1	1 (1 point)
Linestring	3002	2	1 (straight line)
Planar Polygonal Surface	3003	1003 (outer) or 2003 (inner)	1 (straight lines)
Composite Surface	3003	1006	N: number of planar-polygon surfaces that follow.

Composite Surface



SDO_GEOMETRY:

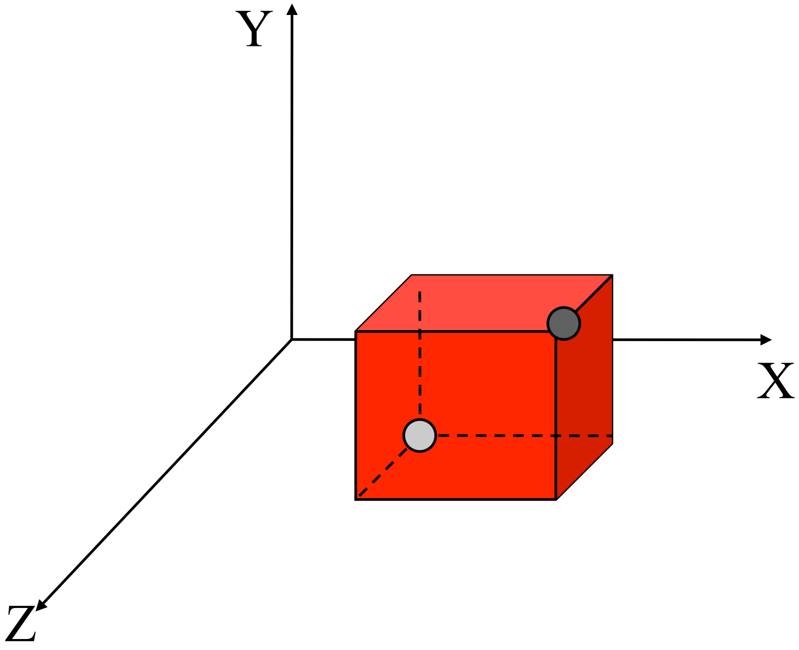
3003 – 3-dimensional surface,
SRID,
NULL,
SDO_ELEM_INFO_ARRAY
(
1 – starting offset,
1006 – composite surface
2 – 2 elements for this comp,
1,1003,3 – first element rectangle
7, 1003, 3 – second rectangle
),
SDO_ORDINATE_ARRAY(
2,0,2, 4,2,2, 2,0,2, 4,0,4)

3d SDO_GEOMETRY: Gtypes,Etypes for Solids

Normals for a solid always point outward: vertices specified accordingly

Name	GTYPE	ETYPE	Interpretation
Solid	3008	1007 (no 2007)	1 Followed by: 1 outer surface and N inner surfaces
Solid Box	3008	1007 (no 2007)	3: Axis aligned box Specified by the min x,y,z and max x,y,z
Composite Solid	3008	1008	N: number of ‘composing’ solid elements followed by descriptions of each. Composite solid defines a single volume.

Solid Box Example

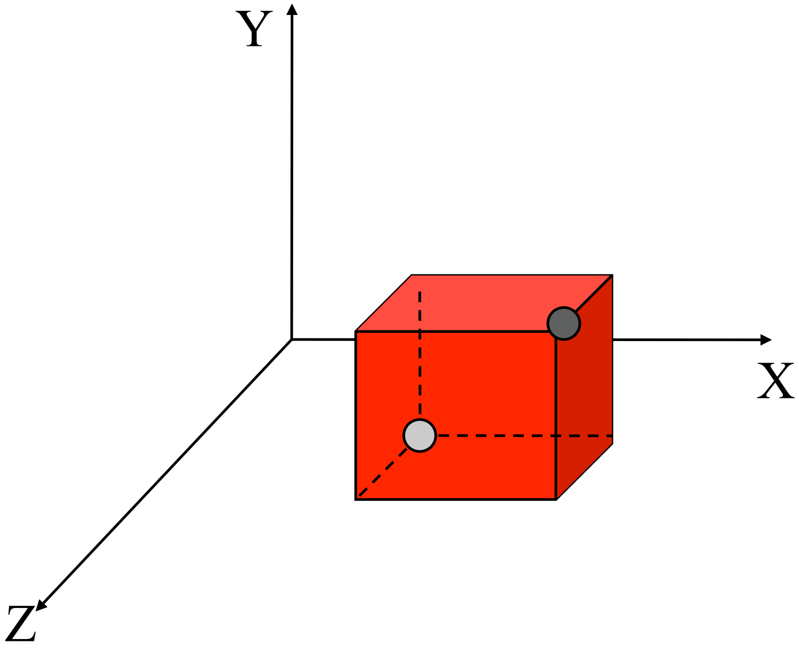


○ (2,0,2
)

● (4,2,4
)

```
SDO_GEOMETRY:
        3008 – 3-dimensional solid,
        SRID,
        NULL,
        SDO_ELEM_INFO_ARRAY
        (
          1 – starting offset,
          1007 – solid element
          3 – Axis-aligned box,
        ),
        SDO_ORDINATE_ARRAY(
          2,0,2, -- first end point
          4,2,4 – second endpoint)
```

Solid As a Closed Surface



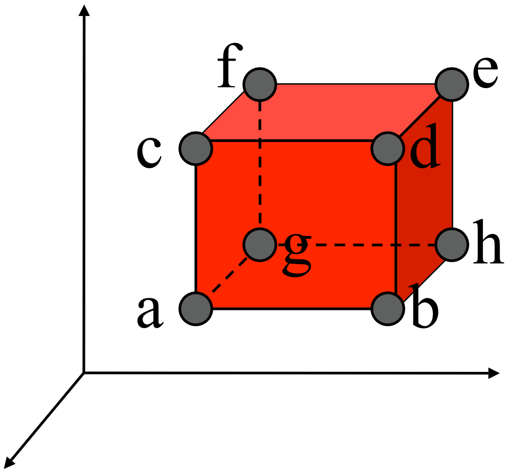
○ (2,0,2)

● (4,2,4)

SDO_GEOMETRY:

3008 – 3-dimensional solid,
SRID,
NULL,
SDO_ELEM_INFO_ARRAY
(
1 – starting offset,
1007 – solid element
1 – BREP solid,
1, 1006, 6
– 1 exterior surface w 6 faces
…),
SDO_ORDINATE_ARRAY(
ordinate specification for each
face – next slide)

Solid Example



Orientation of faces for exterior surface:
-- Such that normal points outward the solid

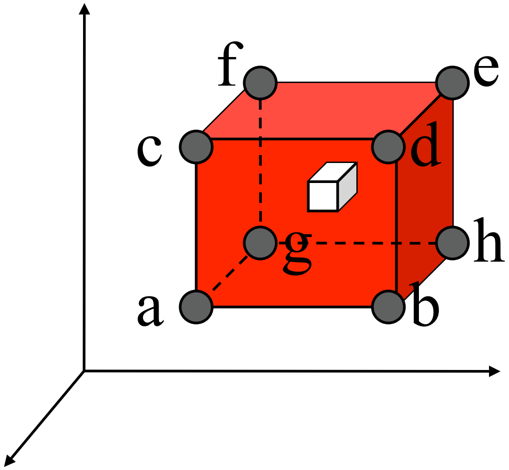
-- Example:

Face (abcd): ordinates for a,b,d, c (ccw)
Face (efgh): ordinates for e,h,g,f (cw)

-- Likewise:

Face (cdef): ordinates for c,d,e,f (ccw)
Face (bdeh): ordinates for b,h,e,d (ccw)
Face (abgh): ordinates for a,g,h,b (cw)
Face (acfg): ordinates for a,c,f,g (cw)

Solid with a Hole Example



Interior Hole:
(aH,bh,…hH)

Orientation of faces for interior surface:
-- Reverse as that of the exterior surface
-- Such that normal points outward the solid (I.e., into the hole/void)
-- Example:
Face (aHbHcHdH):
ordinates for aH,cH,dh, bH (**cw**)
-- No face of the interior hole touches to a face of the exterior in more than 2 points.

3D SDO_GEOMETRY: Collections

Name	GTYPE	Element_Info_array has descriptions of
Multi-Point	3005	N disjoint points
Multi-linestring	3006	N linestrings
Multi-surface	3007	N disjoint surfaces (may touch at points)
Multi-solid	3008	N disjoint solids (may share a point, edge)
Collection	3004	Disjoint set of points, lines, surfaces, and/or solids (heteregenous collection)

Operations on 3D SDO_GEOMETRY

- Spatial Indexing: 3D R-trees
- SQL Operator support
  - SDO_FILTER, SDO_ANYINTERACT, SDO_WITHIN_DISTANCE, SDO_NN only
- PL/SQL Functions
  - VALIDATE_GEOMETRY
  - VALIDATE_LAYER
  - SDO_DISTANCE,
  - SDO_GEOM.ANYINTERACT
  - SDO_AREA, SDO_VOLUME
  - SDO_CS.TRANSFORM

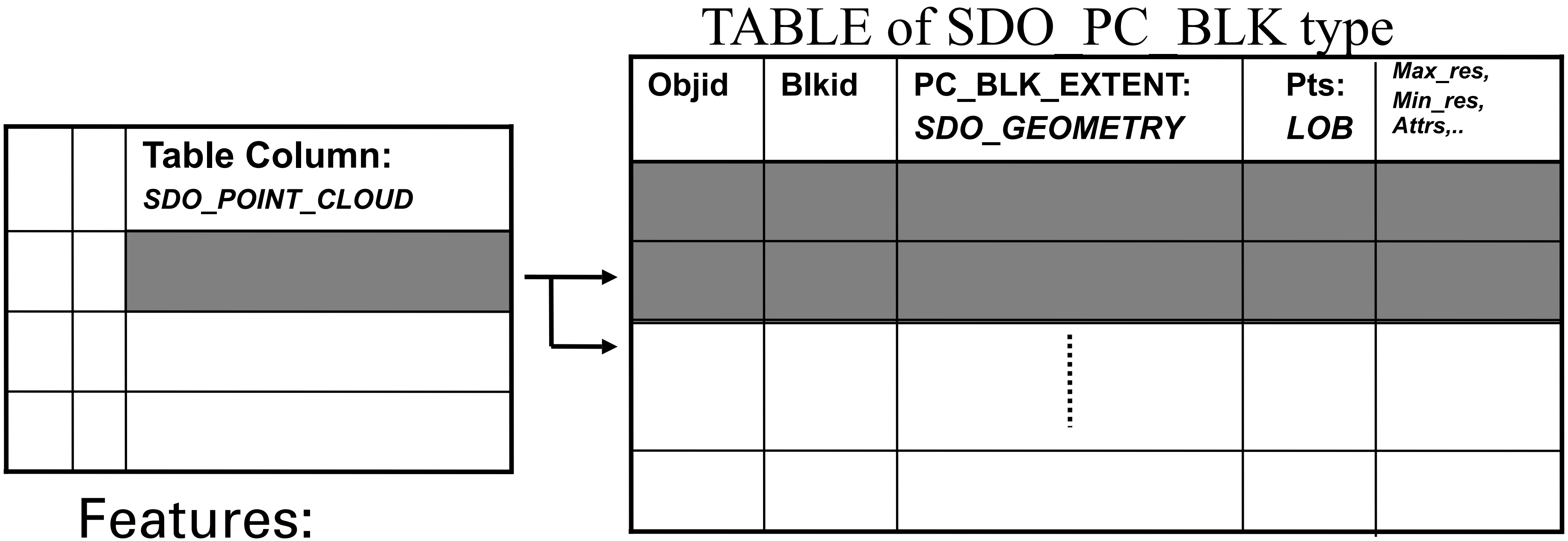
J3D_Geometry: Java Interface

- Extends JGeometry class
- Supported Methods:
  - Validate(tolerance): Returns true or false
  - Distance(J3D_geometry): Returns distance
  - Anyinteract(J3D_Geometry): Returns true or false
  - Length(): returns the length of a line geometry
  - Area(): returns area for a surface geometry
  - Volume(): returns volume of a solid geometry

Specialized Types for High Density 3D Point Data

- Large volumes of point data acquired by sensors
  - LIDAR
  - Sensors used to collect data inside buildings
- Millions of points used to model a scene
- SDO_GEOMETRY is not suitable for such data
- POINT_CLOUD data type introduced to efficiently manage this type of point data
- Surface models can be generated from these point clouds

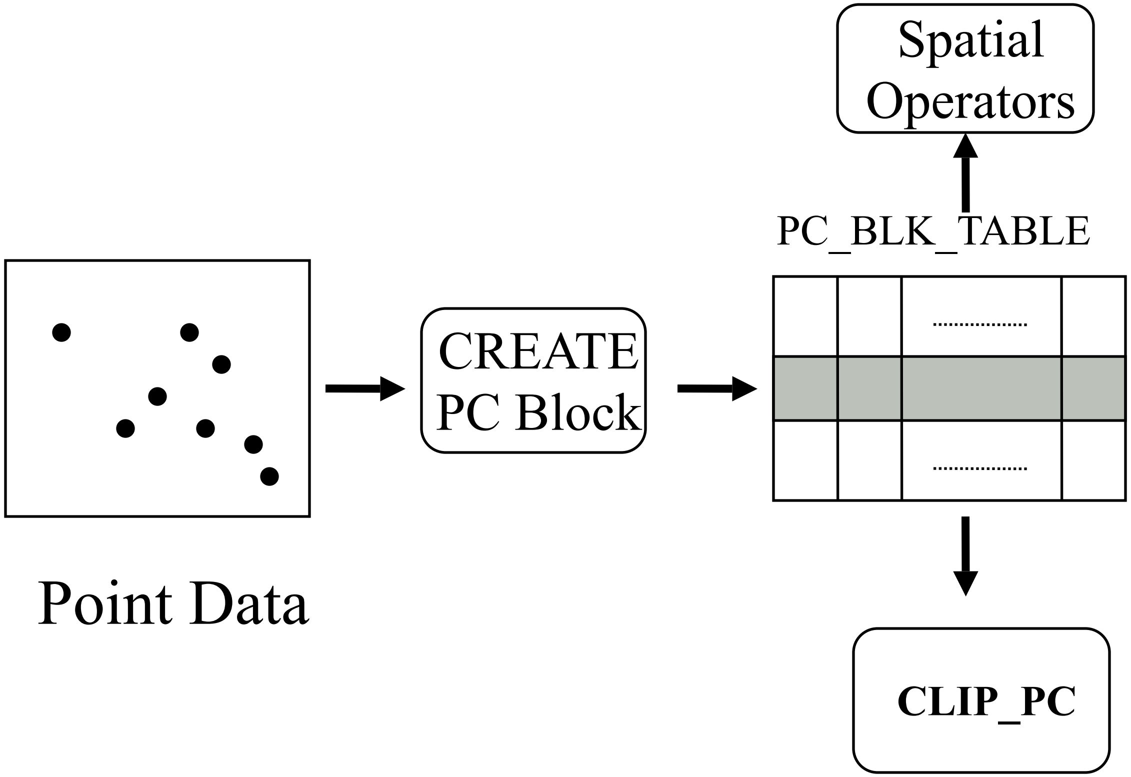
SDO_POINT_CLOUD in Oracle



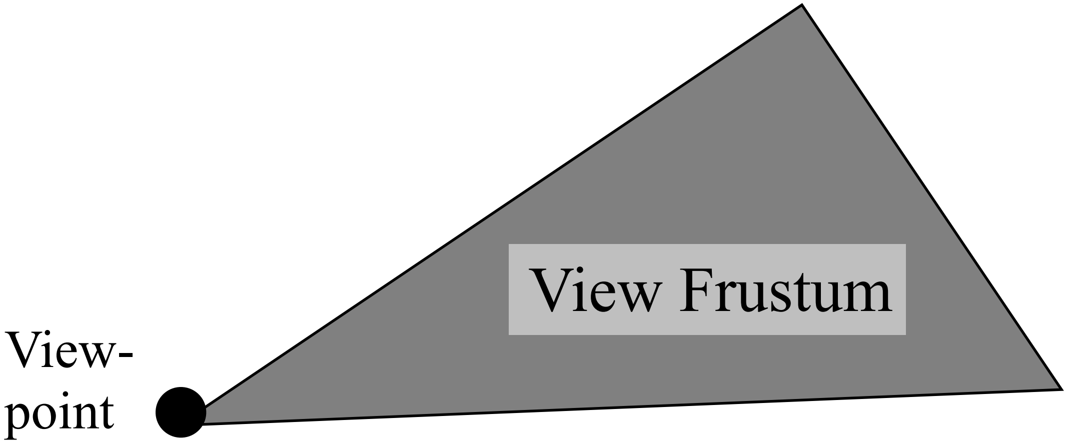
Features:

- Partition into multiple blocks for granularity of access
- Each “sdo_point_cloud” column is associated with “only 1” block table (of SDO_PC_BLK type).

Operations


Spatial Operators
PC_BLK_TABLE
CREATE PC Block
Point Data
CLIP_PC

Visibility Query



- Farther objects in lower resolution
- Nearer objects in higher resolution

Query specifies one or more

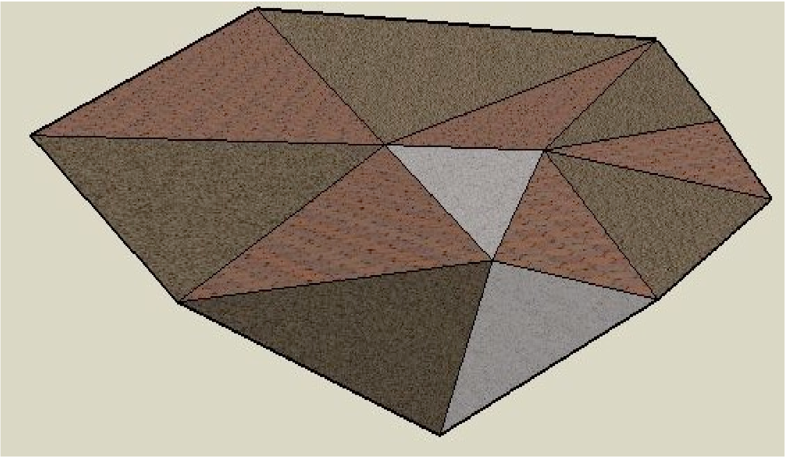
<query solid , [min_res, max_res]>

SDO_POINTCLOUD

- Operations
  - Creation of a point cloud using a set of points
  - Spatial query with <frustum window, interval range>
  - Read point cloud block as an SDO_GEOMETRY
  - System data management
    - Blk_table *automatically* cleaned up
      - When an sdo_point_cloud (row) is deleted from base_ table
      - When base_table (or column) is dropped
      - When base_table is truncated

TIN: Triangulated Irregular Network

- What is a TIN?
  - Vector-based topological data model used to represent terrain/surface
  - Contains a network of irregularly spaced triangles
  - 3D surface representation derived from irregularly spaced points
  - Each sample point has an x, y coordinate and a z value or surface value



Node No	X	Y	Z
1	5	6	3
2	3	6	5
3	1	5	6
4	4	4	4
5	6	5	3
6	2	2	2
.	.	.	.

SDO_TIN in Oracle

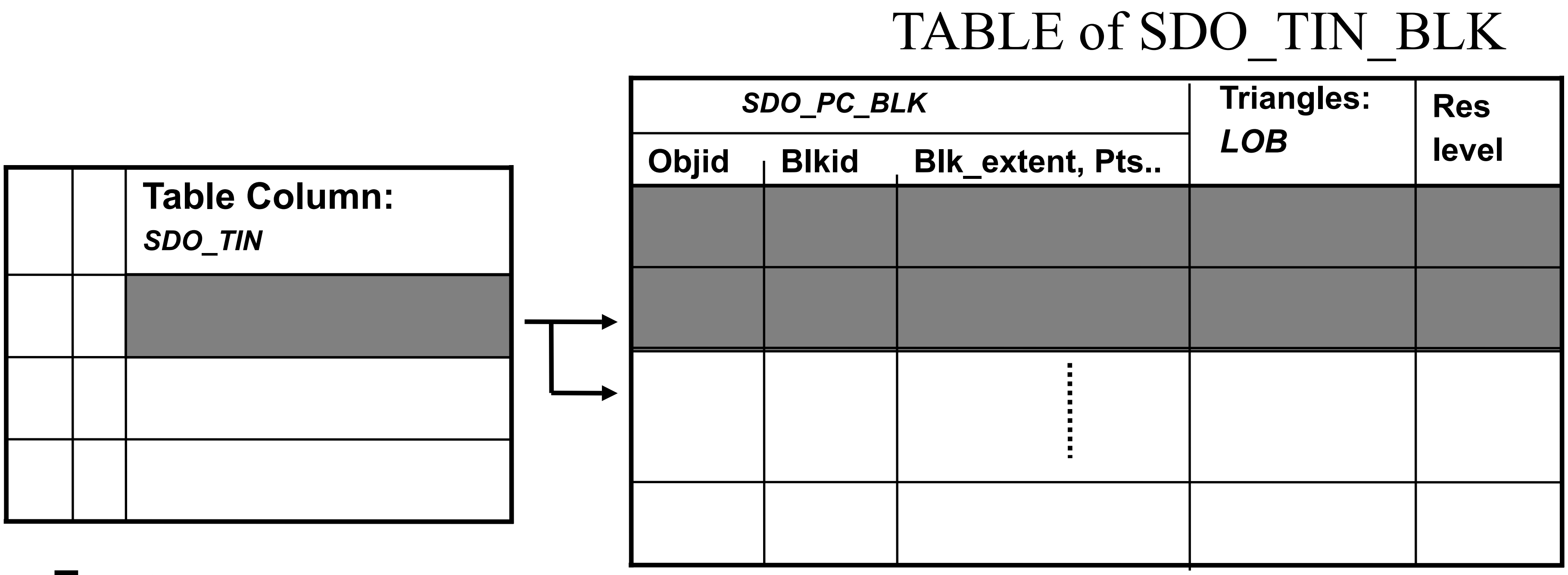


TABLE of SDO_TIN_BLK

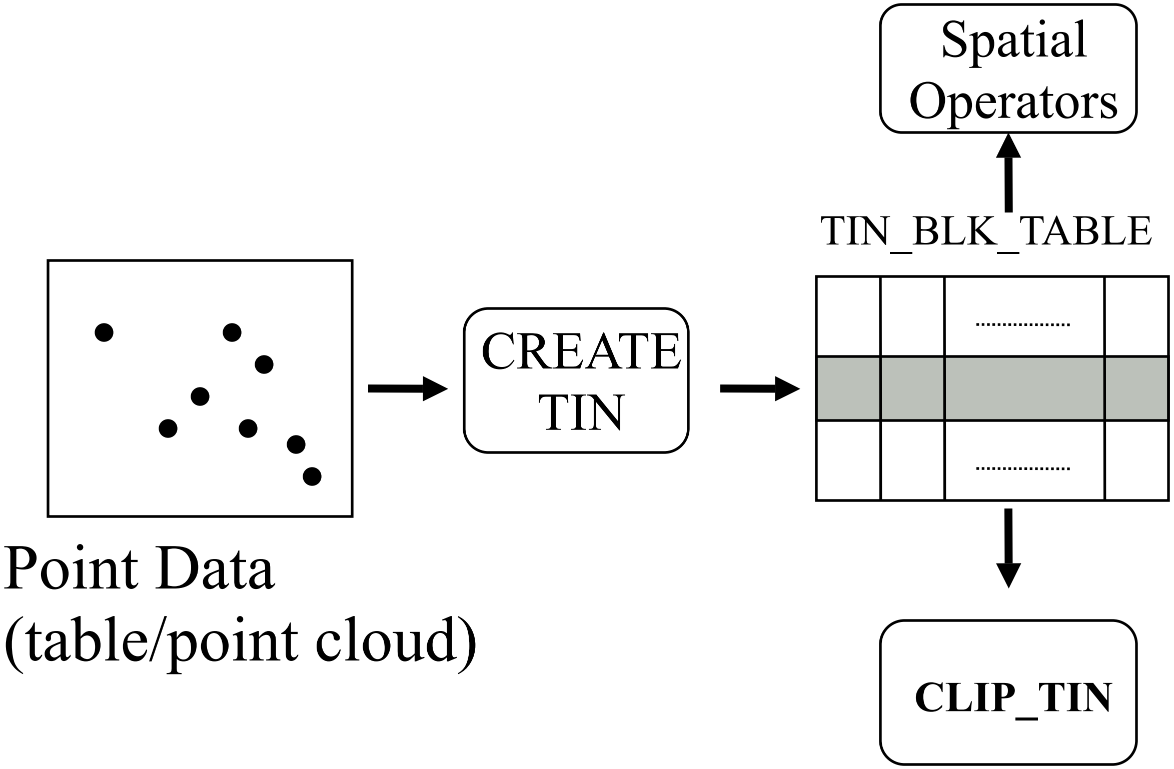
		Table Column: ***SDO_TIN***

SDO_PC_BLK			**Triangles:** ***LOB***	**Res** **level**
Objid	**Blkid**	**Blk_extent, Pts..**		
		⋮		

Features:

- Partition into multiple blocks for granularity of access
- Can store upto 4Billion*2Billion points
- Each “sdo_tin” column is associated with “only 1” block table (of SDO_TIN_BLK type)

Operations


Spatial Operators
TIN_BLK_TABLE
CREATE TIN
....
....
Point Data (table/point cloud)
CLIP_TIN

SDO_TIN

- Create TINs for millions of points
- Operations
    - Creation of a TIN using a set of points
    - Spatial query with <frustum window, interval range>
    - Read a TIN block as an SDO_GEOMETRY
    - Blk_table *automatically* cleaned up
        - When an sdo_point_cloud (row) is deleted from base_ table
        - When base_table (or column) is dropped
        - When base_table is truncated

Creating TINs

- CREATE_TIN
  - Initialized tin object
  - Input *table* or *view* of points
  - An optional output “pts” table (with addl. <ptn_id, pt_id> columns) to store the points in a clustered fashion
    - Typically an IOT pre-created by the user
    - Useful to query based on non-spatial attributes of data
  - An optional output “triangles” table to store the triangles in a clustered fashion
    - Users can add/associate triangle-specific information
    - Typically an IOT pre-created by the user

3D Coordinate System Functions

Same use as 2D Coordinate Systems:

A reference system for spatial operations

- Associate a coordinate system with 3D data
  - SDO_GEOMETRY
- Support transformations from one to another coordinate system
- Compute distances, and other spatial relationships between two objects within the same coordinate system

3D Coordinate Systems

Following EPSG types are being supported:

- Vertical Coordinate Systems (w.r.t to sea-level etc.): essentially 1-d coordinate system
- Geocentric: 3-d cartesian
- Geographic-2d: 2-d ellipsoidal
- Geographic-3d: 3-d ellipsoidal
- Compound Coordinate System

Spatial Web Services

Web Services

A Web Service is application or business logic that is accessible using standard Internet protocols.

Web Services represent black-box functionality that can be used and reused without regard to how the service is implemented.

Spatial Web Services: When & How

- Model 1: Spatial is primary focus of web service
  - Perform operations on spatial data
  - Incorporate real time information, personalization, and presence
  - Result may be spatial (e.g. a map, data in specified exchange format)
  - Often it’s a report, or next step in a business process
  - Example: Return driving directions between two addresses

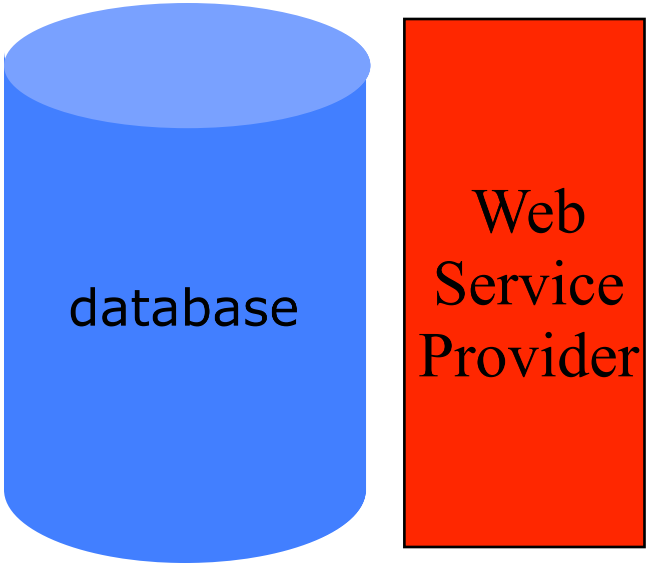
Spatial Web Services: When & How

- Model 2: Business web service with a spatial component
  - Perform business search, query or workflow function
  - Spatial is not the primary focus of the service
  - Incorporate operational, financial, real time information
  - Result likely to be a report, or next step in a business process
  - May be spatial (e.g. a map, data in specified exchange format)
  - Example: Compute the sum of all sales within a particular sales region

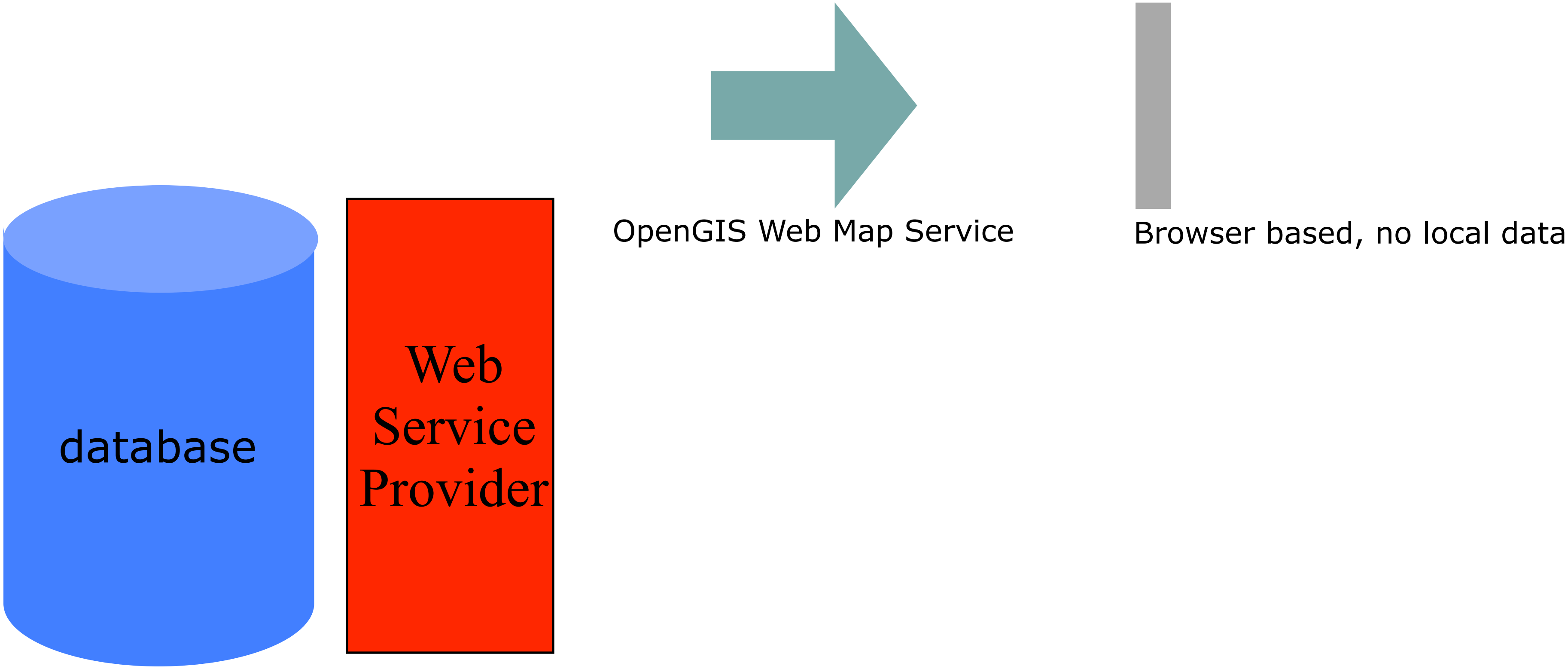
Common spatial web services

- Locator Services
  - Address finder (geocoding/reverse geocoding)
  - Route finder (driving directions)
- Mapping Services
  - Display base map
  - Overlay theme on base map
- Query Services
  - Find POI within user specified distance
  - Find closest POI to user specified location

Web Services **use cases**



Web Services **use cases**



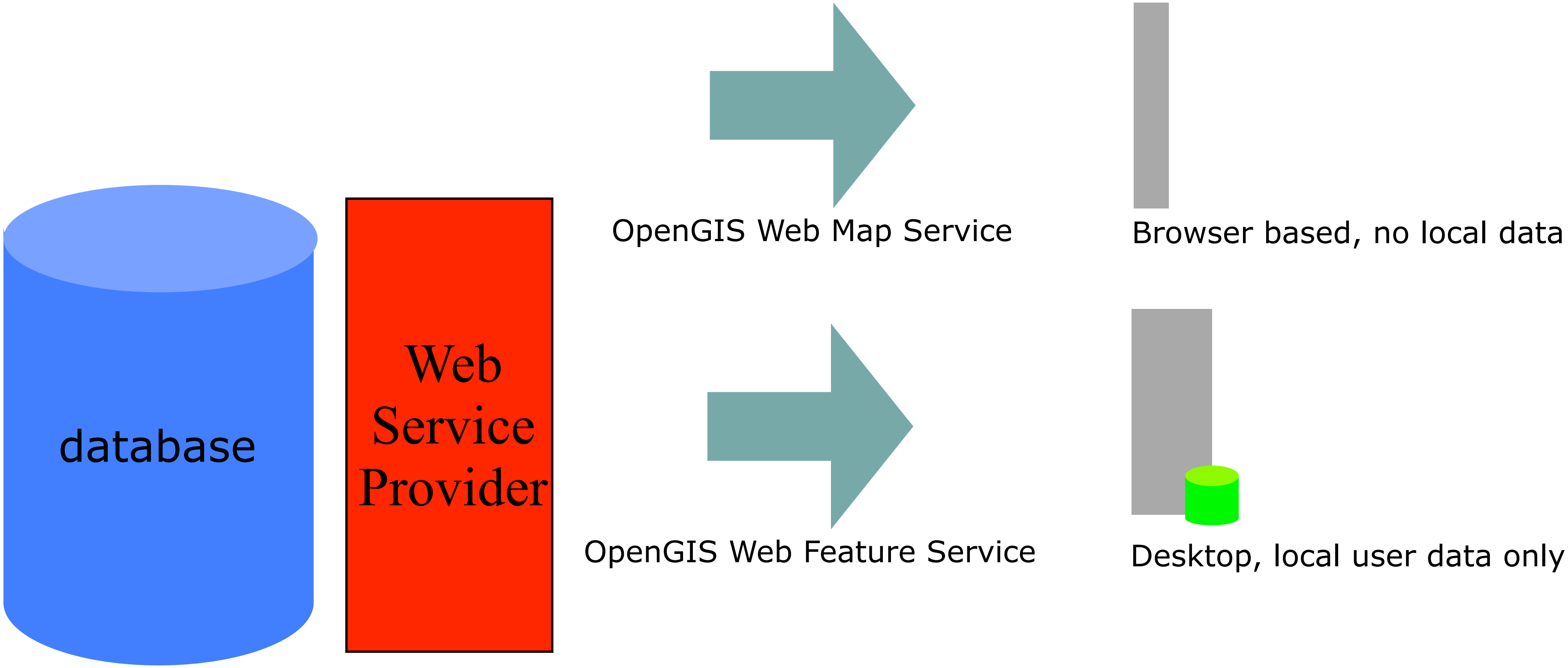
database

Web Service Provider

OpenGIS Web Map Service

Browser based, no local data

Web Services **use cases**



database

Web Service Provider

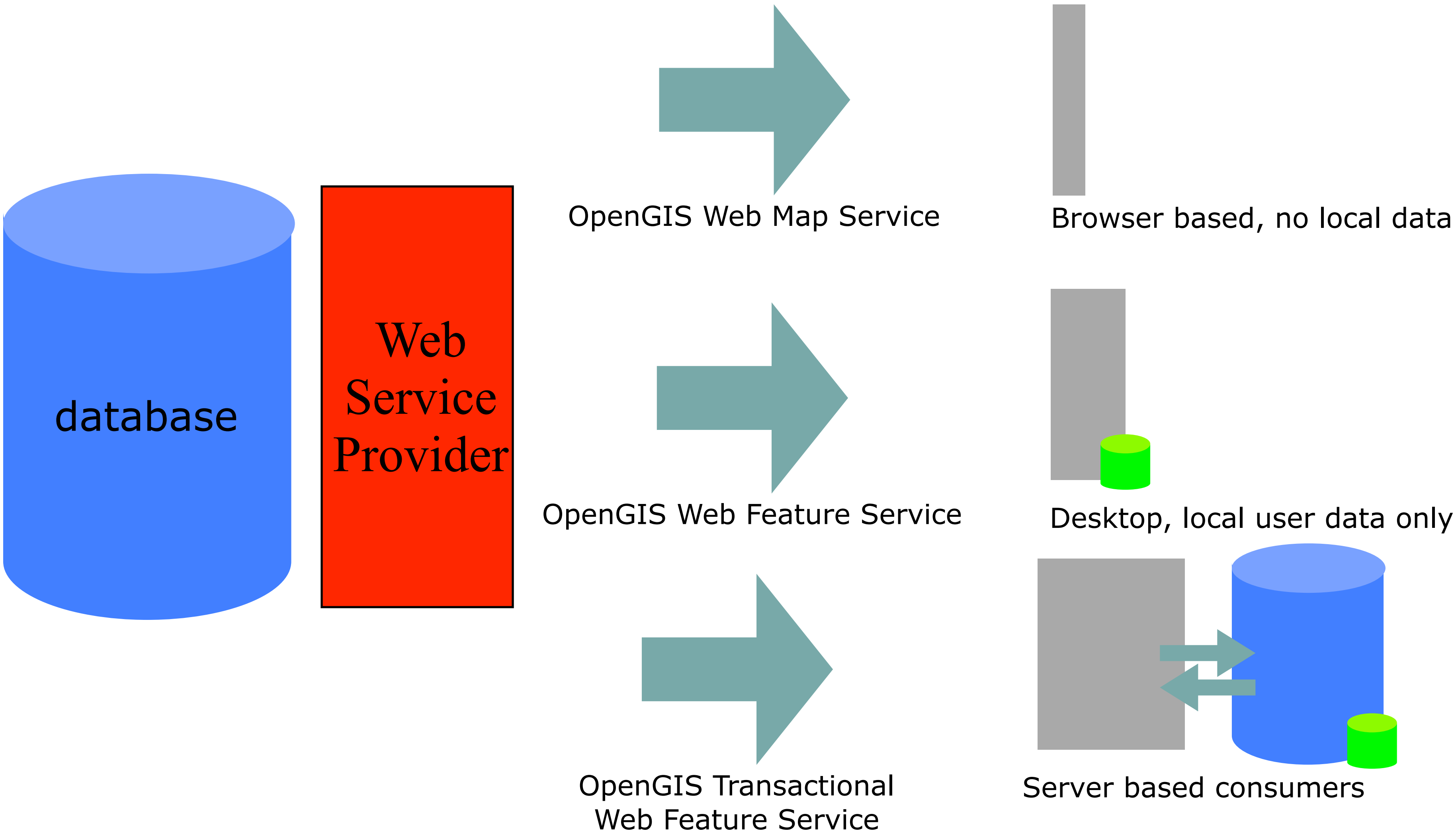
OpenGIS Web Map Service

Browser based, no local data

OpenGIS Web Feature Service

Desktop, local user data only

Web Services **use cases**



database

Web Service Provider

OpenGIS Web Map Service

Browser based, no local data

OpenGIS Web Feature Service

Desktop, local user data only

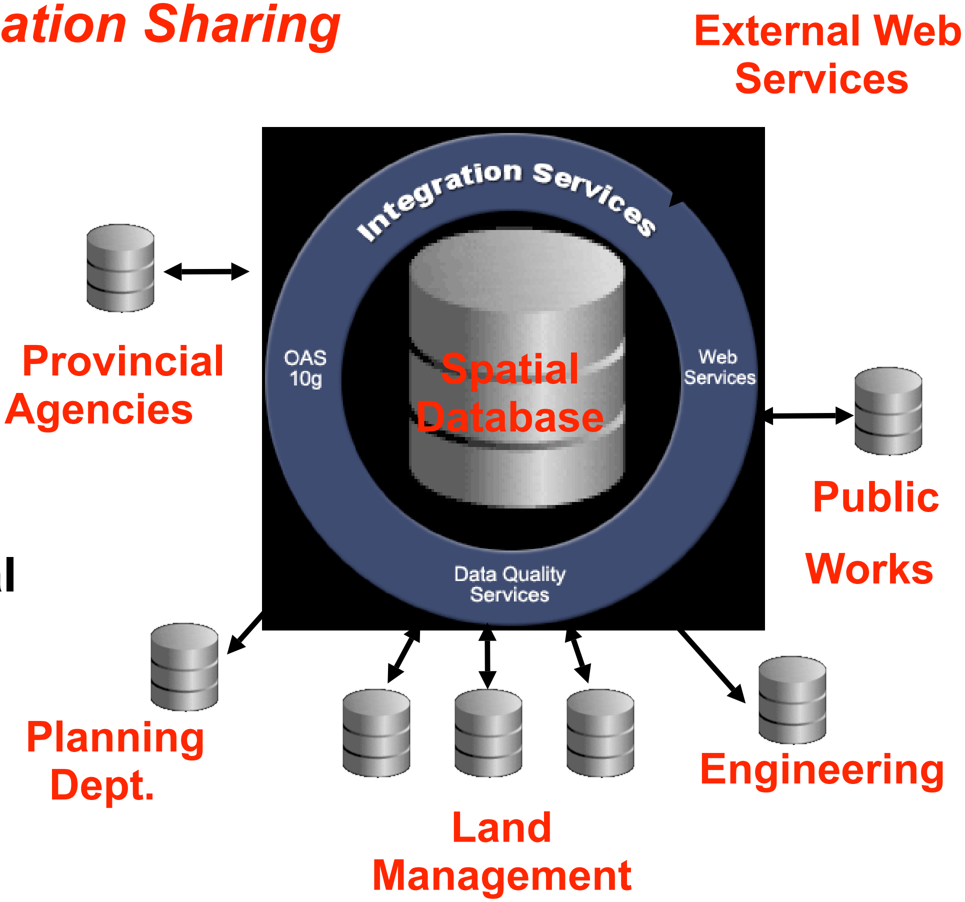
OpenGIS Transactional
Web Feature Service

Server based consumers

Consolidation Value Proposition

Integrated and Assured Information Sharing

- Single source of truth
- High Performance
- Strong security
- Centralized geoprocessing
- Centralized maintenance
- Enhanced business and operational intelligence
- Creation of a Web centric, spatially enabled, real time enterprise



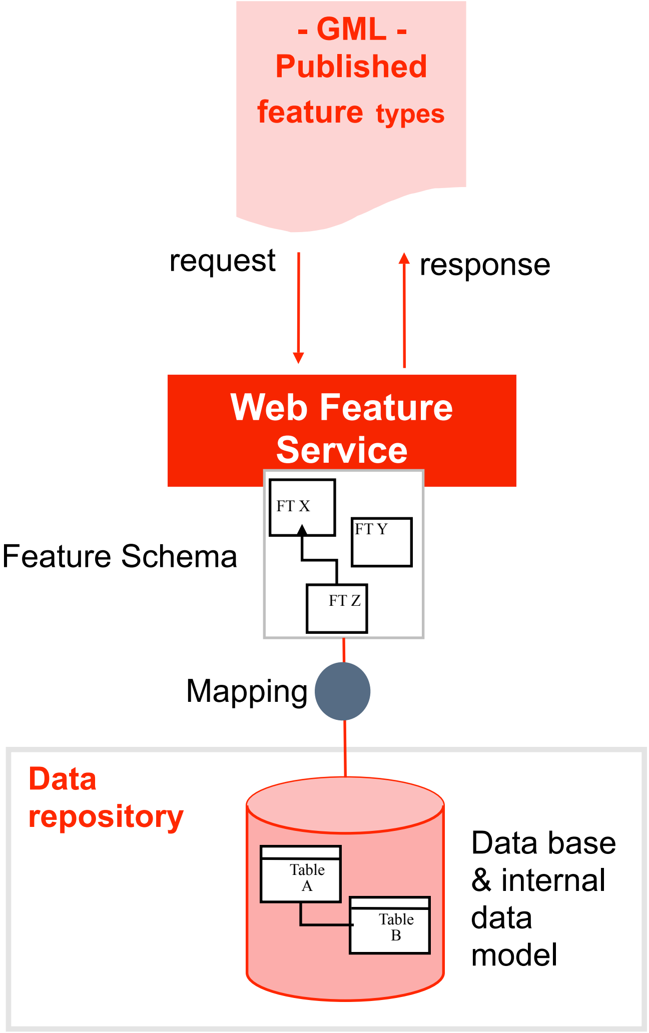
WFS (Web Feature Service)

A OGC standard (and ISO 19142) – current version 1.1

A WFS is a Web interface that allows one to publish and deploy geographic feature data locally or across the Internet.

2 types of services: standard WFS and Transactional WFS (WFS-T)

WFS-T operations include the ability to insert, delete, update, get and query features on spatial and non-spatial constraints



Features of a WFS

- Access/search/update/delete geo-spatial feature instances based on spatial/non-spatial search criteria using a standard interface over the web
- Access/Update in a secure way with proper authentication and authorization
- Manage feature privileges at a instance level
- Real-time transfer of feature instances in a platform/ programming language independent way

Oracle Spatial WFS

- Use SOAP/XML over HTTP for Request/Response
- Oracle Spatial for Feature instance Storage/Retrieval
- Implement GML filter specification for feature search
- Use LDAP for authentication, Oracle Label-based security for instance-level privilege mgmt and WSS/ SSL for secure transfer of feature data
- Implement token-based locking of feature instances to support WFS locking protocol
- Implement feature cache in middle-tier to reduce volume of spatial data transfer from DB to middle-tier, and make WFS request processing more efficient.

Publishing Feature Types

- Support publishing of feature types from database data sources(tables, views)
  - Complex Type columns
  - Nested Table/VARRAY columns
  - XMLType Columns
- Support publishing feature types from external data sources (external XSDs)

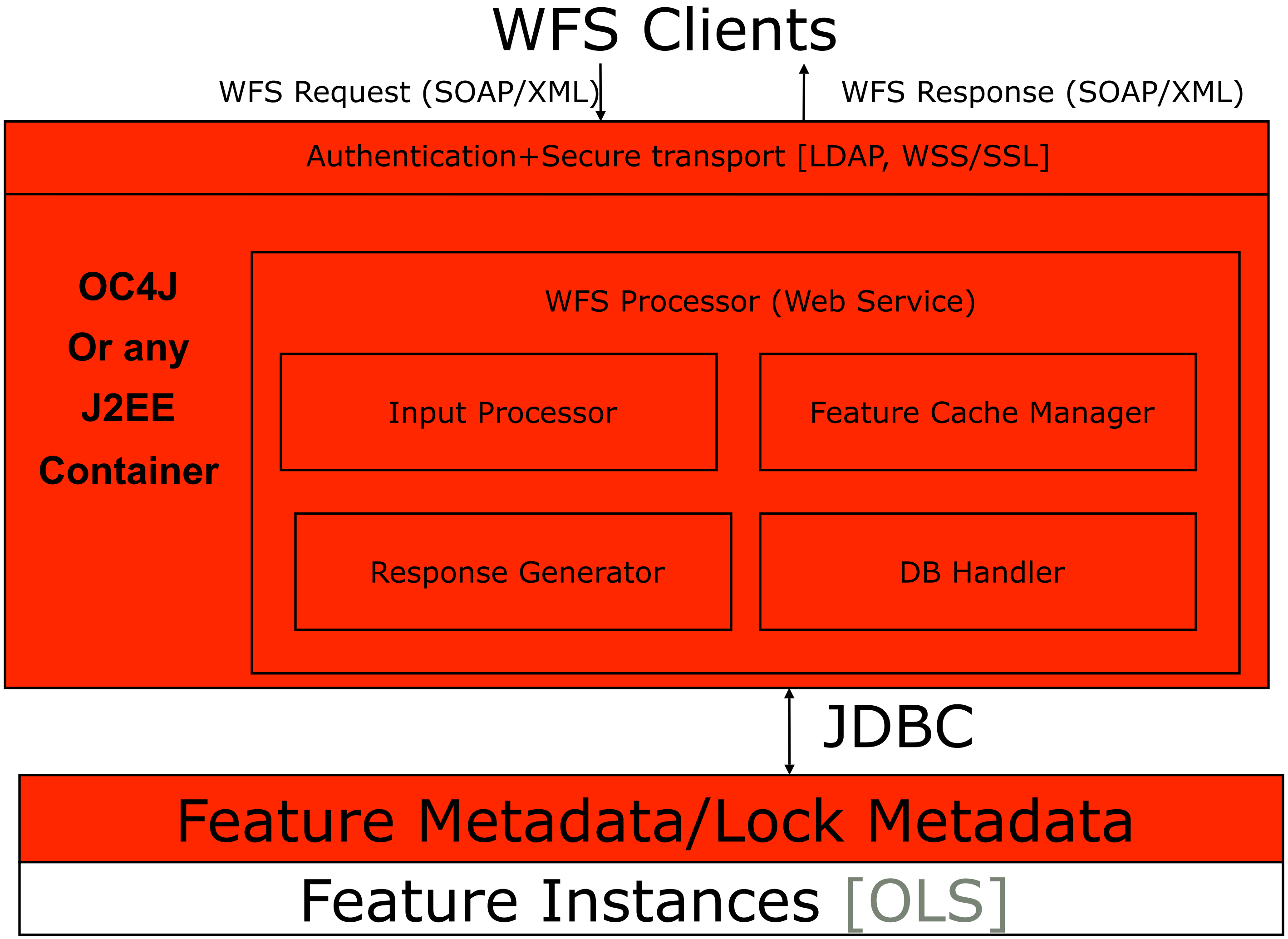
Publish Feature Types

- Relational datasource (e.g. table)
  - PLSQL API to publish the content of a table with Spatial Column to a feature type which is a subtype of gml:_Feature
    - Columns Map to Feature Type Tags
    - Column Types Map to Tag Types in XML
    - User-defined object map to ComplexTypes in XML
    - Type naming in chosen by default
- XSD Document based datasource
  - Java API to register feature type XSDs and feature type metadata
    - Register spatial paths on which spatial index will be built
    - Register non-spatial paths on which XDB index will be built
    - Feature Type Registration XSD, captures all feature type metadata parameters

WFS Operations

- Basic
  - Get Capabilities – get the metadata about the types / operations a feature server supports
  - DescribeFeature - get the structural information about a feature type
  - GetFeature – query different parts of feature instances
- Transactional operations
  - GetFeatureWithLock – get a set of features, and lock some/all of them for a certain period of time.
  - LockFeature – lock a set of feature instances
  - Transaction
    - Insert new feature instances
    - Update existing feature instances based on filter criteria
    - Delete existing feature instances based on filter criteria

Architecture



WFS Clients

WFS Request (SOAP/XML)

WFS Response (SOAP/XML)

Authentication+Secure transport [LDAP, WSS/SSL]

OC4J

Or any

J2EE

Container

WFS Processor (Web Service)

Input Processor

Feature Cache Manager

Response Generator

DB Handler

JDBC

Feature Metadata/Lock Metadata

Feature Instances [OLS]

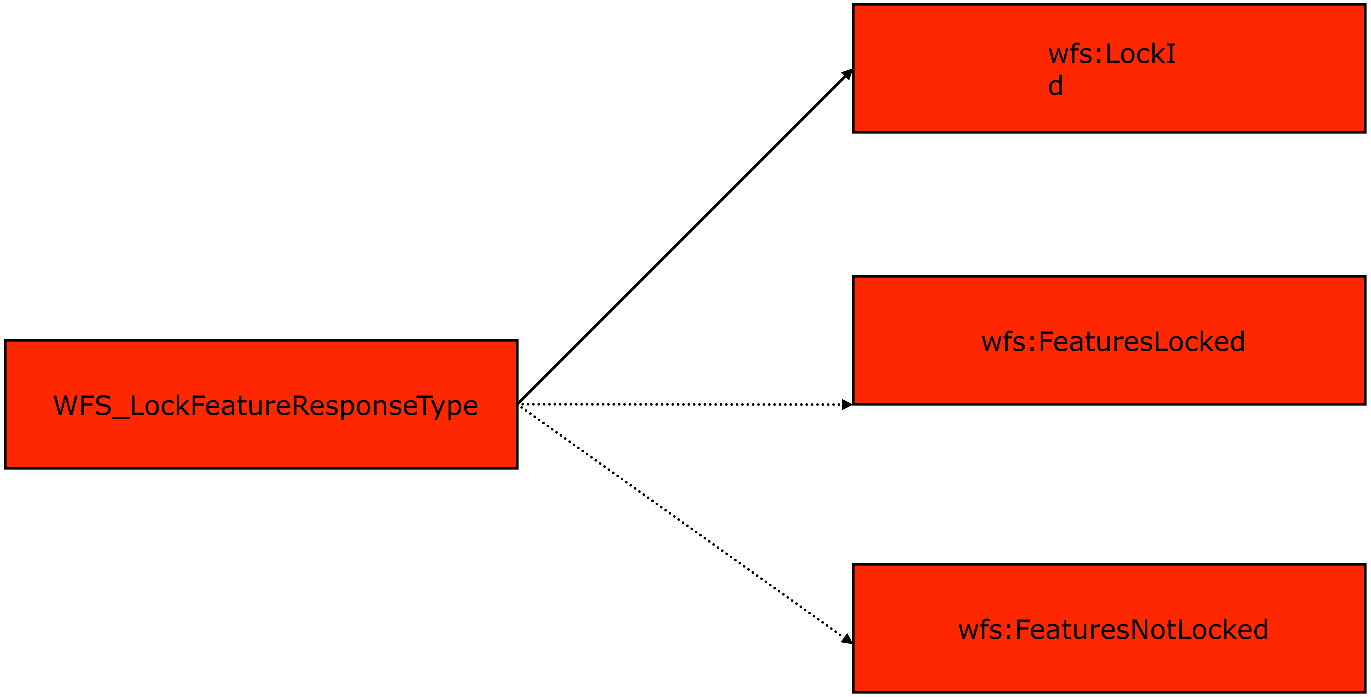
Caching

- Provide main-memory storage of spatial objects
- Helps reduce frequent transfer of spatial object from database to memory
- In-memory locks for update cache entries consistently

Locking

- DB Locking
- Lock duration in minutes and spans db transaction boundary
- Token-based locking semantics
- Unlock rows when lock expires
- Define triggers on feature tables/views to make sure that the user in the current session has shown a non-expired lock token, which was obtained previously for updating/deleting the concerned rows
- Locking logic will be enforced uniformly for Java or PLSQL interfaces

Lock Feature Response



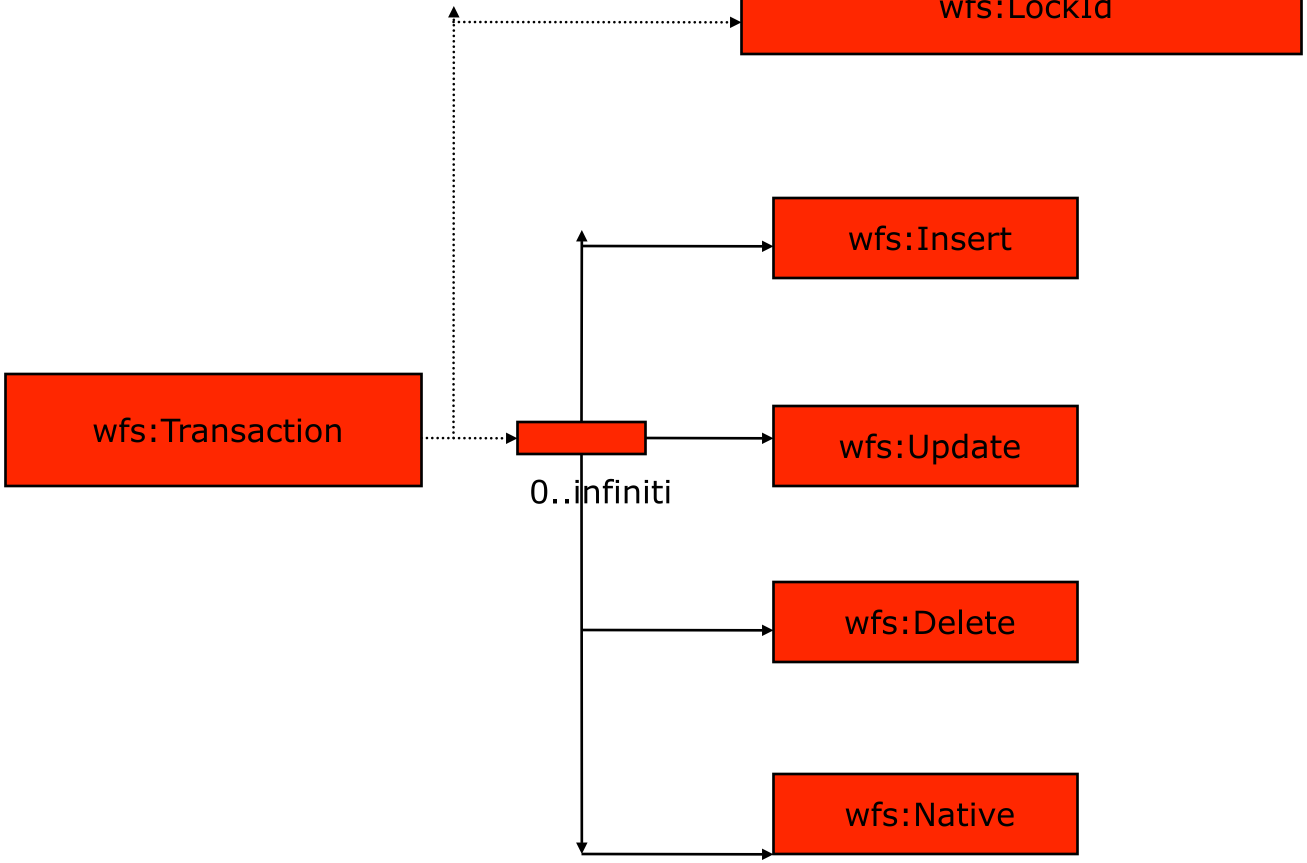
WFS_LockFeatureResponseType

wfs:LockI
d

wfs:FeaturesLocked

wfs:FeaturesNotLocked

WFS Transaction



wfs:LockId

wfs:Insert

wfs:Transaction

wfs:Update

0..infiniti

wfs:Delete

wfs:Native

WFS Use Cases

- Type Consumers
  - Get Server Capabilities
  - Describe Feature Type
  - GetFeatures (with proper filter)
  - GetFeatureWithLock
  - LockFeature
  - Transaction
    - Insert/Update/Delete
- Type Supplier
  - Publish Types
  - Define type access control privileges

Feature Type Example

```
<complexType name="ROADL_1M_Type">
<complexContent>
<extension base="gml:AbstractFeatureType">
<sequence>
   <element name=“NAME” type=“string” nillable=“false”/>
   <element name=“PATH"  type="gml:LineStringPropertyType"   nillable="false"/>
   <element name="SURFACE_TYPE" nillable="true" minOccurs="0">
     <simpleType>
        <restriction base="string">
           <maxLength value="30"/>
         </restriction>
    </simpleType>
   </element>
   <element name="NLANES" nillable="true" minOccurs="0">
        <simpleType>
          <restriction base="integer">
              <totalDigits value="2"/>
           </restriction>
        </simpleType>
    </element>
</sequence> </extension> </complexContent>
</complexType>
```

Feature Instance Example

```
<gml:featureMember>
        <ROADL_1M fid="ROADS_1M.100” >
          <NAME>HYW 401</NAME>
           <PATH>
                 <gml:LineString  srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
                    <gml:coordinates decimal="." cs="," ts=" ">-59.478340,-52.226578
    -59.484871,-52.223564 -59.488991,-52.198524 -59.485958,-52.169559
    -59.480400,-52.152615 -59.465576,-52.141491 -59.462002,-52.136417
    -59.447968,-52.127190 -59.422928,-52.120701 -59.411915,-52.117844
    -59.397972,-52.116440 -59.371311,-52.121300   </gml:coordinates>
                  </gml:LineString>
          </PATH>
           <SURFACE_TYPE>ASPHALT</SURFACE_TYPE>
           <NLANES>12</NLANES>
          </ROADL_1M>
  </gml:featureMember>
```

Get Feature Request

```
  <?xml version="1.0" ?>
<GetFeature
version="1.0.0"
service="WFS"
handle="Example Query"
xmlns="http://www.opengis.net/wfs"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:gml="http://www.opengis.net/gml"
xmlns:myns="http://www.someserver.com/myns"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd">
<Query typeName="myns:ROADS_1M">
    <ogc:PropertyName>myns:PATH</ogc:PropertyName>
     <ogc:PropertyName>myns:SURFACETYPE</ogc:PropertyName>
    <ogc:PropertyName>myns:NLANES</ogc:PropertyName>
     <ogc:Filter>
         <ogc:Within>
               <ogc:PropertyName>myns:PATH</ogc:PropertyName>
               <gml:Box>
                    <gml:coordinates>50,40 100,60</gml:coordinates>
                </gml:Box>
         </ogc:Within>
     </ogc:Filter>
    </Query>
 </GetFeature>
```

Get Feature Response

```
<?xml version="1.0" ?>
<wfs:FeatureCollection
xmlns="http://www.someserver.com/myns"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd
http://www.someserver.com/myns ROADSRAILS.xsd">
     <gml:boundedBy>
            <gml:Box srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
                     <gml:coordinates>0,0 180,360</gml:coordinates>
             </gml:Box>
      </gml:boundedBy>
<gml:featureMember>
     <ROADS_1M fid="ROADS_1M.100">
           <PATH>
               <gml:LineString gid="1"
                     srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
                      <gml:coordinates>10,10 10,11 10,12 10,13</
    gml:coordinates>
               </gml:LineString>
          </PATH>
     <SURFACE_TYPE>ASPHALT</SURFACE_TYPE>
     <NLANES>4</NLANES>
    </ROADS_1M>
</gml:featureMember>
</wfs:FeatureCollection>
```

Insert Feature Instance

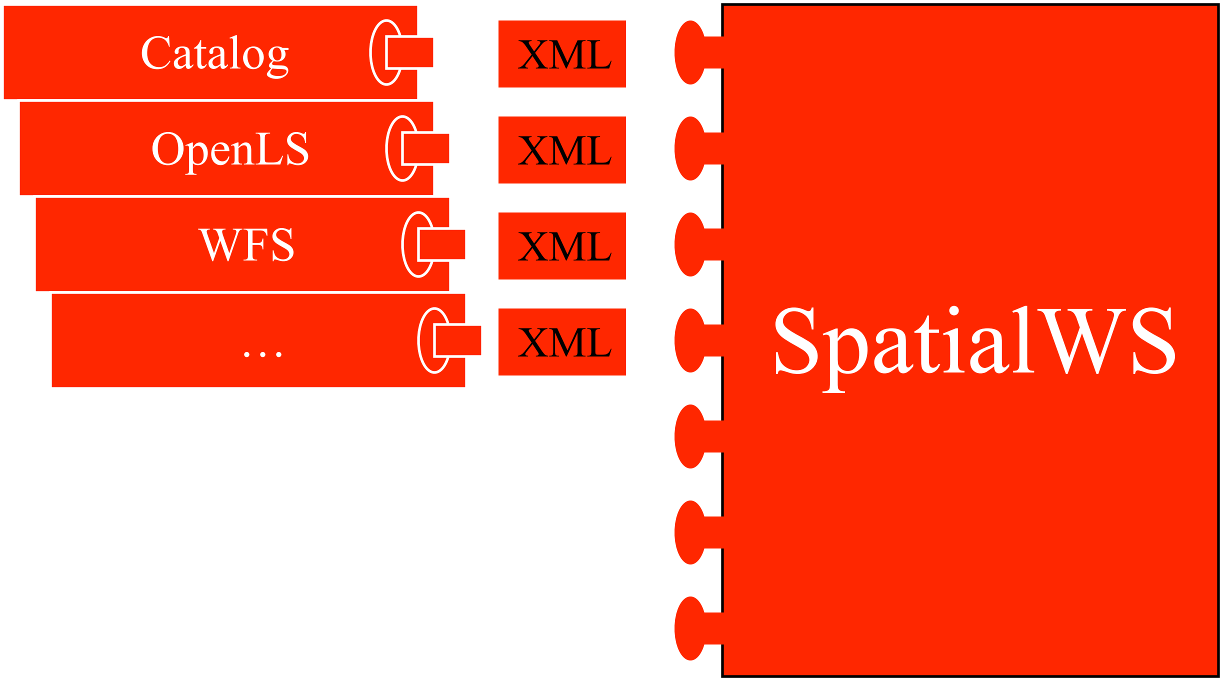
```
 <wfs:Insert handle="ComplexInsert">
<ROADL_1M>
<NAME>Highway 401</NAME>
<PATH>
        <gml:LineString gid="e3"
            srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
           <gml:coordinates>...</gml:coordinates>
        </gml:LineString>
</PATH>
<SURFACE_TYPE>Asphalt</SURFACE_TYPE>
<NLANES>12</NLANES>
</ROADL_1M>
</wfs:Insert>
```

Web Services Security

Security

- Identification - Who are you?
- Authentication - How do I know it is you?
- Authorization - Are you allowed?
- Integrity - Is the data tampered?
- Confidentiality - Did anyone else read it?
- Auditing - Can you show what happened?
- Non-Repudiation - Can you prove it happened?

Share the Security Platform



OGC compliant
SOAP

WSS,
VPD

Proxy Auth / App user mgm

Spatial WS Functionality

- Pluggable component handles
    - XML request / response
- Component can ignore
    - SOAP envelope
        - Will comply w/ OGC
        - User name & pwd/certificate
        - Encryption & signatures
    - User authorization
        - Managed by DB including VPD
        - When needed: J2EE security model
    - Connection mgm
        - Proxy authentication
        - App user mgmt
        - Multi-user login

Propagation Of Identity

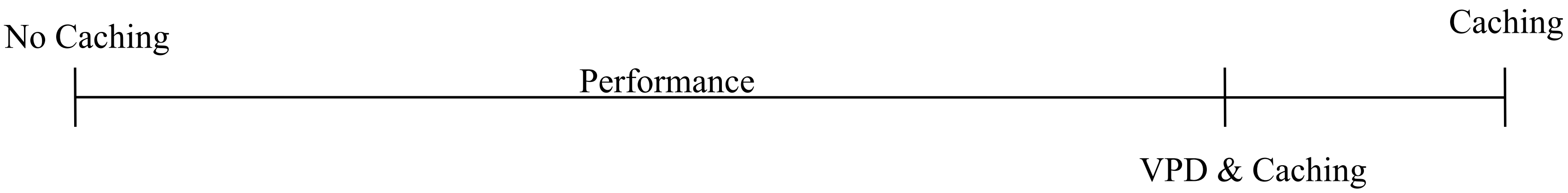
- Client
  - JAX-RPC
    - Username
    - Password
- SOAP header
  - WSS
- Oracle JAZN/LDAP/XML
- SpatialWS
  - Proxy Auth
  - App User mgm
  - (Group of users)
- DB connection
  - Grant
  - VPD

Spatial Web Services Security

- Data Access Authorization in DB
  - VPD
  - Oracle Label Security
- User authentication in mid-tier
- Similar approach is used for handling versioning with Workspace Manager
- Issues with Cache

Caching And VPD
Basic Solution

- Verify visibility in the DB
  Select **id** from … where …
- Lookup record in cache



No Caching

Caching

Performance

VPD & Caching

Java API

Java API

- 2D Simplify
- Projection to Local Tangent Plane for Geodetic data
  - And its inverse operation
- Arc Densification
- Affine Transformations
  - Shift, rotate, translate, scale
- Element Extractor
- 2D Buffer
- GML3 Geometry Support
  - From GML3 Geometry: GML3g.java
  - To GML3 Geometry: GML3.java
  - These classes will be part of sdoutl.jar
  - PL/SQL interfaces to support GML3 conversion

J3D_Geometry: Java API

- Extends JGeometry class
- Supported Methods:
  - Validate(tolerance): Returns true or false
  - Distance(J3D_geometry): Returns distance
  - Anyinteract(J3D_Geometry): Returns true or false
  - Length(): returns length for 3D line
  - Area(): returns area for a surface geometry
  - Volume(): returns volume of a solid geometry
- *Iterator* class to access elements of a J3D_Geometry
  - Optional “extract_level”
  - nextElement(): returns the next element of the input J3D_Geometry (as a J3D_Geometry) or null at the end

GeoRaster

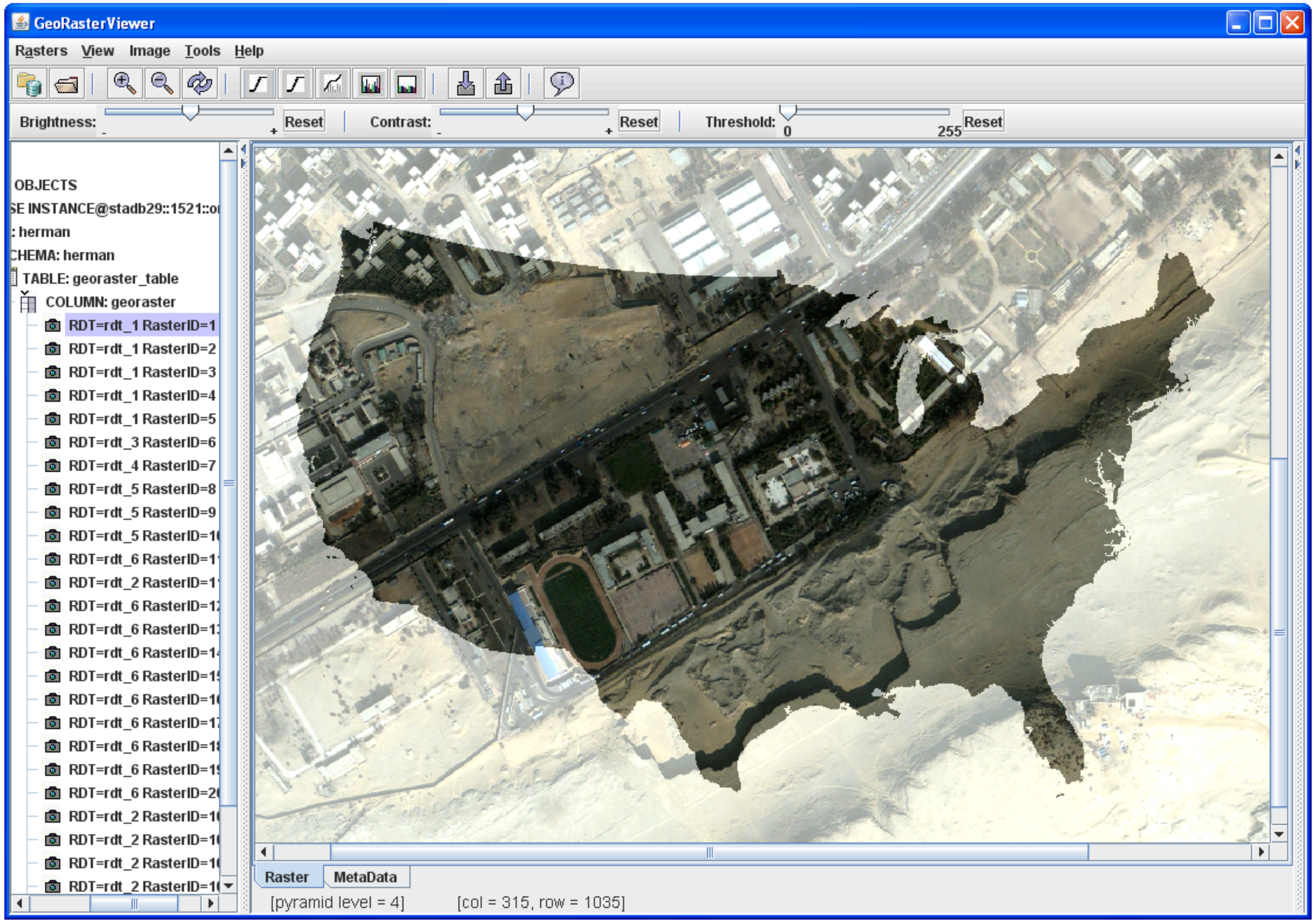
System Manageability Enhancements

- Automated creation and enhanced monitoring and management of GeoRaster DML triggers to improve usability and ensure data integrity
- Enhanced internal database mechanism to monitor DDL events and activities on GeoRaster sysdata table to improve manageability, data safety, reliability, robustness, and usability
- New SDO_GEOR_ADMIN package providing tools to automate and ease GeoRaster database upgrade and migration
- Supports Oracle Workspace Manager for raster versioning and Label Security for row-level data security

New Metadata and Raster Support

- Supports a generic functional fitting georeferencing model
  - georeference un-rectified or raw airborne photos and satellite images
  - supports up to a power of 5 and 3-D model coordinates
  - It includes special models such as DLT and RPC
- Supports bitmap masks for any GeoRaster objects and their individual bands/layers
  - masks are stored inside the GeoRaster objects
  - Pyramids can be generated for masks as well.

Bitmap Mask Example



OBJECTS
E INSTANCE@stadb29::1521::o
: herman
CHEMA: herman
TABLE: georaster_table
COLUMN: georaster
RDT=rdt_1 RasterID=1
RDT=rdt_1 RasterID=2
RDT=rdt_1 RasterID=3
RDT=rdt_1 RasterID=4
RDT=rdt_1 RasterID=5
RDT=rdt_3 RasterID=6
RDT=rdt_4 RasterID=7
RDT=rdt_5 RasterID=8
RDT=rdt_5 RasterID=9
RDT=rdt_5 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_2 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_6 RasterID=2
RDT=rdt_2 RasterID=1
RDT=rdt_2 RasterID=1
RDT=rdt_2 RasterID=1
RDT=rdt_2 RasterID=1

```
[pyramid level = 4]        [col = 315, row = 1035]
```

New Metadata and Raster Support

- Supports multiple NODATA Values and multiple NODATA Value Ranges for any GeoRaster objects and their individual bands/layers
- Supports empty raster blocks
  - support for special sparse data type
  - saves storage space of large mosaic (virtually no disk space is needed for these blocks)
  - improved raster processing performance
  - Pyramids can contain empty raster blocks

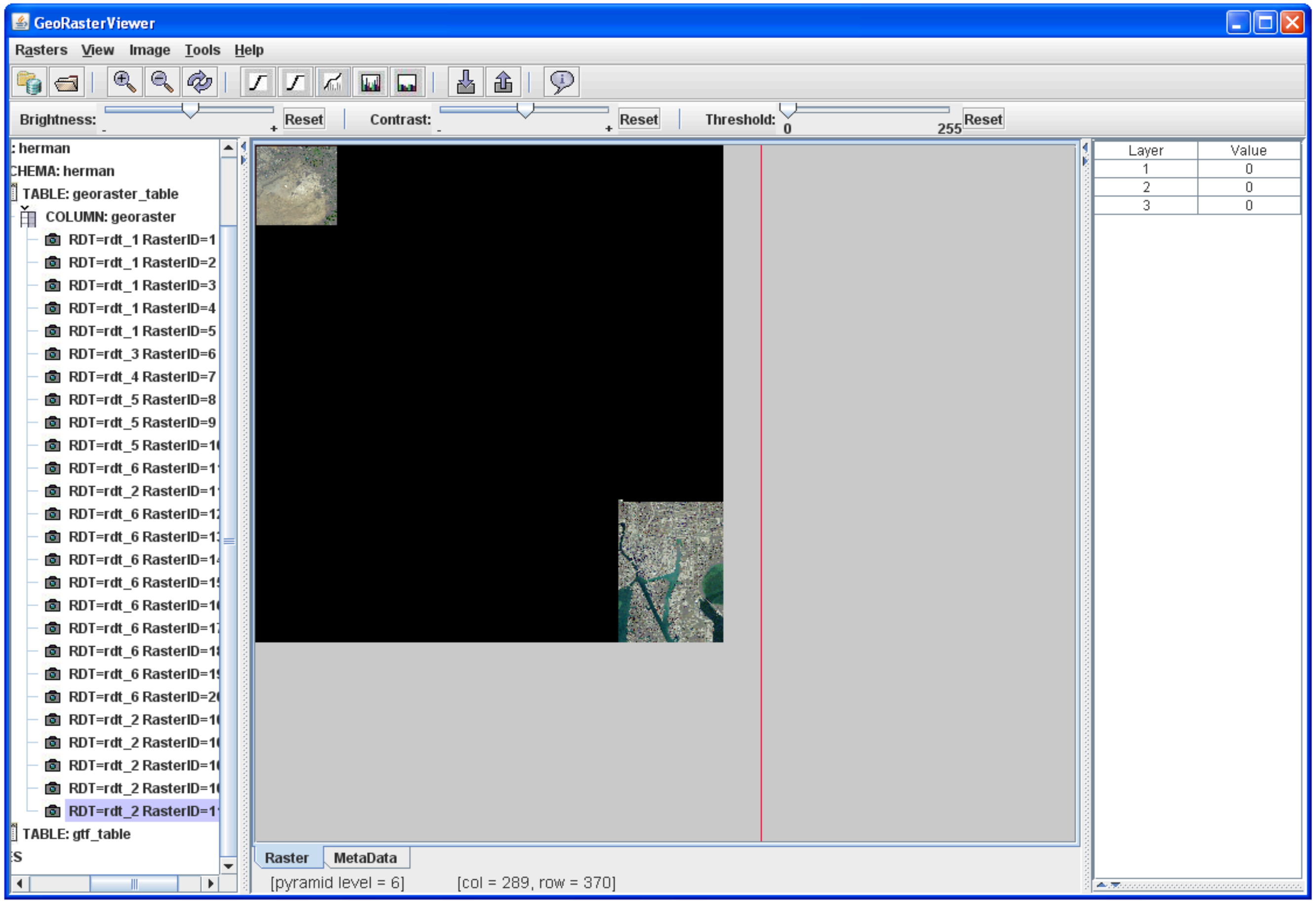
New Functions

- Union/merging of multiple GeoRaster objects or multiple layers
- Partial update/edit of raster data inside a GeoRaster object
- GeoRaster template functions to ease third-party software integration so that the developers don’t need to directly deal with database BLOB and XMLType
- Statistic analysis and histogram generation
- PL/SQL API enhanced to support the new georeferencing models, bitmap masks, NODATA types and empty raster blocks

New Functions

- Supports sub-cell or sub-pixel addressing (floating row and column numbers) in the GeoRaster cell space
  - It was only internally supported in 10g
- Supports random raster blocking sizes
  - which don't have to be a power of 2 anymore
  - It's more flexible and helps save storage space.
- Enhanced mosaic
  - Allows gaps, overlaps and missing raster tiles.
  - Gaps and missing tiles are stored as empty raster blocks (no disk space needed) as appropriate.

Sample Mosaic With Empty Raster Blocks



: herman
CHEMA: herman
TABLE: georaster_table
COLUMN: georaster
RDT=rdt_1 RasterID=1
RDT=rdt_1 RasterID=2
RDT=rdt_1 RasterID=3
RDT=rdt_1 RasterID=4
RDT=rdt_1 RasterID=5
RDT=rdt_3 RasterID=6
RDT=rdt_4 RasterID=7
RDT=rdt_5 RasterID=8
RDT=rdt_5 RasterID=9
RDT=rdt_5 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_2 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_6 RasterID=1
RDT=rdt_6 RasterID=2
RDT=rdt_2 RasterID=1
RDT=rdt_2 RasterID=1
RDT=rdt_2 RasterID=1
RDT=rdt_2 RasterID=1
RDT=rdt_2 RasterID=1
TABLE: gtf_table
S

Layer	Value
1	0
2	0
3	0

[pyramid level = 6] [col = 289, row = 370]

Object Size = 2.15 GB

Space Used = 0.34 GB

Space Saved = 1.81 GB

Storage Ratio = 16.14%

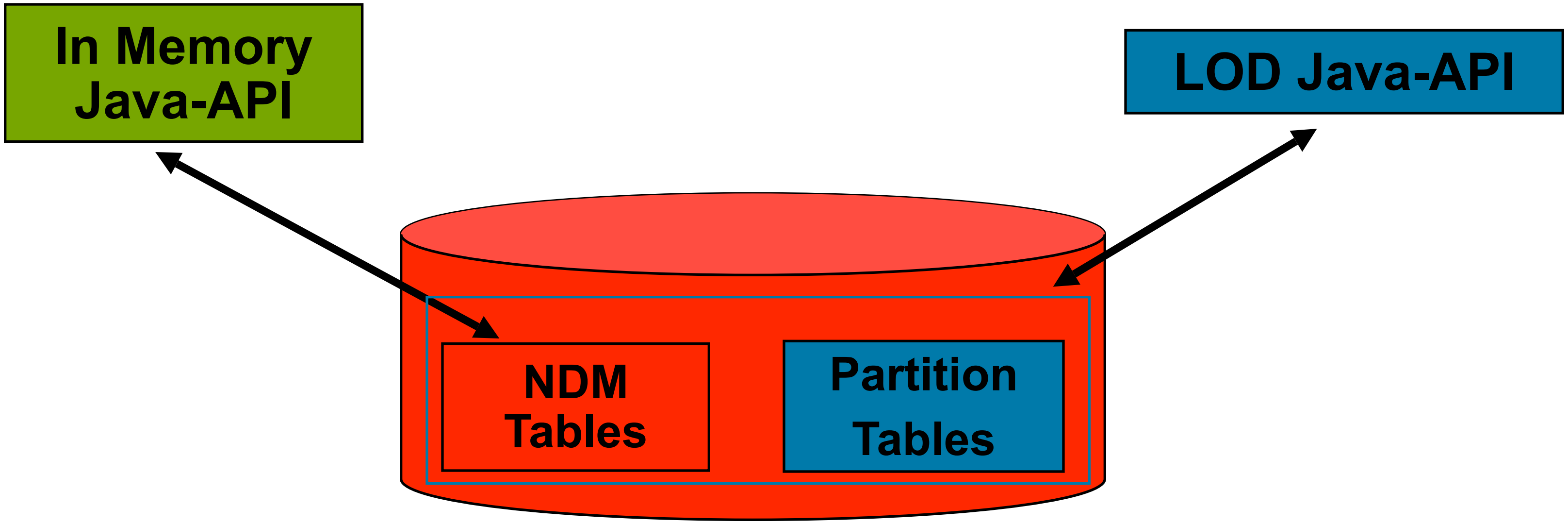
New Functions - Tools

- Loading and exporting more formats
  - GeoTIFF file format. Supports its geometadata, including EPSG coordinate systems
  - JPEG 2000 file format (limited. command-line only)
  - Digital Globe RPC text file format
- GeoRaster Viewer is enhanced to display
  - Masks and related operations
  - Coordinates from new georeferencing models.
  - Empty raster blocks, etc

Network Data Model

Two approaches to Network Analysis

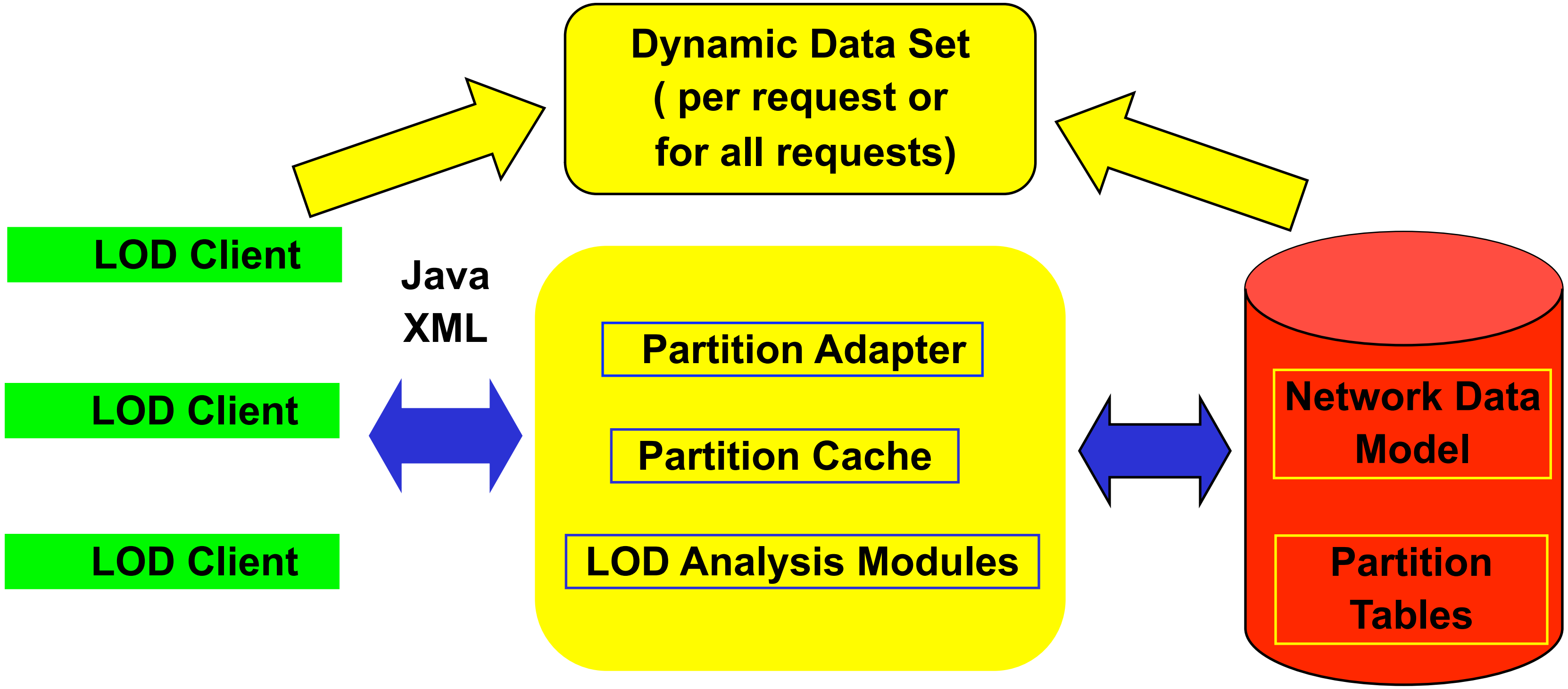
- Load-On-Demand Analysis
  - Handle network analysis on very large networks
- In-Memory Analysis
  - New modeling and analysis features for in-memory approach



NDM: Load On Demand Analysis

- Provide a scalable solution to network analysis
  - Handle large partitioned networks
- Provide a flexible solution to network modeling and analysis
  - Support network constraints
  - Support user defined data
  - Support dynamic changes
- Use the same NDM data models in database
  - Same NDM network data model
  - Partitioning data (partition table + partition blobs)
- LOD APIs are different from NDM in-memory APIs
  - LOD does not pre-load the whole network
  - It only loads the parts that are needed during analysis

NDM LOD Architecture



Dynamic Data Set
(per request or
for all requests)

LOD Client

LOD Client

LOD Client

Java
XML

Partition Adapter

Partition Cache

LOD Analysis Modules

Network Data
Model

Partition
Tables

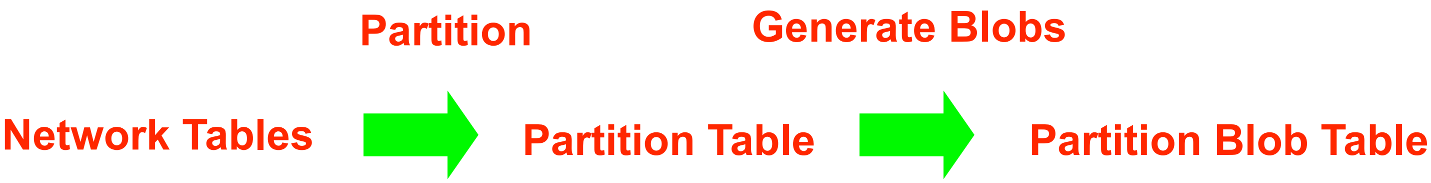
NDM LOD Clients

NDM LOD Analysis Engine

NDM Network Data Models
LOD Partition Data

NDM LOD Network Partitions

- Network partitions are the basic units in NDM LOD
- NDM provides a spatial partitioning utility to help partition spatial networks
- The partition result is stored in the specified partition table
  - To further speed up network partition loading, NDM converts relational partitions into partition blobs and store them in the partition blob table
- Partition size is defined by users (max. no of nodes)
- Support partition by link_level (link priorities)
- User can partition their own networks (NODE_ID, PARTITION_ID)



Partition

Generate Blobs

Network Tables

Partition Table

Partition Blob Table

NDM LOD Dynamic Data Set

- Network Partitions contain mainly static network data
- What if we would like to change some network element attributes such as state or cost? (roads under construction or traffic delays)
  - Need to take dynamic changes of the underlying network into account
- These changes might be only valid for a single query
  - What-if scenarios
  - Using a dynamic data set adding/removing/overwriting these elements in network partitions
  - The size of the dynamic data set is usually small
  - The set can be pre-fetched before analysis if needed

NDM LOD Analysis Engine

- LOD Analysis Java API supports the following
  - Shortest Path
  - Nearest (Reaching) Neighbors
  - Within (Reaching) Cost
  - Reachable and Reaching Nodes
  - Hierarchical shortest path
- Analysis is based on connection information only
  - Spatial information can be stored as a user defined data
- Network Constraints are supported in the above analysis functions
- Users implement user constraints on top of analysis information passed to them

New Features for In-Memory Approach

- User defined data
- Duration modeling (on nodes and links, duration column)
  - Duration as an accumulated attribute (like cost)
- Sub-network support using SQL-like filters
  - Speed up network loading and analysis
- Path arithmetic support
  - Path addition, subtraction, intersection, and comparison based on connectivity
- New analysis functions
  - Partial link support on paths (sub-path), trace-out
- Workspace manager support
  - Transform networks between different work spaces

Routing Engine

- Driving directions in different languages
  - German, French, Spanish, Italian
- Generation of turn-specific geometries is supported
  - Helps in adding turn specific YP data to the LBS application
- Route as a set of Edges
  - The computed route can be returned as a set of edges which can be used in further analysis
- Works with TeleAtlas and NAVTEQ data sets

Workspace Manager

Valid Time Enhancements

- Provide Valid Time (VT) wm_period datatype optionally as two scalar types

```
EXECUTE DBMS_WM.SetSystemParameter
(‘USE_SCALAR_TYPES_FOR_VALIDTIME’, 'ON');
```

- Allow initial VT ValidFrom and ValidTill dates to be specified when a table is version-enabled

```
EXECUTE DBMS_WM.EnableVersioning (‘employee', ‘NONE’,
FALSE, TRUE, ‘UNLIMITED’, WMSYS.WM_PERIOD(TO_DATE
('01-01-2006', 'MM-DD-YYYY'), DBMS_WM.UNTIL_CHANGED));
```

- Allow editing of the VT time range for a feature in a parent table of a Referential Integrity Constraint

History Enhancements

- Provide a unique key for every VT & history row

```
EXECUTE DBMS_WM.SetSystemParameter
(‘ADD_UNIQUE_COLUMN_TO_HISTORY_VIEW’, 'ON');
```

- Track which workspace creates/retires a history row

```
EXECUTE DBMS_WM.SetSystemParameter
(‘KEEP_REMOVED_WORKSPACES_INFO’, 'ON');
```

- New views ALL_REMOVED_WORKSPACES and USER_REMOVED_WORKSPACES

Other Workspace Enhancements

- Oracle Spatial Georaster long transaction editing support
- Merge version-enabled tables involved in a Referential Integrity Constraint together as a set

```
EXECUTE DBMS_WM.MergeTable ('NEWWORKSPACE',
'user3.table1, user3. table2’, 'last_name = ''Smith''');
```

- Workspace ID Column Added to Views

Performance Enhancements

- Row level locking for Merge operations to improve concurrency (shared lock on parent workspace)

```
EXECUTE DBMS_WM.SetSystemParameter
(‘ROW_LEVEL_LOCKING’, 'ON');
```

- Optionally compresses the parent workspace savepoints for RemoveWorkspace

```
EXECUTE DBMS_WM.SetSystemParameter
(‘COMPRESS_PARENT_AFTER_REMOVE’, 'ON');
```

- Constrain memory used for Merge operations

```
EXECUTE DBMS_WM.SetSystemParameter
(‘TARGET_PGA_MEMORY’, '8388608');
```

Performance Enhancements

User-specified hints for workspace operations

AddUserDefinedHint

This example specifies a full table scan on the TABLE1 table and any associated Workspace Manager infrastructure tables when a SQL statement specifies hint ID 1101 with the SCOTT.TABLE1 table.

```
EXECUTE DBMS_WM.AddUSerDefinedHint (1101, 'scott.table1', 'full
(t1)');
```

RemoveUserDefinedHint

This example removes, for the SCOTT.TABLE1 table, the user-defined hint from SQL statements associated with the hint with the hint ID 1101, and causes the default hint to be used

```
EXECUTE DBMS_WM.RemoveUSerDefinedHint (1101, 'scott.table1');
```

Security & Database Support

- Version-enable a table on which a Oracle Label Security policy is defined
- Allow online Alter Table physical space mgt.
- Support the [NOT] NULL constraint option
- AlterVersionedTable alter_option supports rebuilding an index

```
EXECUTE DBMS_WM.AlterVersionedTable ('scott.my_table',
'REBUILD_INDEX', 'index_owner=scott, index_name=my_index,
noreverse’);
```

ORACLE®