
ORACLE®

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decision. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Oracle Spatial
11g Planned features

Overview

- 3-D Types and Functions
- Web Services
- GeoRaster
- Network Data Model
- Q&A

Planned functionality for 3D Data Management in Oracle Spatial

Background on 3-d Spatial

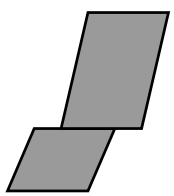
- 3D Types for points/lines/polygons
- Specialized types for large volumes of 3D point data
  - Represent scenes as a set of 3-d points obtained using laser scanners etc. (point clouds)
  - surface representation using TINs
- Address growing number of 3-d applications
  - GIS, CAD/CAM
  - VR, Medical Applications
  - City Modeling

Oracle Spatial 3D Functionality

- SDO_GEOMETRY for 3D Data
- SDO_POINT_CLOUD
- SDO_TIN
- 3D Coordinate Systems

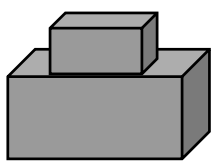
SDO_GEOMETRY for 3D Data

- 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 an 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



SDO_GEOMETRY for 3-D

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

Operations on 3D SDO_GEOMETRY

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

JGeometry: 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

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

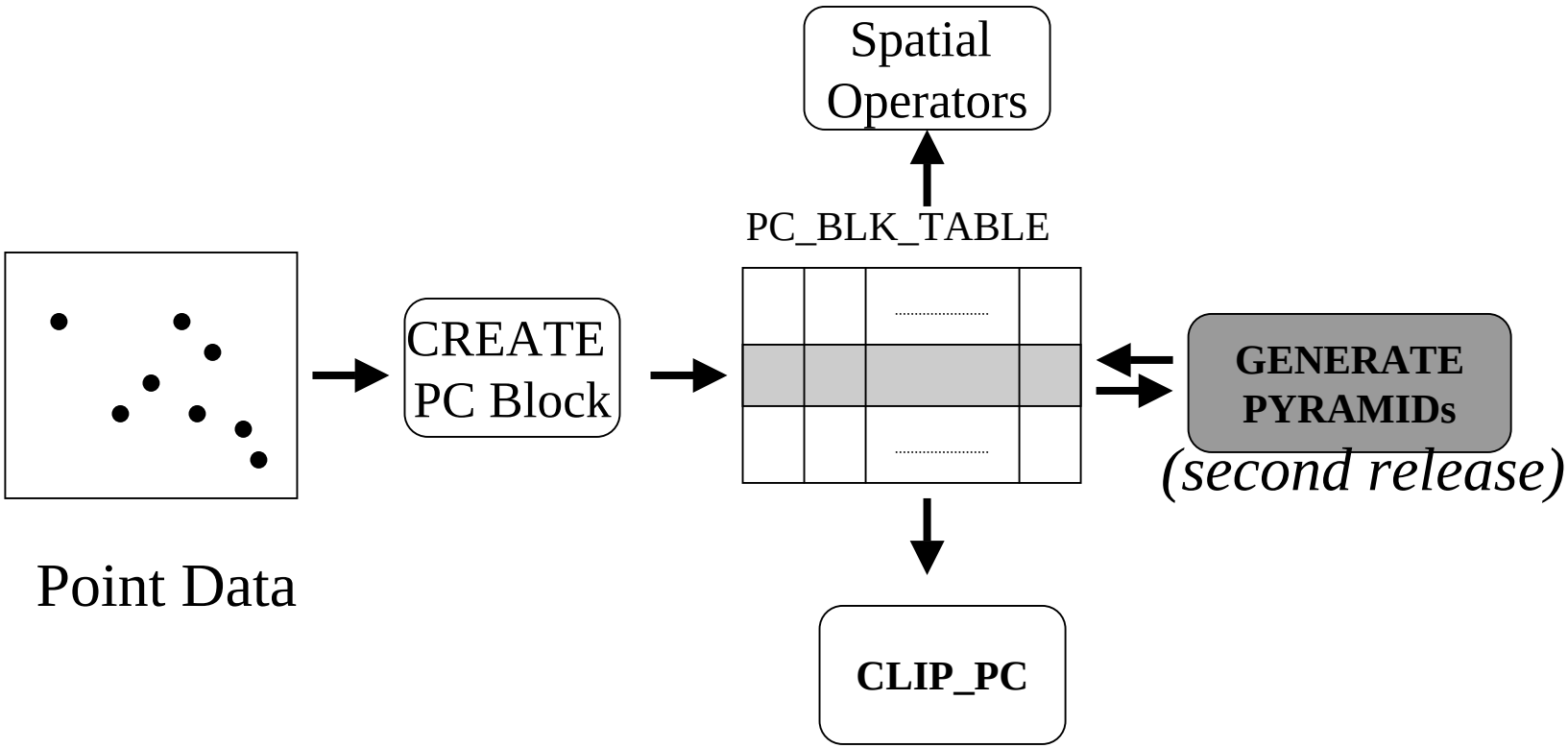
Specialized Types for High Density 3D Point Data

- 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
- TIN data model is built on top of this type

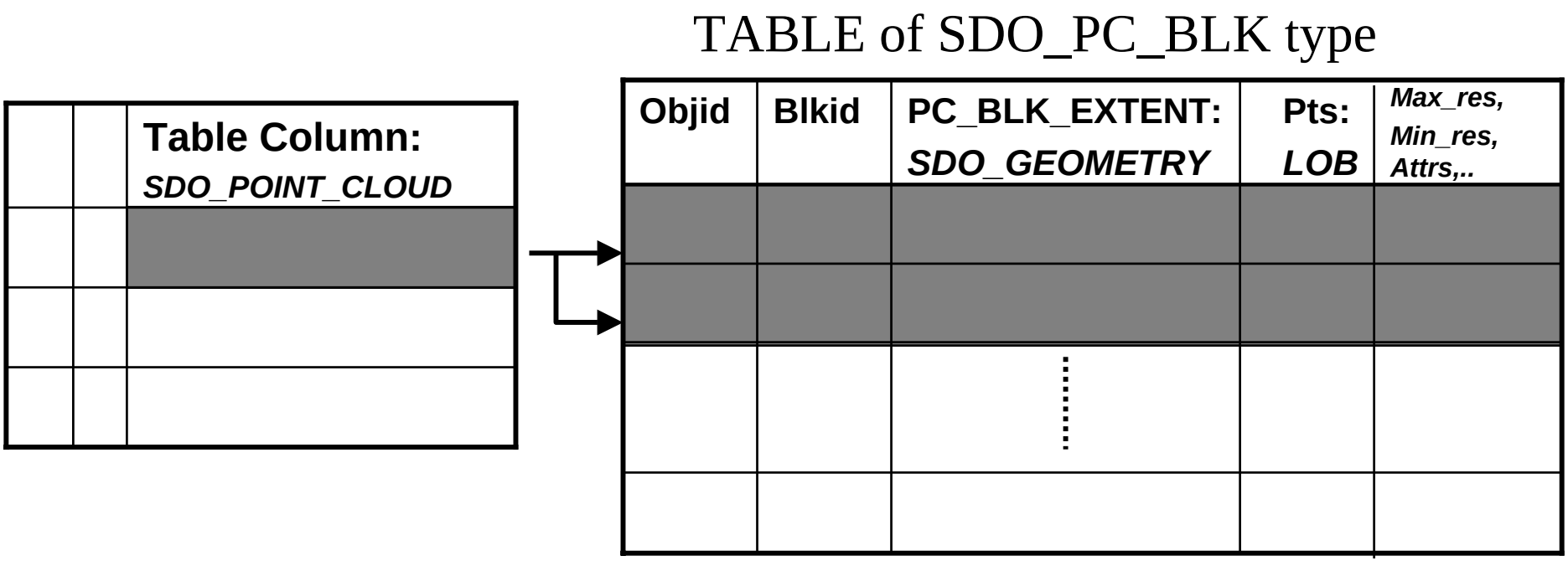
SDO_POINTCLOUD

- Features:
  - Points in a block can be stored in a “compressed” format w.r.t MBR
  - Block associated with an “interval” of resolution levels
  - Can store LIDAR data (LAS format)
  - Partition into multiple blocks for granularity of access
  - Spatial query with <frustum window, interval range>
  - Clipping, other operations

Operations

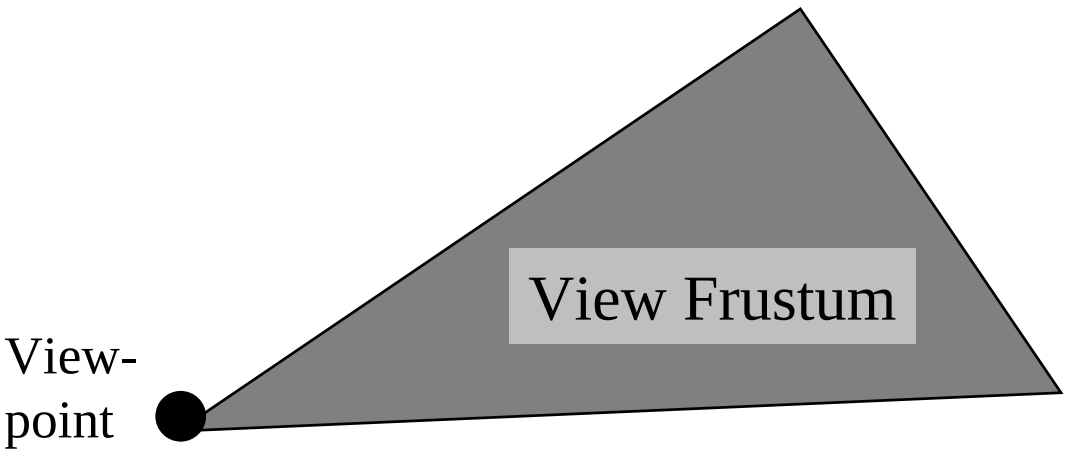

Spatial Operators
PC_BLK_TABLE
CREATE PC Block
GENERATE PYRAMIDs
(second release)
Point Data
CLIP_PC

SDO_POINT_CLOUD: Storage



- Not all columns shown. Exact details may change
- <Min_Res, Max_res> associated with blocks
- Pts can be stored w.r.t. pc_blk_extent (to reduce # of digits/space)

Visibility Query

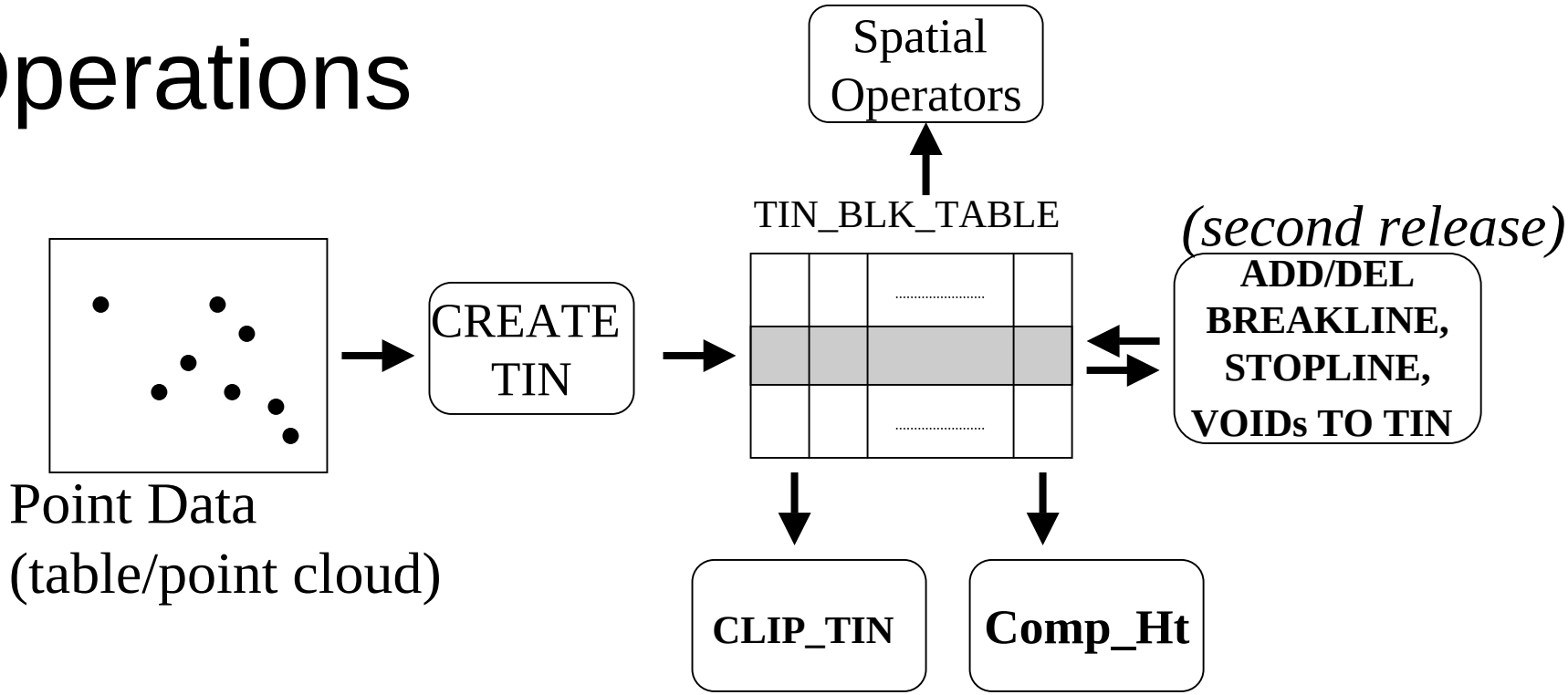


- Farther objects in lower resolution
- Nearer objects in higher resolution
Query specifies one or more
<query solid (frustum), [min_res, max_res]>

SDO_TIN

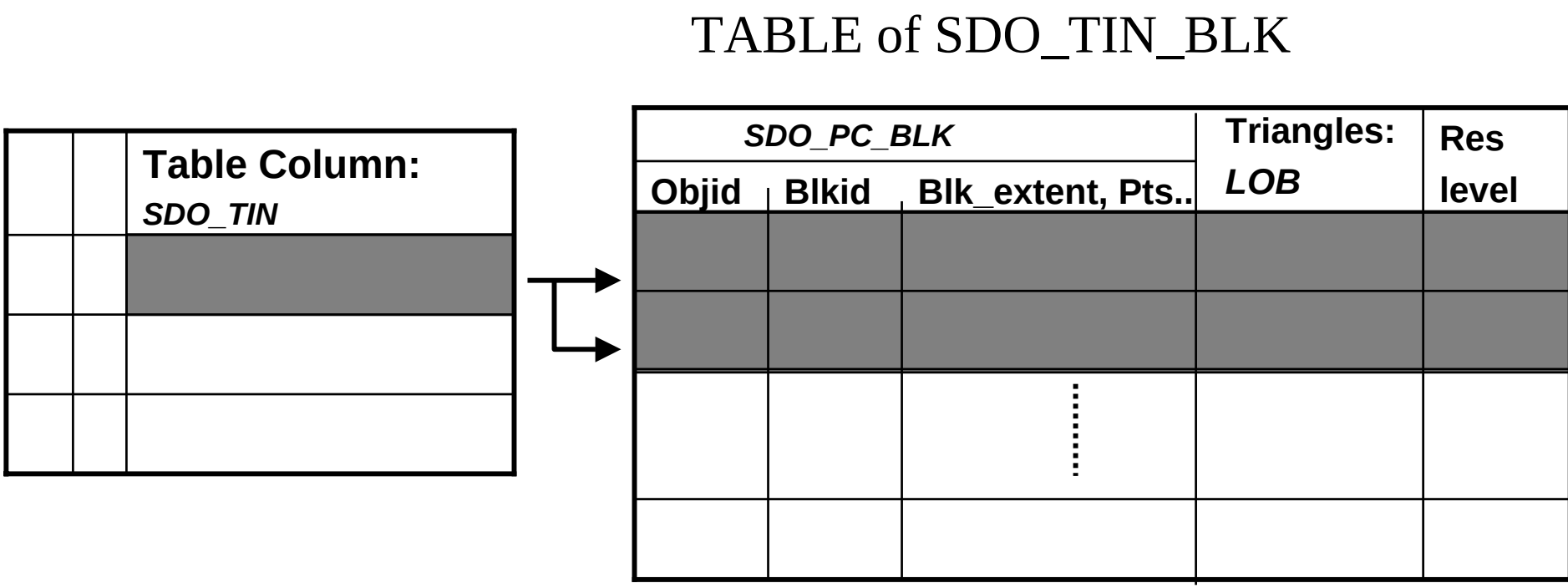
- Triangulated Irregular Network for representing surfaces
- Operations/Functions
  - Create TINS using points, break lines, stop lines and void regions
  - Validation
  - Add/delete breaklines, stoplines associated with a TIN
  - Clipping, compute_height, multi-resolution (future)

Operations



- TIN Creation for a given set of points, breaklines,…
- Read/Write Point/Triangle Data to the TIN_BLK_Table
- (Re-)Compute Triangles for a set of blocks using the points
- Querying the TIN to retrieve/clip the blocks
- Update TIN: add/delete points/lines/voids

SDO_TIN: Storage



- Not all columns shown. Exact details may change
- Res_level is set to 0. Future versions will use it to denote multi-resolution blocks.
- Pts can be stored w.r.t. tin_blk_extent (to reduce # of digits/space)

SDO_TIN

- Compact and scalable representation/storage for large TINS
  - 4Billion*4Billion Points
- Querying functions
  - Querying using a window + resolution-level
  - Clipping of the appropriate TIN blocks
- More functionality in subsequent releases
  - Storage of Multi-resolution TINs

3D Coordinate Systems

Following EPSG types are to be supported:

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

3D Coordinate System

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, wherever applicable
- Compute distances, and other spatial relationships between two objects within the same coordinate system

Planned functionality for geospatial web services in Oracle Spatial

Web Feature Service 1.0

Requirements

- 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

Our Approach

- 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
- WSS/SSL for secure transfer of feature data

Our Approach (Contd)

- Support publishing 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
- Implement token-based locking of feature instances
- Feature cache in middle-tier to reduce volume of spatial data transfer from DB to middle-tier
  - makes WFS request processing more efficient

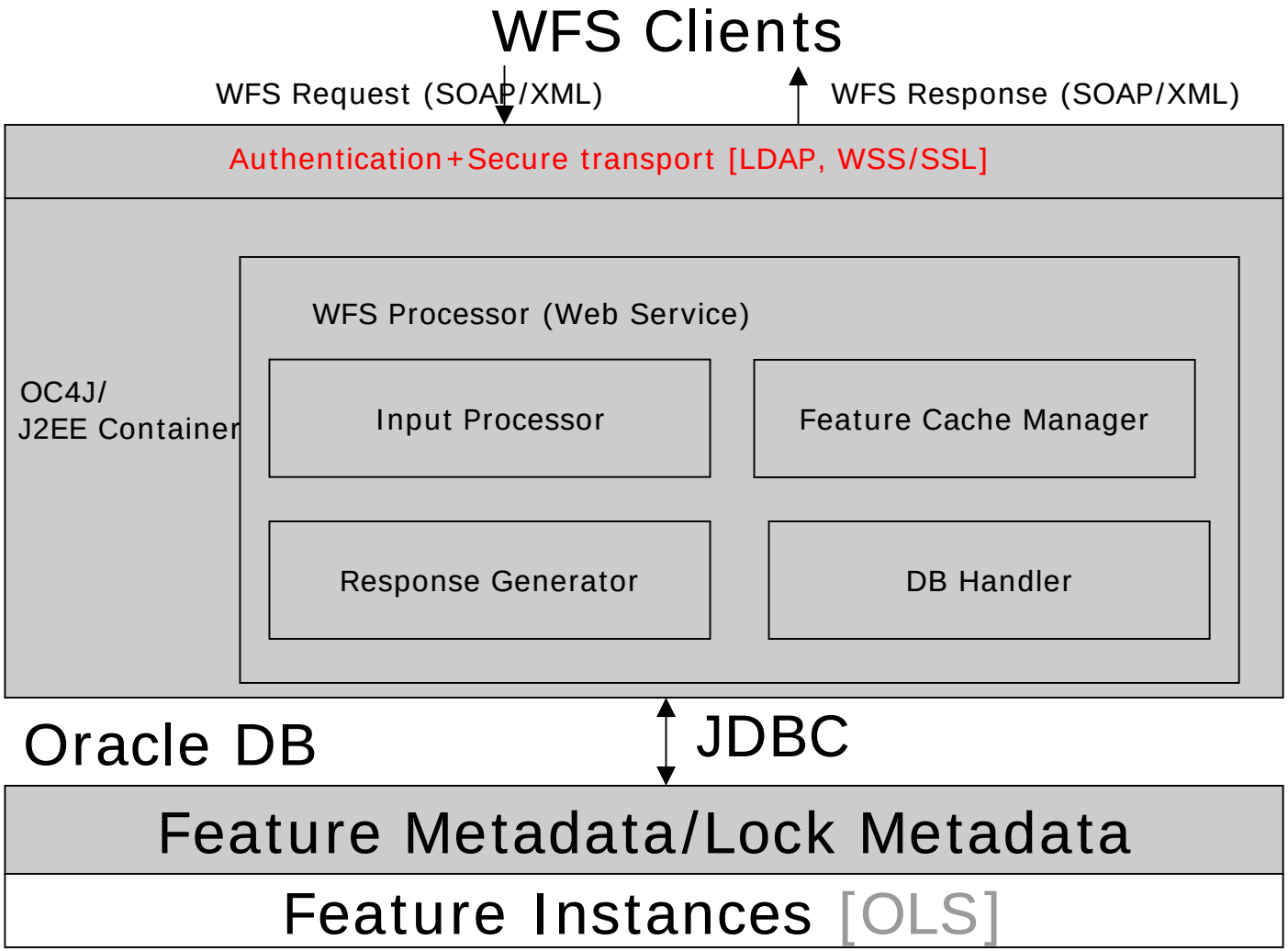
WFS Operations

- 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

WFS Operations (Contd.)

- Transactional WFS
  - 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



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.

Caching APIs

- Lookup (type, IdList)
- Insert (feature rows)
- Update (type, propertyList, IdList)
- Delete (type, IdList)
- loadCache(Connection)
- loadCache(Connection, type)
- Lookup/Update/Delete (filter-based)

Locking

- DB Locking
- Lock duration in minutes
  - Spans db transaction boundary
- Token-based locking semantics
- Unlock rows when lock expires

Locking APIs

- lockRows(type, filter) return token;
- transferToken(token);

…update/delete feature rows

- unlockRows(type, filter);

Update/Delete without proper lockRows/transferToken will be unsuccessful

Locking Approach

- 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.

WFS Metadata

- Feature Types
- Feature Type Tags
- Feature Type Attributes
- Complex Types
- Spatial Operators supported
- Functions supported
- Service Metadata

Publish Feature Types

- Default
  - 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
  - Java/PLSQL APIs

Publish Types (Contd.)

- Custom
  - Feature Type Mapping
  - Feature Type Tag Mapping
  - Feature Type Attribute Mapping
  - Complex/Nested Types Mapping
  - GML feature fields map to SDO_GEOMETRY type in the db
  - Complex type maps to db object types
  - Custom Storage Mapping XSD

Summary

- WFS 1.0 (GML 2.1.2)
- Basic:
  - GetCapabilities
  - GetFeature, DescribeFeature
- Transactional:
  - Transactions (Insert, update, delete)
  - LockFeature
  - GetFeatureWithLock

Catalog Service Web 2.0

Requirements

- Access/search/update/delete geo-spatial catalog records 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 catalog record privileges at a instance level
- Real-time transfer of catalog instances in a platform/programming language independent way

Our Approach

- Use SOAP/XML over HTTP for Request/Response
- Oracle Spatial for Record instance Storage/Retrieval
- Implement GML filter specification for record search
- Use LDAP for authentication, Oracle Label-based security for instance-level privilege mgmt, and WSS/SSL for secure transfer of record data
- Support brief/summary/full output views of record instances.We will also support custom views of record instances, by dynamically generating XSL transformations based on the CSW Query specification

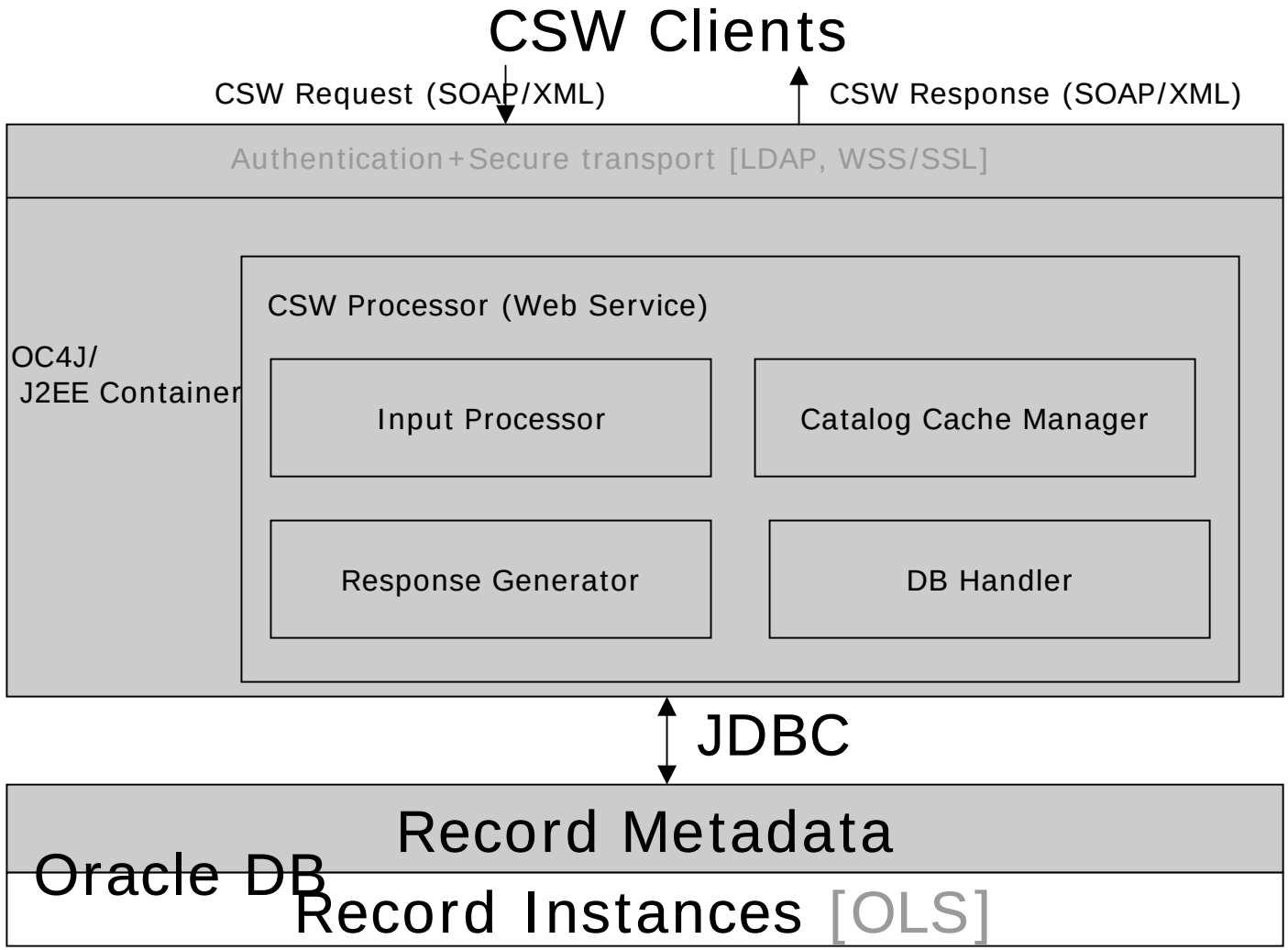
CSW Operations

- Discovery
  - Get Capabilities – get the metadata about the types / operations a catalog server supports
  - DescribeRecord - get the structural information about a catalog record type
  - GetRecords– query different parts of record instances

CSW Operations (Contd.)

- Publication WFS
  - Transaction (Push records)
    - Insert new record instances
    - Update existing record instances based on filter criteria
    - Delete existing record instances based on filter criteria
  - Harvest (Pull records from external source)

Architecture

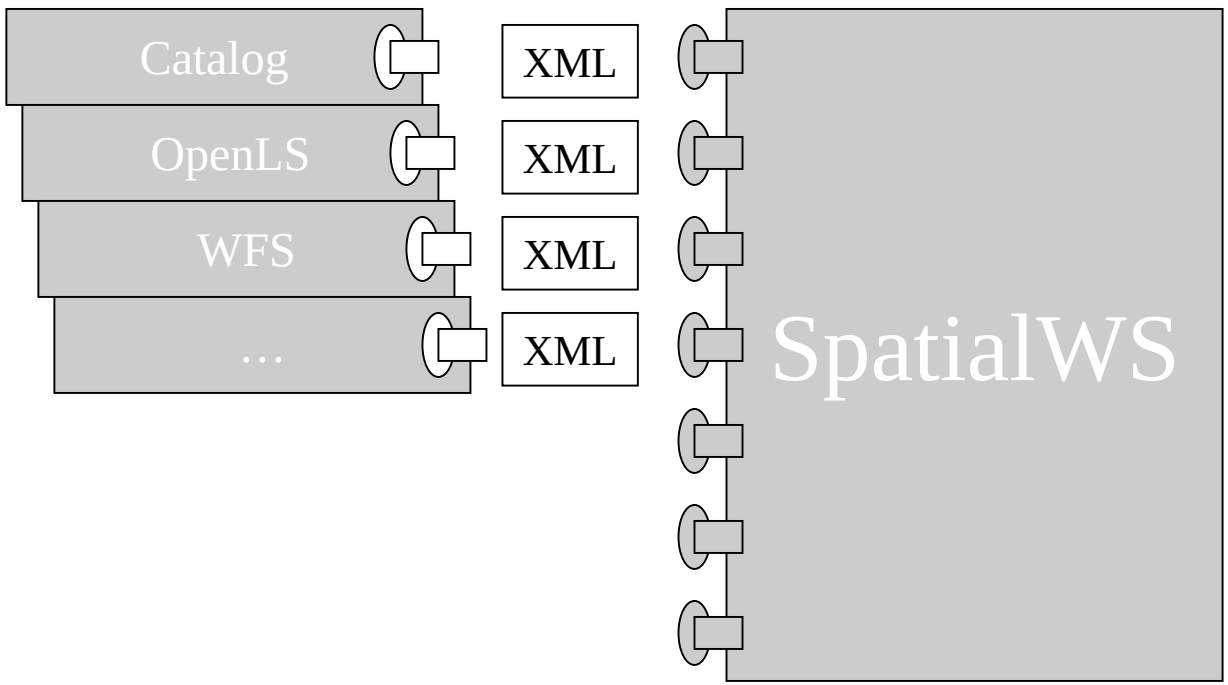


Summary

- CSW 2.0 (GML 2.1.2)
- Discovery:
  - GetCapabilities
  - DescribeRecords, GetRecords
- Publication:
  - Transactions (Insert, update, delete)
  - Harvest

SpatialWS Security Integration Pluggable Components

What…



OGC compliant
SOAP
WSS,
VPD

Proxy Auth/App user mgmt

(Multi user login)

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

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

GeoRaster: 11g enhancements

- Raster processing
  - GeoTIFF import, export
  - New georeferencing techniques (RPC, DLT)
  - partial raster updates, multiple NODATA values, sparse raster support
- Compression
  - JPEG 2000
  - Plugin architecture for 3rd party formats
- Workspace Manager Support
- OLS Support

NDM: 11g Enhancements

- Scalable Searching on large networks
  - Large networks are supported (millions of nodes/links)
  - Network is partitioned using logical or spatial keys
  - Partitions are loaded on demand to keep the in memory foot print small
- New Features
  - Trace out
  - Partial links as end nodes in a path result
- Workspace Manager Support

Product Development Contacts

- Development Manager: Siva.Ravada
- Product Manager : Jayant.Sharma