

# Scale, Share and Store your Models with CDO 2.0

Eclipse Live Webinar, January 29, 2009

**Eike Stepper**

[stepper@esc-net.de](mailto:stepper@esc-net.de)  
<http://www.esc-net.de>  
<http://thegordian.blogspot.com>

**Berlin, Germany**



**Ed Merks**

[Ed.Merks@gmail.com](mailto:Ed.Merks@gmail.com)  
<http://www.macromodeling.com>  
<http://ed-merks.blogspot.com>

**Ballantrae, Ontario, Canada**

# Agenda



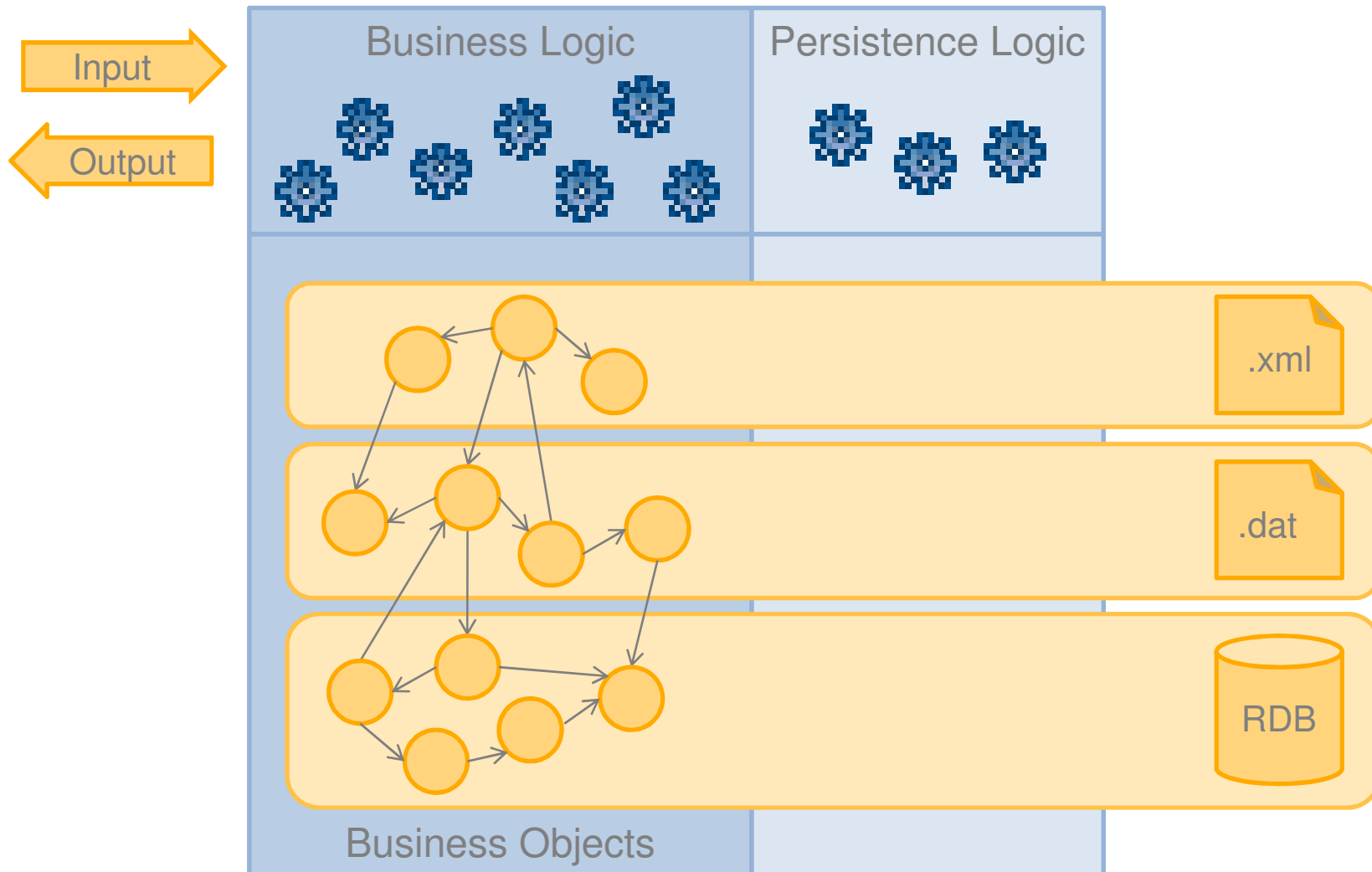
Eike



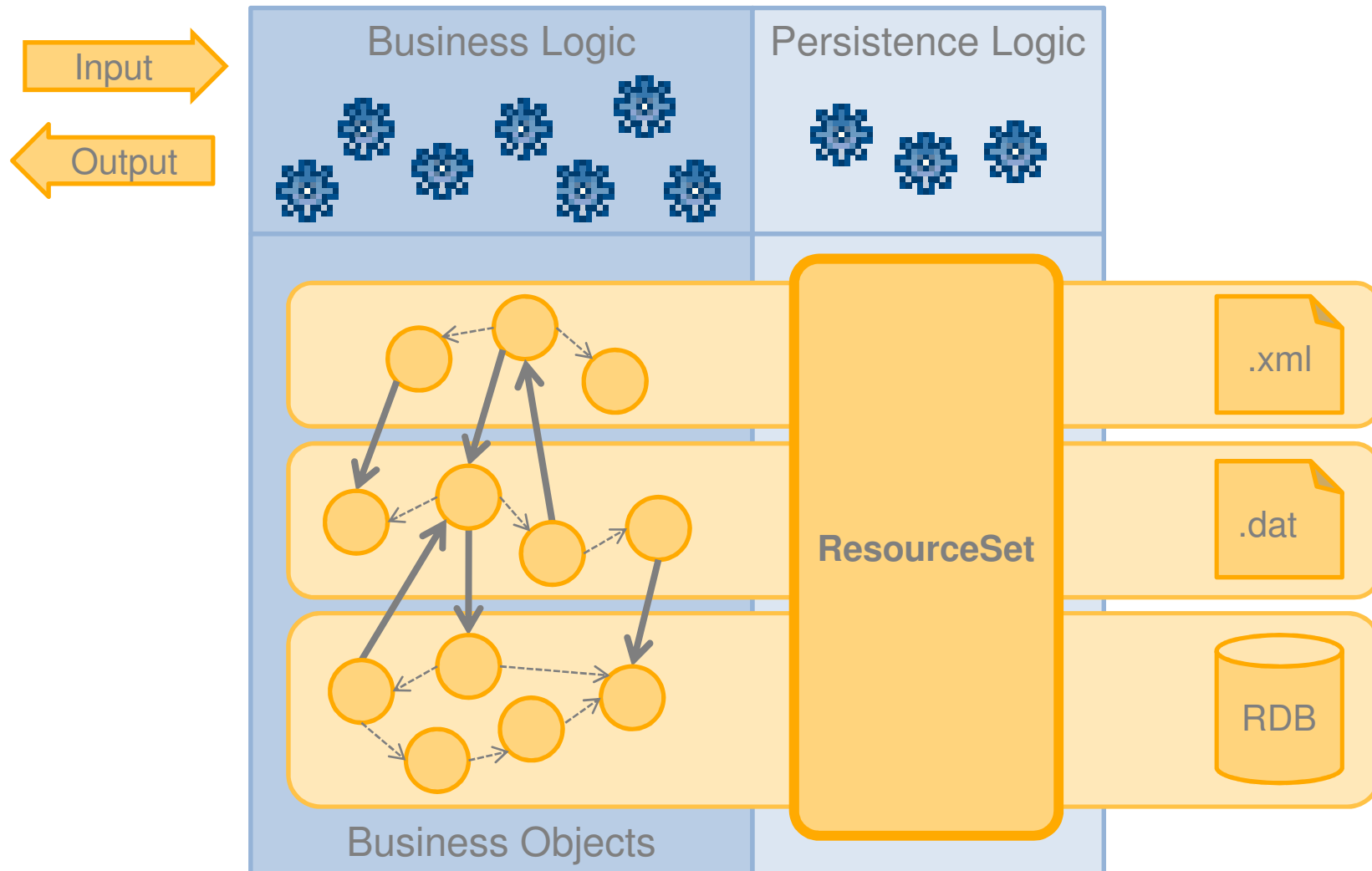
Ed

	<b>Typical Application</b>
<i>answers questions</i>	<b>The Wonder of EMF</b>
	<b>EMF Snippet</b>
<b>Typical CDO Deployment</b>	
<b>Typical CDO Application</b>	<i>answers questions</i>
<b>Typical CDO Server</b>	
<b>Live Demonstration</b>	
<b>Live Discussion</b>	
<b>(CDO Snippets)</b>	
<b>(Next Steps)</b>	

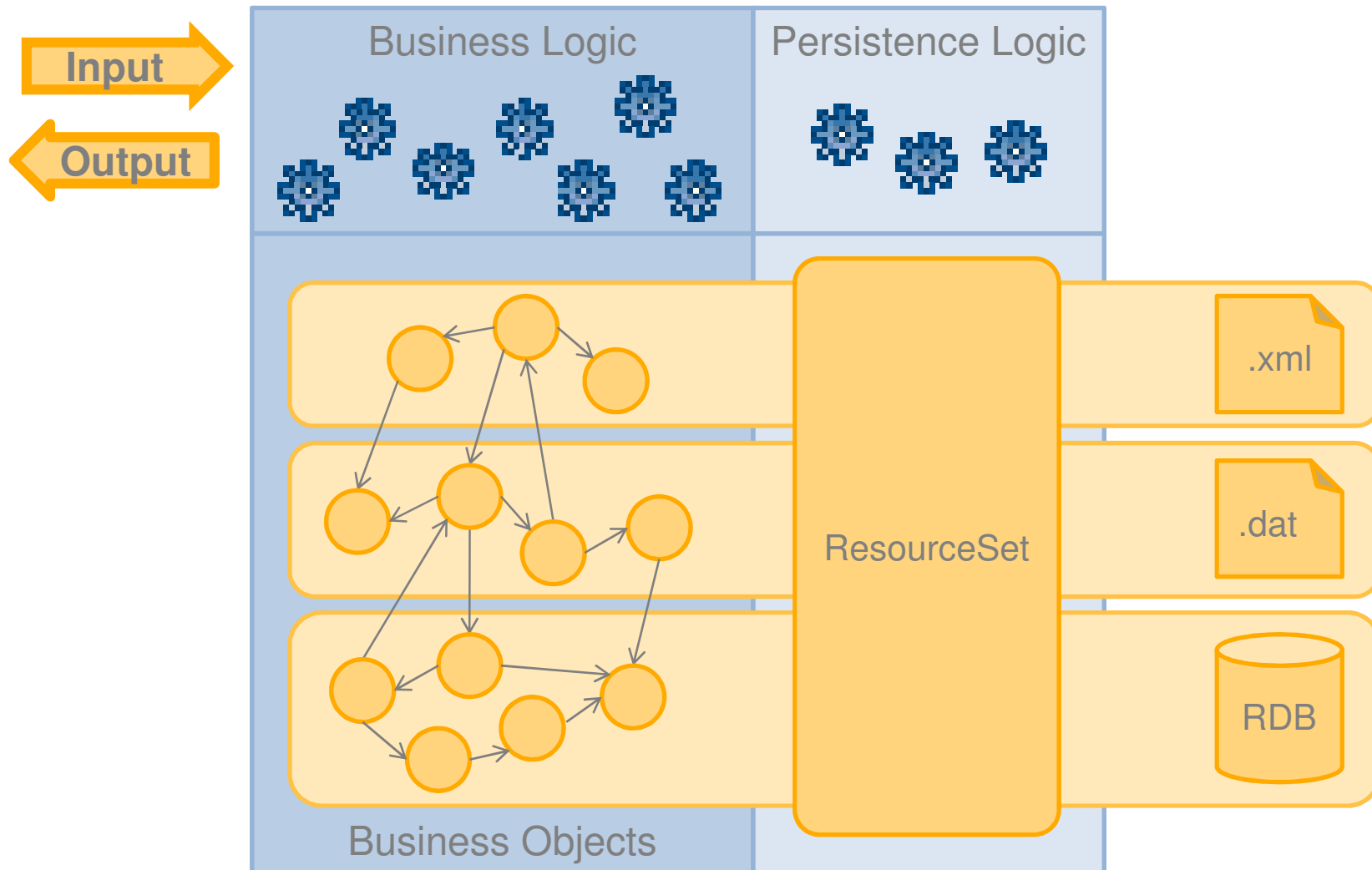
# Typical Application



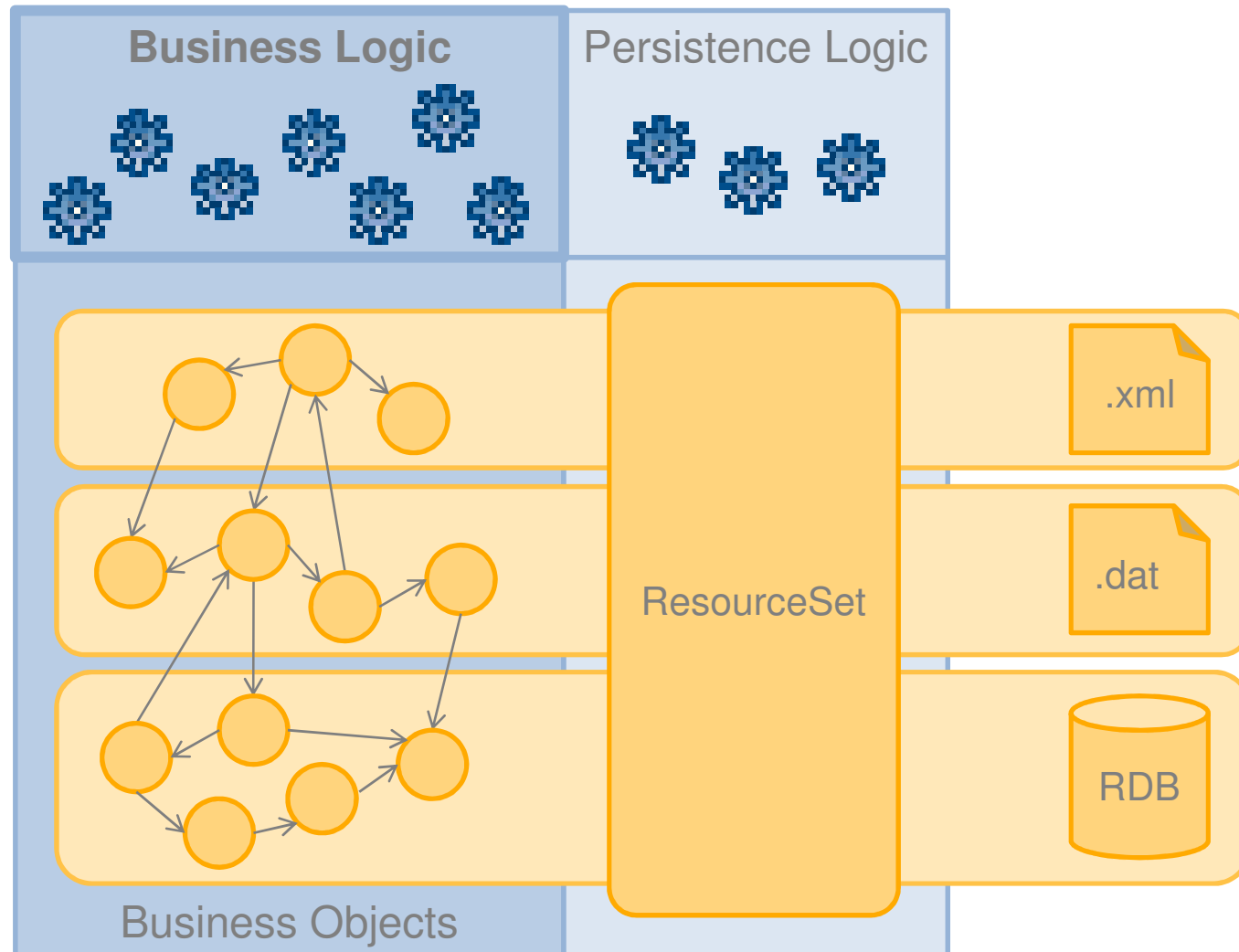
# Cross References by EMF



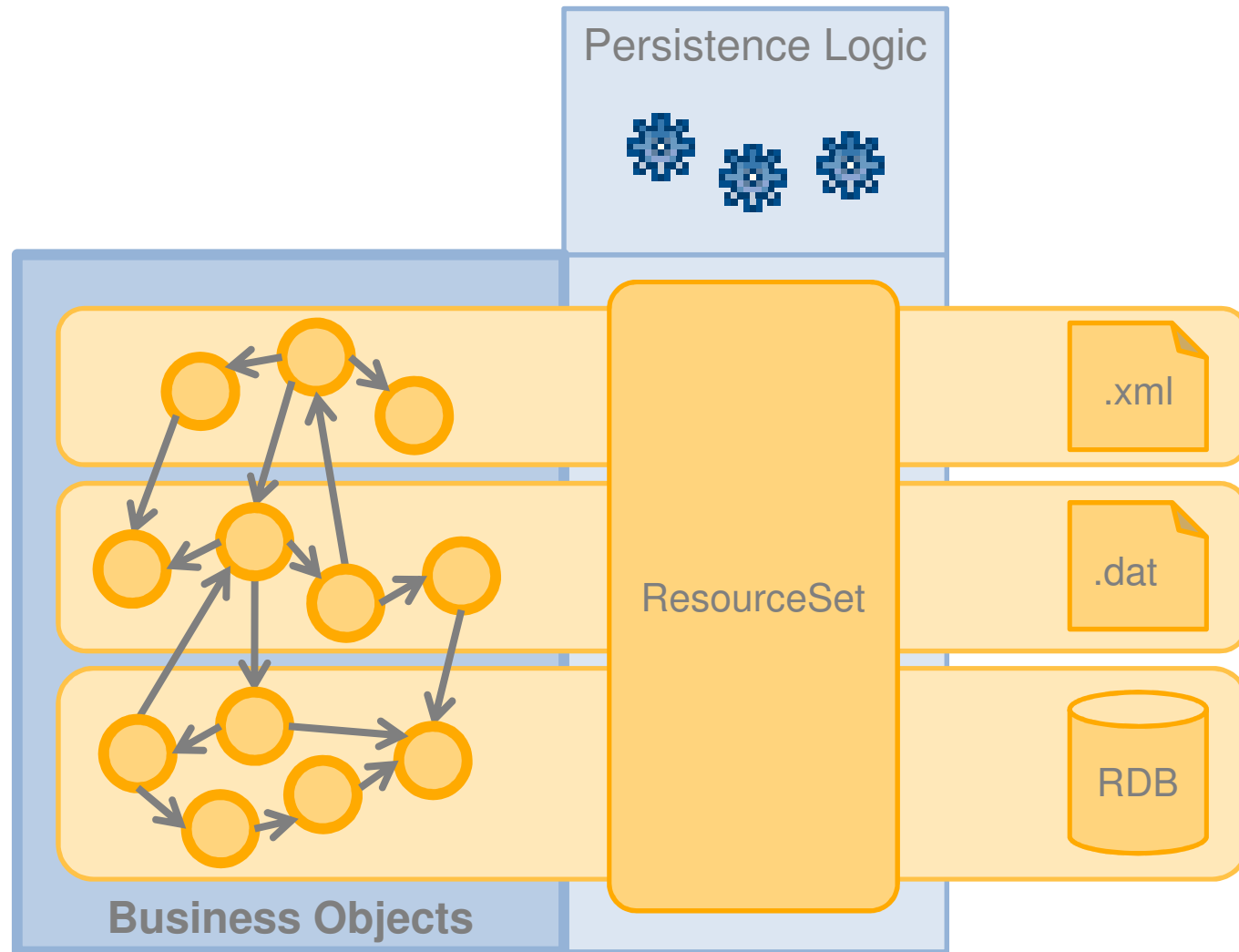
# Model View Controller by EMF



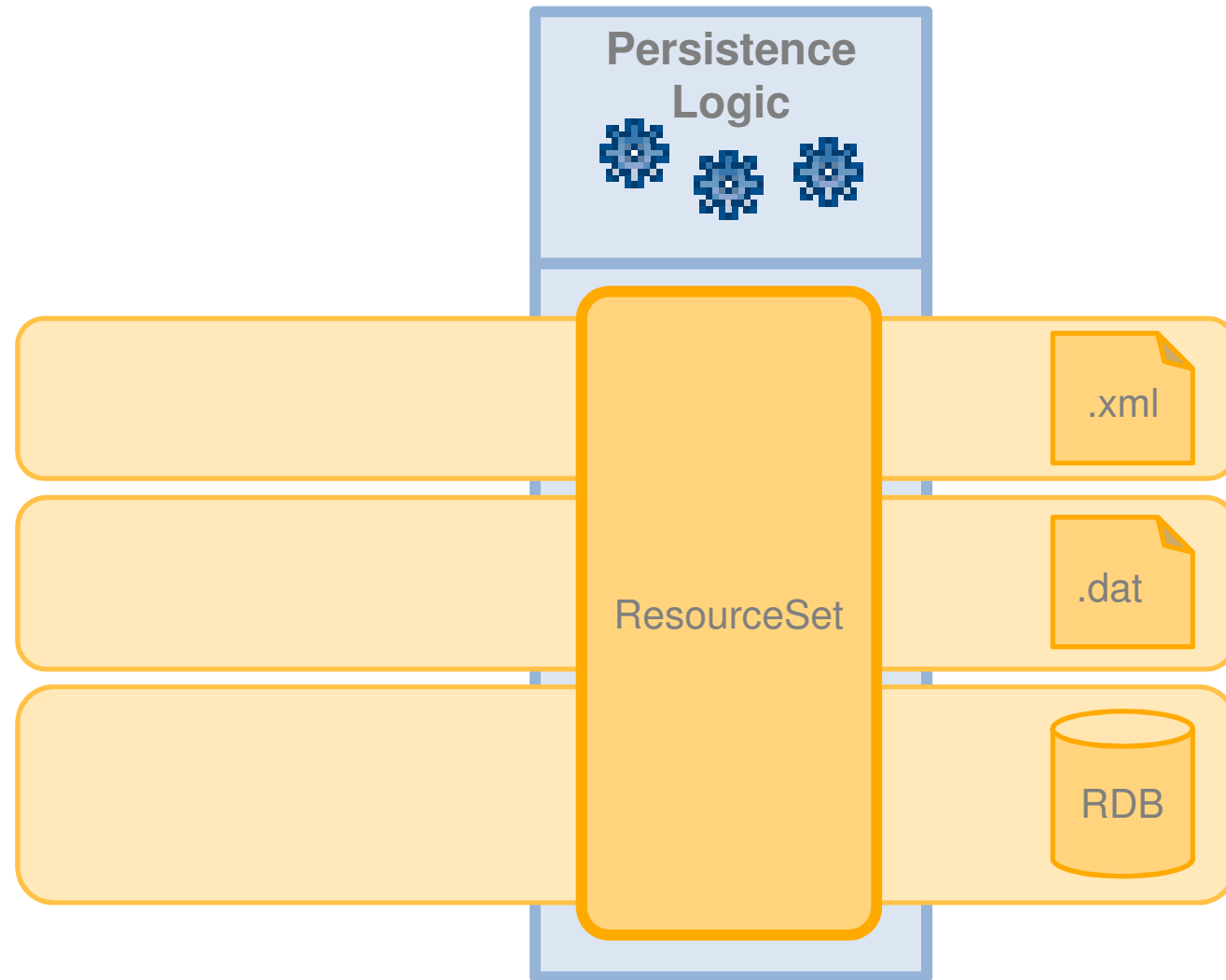
# Business Logic is Yours



# Models Generated by EMF



# Persistence Handled by EMF

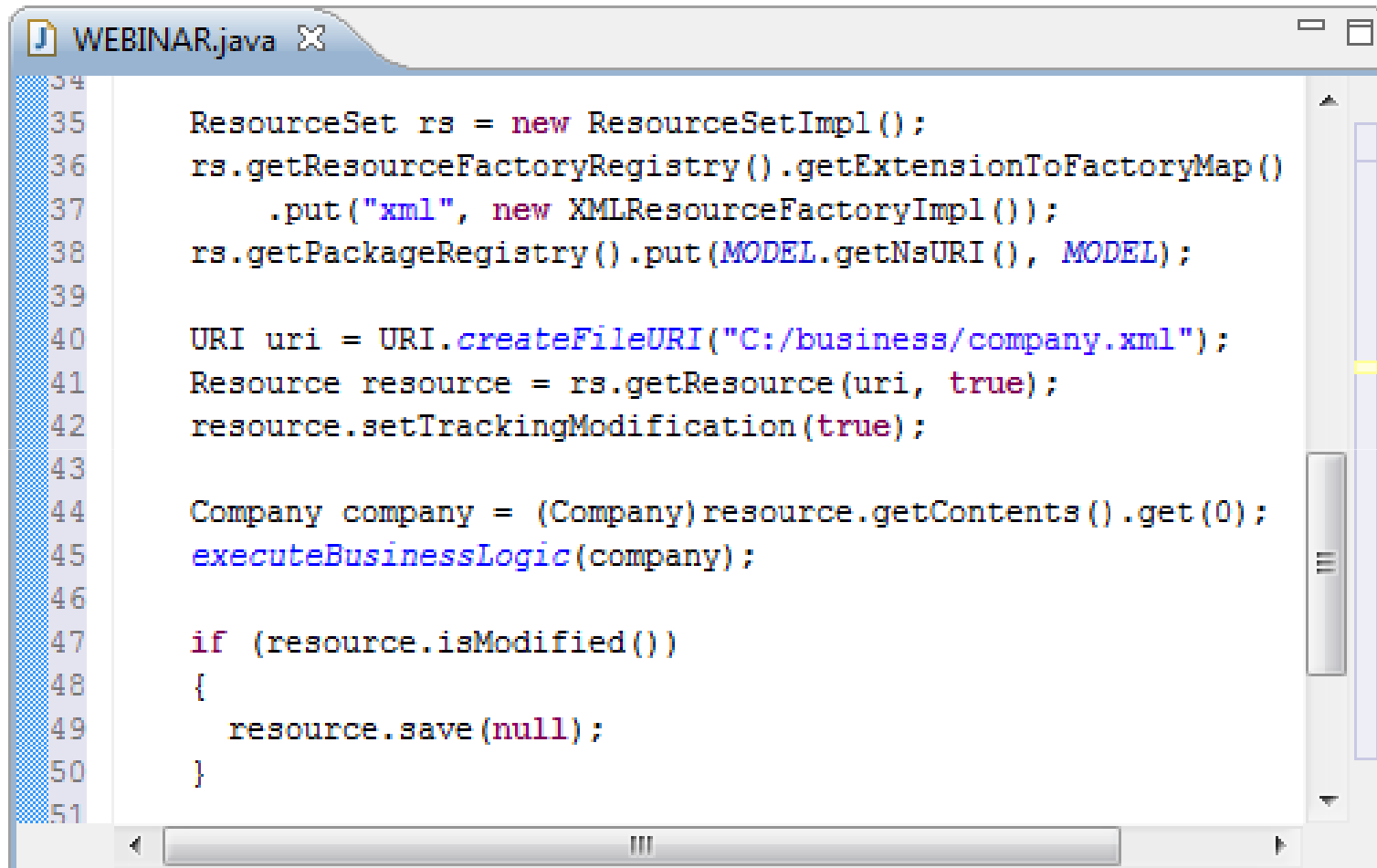




# Default Serialization by EMF

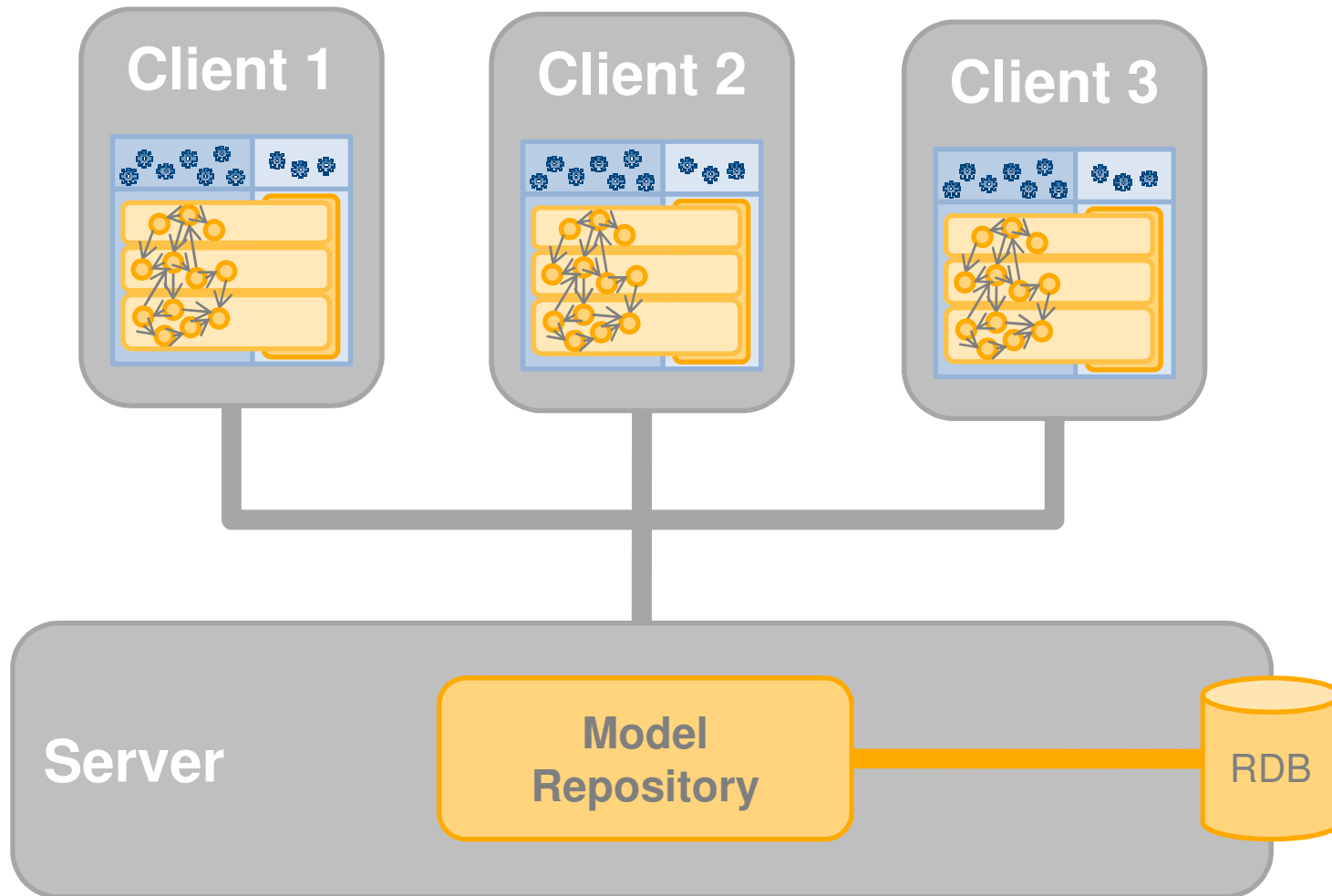


# EMF Snippet

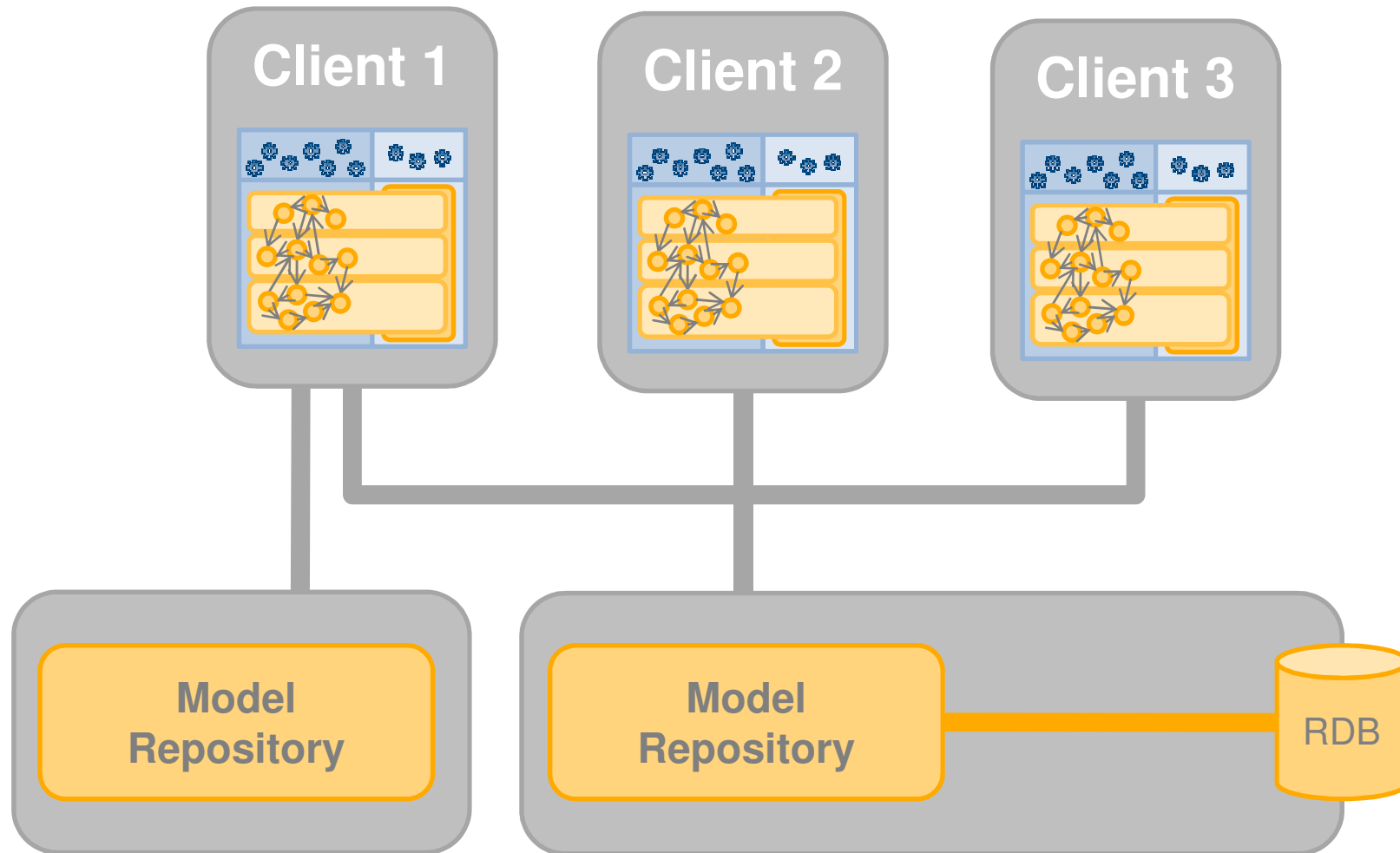


```
34  
35 ResourceSet rs = new ResourceSetImpl();  
36 rs.getResourceFactoryRegistry().getExtensionToFactoryMap()  
37     .put("xml", new XMLResourceFactoryImpl());  
38 rs.getPackageRegistry().put(MODEL.getNsURI(), MODEL);  
39  
40 URI uri = URI.createFileURI("C:/business/company.xml");  
41 Resource resource = rs.getResource(uri, true);  
42 resource.setTrackingModification(true);  
43  
44 Company company = (Company)resource.getContents().get(0);  
45 executeBusinessLogic(company);  
46  
47 if (resource.isModified())  
48 {  
49     resource.save(null);  
50 }  
51
```

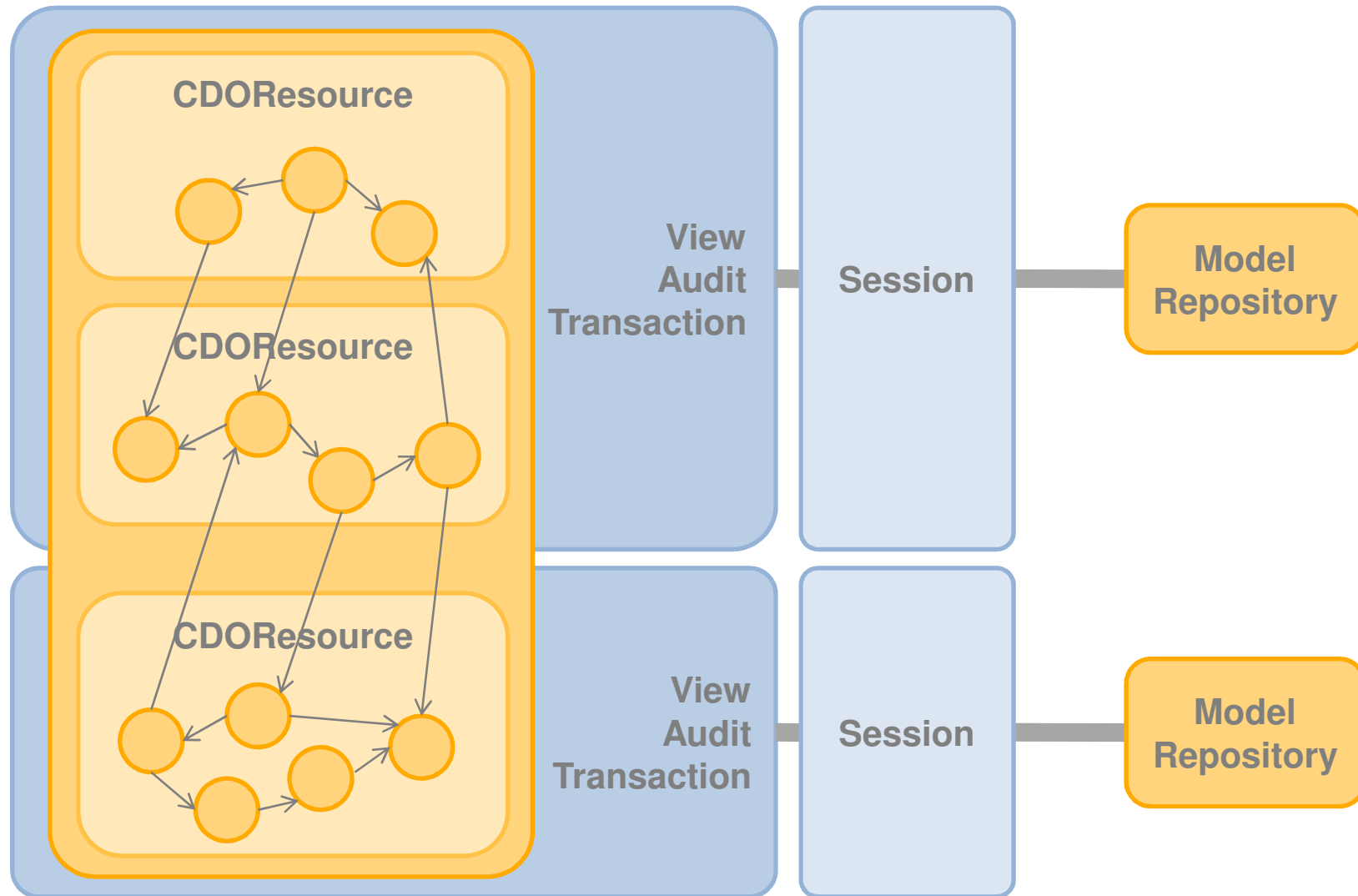
# Typical CDO Deployment



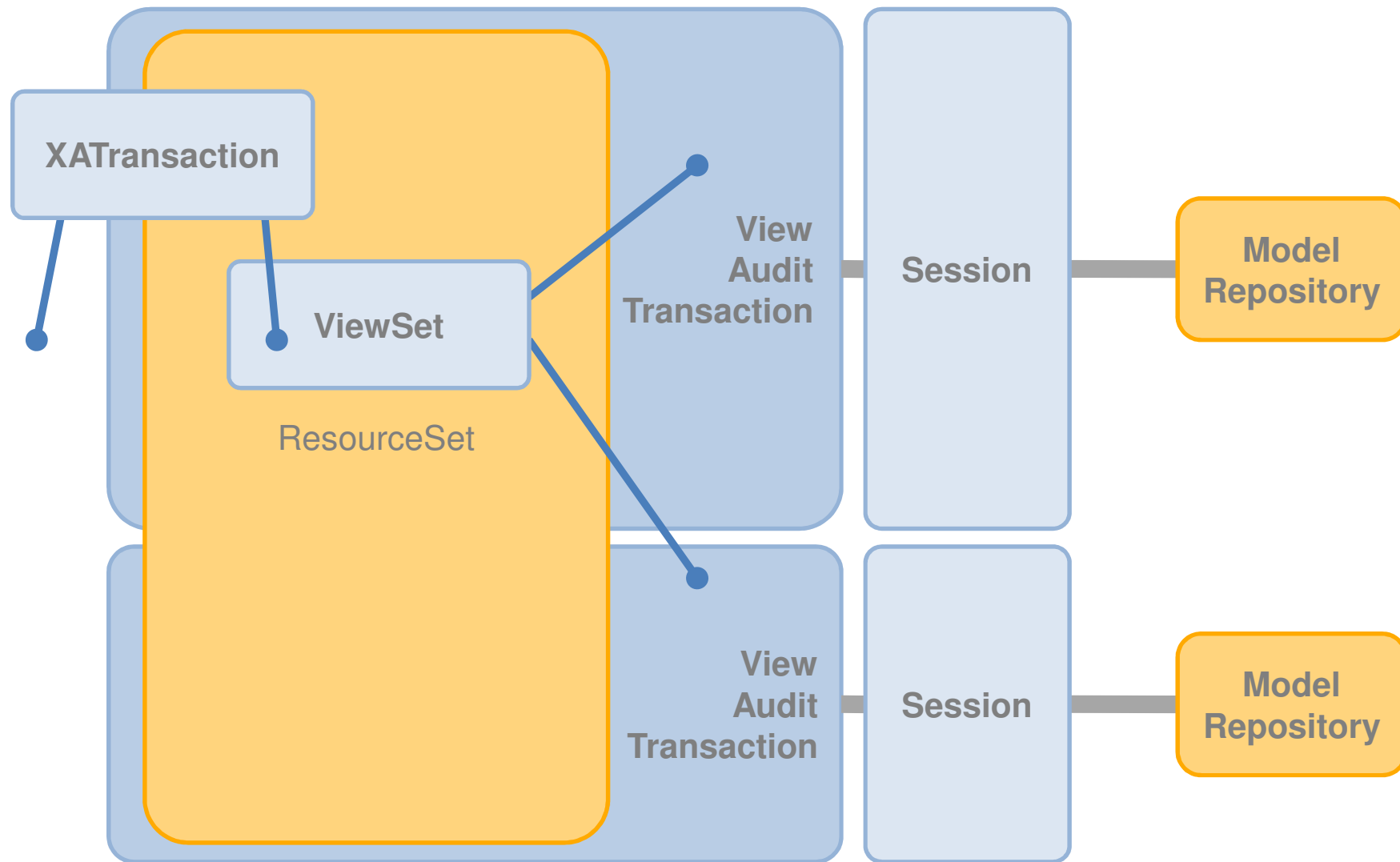
# Multiple Repositories



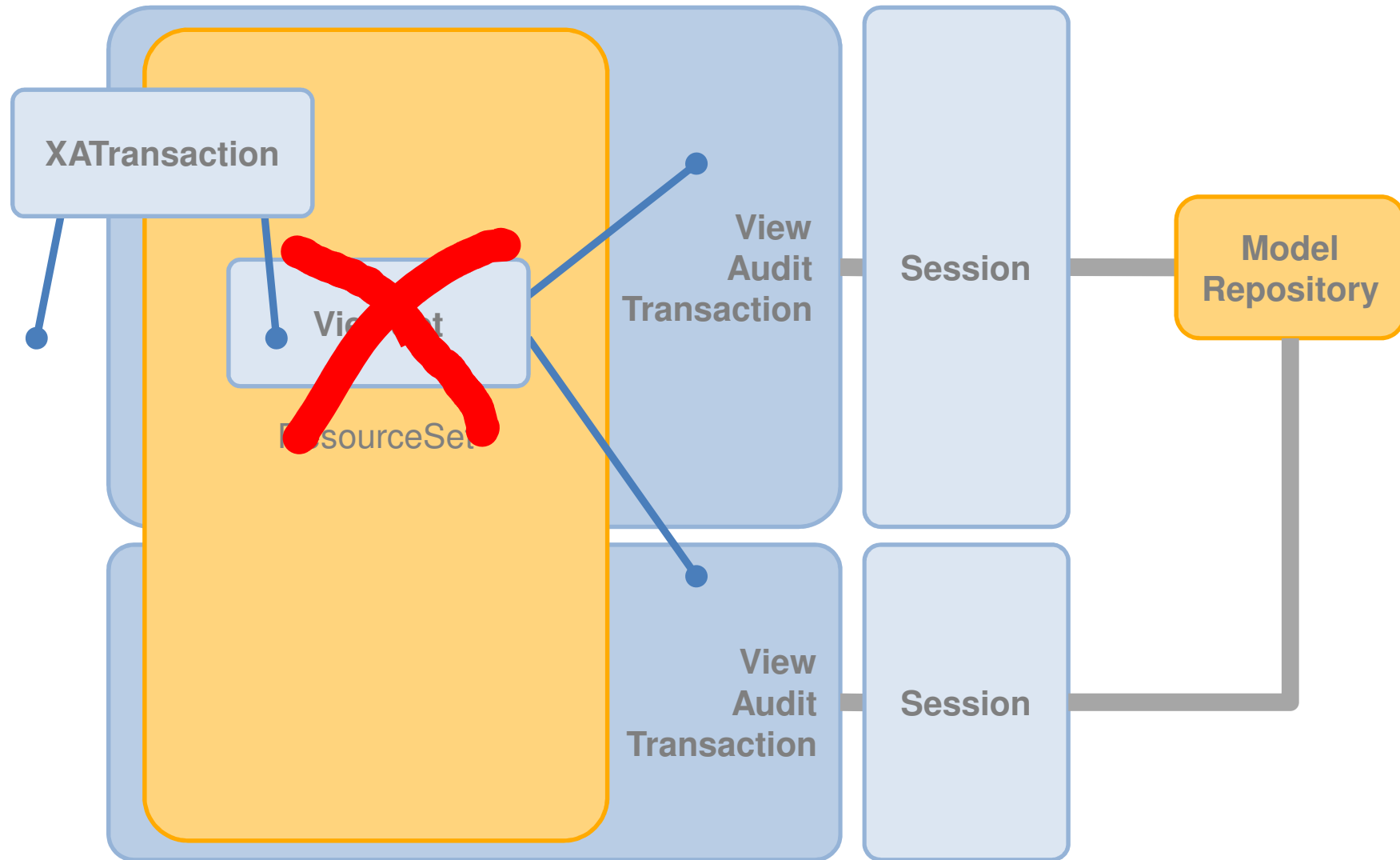
# Typical CDO Application



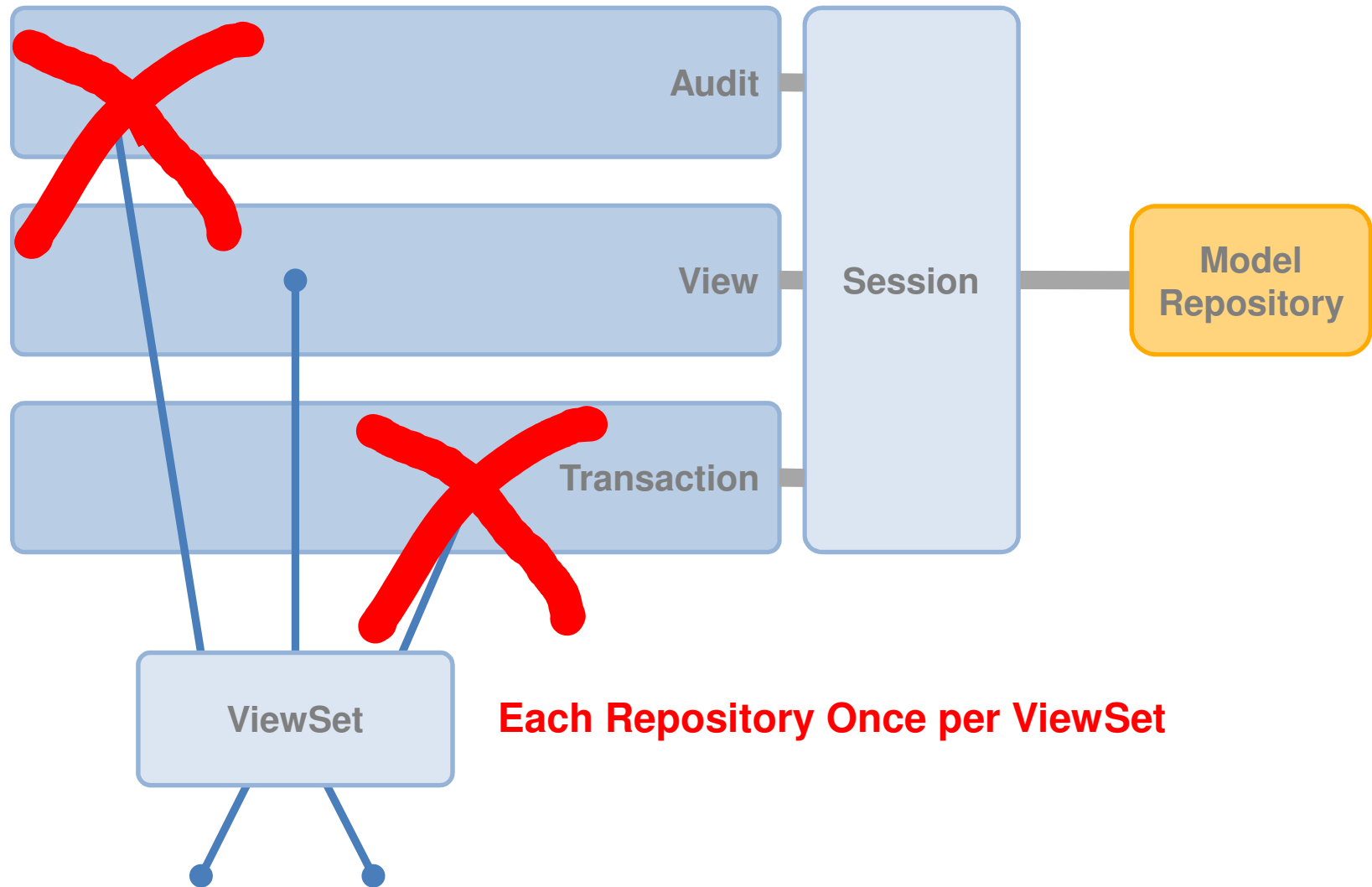
# Integration with ResourceSet



# Each Repository Once per ViewSet

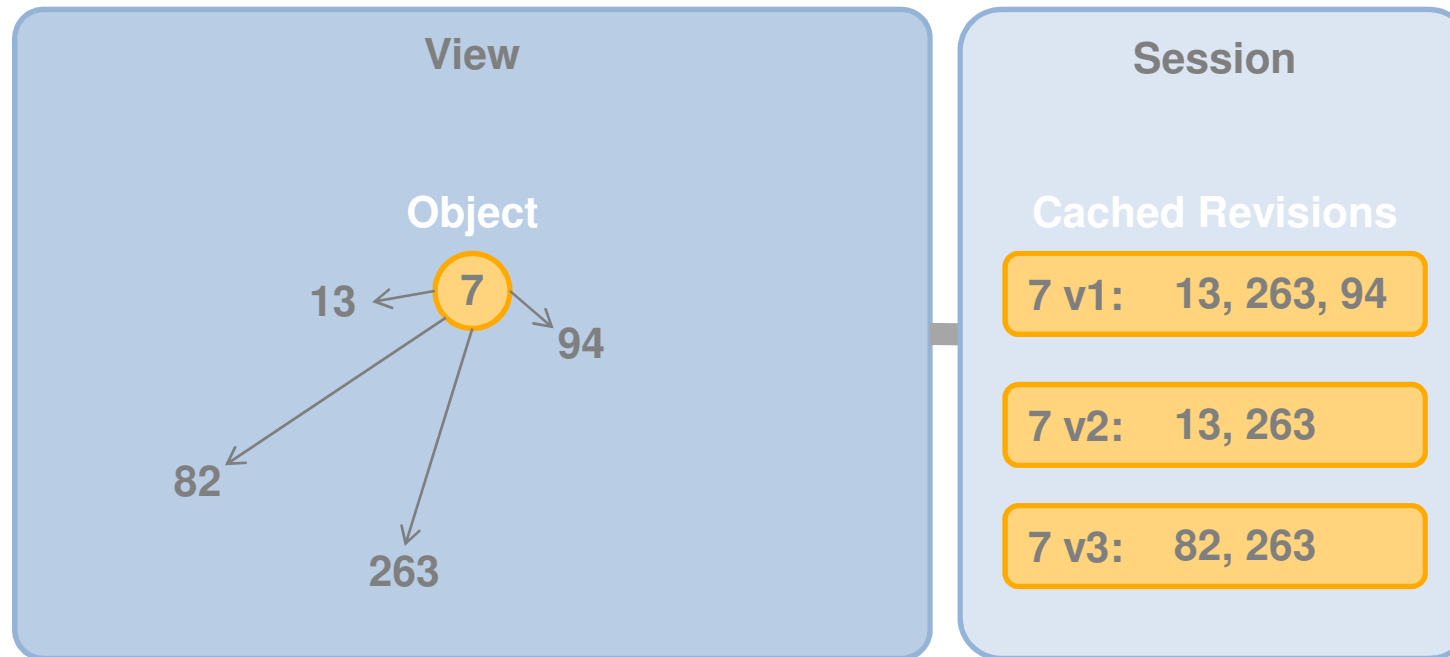


# Multiple Views per Session

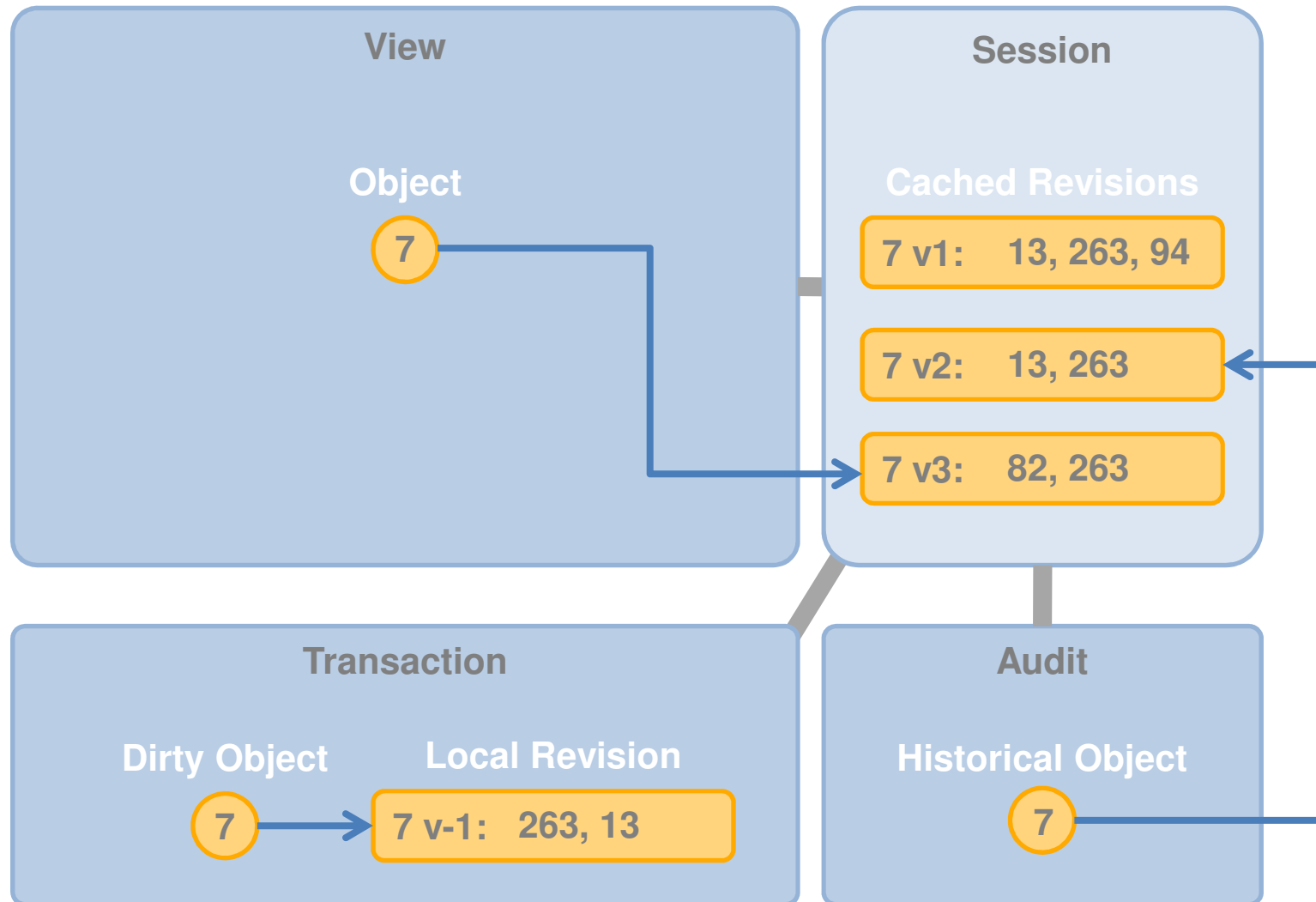




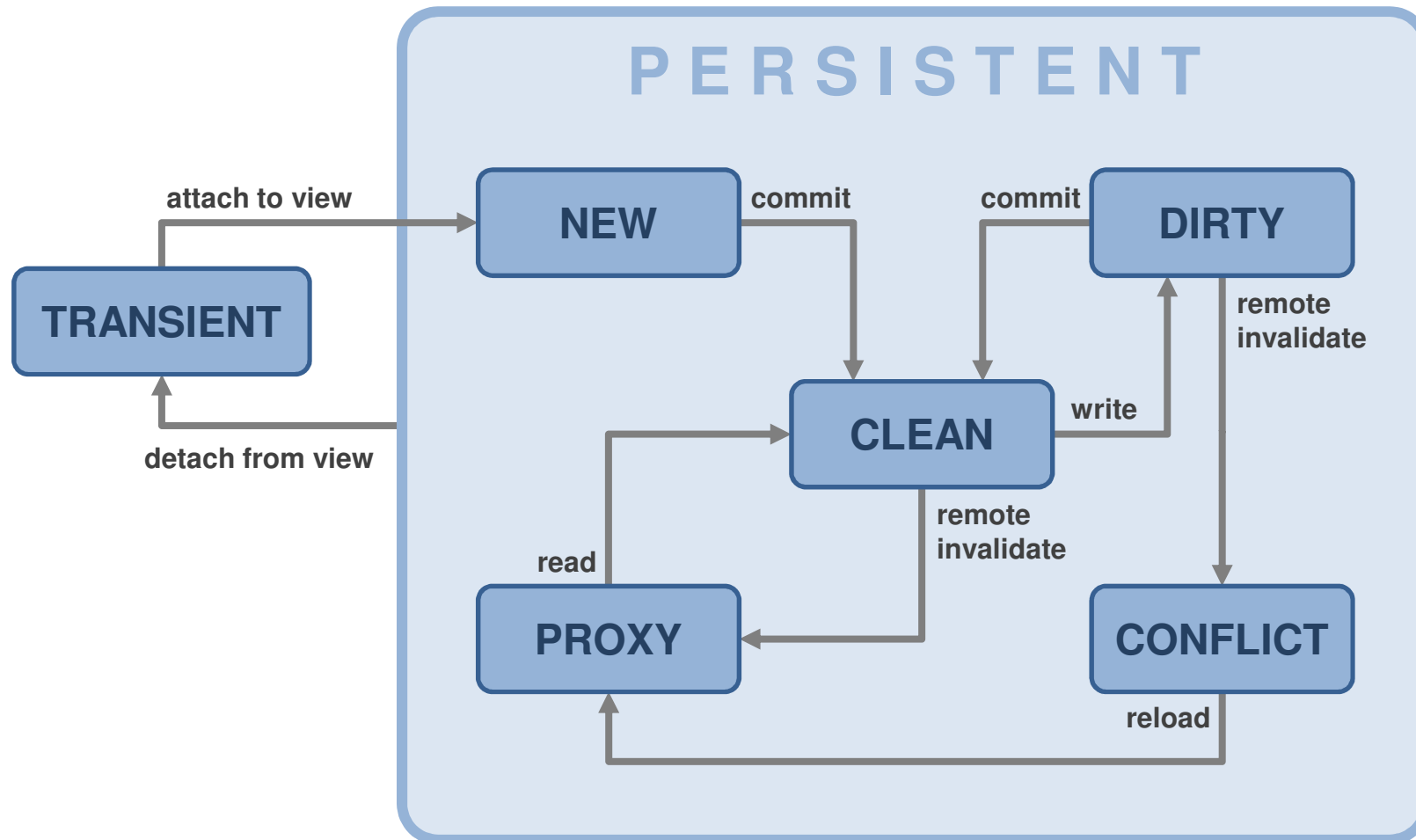
# Scalability through Revisions



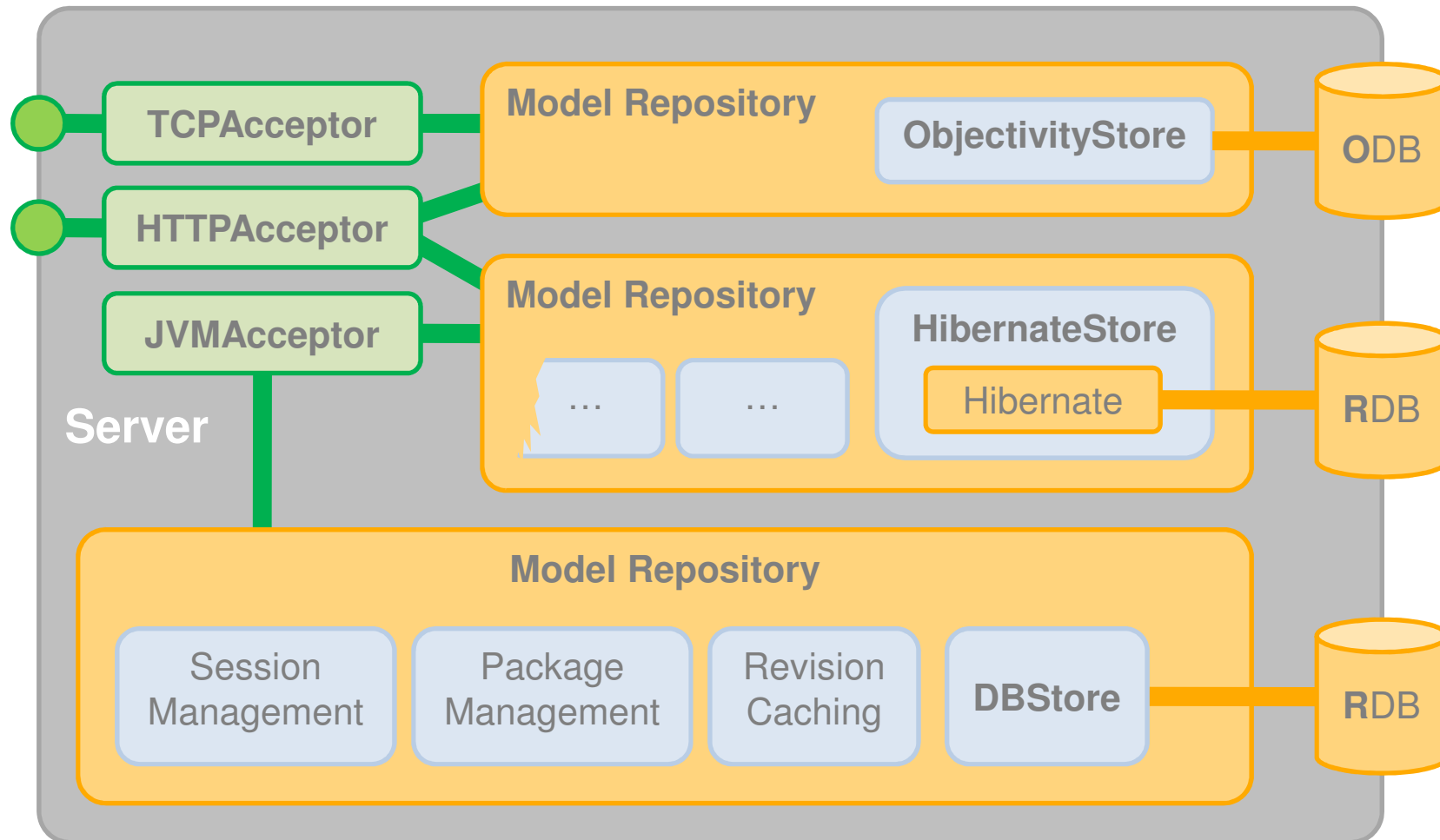
# Views are Revision Selectors



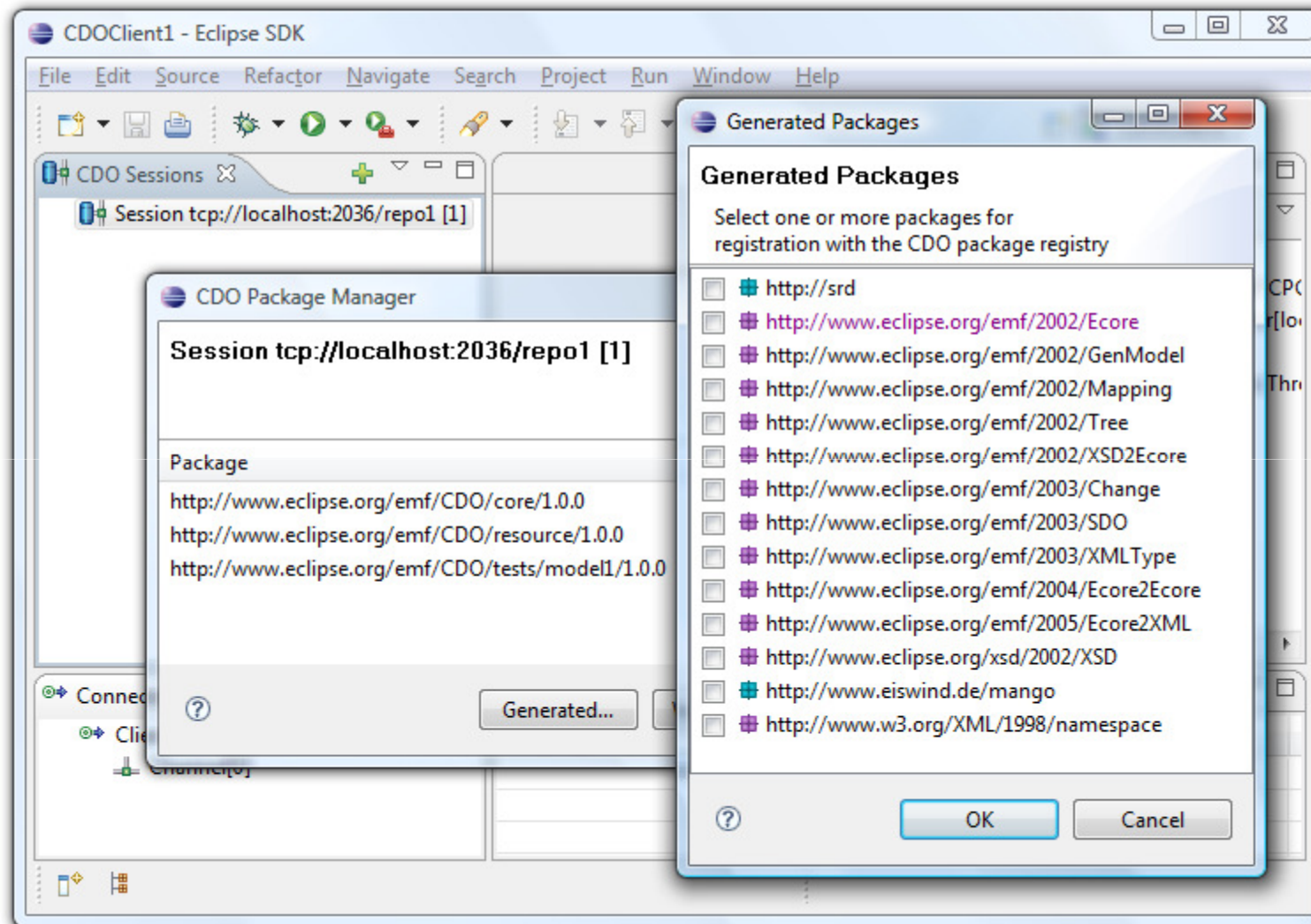
# Object State Machine



# Typical CDO Server



# Live Demonstration



# Opening a Session

```
public CDOSession openSession()
{
    // Create TCP connector through wiring container
    IConnector connector = (IConnector)IPluginContainer.INSTANCE
        .getElement("org.eclipse.net4j.connectors", "tcp", "dev.foo.com:2036");

    // Create CDO session configuration
    CDOSessionConfiguration config = CDONet4jUtil.createSessionConfiguration();
    config.setConnector(connector);
    config.setRepositoryName("repo1");
    config.setLazyPackageRegistry();

    // Open CDO session
    return config.openSession();
}
```

# Opening Views, Audits and Transactions

```
CDOSession session = openSession();
```

```
// Open multiple views on a session
```

```
CDOView view = session.openView();
```

```
CDOAudit audit = session.openAudit(new Date("2009-01-19").getTime());
```

```
CDOTransaction transaction= session.openTransaction();
```

```
// Use own ResourceSet
```

```
ResourceSet resourceSet = new ResourceSetImpl();
```

```
CDOTransaction transaction2= session.openTransaction(resourceSet);
```

```
// Associate transactions with a distributed transaction
```

```
CDOXATransaction xa = CDOUtil.createXATransaction();
```

```
xa.add(transaction.getViewSet());
```

```
xa.add(transaction2.getViewSet());
```

# Structured Resources / Queries

```
// Open CDO session and view
```

```
CDOSession session = openSession();
```

```
CDOView view = session.openView();
```

```
// Navigate through the resource folder structure
```

```
for (CDOResourceNode node : view.queryResources(null, "business", false))
```

```
{
```

```
    if (node instanceof CDOResourceFolder)
```

```
    {
```

```
        EList<CDOResourceNode> subNodes = ((CDOResourceFolder)node).getNodes();
```

```
    }
```

```
    else
```

```
    {
```

```
        EList<EObject> contents = ((CDOResource)node).getContents();
```

```
    }
```

```
}
```

```
// Close the session when done
```

```
session.close();
```



# Explicit Locking

```
CDOSession session = openSession();  
CDOTransaction transaction = session.openTransaction();  
transaction.setAutoReleaseLocksEnabled(true);
```

```
// Lock a single object for writing
```

```
CDOResource resource = transaction.getResource("/my/resource");  
resource.cdoWriteLock().lock();
```

```
// Modify that object
```

```
resource.getContents().add(new Company());  
resource.getContents().add(new Company());  
resource.getContents().add(new Company());
```

```
// Commit atomically and release all locks
```

```
transaction.commit();  
session.close();
```

# Save Points

```
CDOSession session = openSession();  
CDOTransaction transaction = session.openTransaction();
```

```
// Create and populate a resource
```

```
CDOResource resource = transaction.getOrCreateResource("/my/resource");  
resource.getContents().add(new Customer());  
resource.getContents().add(new Customer());
```

```
// Set save point, modify and rollback
```

```
CDOSavepoint savepoint = transaction.setSavepoint();  
resource.getContents().add(new Supplier());  
resource.getContents().add(new Supplier());  
transaction.rollback(savepoint);
```

```
// Commit only the first changes (customers)
```

```
transaction.commit();  
session.close();
```

# Passive Updates

```
CDOSession session = openSession();  
CDOView view = session.openView();  
CDOResource resource = view.getResource("/my/resource");
```

```
// Decouple from passive updates
```

```
session.options().setPassiveUpdateEnabled(false);
```

```
for (EObject object : resource.getContents())
```

```
{
```

```
    Company company = (Company)object;
```

```
    // Work with local model...
```

```
}
```

```
session.refresh();
```

```
// Work with refreshed model...
```

```
// Stay in sync from here
```

```
session.options().setPassiveUpdateEnabled(true);
```

# Change Subscriptions

```
CDOSession session = openSession();
CDOView view = session.openView();

// Subscribe to repository for CDO adapters
view.options().setChangeSubscriptionPolicy(CDOAdapterPolicy.CDO);

// Define your CDO adapter
class MyAdapter extends AdapterImpl implements CDOAdapter
{
    @Override
    public void notifyChanged(Notification msg)
    {
        System.out.println("Modified remotely: " + msg.getNotifier());
    }
}

// Attach your adapter to any object to trigger a particular subscription
CDOResource resource = view.getResource("/my/resource");
resource.eAdapters().add(new MyAdapter());
```

# Query Framework

```
CDOSession session = openSession();  
CDOView view = session.openView();
```

```
// Create query
```

```
CDOQuery query = view.createQuery("SQL",  
    "SELECT cdoId FROM Company WHERE name LIKE ${name}");  
query.setParameter("name", "Foo%");  
query.setMaxResults(35);
```

```
// Send query to server and iterate the result asynchronously
```

```
for (Iterator<Company> it = query.getResultAsync(Company.class); it.hasNext())  
{  
    Company company = it.next();  
    System.out.println(company);  
}
```

```
// Closing the view closes all open queries
```

```
session.close();
```

# NEXT STEPS

## Offline mode

- Checkout
- Re-sync

## Legacy model support

- Ecore
- UML
- 3rd party models

## OCL server-side

- As common query language
- For commit validation

## Model code server-side

- Custom data types (persistence)
- Operations (e.g. in OCL)

## Integration

- GMF integration
- Workspace integration
- Team integration
- CDO Explorer

