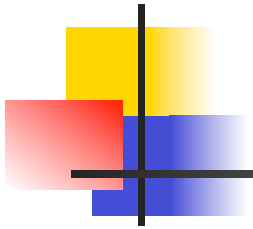
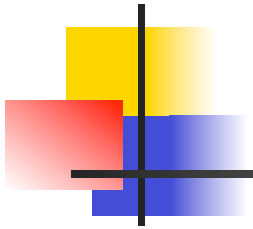
Security Integration Between OBI EE and Hyperion Essbase

Venkatakrishnan J, BI Architect, Oracle



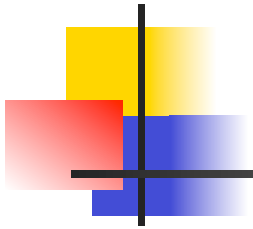
Who Am I?

- BI Architect with Oracle
- Part of a group called NAC within Oracle
- Over 6+ years of Oracle BI Experience
- Blog at http://oraclebizint.wordpress.com
- Technical Reviewer of an Essbase book (to be released in November)
- 8 Successful big implementations(all in production) using Oracle BI EE Plus, Hyperion Essbase, Oracle EPF, BI Applications



Security Integration – Key Pieces

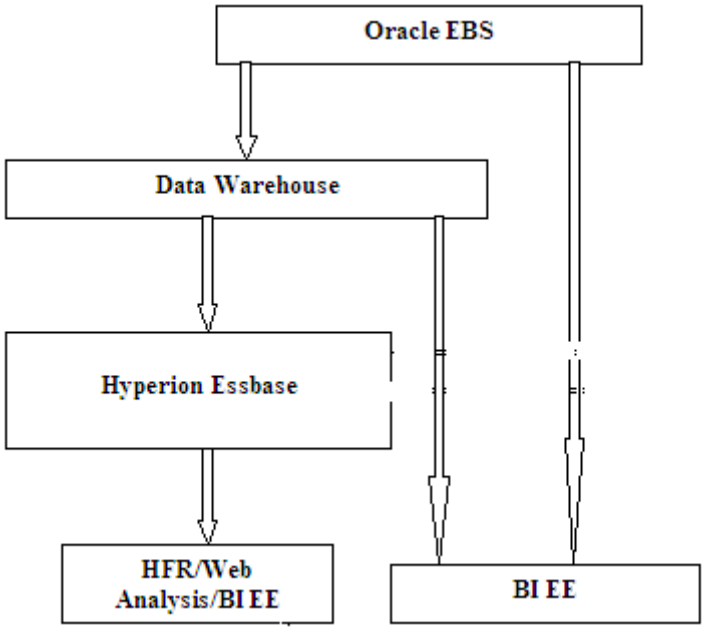
- Authentication
- Authorization
- Single Sign-on

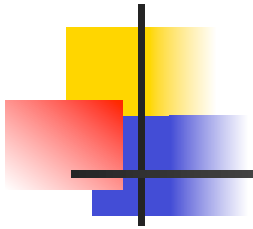


Security Integration – Current Known Issues with 9.3.1 and 10.1.3.4

- No Single Sign-on across products.
- Both support external authentication
- No common authorization platform
- Introduction of EPM 11.1.1.0 and BI EE 10.1.3.4 solves the SSO issue. But issues still remain with regard to authorization.

Overview of Data Architecture – Common Scenario

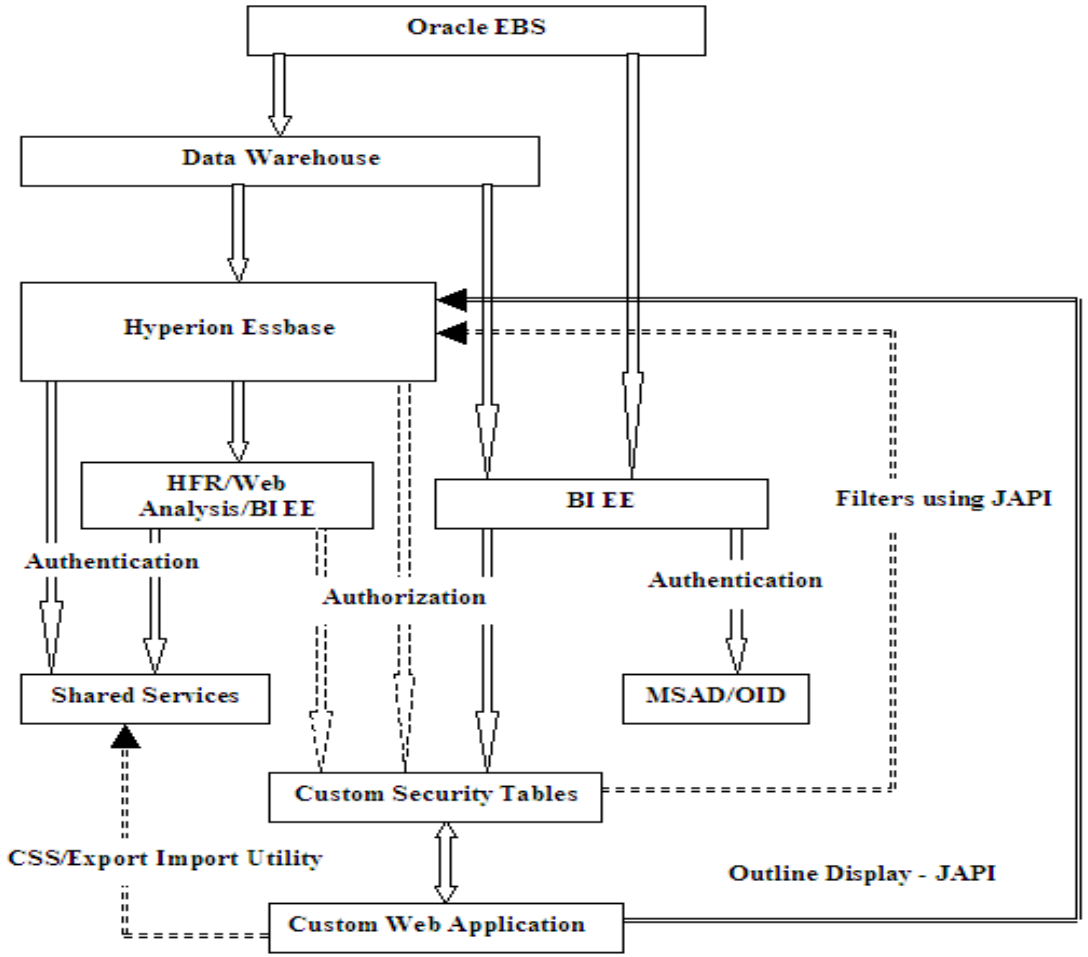


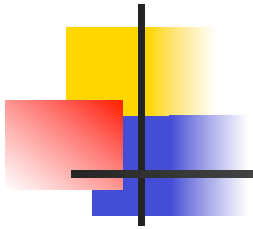


Key Security Requirements/Features

- Common Authorization Interface
- A web based interface for handling security
- Security handled through parent-child hierarchies
- Security to be uniform across BI EE and Essbase for compliance reasons
- Securing on a parent should give access to all its children

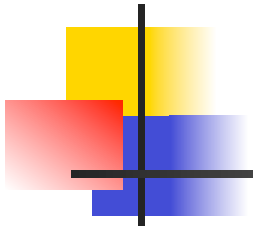
Security Implementation - Overview





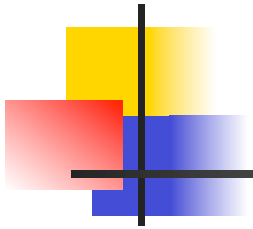
How does it Work?

- The custom Application contains 3 interfaces
  - Group Creation Interface
  - User Group Assignment Interface
  - Group Security Member Assignment Interface
- The Application itself can be made to authenticate against any User Source
- Essbase, HFR and Web Analysis authenticate/ authorize using Shared Services and Essbase Filters
- BI EE authenticates against MSAD/OID and authorizes against custom security tables



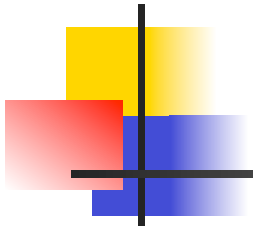
How does it Work? – Group Creation

- Security Administrators login to the Custom Application
- They create a custom group
- The group is first inserted into the custom security table CUSTOM_SECURITY_GROUPS
- Then the group is created in shared services using CSS/Export Import utility
- Any failure anywhere, automated rollbacks happen
- Similar to Hyperion Planning Security(only to an extent) where security info is maintained in Interface tables as well as custom Filters in Essbase



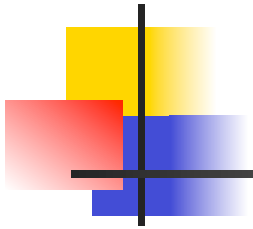
How does it Work? – User Group Assignment

- Users are obtained from FND_USERS table of EBS or from a custom package if the user source is OID/MSAD
- The users are then assigned to the custom groups that have been created
- The assignment is first inserted into custom security table CUSTOM_SECURITY_USER_GROUP
- Then it is inserted into shared services using CSS/Export Import utility
- Any failure anywhere, rollbacks happen automatically to ensure that the security table and shared services are in sync



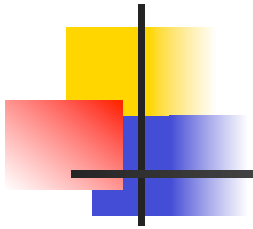
How does it Work? - Group Member Assignment

- The application would display all the members as is from the Essbase Outline(after choosing an Essbase App and Database) and would provide an option to choose a group.
- Users can choose all the members(from all the dimensions or specific dimensions) that they want to secure on.
- On submit, the custom members are inserted into the custom security table CUSTOM_SECURITY_GROUP_MEMBERS
- Using JAPI, a custom filter would be created using the members chosen. The filter would then be assigned to the group chosen. Before assigning access, shared services and Essbase would be synchronized.



How does it Work? – Essbase/HFR/Web Analysis

- When an user logs into Essbase Excel-add in/ HFR/Web Analysis, he/she would be authenticated against shared services.
- Shared Services would automatically pick up the group the user is assigned to and then would pass on the credentials to Essbase.
- When the user requests for any data, the essbase filter created for that group would kick in and would secure the data



How does it Work? – BI EE

- When an user logs into BI EE, he/she would be authenticated against OID/MSAD.
- The USER variable would be set and the corresponding group would be obtained from the CUSTOM_SECURITY_USER_GROUPS table
- The LTS of the physical source to be secured would have a filter on the members that are part of the group.
- To provide access to child members as well, the hierarchy would be traversed using CONNECT BY and filters would be applied appropriately

Custom Security Tables

Column Name	Nullable	Data Type	Data Default	COLUMN ID
USER_NAME	Yes	VARCHAR2(100 BYTE)	(null)	1
GROUP_NAME	Yes	VARCHAR2(100 BYTE)	(null)	2
IS_ACTIVE	Yes	VARCHAR2(10 BYTE)	(null)	3
CREATED_DATE	Yes	DATE	(null)	4
CREATED_BY	Yes	VARCHAR2(100 BYTE)	(null)	5
LAST_UPDATE_DATE	Yes	DATE	(null)	6
LAST_UPDATED_BY	Yes	VARCHAR2(100 BYTE)	(null)	7

CUSTOM_SECURITY_USER_GROUPS

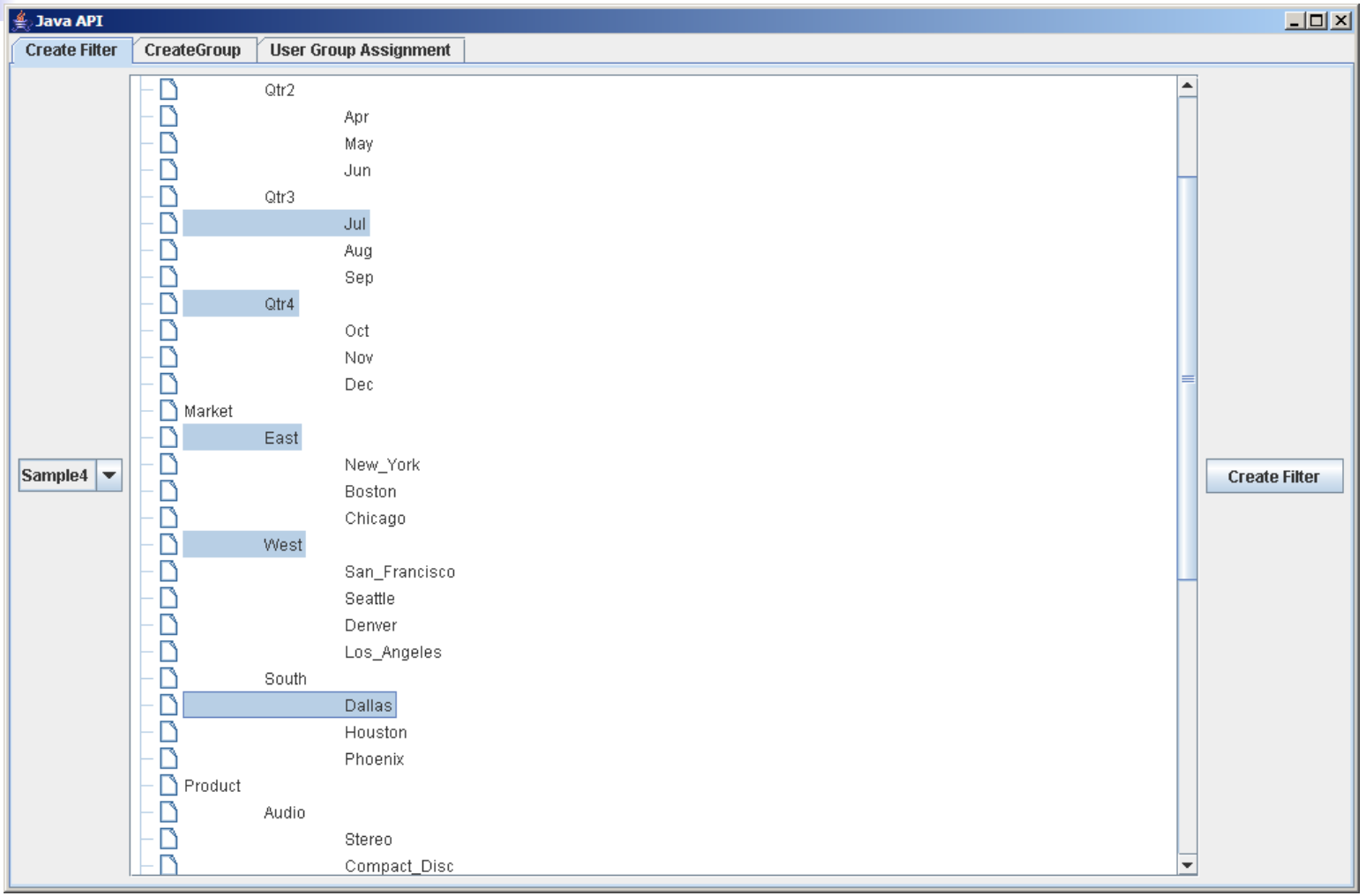
Column Name	Nullable	Data Type	Data Default	COLUMN ID
GROUP_NAME	No	VARCHAR2(100 BYTE)	(null)	1
IS_ACTIVE	No	VARCHAR2(10 BYTE)	(null)	2
CREATED_DATE	Yes	DATE	(null)	3
CREATED_BY	Yes	VARCHAR2(100 BYTE)	(null)	4
LAST_UPDATE_DATE	Yes	DATE	(null)	5
LAST_UPDATED_BY	Yes	VARCHAR2(100 BYTE)	(null)	6

CUSTOM_SECURITY_GROUPS

Column Name	Nullable	Data Type	Data Default	COLUMN ID
GROUP_NAME	Yes	VARCHAR2(100 BYTE)	(null)	1
SECURITY_DIM	Yes	VARCHAR2(100 BYTE)	(null)	2
DIM_VALUE	Yes	VARCHAR2(255 BYTE)	(null)	3
IS_ACTIVE	Yes	VARCHAR2(10 BYTE)	(null)	4
CREATED_DATE	Yes	DATE	(null)	5
CREATED_BY	Yes	VARCHAR2(100 BYTE)	(null)	6
LAST_UPDATE_DATE	Yes	DATE	(null)	7
LAST_UPDATED_BY	Yes	VARCHAR2(100 BYTE)	(null)	8

CUSTOM_SECURITY_GROUP_MEMBERS

Group Creation - Technical Details



Group Creation - Technical Details - JAPI

```
// Create JAPI instance.
ess = IEssbase.Home.create(IEssbase.JAPI_VERSION);

// Sign On to the Provider
IEssDomain dom
    = ess.signOn(s_userName, s_password, false, null, s_provider);

// Open connection with olap server and get the cube.
olapSvr = (IEssOlapServer)dom.getOlapServer(s_olapSvrName);
olapSvr.connect();
IEssCube cube = olapSvr.getApplication("Demo").getCube("Basic");

listOutlineMembers(cube);
System.out.println("\nOutline Viewing sample complete.");
catch (EssException x) {
System.err.println("Error: " + x.getMessage());
statusCode = FAILURE_CODE;
```

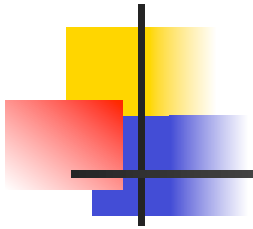
Group Creation - Technical Details - JAPI

```
boolean playWithLineStyle = false;
String lineStyle = "Angled";
try {

    otl = cube.openOutline();
    System.out.println("Outline information\n--------------------");
    System.out.println("Count of enabled DTS members: " + otl.getCountEnabledDTSMembers());

    System.out.println("\nListing all ouline members in cube " +
        cube.getApplication().getName() + "/" + cube.getName());
    System.out.println("-------------------------------------------------");
    DefaultMutableTreeNode top = new DefaultMutableTreeNode("Demo");
    IEssIterator dims = otl.getDimensions();
    for (int i = 0, cntTabs = 0; i < dims.getCount(); i++) {
        IEssDimension dim = (IEssDimension)dims.getAt(i);
        listOutlineMembers_helper(dim.getDimensionRootMember(), cntTabs, top);
    }
    otl.close();
    otl = null;
```

Group Creation - Technical Details - JAPI

```
String [] rowStrings = {"@IDESCENDANTS(Scenario)", "@IALLANCESTORS (Scenario)"};
IEssCube.IEssSecurityFilter filter = cube.createSecurityFilter("TestFilter1");

System.out.println("\nList after Creating ...");
getFilterList(cube);

IEssCube.IEssSecurityFilter setFilter = cube.setSecurityFilter(filter.getName(), true, IEssCube.EEssCubeAccess.READ_WRITE_CUBE_DATA);
for (int i =0; i < rowStrings.length; i++) {
    setFilter.setFilterRow(rowStrings[i], (short)EssGlobalStrings.ESS_ACCESS_WRITE);
}
setFilter.setFilterRow("", (short)0);

System.out.println("\nList after Setting ...");
getFilterList(cube);

setFilter.copyFilter(filter.getName() + "Copy");
System.out.println("\nList after Copying ...");
getFilterList(cube);
```

DEMO



Summary

- The approach discussed here requires atleast 4 weeks of development effort to tie all the pieces together
- Points of failure more due to custom integration
- EPM 11 provides more seamless shared services JAPI integration.
- More enhancements possible to cater to different kinds of security requirements