



ORACLE[®]

Building a Secure Web Service using Oracle Spatial and JDeveloper





Overview

- Background
- Spatial Web Services
 - WFS 1.0
 - Web Service Security
 - Demo: Building a Secure Web service
 - Server side management (using PLSQL/Java/EM)
 - Building a Secure Client (using JDeveloper)
- OpenLS 1.1
- CSW 2.0.0
- WMS 1.1
- Summary



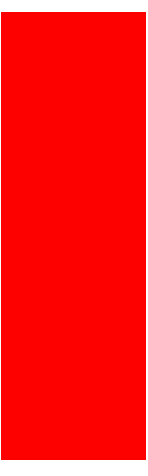
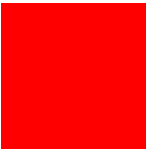
Background

- Open GIS Consortium (OGC) Standards
 - Web Feature Service
 - Access/search/update/delete geo-spatial feature instances based on spatial/non-spatial search criteria using a standard interface over the web
- Location Services
 - Routing
 - Mapping
 - Geocoding
 - Directory Services



Background (Contd)

- Catalogue Service for Web
 - Defines a common interface that enables diverse but conformant applications to perform discovery, browse and query operations against catalog servers.
- Web Map Service
 - Request/Provide maps
 - Request/Provide information about content of a map
- Make services secure
 - Authentication
 - Authorization
 - Transport-level security



Spatial Web Services



Web Feature Service (WFS)



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 an instance level
- Real-time transfer of feature instances in a platform/programming language independent way.



Approach

- Use
 - SOAP for Request/Response
 - XML over HTTP Post Method for Request/Response
- Oracle Spatial for Feature instance Storage/Retrieval.
- Implement OGC filter specification for feature search
- Use WSS/LDAP for authentication, Row level security for instance-level privilege mgmt and WSS for secure transfer of feature data.



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 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.



WFS Operations

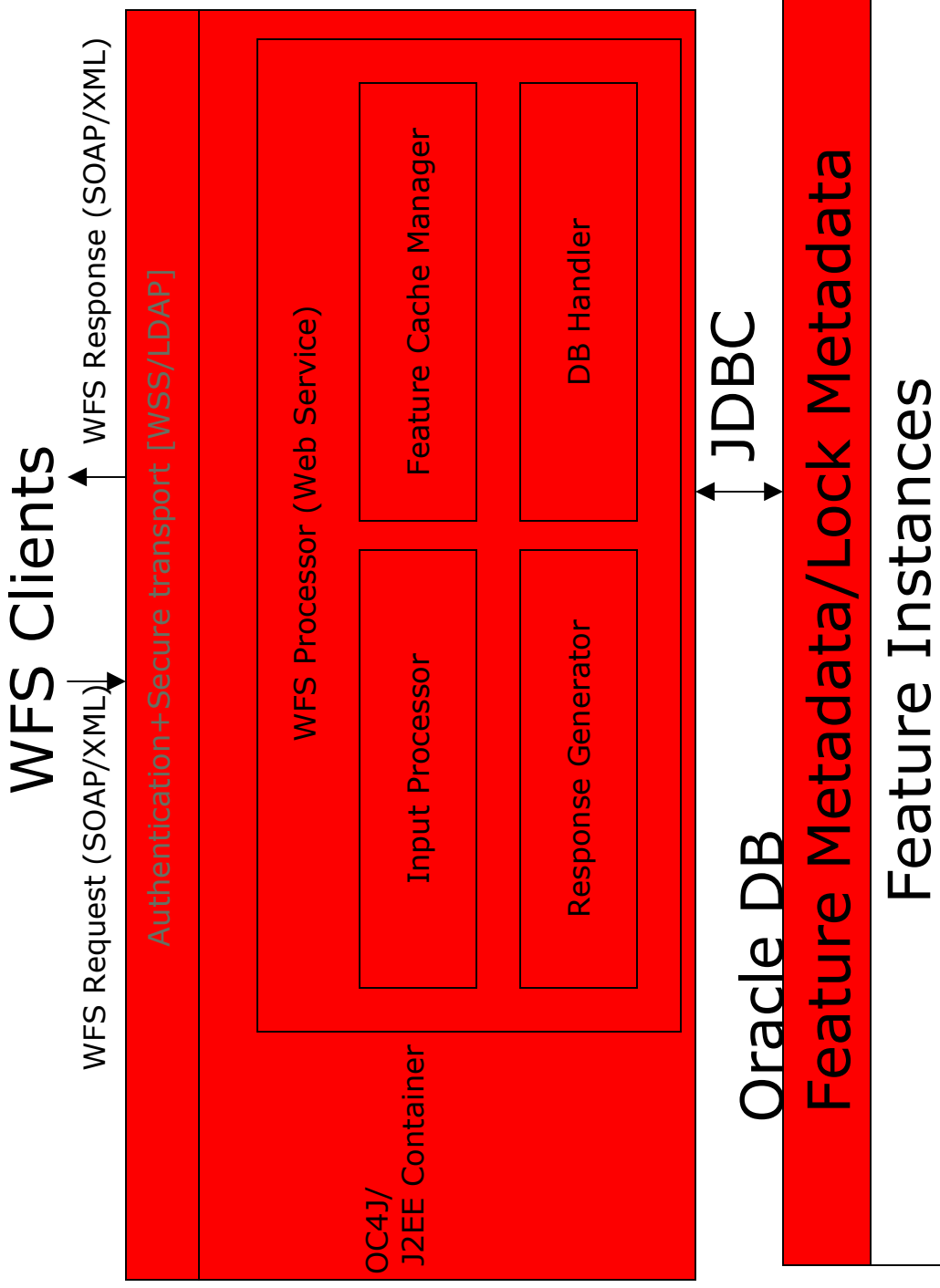
- Basic
 - Get Capabilities – get the metadata about the types / operations that a feature server supports
 - DescribeFeature - get the structural information about a feature type
 - GetFeature – query different parts of feature instances



WFS Operations (Contd.)

- Transactional WFS
 - GetFeatureWithLock – get a set of features and lock them
 - LockFeature – lock a set of feature instances, and lock some/all of them for a certain period of time.
- 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.



Locking

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



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



WFS Metadata (Contd.)

- Type Metadata populated automatically during type publication
- Type Metadata used to serve DescribeFeatureType response
- Capabilities metadata used to serve GetCapabilities response



Publish Feature Types

- Relational data source (e.g. table)
 - PLSQL API which will:-
 - Publish the content of a table with Spatial Column(s) to a feature type which is a subtype of gml:_Feature such that
 - 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



Publish Types (Contd.)

- XSD Document based data source
 - Java API which will:-
 - 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
- Java/PLSQL APIs to:-
 - Grant WFS Metadata access to WFS Users
 - Grant WFS Feature type access to WFS Users



Use Case (Type Suppliers)

- Publish Types
- Define type access control privileges



Use Case (Type Consumers)

- Get Server Capabilities
- Describe Feature Type
- GetFeatures (with proper filter)
- GetFeatureWithLock
- LockFeature
- Transaction
 - Insert/Update/Delete



Summary

- WFS support based on Oracle Spatial database
- Support for WFS basic/transaction interfaces
- Support for Spatial and XDB based indexing on feature instances.
- Support for registering Relational and Document based Feature Type
- Support for token-based WFS locking



Web Services Security

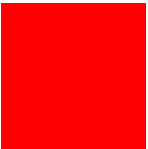


Issues addressed

- Authentication (WSS)
- Authorization
- Transport Security (WSS)

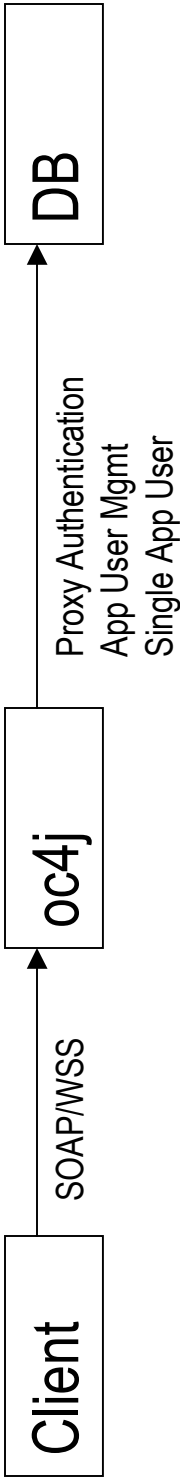
Authentication (WSS)

```
<env:Envelope ... >
  <env:Header>
    <wsse:Security ... >
      ...
      <dsig:Signature ... >
        ...
        </dsig:Signature>
      <wsse:UsernameToken ... >
        <wsse:Username>SpatialWsUser0</wsse:Username>
        <wsse:Password ... >MzSmymXK7mIiH4NU6h4lVS+IVb8=</wsse:Password>
        ...
      </wsse:UsernameToken>
    </wsse:Security>
  </env:Header>
  <env:Body ... >
    <ns0:makeSpatialRequestElement>
      <XLS ...> ...
    </XLS>
  </ns0:makeSpatialRequestElement>
</env:Body>
</env:Envelope>
```



Authorization

Propagation Of User ID





Authorization (Contd)

- User views
- VPD
- Enforced in the DB



Transport Security

- Message Integrity (Signing/Signature verification)
- Message Privacy (Encryption/Decryption)



Demo: Building a Secure Web Service



Server-side management

- Deploy and configure spatial web service (sdows.ear)
- Publish a feature type in WFS (Document-based) using Java API provided by Oracle Spatial
 - This function can be performed by a WFS Admin user
 - Specify feature type XSD
 - Specify the Xpaths on which to build spatial index
 - Specify the Xpaths on which to build XMLTable index
 - Specify other feature type metadata
 - Provide the above as XML/Java parameters to Java API
 - Grant privileges to WFS User (who will perform WFS query and transaction operations)



Building a Secure Client

- Generate a Client proxy from SpatialWS.wsdl (using Oracle Jdeveloper)
- Secure the Client proxy (using Oracle Jdeveloper)
- Customize client to invoke the Spatial Web service (WFS in this case) multiple times with different WFS operations in a certain sequence.
- Generate service output in log files



Open Location Services (OpenLS)



Functionality

- Location Utility Service (uses Oracle Geocoder)
- Presentation Service (uses Oracle MapViewer)
- Route Service (uses Oracle Route Server)
- Directory Service



Location Utility Service

Sample Request

```
<XLS ... >
...
<Request ... methodName="GeocodeRequest" ... >
  <GeocodeRequest>
    <Address countryCode="US">
      <StreetAddress>
        <Building number="400" />
        <Street><b>Post Street</b></Street>
      </StreetAddress>
      <Place type="CountrySubdivision"><b>CA</b></Place>
      <Place type="Municipality"><b>San Francisco</b></Place>
      <PostalCode><b>94102</b></PostalCode>
    </Address>
    <Address countryCode="US">
      <StreetAddress>
        <Building number="233" />
        <Street>Winston Drive</Street>
      </StreetAddress>
      <Place type="CountrySubdivision"><b>CA</b></Place>
      <Place type="Municipality"><b>San Francisco</b></Place>
      <PostalCode>94132</PostalCode>
    </Address>
  </GeocodeRequest>
</Request>
</XLS>
```



Location Utility Service

Sample Response

```
<xls:XLS ... >
...
<xls:Response ... >
  <xls:GeocodeResponse ... >
    <xls:GeocodeResponseList ... >
      <xls:GeocodedAddress>
        <gml:Point ... >
          <gml:pos dimension="2" srsName="4326">-122.4083257 37.788208</gml:pos>
        </gml:Point>
        <xls:Address countryCode="US">
          <xls:StreetAddress>
            <xls:Building number="400"/>
            <xls:Street>POST ST</xls:Street>
          </xls:StreetAddress>
          <xls:Place type="CountrySubdivision">CA</xls:Place>
          <xls:Place type="Municipality">SAN FRANCISCO</xls:Place>
          <xls:PostalCode>94102</xls:PostalCode>
        </xls:Address>
      </xls:GeocodedAddress>
    </xls:GeocodeResponseList>
  </xls:GeocodeResponse>
</xls:Response>
</xls:XLS>
```



Directory Service

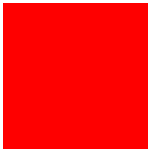
Sample Request

```
<XLS ... >
...
<Request ... methodName="DirectoryRequest">
<DirectoryRequest>
  <POILocation>
    <WithinBoundary>
      <AOI>
        <gml:CircleByCenterPoint ... >
          <gml:pos>-122.405 37.785</gml:pos>
          <gml:radius uom="9001">500</gml:radius>
        </gml:CircleByCenterPoint>
      </AOI>
    </WithinBoundary>
  </POILocation>
  <POIProperties>
    <POIProperty name="SIC_code" value="1234567890" />
  </POIProperties>
</DirectoryRequest>
</Request>
</XLS>
```

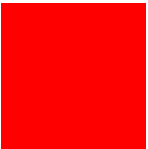
Directory Service

Sample Response

```
<xls:XLS ... >
...
<xls:Response ... >
  <DirectoryResponse ... >
    <xls:POIContext ... >
      <xls:POI ID="2" POIName="Borders Books & More" phoneNumber="415-399-1633" description="Books & more">
        <POIAttributeList ... >
          <ReferenceSystem ... >
            <xls:NAICS ... category="Books" ... />
            <xls:SIC ... category="Book stores" code="1234567890" ... />
            <xls:SIC ... category="Cafes & Cafeterias" code="1234567891" ... />
          </ReferenceSystem>
          </POIAttributeList>
          <gml:Point ... >
            <gml:pos dimension="2" srsName="4326">-122.4083257 37.788208</gml:pos>
          </gml:Point>
          <xls:Address countryCode="US">
            <xls:StreetIntersection>
              <xls:Street>Post St</xls:Street>
              <xls:IntersectingStreet>Powell St</xls:IntersectingStreet>
            </xls:StreetIntersection>
            <xls:Place type="CountrySubdivision">CA</xls:Place>
          ...
          <xls:Place type="Municipality">San Francisco</xls:Place>
          ...
          <xls:PostalCode>94102</xls:PostalCode>
        </xls:Address>
      </xls:POI>
    ...
```

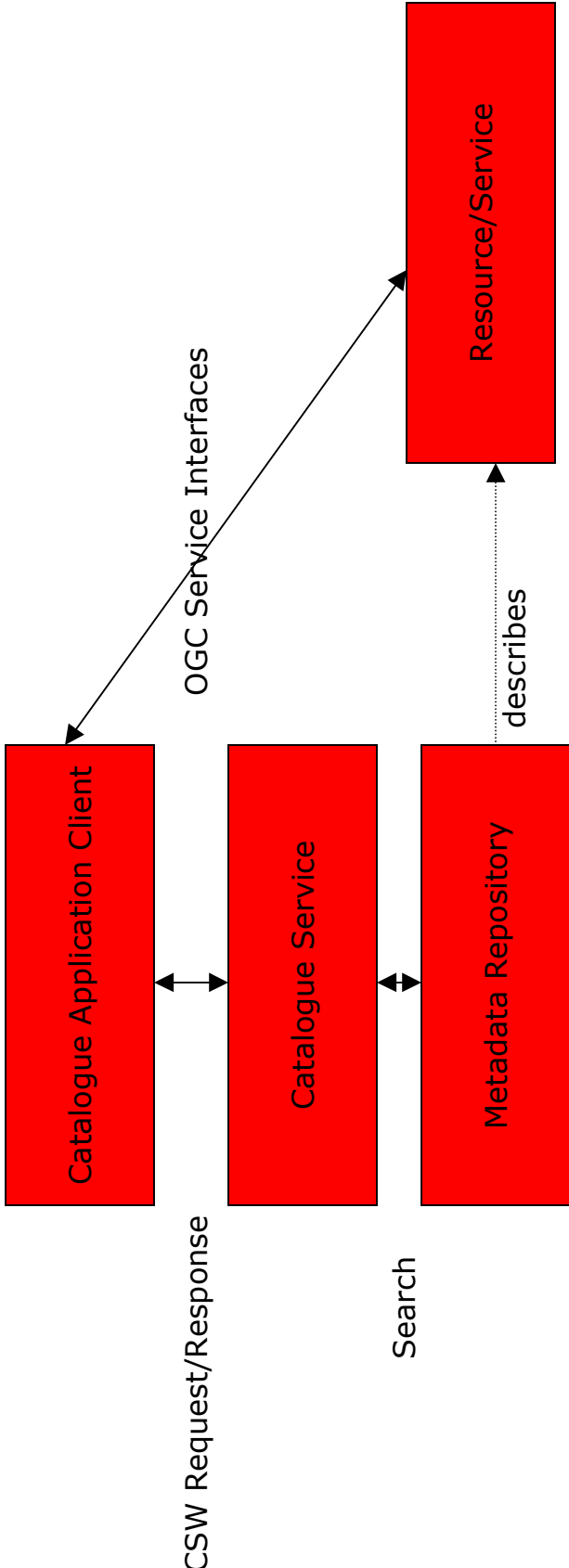


Catalog Service Web (CSW)



Overview

Application Interaction with Catalogue Server





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 an instance level
- Real-time transfer of catalog instances in a platform/programming language independent way.



Our Approach

- Use
 - SOAP for Request/Response
 - XML over HTTP Post Method for Request/Response
- Oracle Spatial for Record instance Storage/Retrieval.
- Implement OGC filter specification for record search
- Use WSS/LDAP for authentication, Row level security for instance-level privilege mgmt, and WSS for secure transfer of record data.
- ResultSet caching will be supported, to retrieve records from a single query across different web requests.



CSW Operations

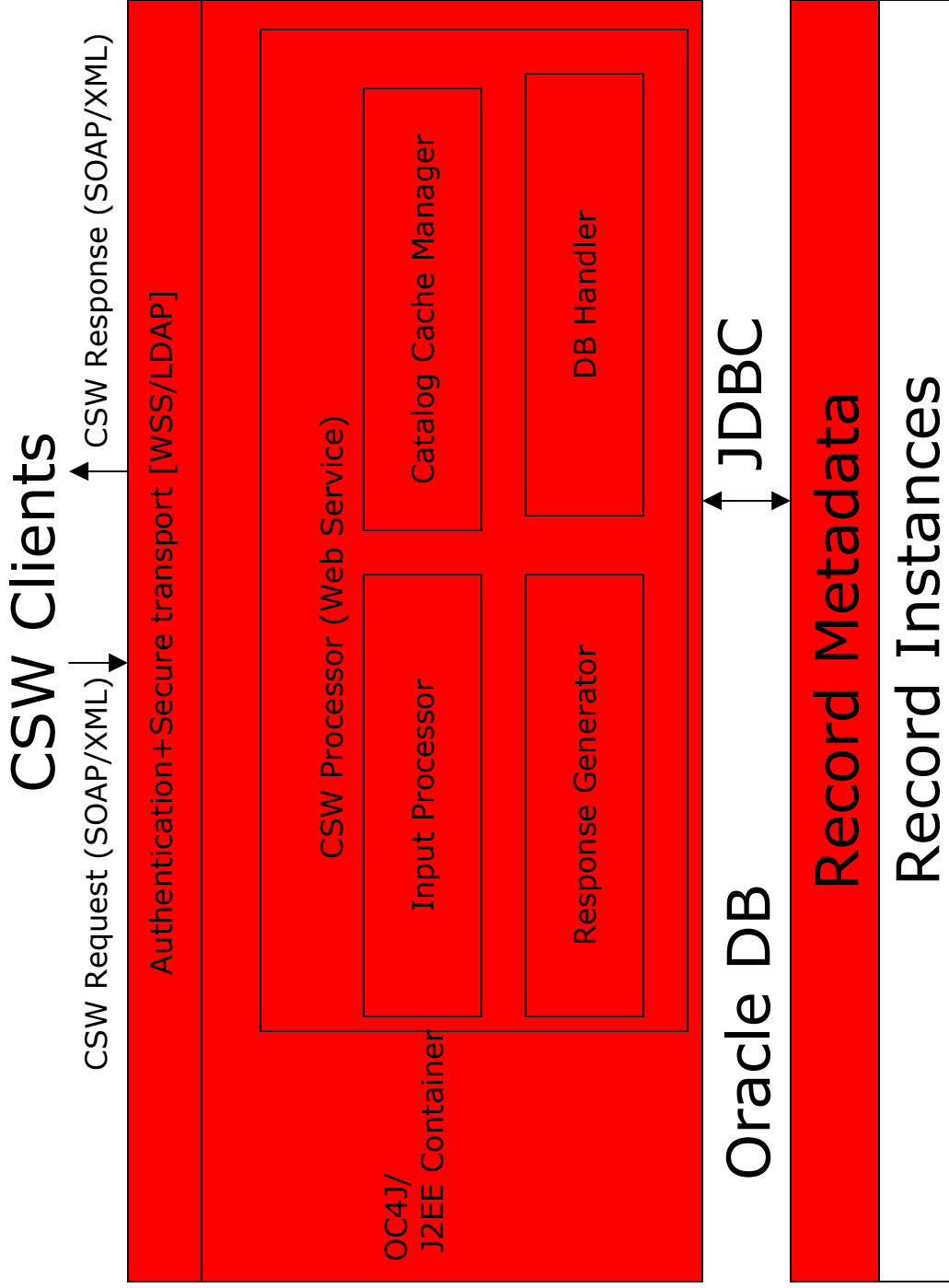
- **Discovery**
 - **Get Capabilities** – get the metadata about the types / operations that a catalog server supports
 - **DescribeRecord** - get the structural information about a catalog record type
 - **GetRecords**– query different parts of record instances
 - **GetRecordById** – query a record instance by identifier
 - **GetDomain** – query about range of values for CSW request type parameters



CSW Operations (Contd.)

- Publication
- Transaction
 - Insert new record instances
 - Update existing record instances based on filter criteria
 - Delete existing record 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.



CSW Metadata

- Record Types Metadata
- Domain Related Metadata
- Capabilities Related Metadata
- Records Metadata
- Type metadata populated during publishing record type
- Type metadata used to serve DescribeRecordType response
- Capabilities metadata used to serve GetCapabilities response



Publish Record Types

- Default (CSW Record Type).
- Custom Record Types, conforming to `csw:AbstractRecord`



Use Case (Type Suppliers)

- Publish Record Types
 - Java API which will:-
 - Register record type XSDs and record type metadata
 - Register spatial paths on which spatial index will be built
 - Register non-spatial paths on which XDB index will be built
 - Record Type Registration XSD, captures all record type metadata parameters
- Java APIs to:-
 - Grant CSW Metadata access to CSW Users
 - Grant CSW Record type access to CSW Users



Use Case (Type Consumers)

- Get CSW Server Capabilities
- Describe Record Type
- GetRecords (with proper filter) [Current support is for OGC Filter specification]
- GetRecordById
- GetDomain
- Do Transaction on Records
 - Insert/Update/Delete



Web Map Service (WMS)



Operations

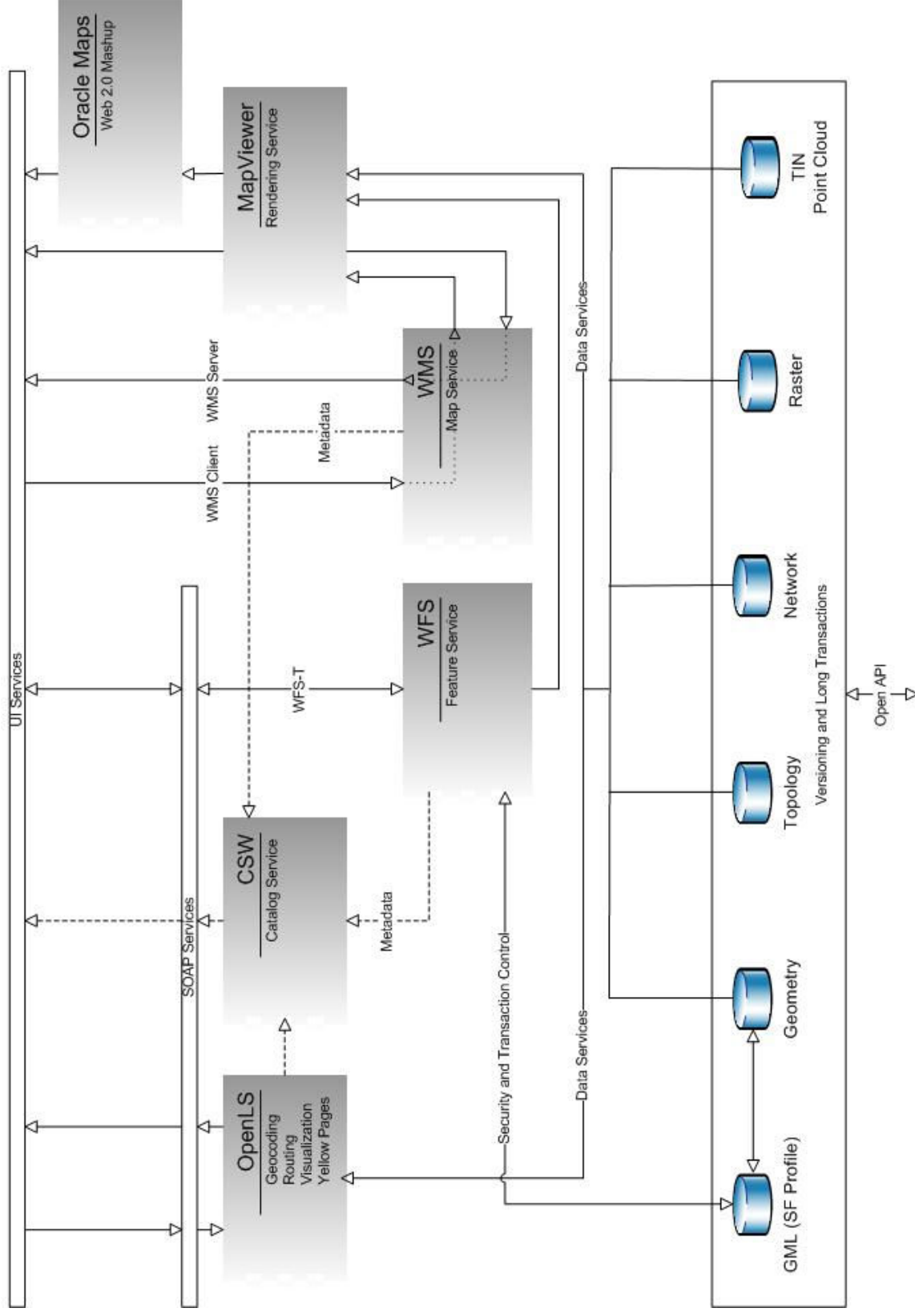
- GetCapabilities
- GetMap
- GetFeatureInfo



Summary

- WFS support based on Oracle Spatial feature repository. It provides OGC standards based access to Oracle Spatial table data for query and manipulation
- Location Service provides OGC standards based access to Mapping, Routing, Geocoding and Directory Service functionality of Oracle Spatial, which promotes interoperability
- CSW support implements OGC Catalogue Service specification based on Oracle Spatial Catalogue repository
- WMS provides OGC standards based access to maps and information about content of a map
- Use WSS/LDAP for authentication, Row level security for instance-level privilege mgmt and WSS for secure transfer of feature data.
- Build a secure web service client using Oracle Jdeveloper

Oracle Spatial Services





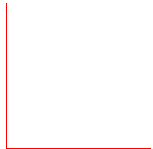
For More Information


search.oracle.com

or

www.oracle.com/database/spatial.html



 | **ORACLE IS THE INFORMATION COMPANY**

