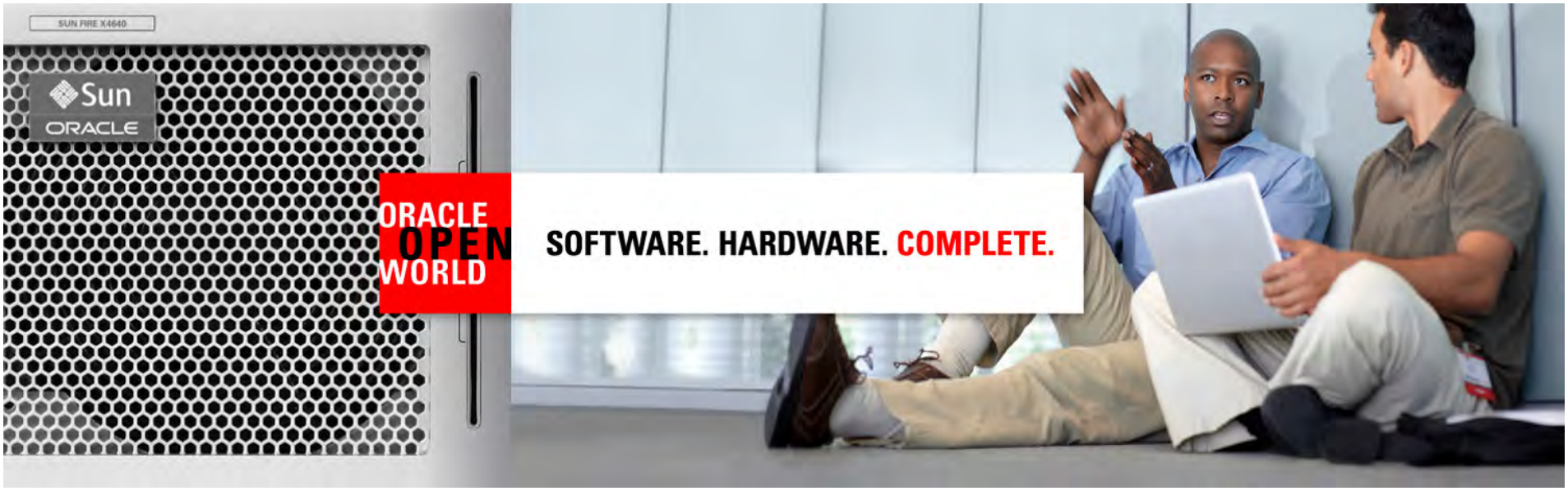

ORACLE®

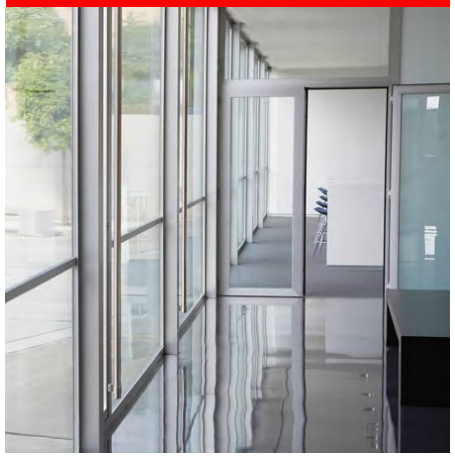


Oracle Spatial 11g Release 2 New Features Deep Dive

Siva Ravada
Director of Development, Oracle Spatial

Michael Smith
US Army Corps of Engineers

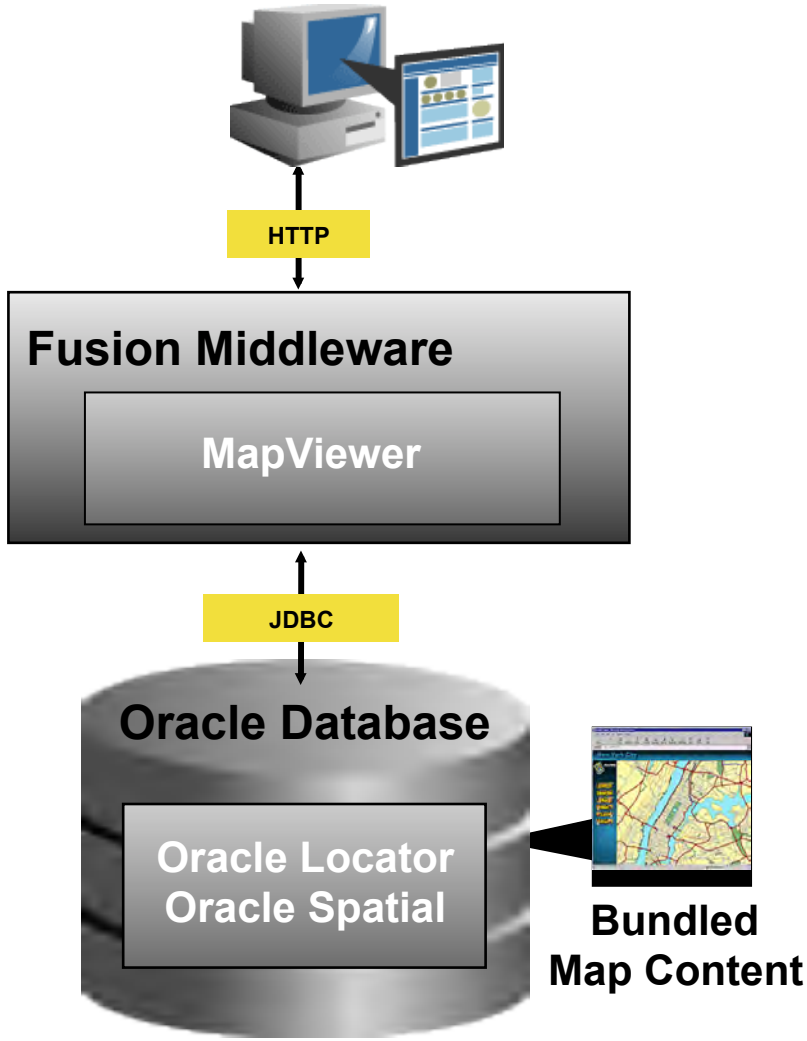
Agenda



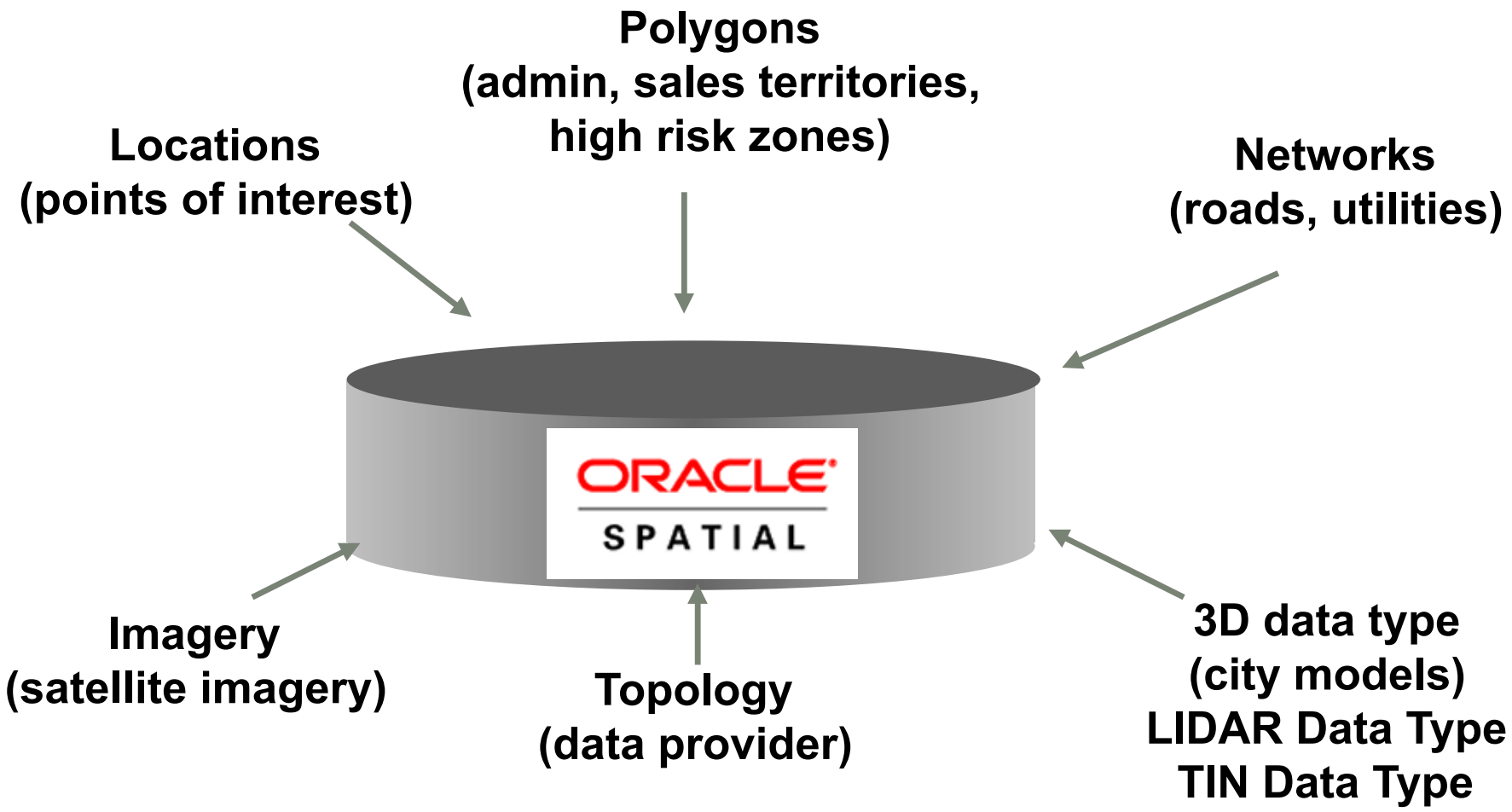
- Introduction to Oracle Spatial
- Spatial Engine
- SQL Developer
- GeoRaster
- Network Data Model
- Moving Objects
- 3D Support

Oracle Spatial Features

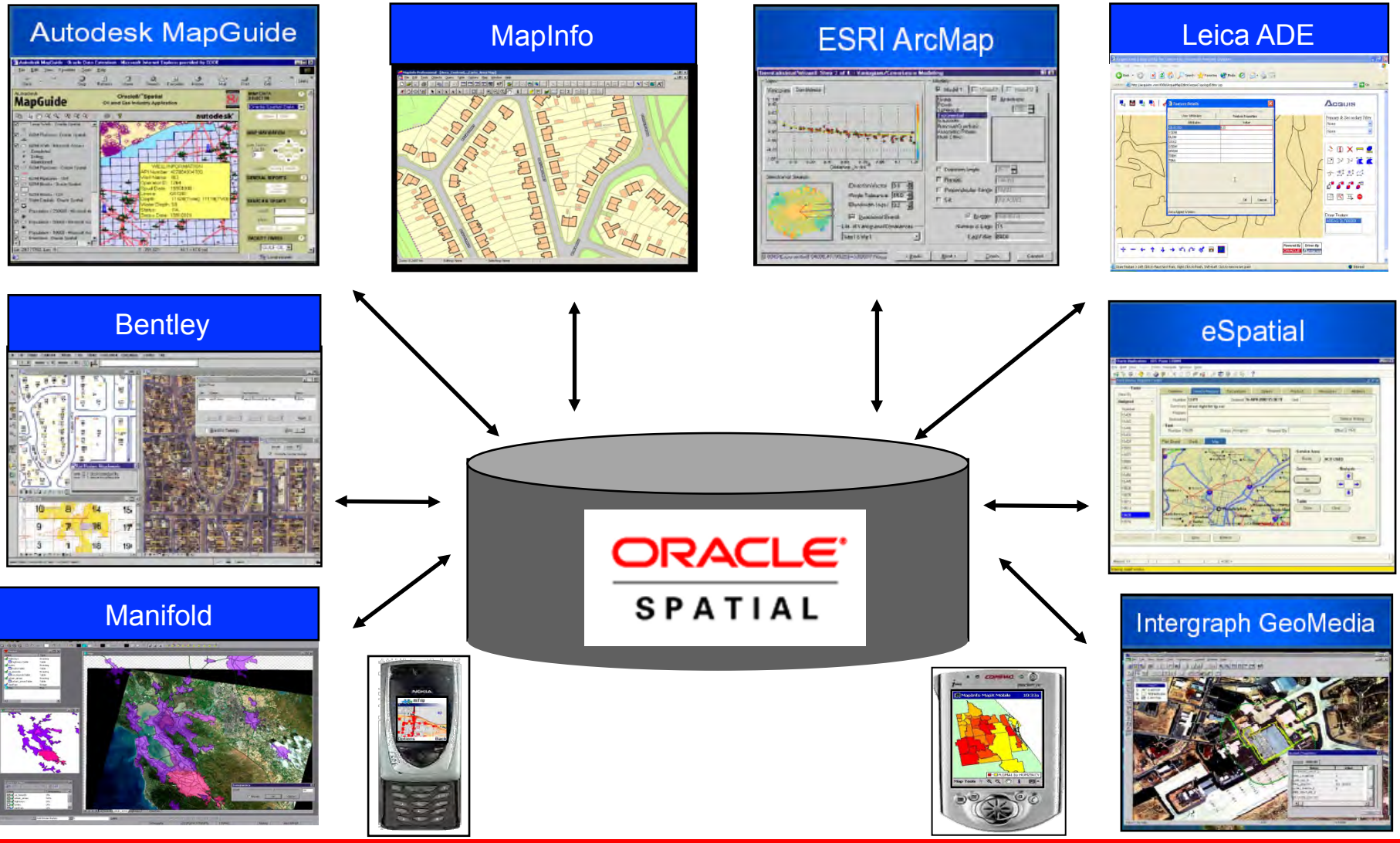
- **Oracle Locator**: Feature of Oracle Database XE, SE, EE
- **Oracle Spatial**: Priced option to Oracle Database EE
- **MapViewer***: Java application and map rendering feature of Fusion Middleware
- **Bundled Map Content**: Major roads, administrative boundaries (city, county, state, country) - worldwide coverage from Navteq



Supports all Geospatial Data types

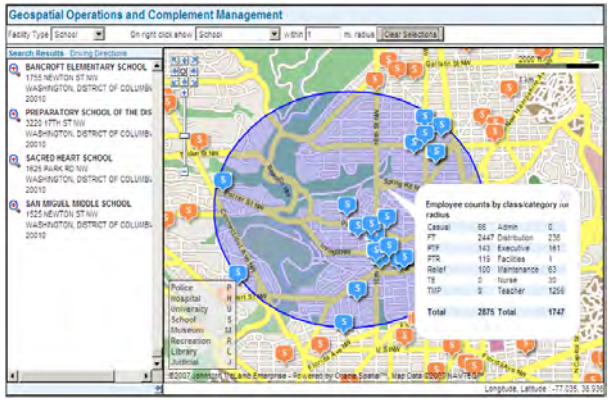


Open Platform for Leading Applications and Tools

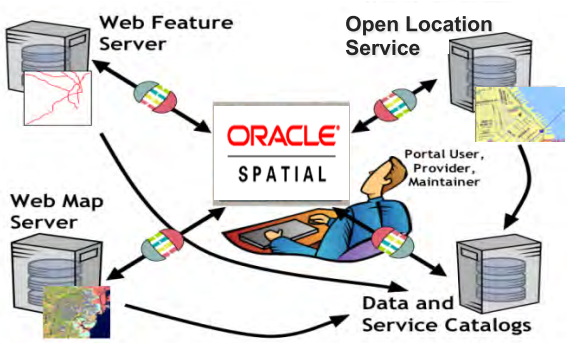


Oracle Spatial 11g Enabled Applications

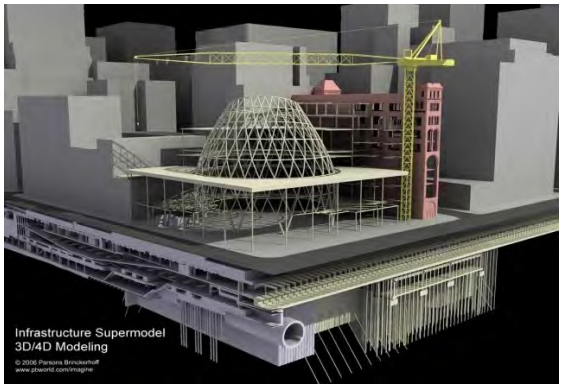
Scrollable, Interactive Maps



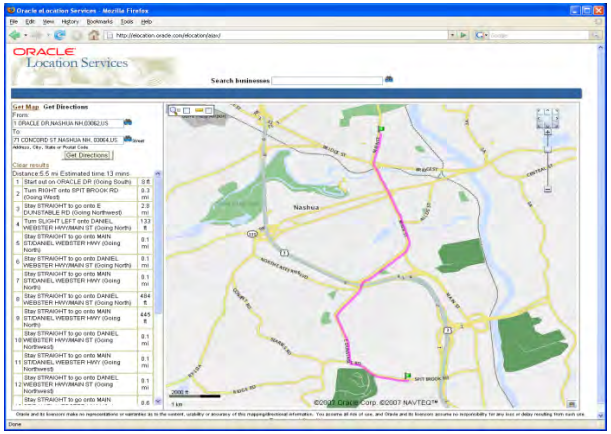
Spatial Web Services



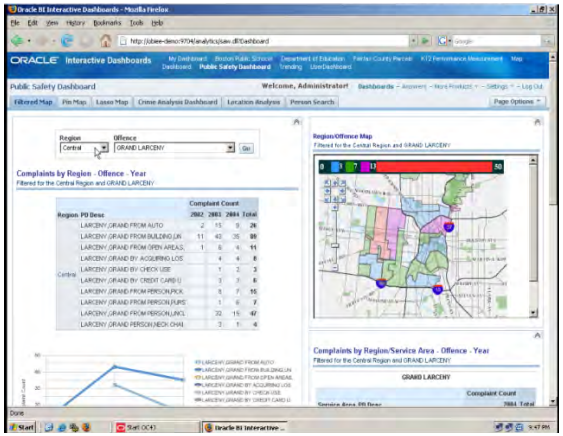
3D, Point Clouds, and LIDAR



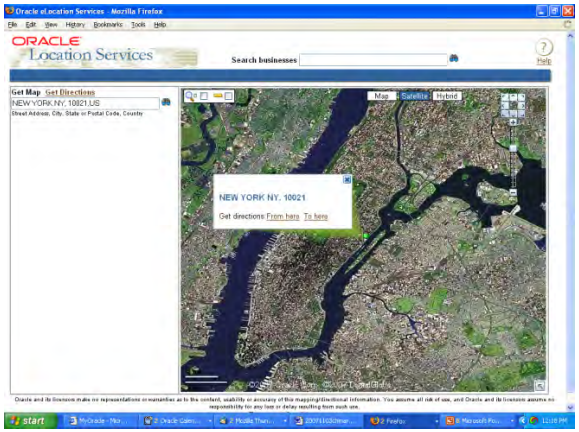
Geocoding & Routing



Oracle BI Dashboards



Raster Imagery



ORACLE


Spatial Engine
SOFTWARE.
HARDWARE.
COMPLETE.
TAPE
~AC ⎓DC
STANDBY
XSCF
ORACLE

Spatial on Exadata

- We achieve performance improvements based on the exploitation of the processing power, bandwidth, and parallelism of the Exadata machine
- Parallel query and partitioning help improve Spatial performance
- All the row level functions are also parallel enabled to take advantage of parallel query
- Speedups of up to 40 times (compared to a 1-core CPU box) for the typical SDO_ANYINTERACT spatial query
- Spatial predicates are not currently pushed into the Exadata storage nodes

Spatial Index Parallel Creation on Exadata

- Build a Local Spatial Index on Partitioned Table: each slave can build a different index partition in parallel.

```
|   0 | CREATE INDEX STATEMENT                    |
|   1 |   DOMAIN INDEX BUILD (LOCAL)              |
|   2 |    PX COORDINATOR                         |
|   3 |     PX SEND QC (RANDOM)                   |
|   4 |      PX PARTITION RANGE ALL               |
|   5 |       DOMAIN INDEX PARTITION BUILD        |
```

Spatial Parallel Query on Exadata

- Use Local Spatial Index on Partitioned Table
  - each slave can work on a different partition in parallel
  - Efficiently uses all resources (CPU, Memory, and I/O) in Exadata.

```
| PX PARTITION RANGE ALL                    |
|   TABLE ACCESS BY LOCAL INDEX ROWID       |
|     DOMAIN INDEX CUSTOM OPTION            |
```

Fast Coordinate System Transformations

- SDO_CS.Transform is upto 10 times faster in 11gR2
  - This speedup is seen if 1000s of transformations are done in a session
  - This is typical for mapping applications where themes are transformed at run time to match the base map SRID
- In the prior releases, the transformation context was created for each transform() call
  - now we use that context between transform() calls with the same source and traget SRIDs

Interior Point

SDO_UTIL.INTERIOR_POINT

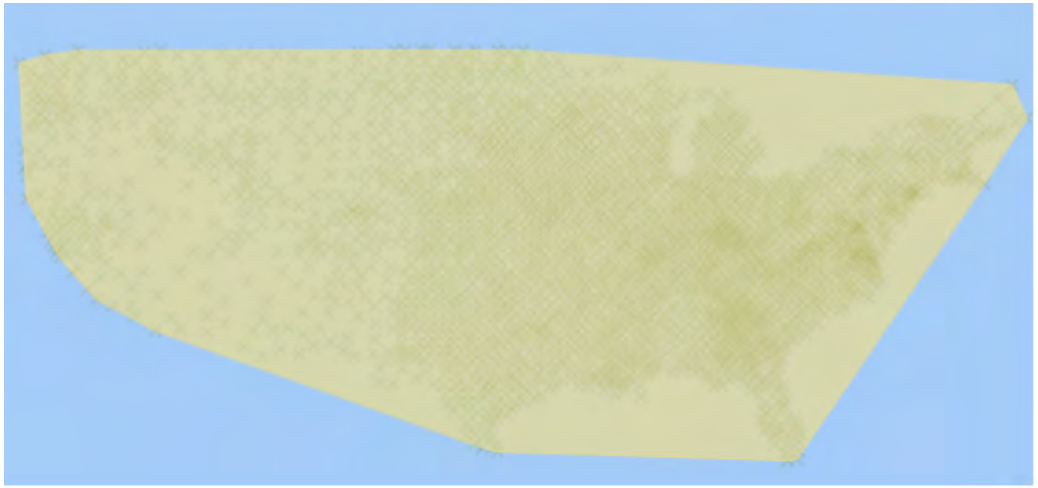
- This computes a point that is interior to the given polygon
- Input geometry should have a polygon type
- This is very useful for applications that look for a suitable label location in polygonal features
- An interior point is returned which is guaranteed to be INSIDE the given polygon, for any shaped polygon
- This is different from sdo_geom.point_onsurface which is not guaranteed to return a point INSIDE the given geometry

Concave Hull

SDO_GEOM.CONCAVEHULL

- This computes a polygon that represents the area occupied by a set of points in the plane
- Input geometry should be a 2D Geometry of any type
  - This is typically a multipoint geometry
- A concavehull is returned which is guaranteed to be a valid polygon (gtype 2003)
- Very useful for generating trade areas from a given set of points of interest (e.g., customers)
- This is different from convexhull

Convex Hull vs Concave Hull



Transparent support for TTS

- Cross-Endian support for TTS and Spatial Index
- SDO_UTIL.PREPARE_FOR_TTS Deprecated
  - When spatial data is transported as TTS, there is no need to do any special processing for Spatial indexes
  - This applies to any TTS export done with 11gR2 database
- SDO_UTIL. INITIALIZE_INDEXES_FOR_TTS now automatically fixes the index if the TTS export is done in a different endian format than the target format

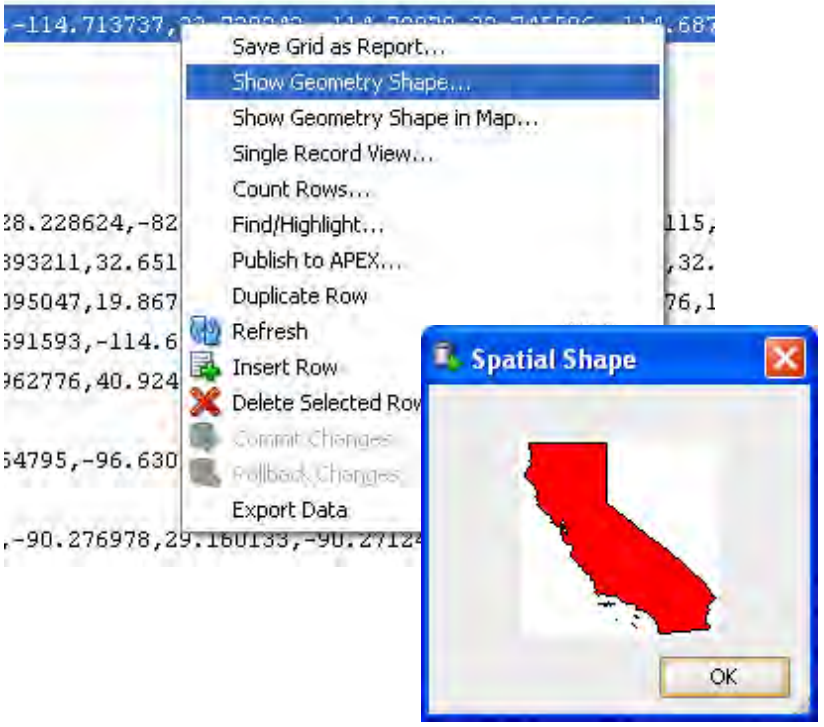

SQL Developer
SOFTWARE.
HARDWARE.
COMPLETE.
TAPE
~AC ⎓DC
STANDBY
XSCF
ORACLE

SQL Developer Support for Spatial

- Support for Spatial Data Management
  - Metadata creation
  - Index creation
  - Validating the Spatial data
  - MapViewer metadata support
- Support for Spatial Data Querying
  - Query results can be viewed graphically

Spatial Data Querying: Single Value

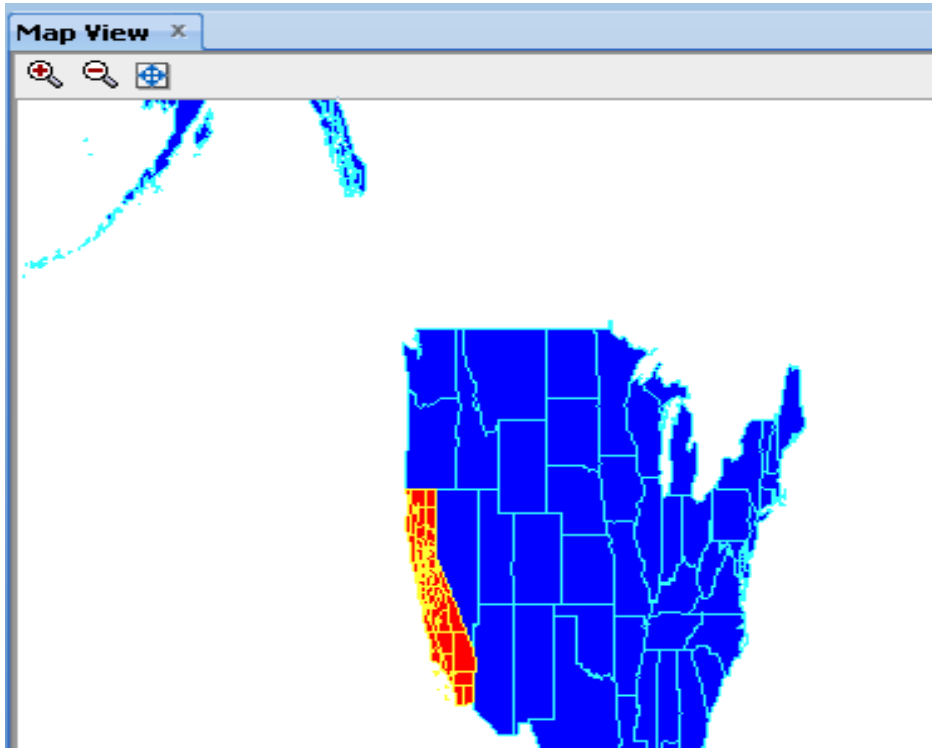
- Spatial shapes can be graphically viewed



Spatial data querying: Map View Display

- Secondary queries are overlayed

Here we show the counties in California over the States layer

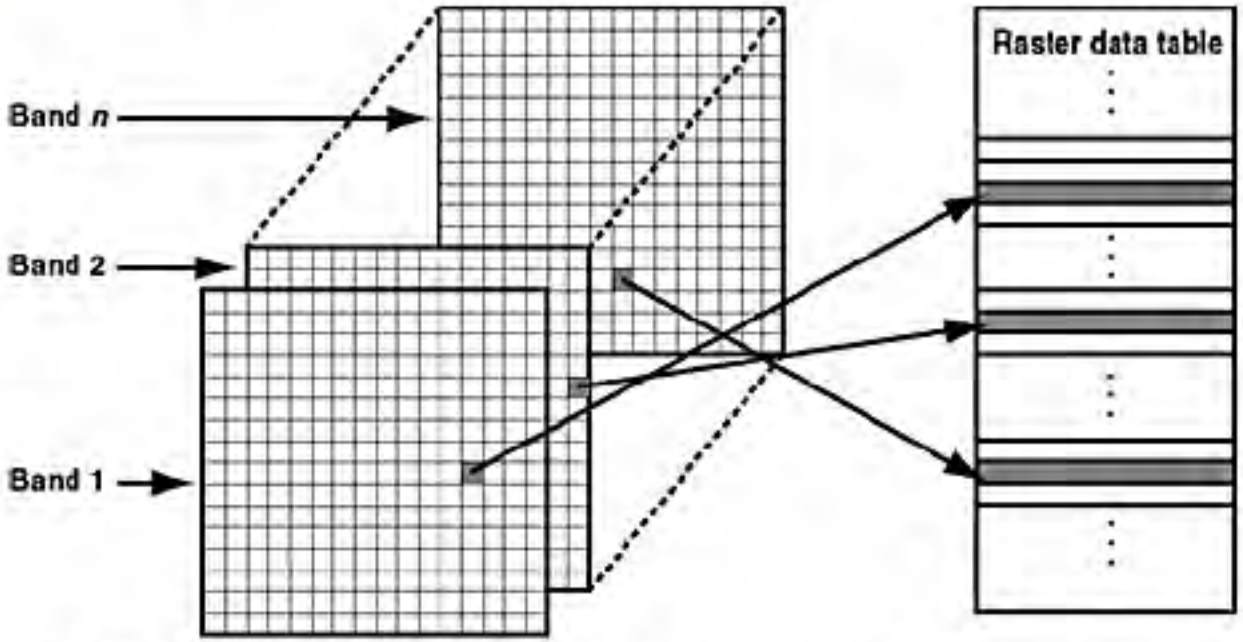



GeoRaster
SOFTWARE.
HARDWARE.
COMPLETE.
TAPE
~AC ⎓DC
STANDBY
XSCF
ORACLE

GeoRaster

- A data type to store raster data
  - Satellite images, remote sensing data, digital elevation models
  - Multi-band, multi-spectral
  - An XML schema to store Metadata
    - Data source, layer information
  - Geo Referencing information
- Functionality
  - storage and indexing of raster data
    - No size limit for each raster object
  - Generate resolution pyramid
  - query and analysis
  - delivery to external consumers
    - Publish as JPEG, GIFF images

Raster Bands, Layers, and Blocks



- RGB Image: 3 Bands (Red, Green, Blue)
  - Consider an Image with 8192X8192 pixels
  - Each block is 256X256 pixles
  - This results in 1024 blocks for each band image
  - 3072 blocks for the whole image
- Hyperspectral image: hundreds of bands

ETL tool – the GDAL GeoRaster Driver

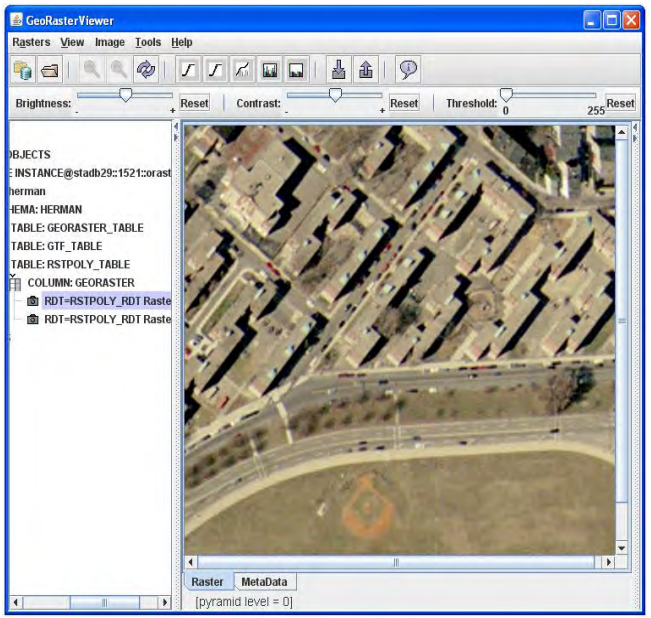
- GDAL is the best open source geospatial ETL tool/ API for raster data
  - It now natively supports importing and exporting many formats, to/from SDO_GEORASTER, including GeoTIFF, JPEG2000, ECW, NITF, HDF, NetCDF, ERDAS IMG, USGS DEM, SPOT, and more.
  - GDAL is written in C++ and has great performance
  - It provides C/C++, Java, Python API for accessing GeoRaster
  - It provides many tools. Two of the important ones are:
    - gdal_translate – utility to translate raster formats to/from GeoRaster objects
    - gdalinfo – utility to view information about a raster, such as a GeoRaster object

Georeferencing Using GCP

- Georeferencing is the process of mapping pixels on the image to ground coordinates
- GCP: stands for Ground Control Points
- GCP Model: GeoRaster supports a generic GCP model. In the current release, 2D cell coordinates, 2D and 3D model coordinates are supported.
- GCP Storage: GeoRaster defines a GCP XML schema and can (optionally) store GCP natively in the metadata of GeoRaster objects.
- GCP Manipulation: GeoRaster provides a set of update and query functions to manipulate GCP's and related data

The GeoRaster Java API

- Sample Applications: using this Java API, users can easily develop simple applications, particularly web-based applications. GeoRaster viewer is a simple application built with this API.



Tools.java | **Loader.java** | **Viewer.java** | **Exporter.java**

The GeoRaster Java API

- The GeoRaster JAVA API
  - *oracle.spatial.georaster:* Provides a complete mapping of the SDO_GEORASTER object type and its metadata to Java objects, and provides support for the core GeoRaster features
  - *oracle.spatial.georaster.sql:* Provides a Java wrapper of the GeoRaster PL/SQL API for some server-side operations
  - *oracle.spatial.georaster.image:* Provides support for generating Java images from a GeoRaster object and for processing the images

- The image package is based on Java 2D and JAI
  - This allows users to leverage all the strength and advanced capabilities from Java 2D and JAI. Users can easily develop web applications and other image processing applications

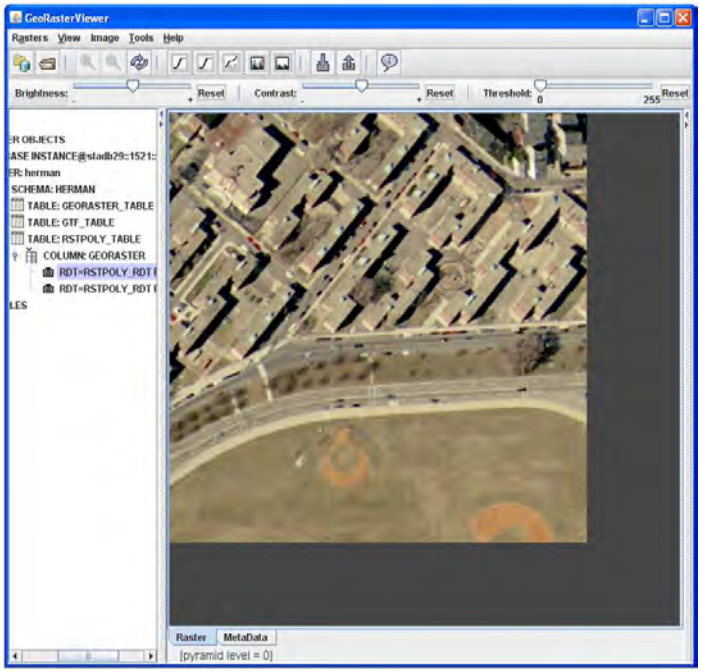
Raster CS Transformations

- SDO_GEOR.reproject: transform GeoRaster raster data from one projection to another projection
  - All oracle spatial coordinate systems are supported
  - Five re sampling methods are supported
    - NN, Bilinear, Cubic, Avarage4 and Average16.
- Supports two options
  - Reproject persistently. Reprojects a GeoRaster object and stores the result as a new GeoRaster.
  - Reproject on-the-fly. This is equivalent to getRasterSubset except the window-based cropping result is transformed into a different projection. The result is stored as a single BLOB.

Raster Reprojection – An Example

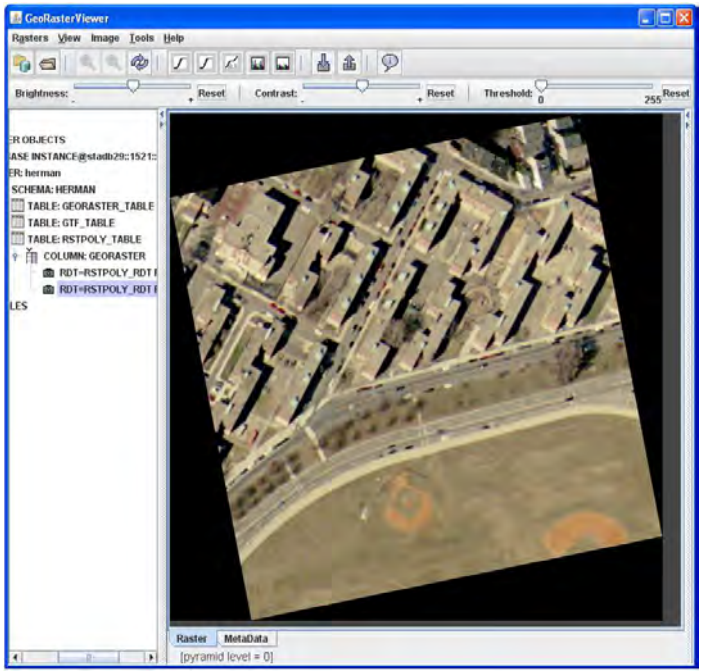
sdo_geor.reproject (gr1, 'resampling=cubic', ‘blocksize=(256,256,3)

interleaving=BIP', 26988, gr2);



From: SRID 26986

"NAD83 / Massachusetts Island"



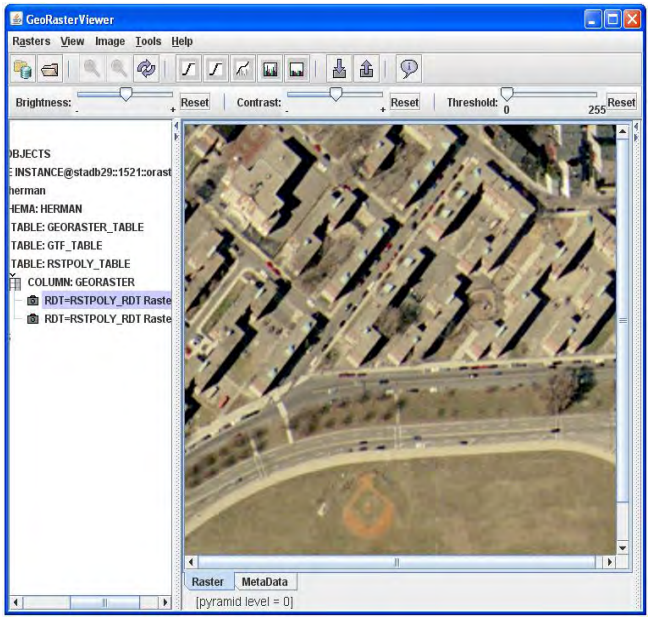
To: SRID 26988

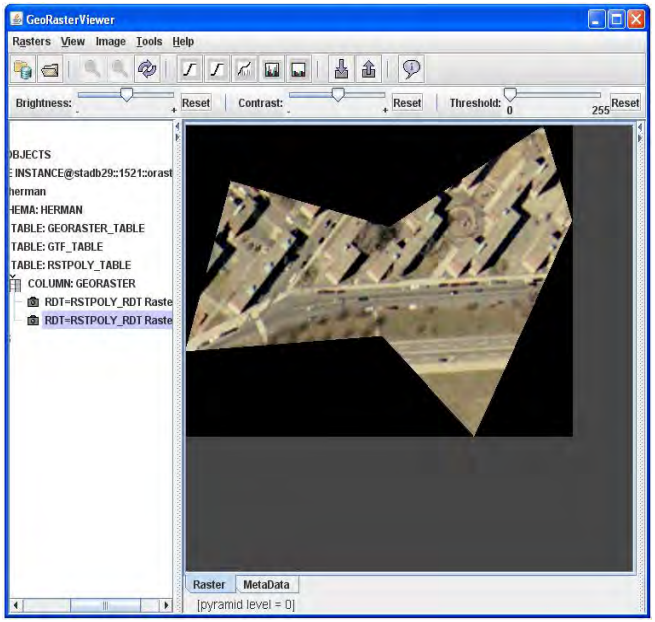
"NAD83 / Michigan North"

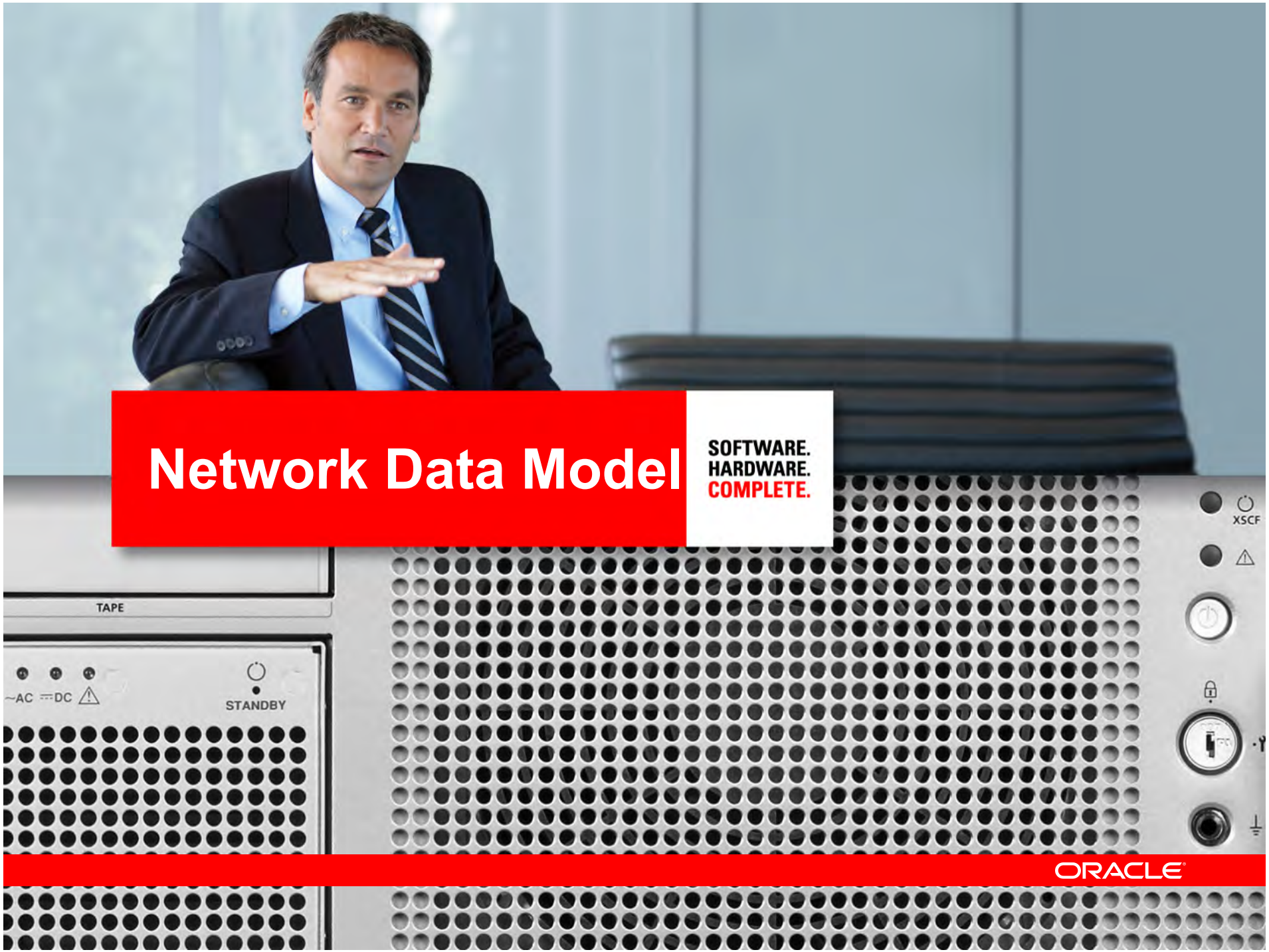
Polygon-based Raster Clipping

SDO_GEOR.getRasterSusbet

- Allows users to clip the query result along the polygon (irregular) boundary






Network Data Model
SOFTWARE.
HARDWARE.
COMPLETE.
TAPE
~AC ⎓DC
STANDBY
XSCF
ORACLE

Sample Application: Shortest Path Analysis

Street Addresses

747 Howard Street, San Francisco, CA →
1099 Lombard Street, San Francisco, CA

Geocoder

Network Nodes and Links

5% on link 1234567 → 10% on link 89101112

NDM Analysis API: Shortest Path

Path Represented in Nodes and Links

Path consisting 500 nodes [….] and 499 links [….]

NDM Network IO API: Read Spatial Path

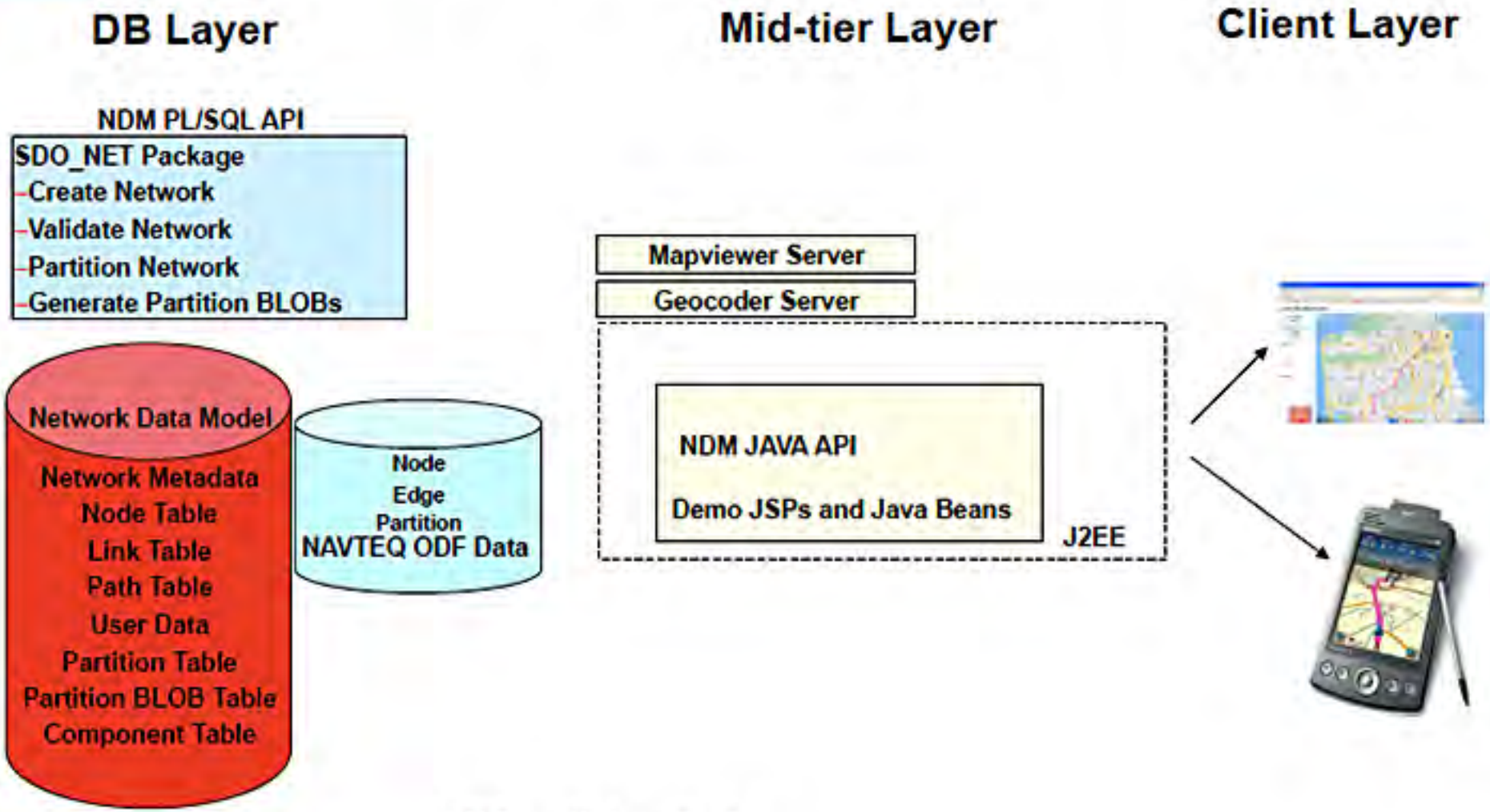
Path Geometry

Path geometry is ….

Oracle Maps Java Script API

Path Displayed On Map

NDM Application Architecture



NDM Demo Web application

NDM Customization

- Users can easily plug in their implementations for
  - Network constraints
    - Trucking restrictions
    - Turn restrictions
    - Avoid toll road
    - Avoid/use Highway
  - Multiple Cost calculators
    - Travel distance
    - Travel time
    - Local/Highway/Scenic Drive preference
    - Penalty of road under construction/traffic congestion
  - Goal nodes
    - Restaurants within 10 min's drive

Multiple Cost Support in Path Analysis

- Path/Subpath analysis now supports multiple costs in a single analysis
- First cost is the one NDM optimizes
- Second cost can be returned as an accumulated value
- Examples
  - Shortest distance paths with travel time
  - Fastest paths with travel distance

Hierarchical A* shortest path Analysis

- Support A* shortest path algorithm
- Provide user defined heuristic cost function
- Support hierarchical shortest path analysis
- Better performance than Dijkstra Algorithm as less number of nodes are explored
  - Uses much less memory than Dijkstra

Traveling Sales Man (TSP) Analysis

- Minimum cost tour that includes all given nodes
- Support Points on Network as nodes to be visited
- Node visit order can be enforced using network constraints
- This is useful scheduling problems where a service representative has to visit a number of customers

Drive Time Polygon Generation

- A spatial representation (polygon) based on minimum cost network coverage
- Concavehull polygon or convexhull polygon (accuracy and performance)
- Example
  - Compute Drive Time Polygon (with travel time as link cost) from a service station
  - Use the polygon to determine if a given address can be reached within a given time using point in polygon operation

Routing with Traffic Patterns

- Fastest Routes are based on speed limits
  - Not very accurate during rush hours
  - A good non-rush route can be the worst rush route!
- Traffic Patterns are historical traffic speeds on roads
  - Can be further classified as speeds on Mon-Thur. Fri. Sat. Sun. holidays, and special events at 15 min, 30 min, 60 min intervals.
- Need to add temporal dimension (start time) in routing with traffic patterns

Routing with Traffic Patterns

- Model Traffic Patterns as user data in NDM
  - Neutral traffic pattern metadata and schema
  - Generate traffic pattern user data automatically
  - Currently Support NavTeq Traffic Patterns
- Link Cost is now a time-dependent function returning actual travel time
  - Start time determines which set of traffic pattern
    - (M-T,Fri.Sat. Sun. Holidays)
  - Link distance/Link speed limit (for non-covered links)
  - Actual Speeds from traffic patterns (interval in a day)
  - Actual Travel Time = Link Distance/Actual Speed

Routing with Traffic Patterns
Fastest Routes at different Start Times

Oracle Spatial Network Data Model Demo

Shortest Path Analysis
Left click for start point, right click for end point, or manually enter node ID, link ID@percentage, or address.

Start 199488837

End 199919135

Network Constraints
(Hold ctrl key for multi-select or de-select)

custom.NoHighwayConstraint
custom.ProhibitedZoneConstraint
oracle.spatial.router.ndm.TruckHeightConstraint
oracle.spatial.router.ndm.TruckLegalConstraint

Prohibited Zone Draw

Link Cost Calculators
custom.TrafficLinkCostCalculator

Keep Previous Results ☑
Reverse Direction ☐

Include Traffic data ☑
Start Time 10:00 PM

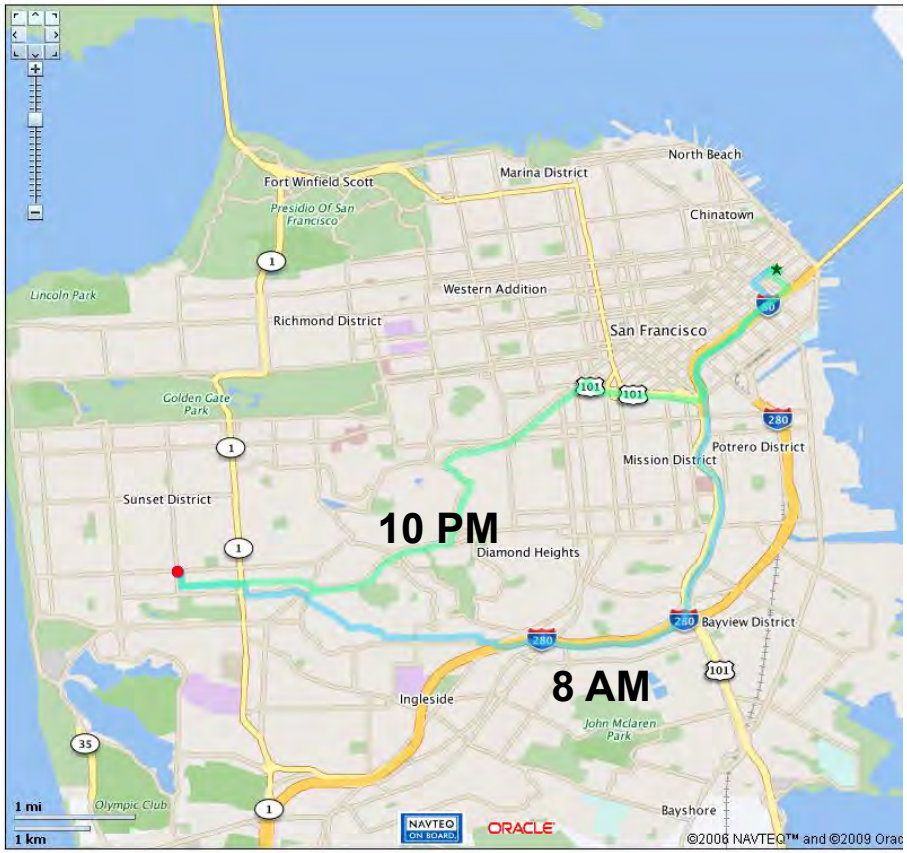
Find Shortest Path

Analysis Result:
(199488837->199919135)
[cost:946.05814, 105 links]

Time to analyze the network: 0.467s.
Time to compute geometries: 0.035s.

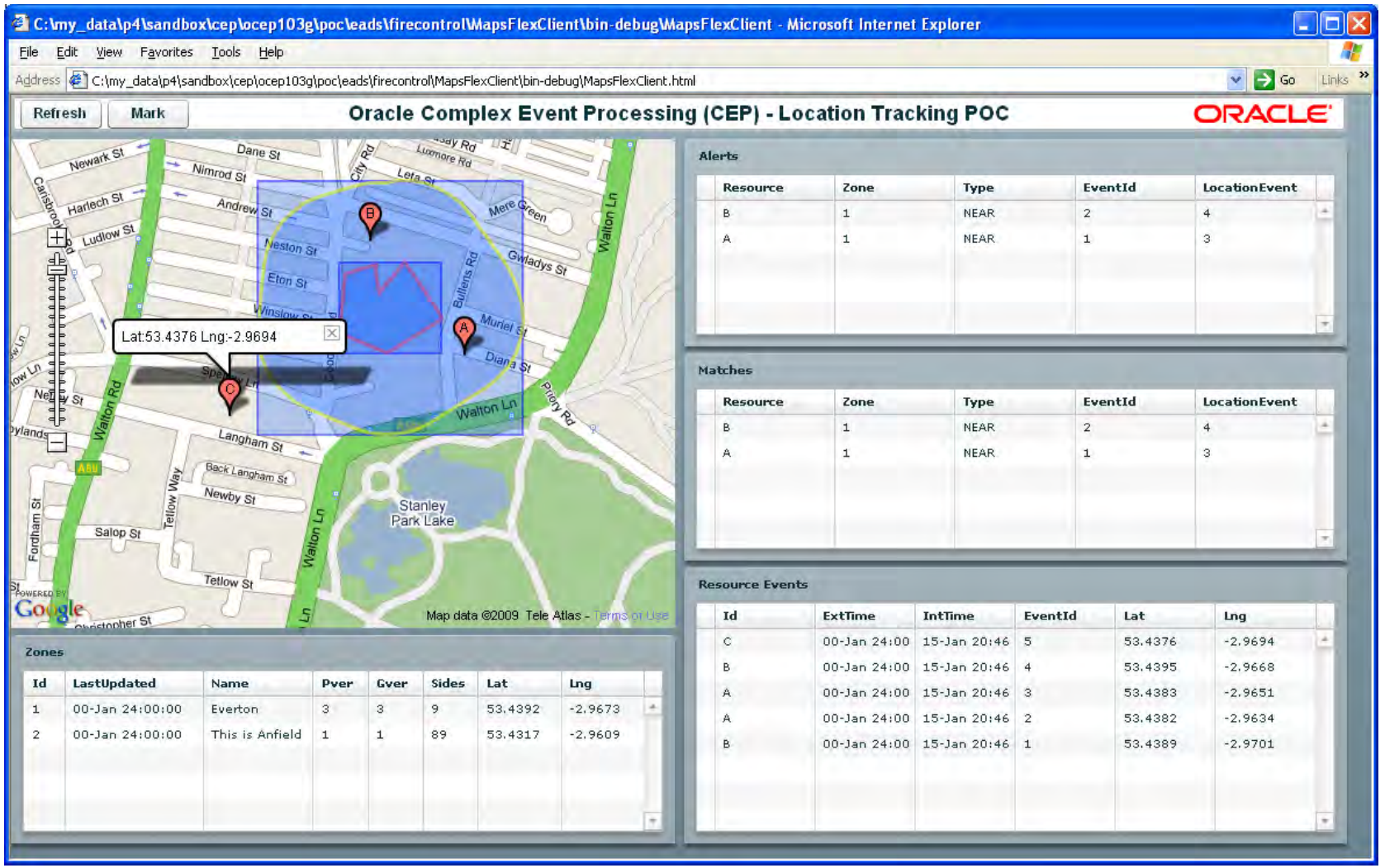
Analysis Result:
(199488837->199919135)
[cost:872.93101, 172 links]

Time to analyze the network: 0.436s.
Time to compute geometries: 0.039s.




Moving Objects
SOFTWARE.
HARDWARE.
COMPLETE.
TAPE
~AC ⎓DC
STANDBY
XSCF
ORACLE

Tracking of Moving Objects

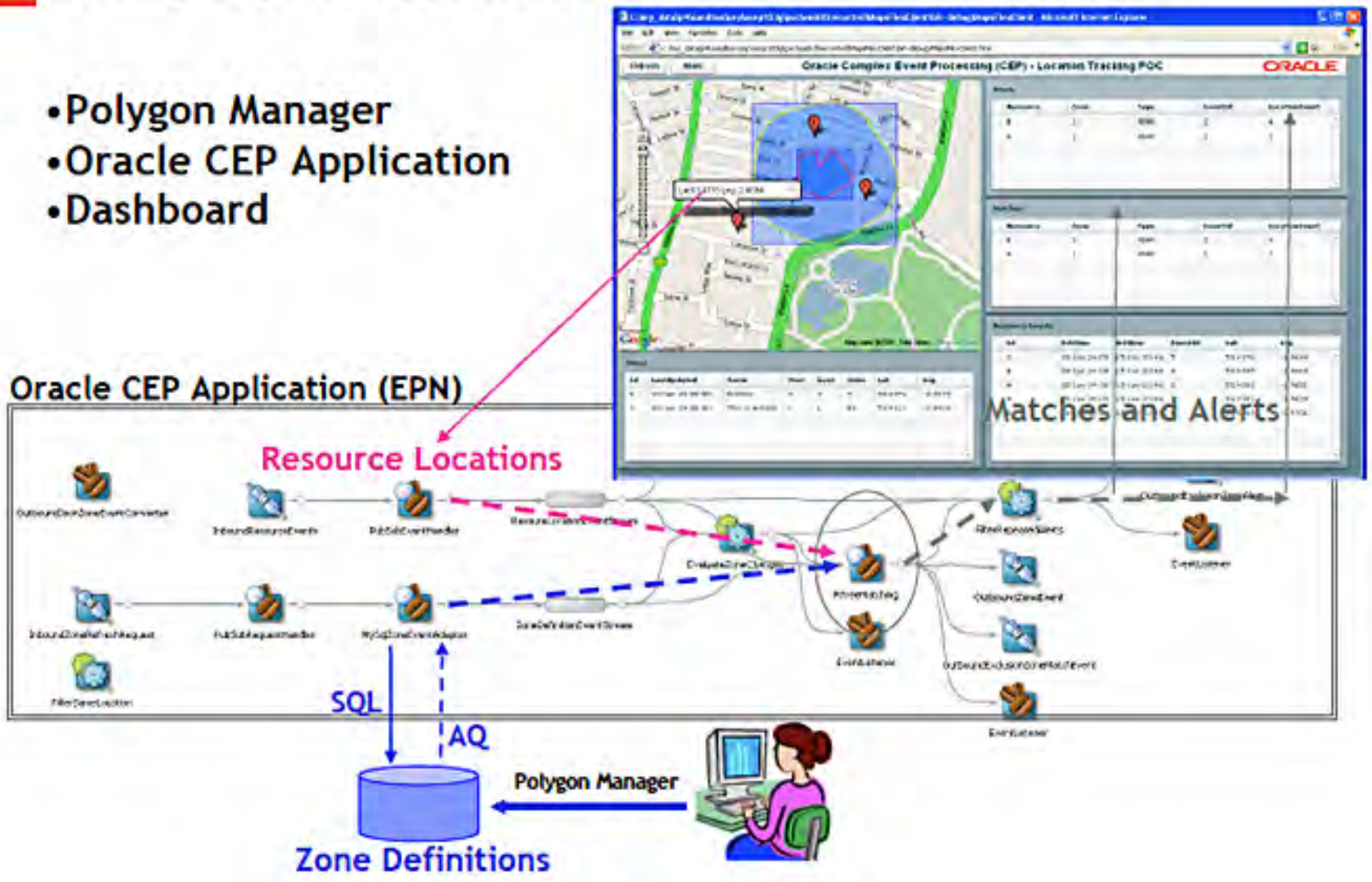


Oracle CEP - Overview

- Designed to target event processing applications
  - Network monitoring and traffic engineering
  - Smart Meters
  - Manufacturing Execution systems
  - Supply chain planning
  - Web logs & Click stream analysis
- Oracle CEP provides a platform to process in-flight & Reference data
  - Domain specific application server capable of extreme low latencies and high throughput
  - Designed to scale-out and be highly available
  - SQL like Query language a.k.a Continuous Query Language

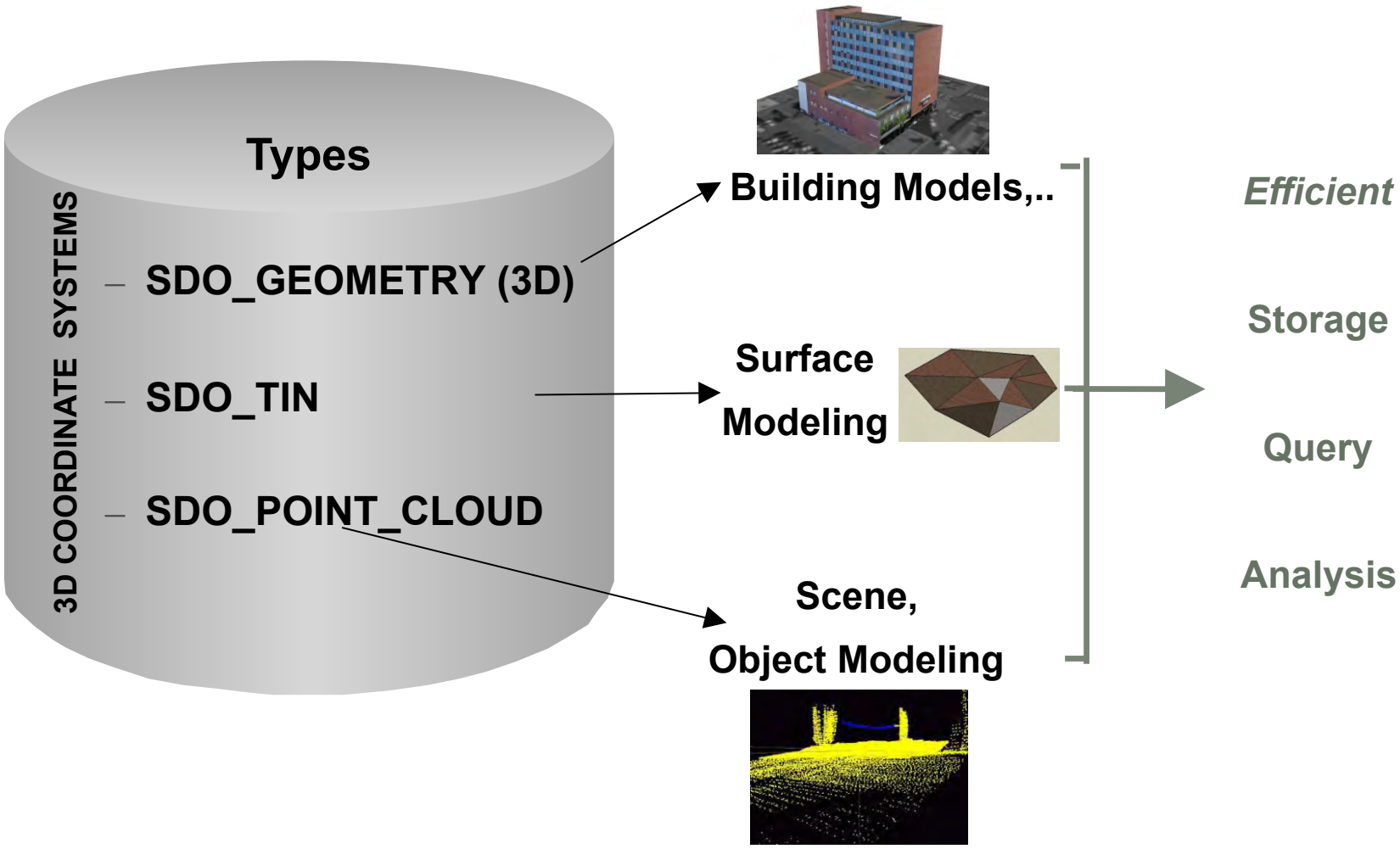
Moving Objects Application

- Polygon Manager
- Oracle CEP Application
- Dashboard



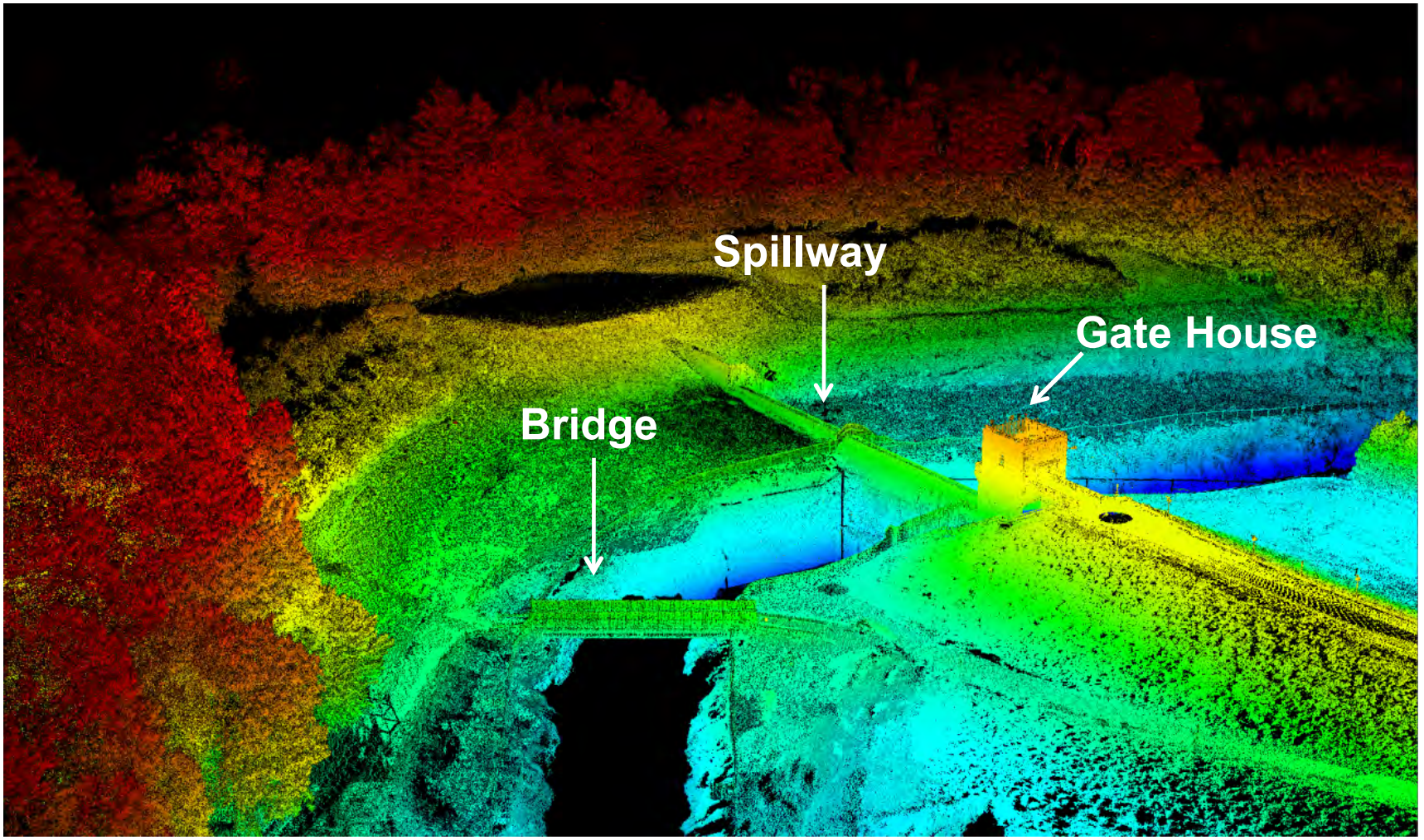

3D Support
SOFTWARE.
HARDWARE.
COMPLETE.
TAPE
~AC ⎓DC
STANDBY
XSCF
ORACLE

3D in Spatial 11g



Raw LiDAR Point Cloud

Color-scaled by elevation (red= high – blue=low)

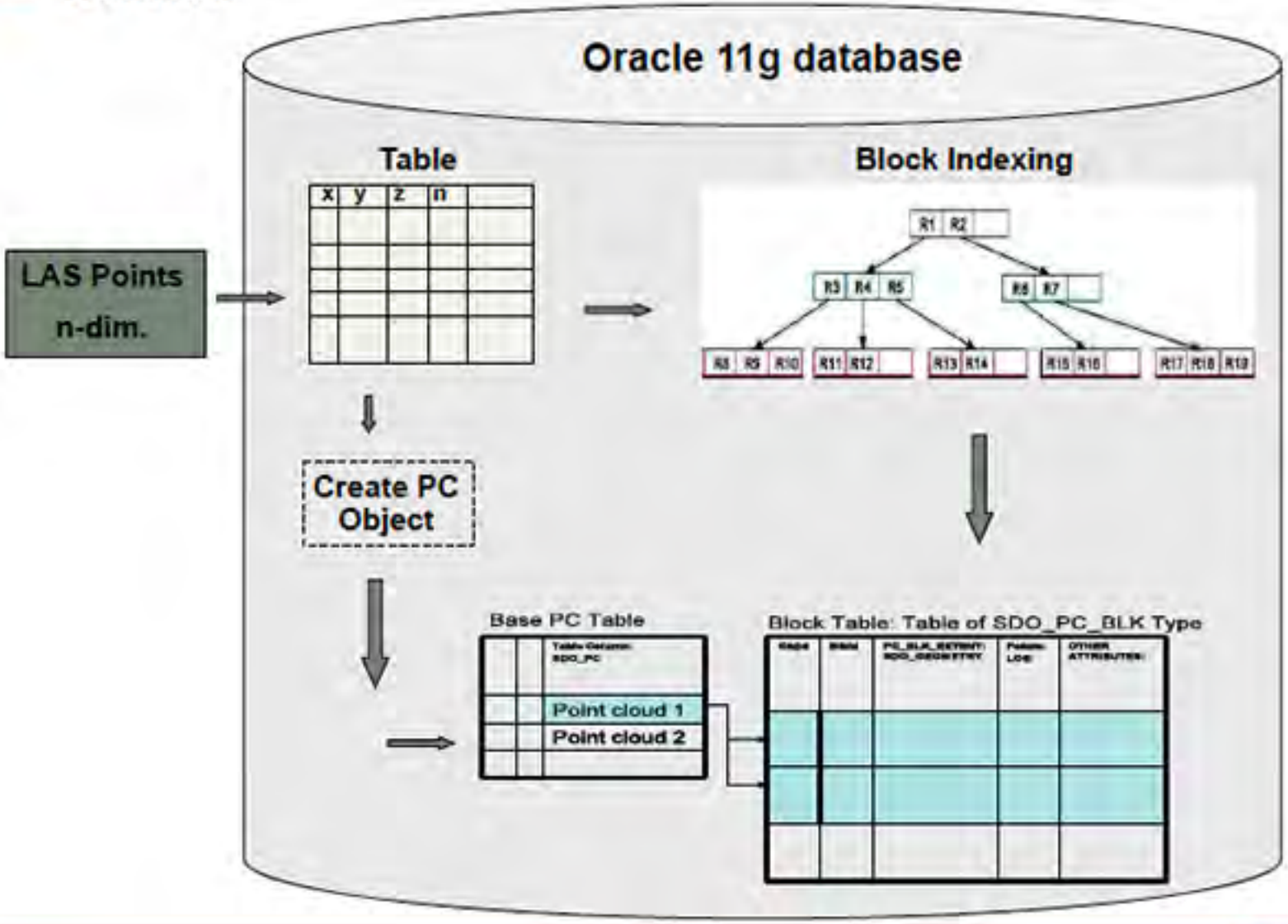


RGB 3D Point Cloud

Oracle Point Clouds (SDO_PC)

- Block organized point clouds.
  - NOT indexed at the point level!!
- Index point cloud data into user defined spatially similar blocks using spatial algorithms (R-tree (2D/3D), KD-Tree or any meeting rules).
- Indexing rules
  - Blocks should not overlap (only in 3D space)
  - Blocks should be filled to optimum capacity
- Each of these blocks has a spatial extent
  - The blocks spatial extent is stored as an indexed geometry (vector)

Oracle Point Cloud Creation Overview – Standard Oracle Workflow



Point Cloud Spatial Data Management / Retrieval

Order of Operation

- **Spatial Operators**
  - Any spatial operator to search through point blocks
    - - SDO_ANYINTERACT
    - - SDO_NN
    - - SDO_FILTER
- **CLIP_PC (Clip Point Cloud)**
    - - 2D or 3D query window
    - - Returns points for any block whose extent intersects the query window
    - - Only points that intersect the query window are returned
    - - Creates a new SDO_PC, can be stored or used in Queries
- **TO_GEOMETRY**
    - - Gets the points (as a Point Cluster) from a PC

Oracle Point Cloud Testing

- Limitations
  - Indexing inside database
    - Indexing algorithm utilized is R-tree only
    - Indexing algorithm was too slow for high density point data
  - Limit on number of dimensions

- Enhancements
  - Added dimensions, up to 12 currently
  - Performance enhancements
    - Retrieval clip_pc, partitioning…
- Enhancements now in 11.2.0.2

Open Source libLAS -> libPC

www.liblas.org

- libLAS is an open source library for reading, writing and manipulating geospatial data encoded in ASPRS LAS file format, version 1.0, 1.1 and 1.2. The libLAS software consists of a base library with bindings available for multiple languages
  - A “quasi” reference implementation of the LAS format.
- Currently utilized by numerous federal and state government agencies along with commercial software vendors as their LAS reader, writer and LAS toolset
- BSD licensed
- Now supports loading LAS data to SDO_PC via LAS2OCI
- Transitioning to libPC supporting *n* Point Cloud format including SDO_PC

Organizations Using libLAS

- Environmental Consulting & Technology, Inc.
- Airborne Research and Survey Facility (at the ARSF Data Analysis Node)
- Dendrometry Laboratory, Laval University, Canada
- National Oceanic and Atmospheric Administration
- OpenTopography (NSF data facility)
- University of California, Davis at the Center for Spatial Technologies and Remote Sensing (CSTARS)
- U.S. Army Cold Regions Research and Engineering Laboratory
- Myriax

Software Using libLAS

- Cadcorp SIS Desktop
- Eonfusion
- LASERDATA LIS
- LizardTech LiDAR Compressor
- TopoDOT Point Cloud Processing Tool Suite
- SAGA GIS

libLAS Enhancements to support Oracle Point Clouds

- External LAS pre-processing
- Oracle loading
- Oracle unloading
- Multi-dimension mapping and support

  - Indexing

    Generate spatial index for a LAS file (external)

    Supports R-tree (3d), KD-tree, and new blocking algorithm

  - Loading - las2oci (*Oracle Call Interface*)

    Load indexed LAS data to Oracle database

    - Invoke pre and post SQL and user defined functions or procedures housed within the database (*e.g. loading metadata or further processing*)

  - Unloading - oci2las

    Generate LAS files from user defined geometries and point clouds retrieved from the database

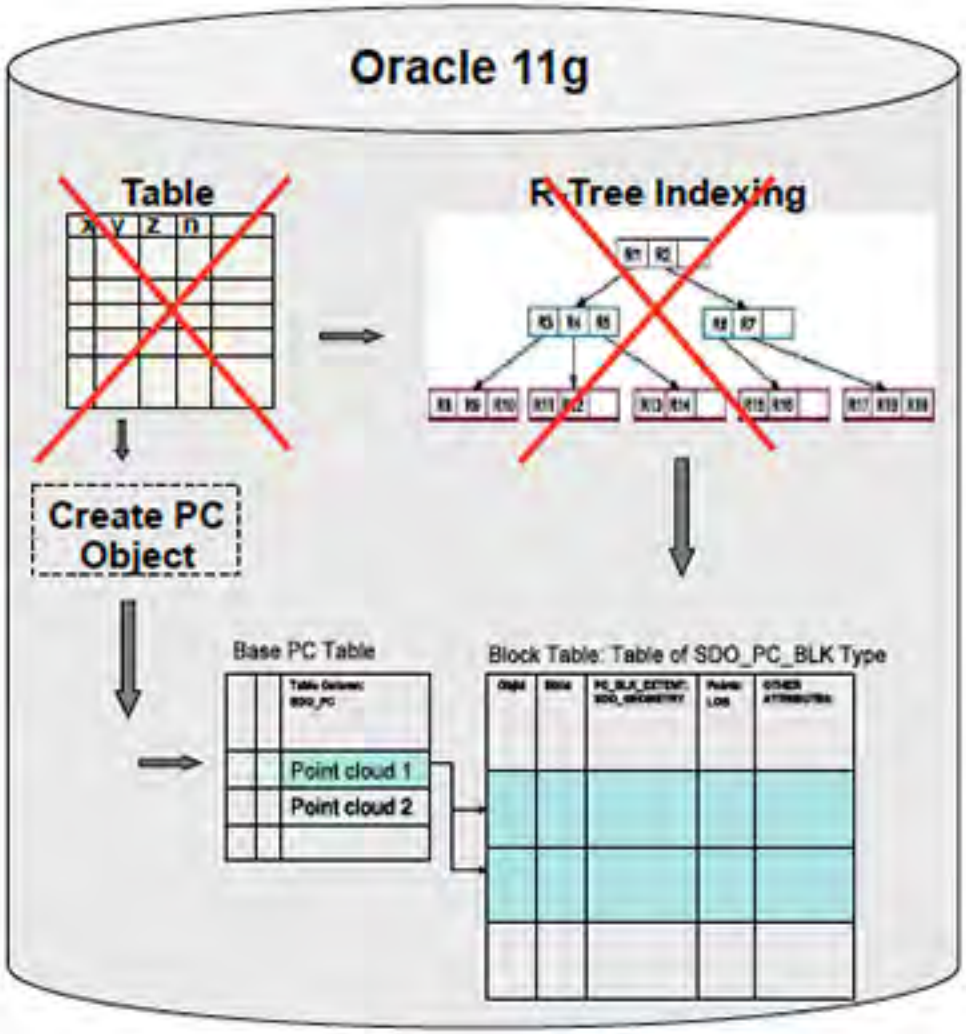
Oracle Point Cloud Creation Overview – New Workflow

Oracle Workflow (old)

1) LAS file
2) Create loading database table
3) Create PC BASE table
4) Initiate PC object
5) Initiate index/load

libLAS workflow (new)

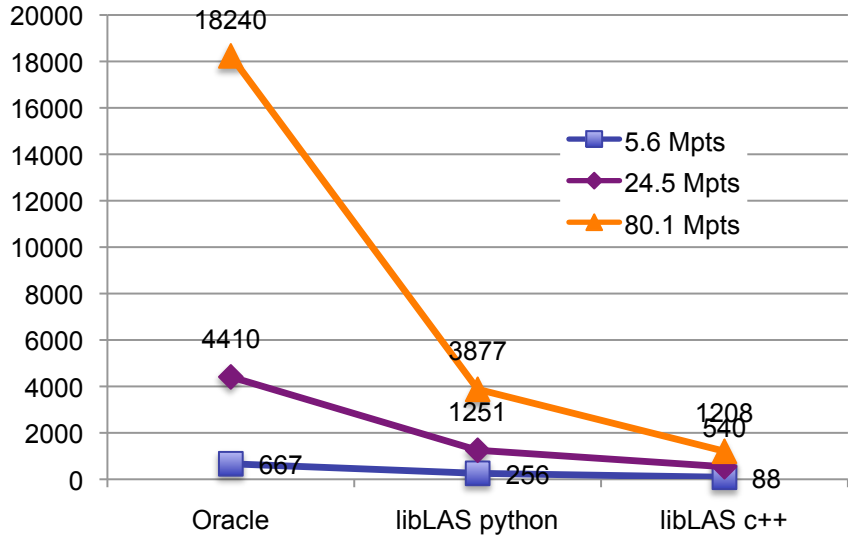
1) Index/block LAS file(s)
2) las2oci
   a) pre-procedures
   b) loading
   c) Post-procedures



Performance Metrics

Filename	Num Points (Millions)	Index Type	Index+load (sec)	Index+load (Min:Sec)	Points/min	File size
Oracle						
355000e4770000n.las	5.6	r-tree	667	11:07	506,974	151 mb
line_27102a.las	24.5	r-tree	4410	73:30	333,333	469 mb
points_a1_baghdad.las	80.1	r-tree	18240	304:47	263,486	2100 mb
libLAS						
355000e4770000n.las	5.6	python kd-tree	256	4:16	1,320,906	151 mb
line_27102a.las	24.5	python kd-tree	1251	20:51	1,175,060	469 mb
points_a1_baghdad.las	80.1	python kd-tree	3877	64:37	1,251,562	2100 mb
libLAS						
355000e4770000n.las	5.6	c++ blocking	88	1:22	3,842,636	151 mb
line_27102a.las	24.5	c++ blocking	540	9:00	2,722,222	469 mb
Points_a1_baghdad.las	80.1	c++ blocking	1208	20:08	3,869,564	2100 mb

Filename	libLAS python index	libLAS c++ index
355000e4770000n.las	261%	758%
line_27102a.las	353%	817%
points_a1_baghdad.las	470%	1509%



SOFTWARE. HARDWARE. COMPLETE.