Eike Stepper

stepper@esc-net.de
http://www.esc-net.de
http://thegordian.blogspot.com

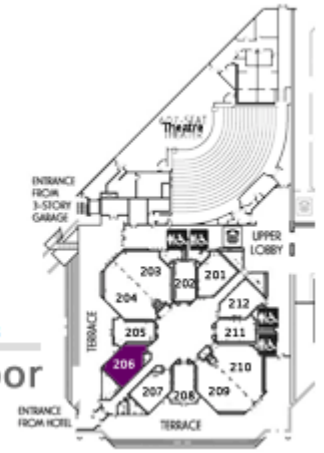
Berlin, Germany

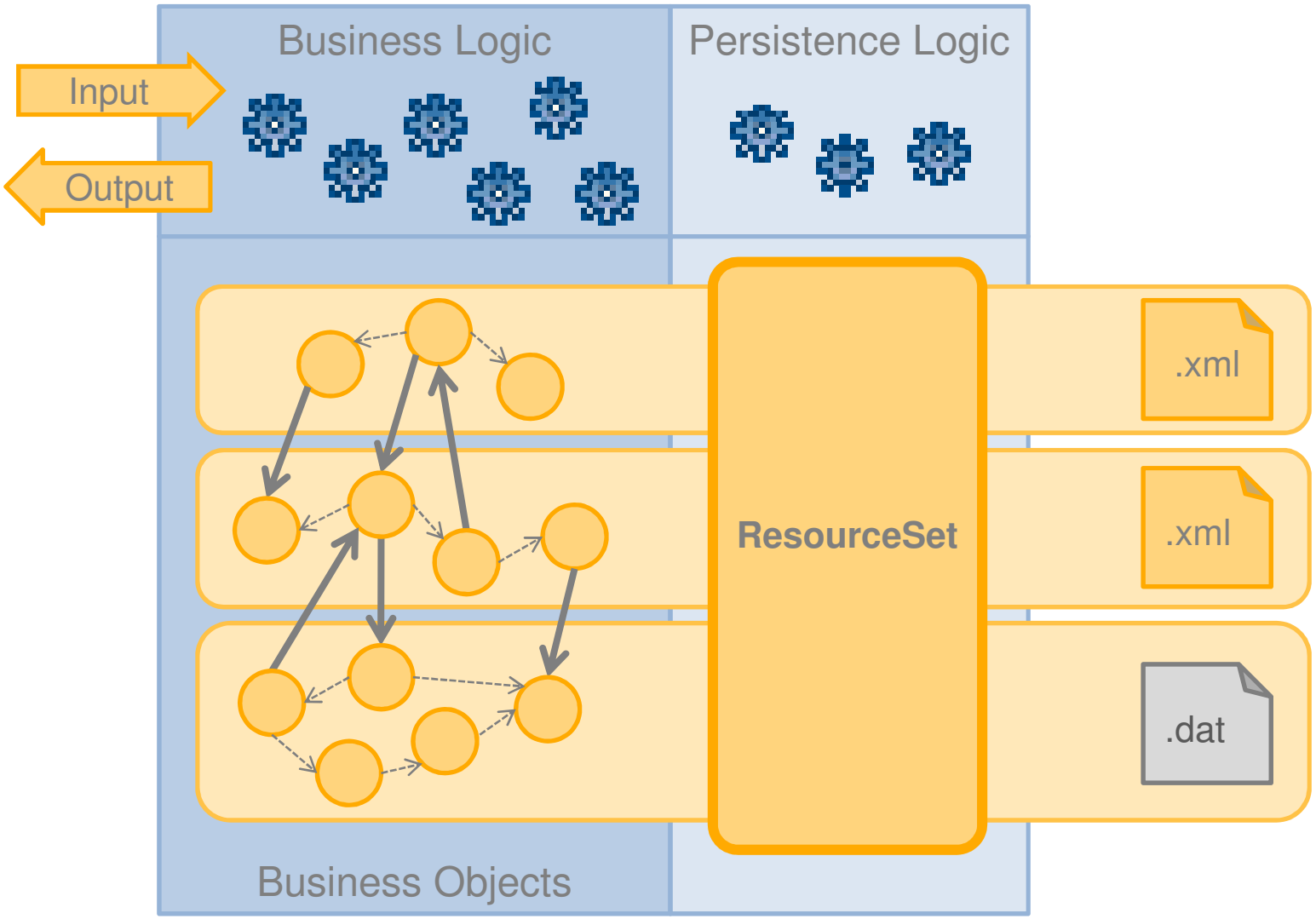
Scale, Share and Store your Models with CDO 2.0

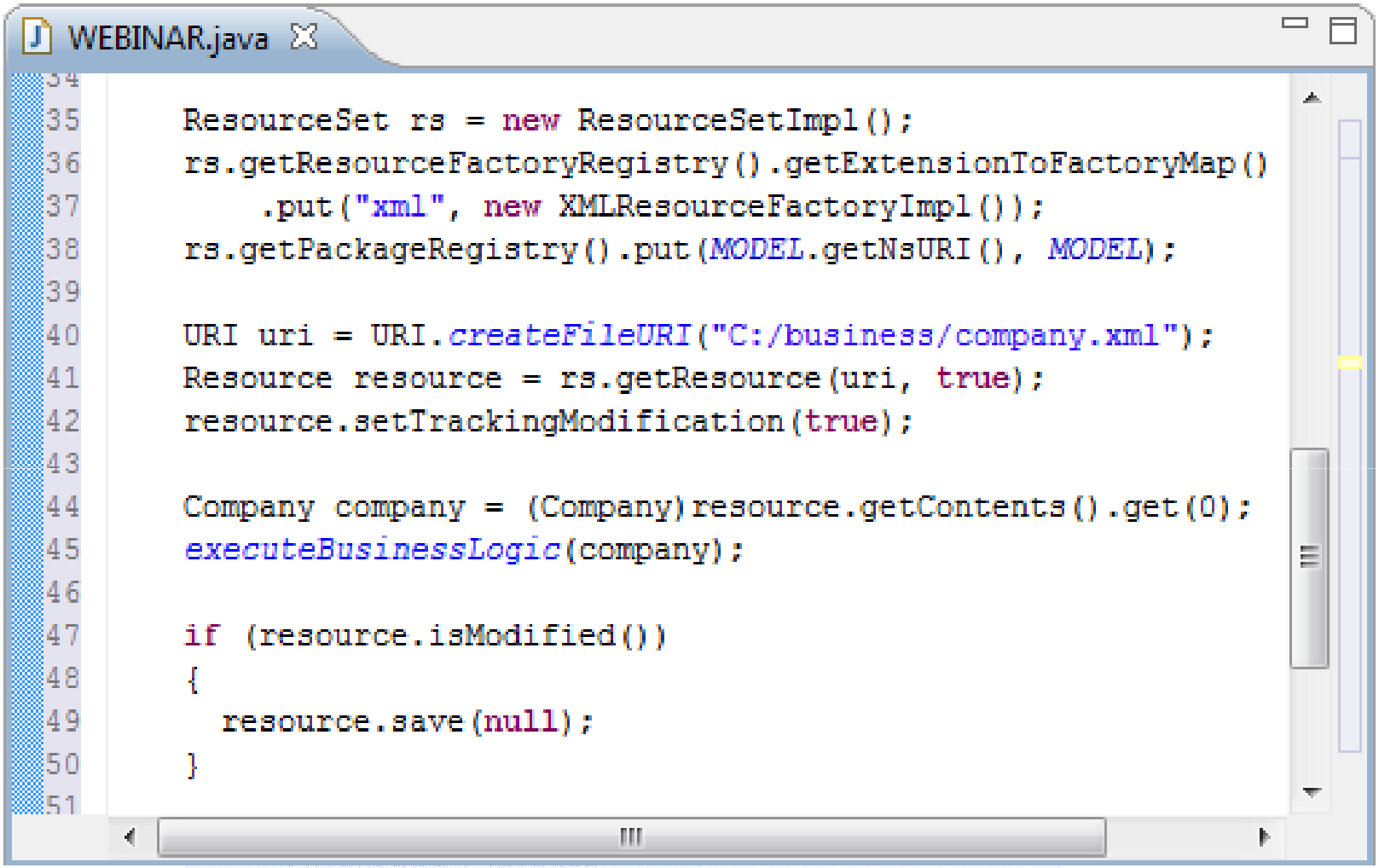
EclipseCon Talk, January 29, 2009

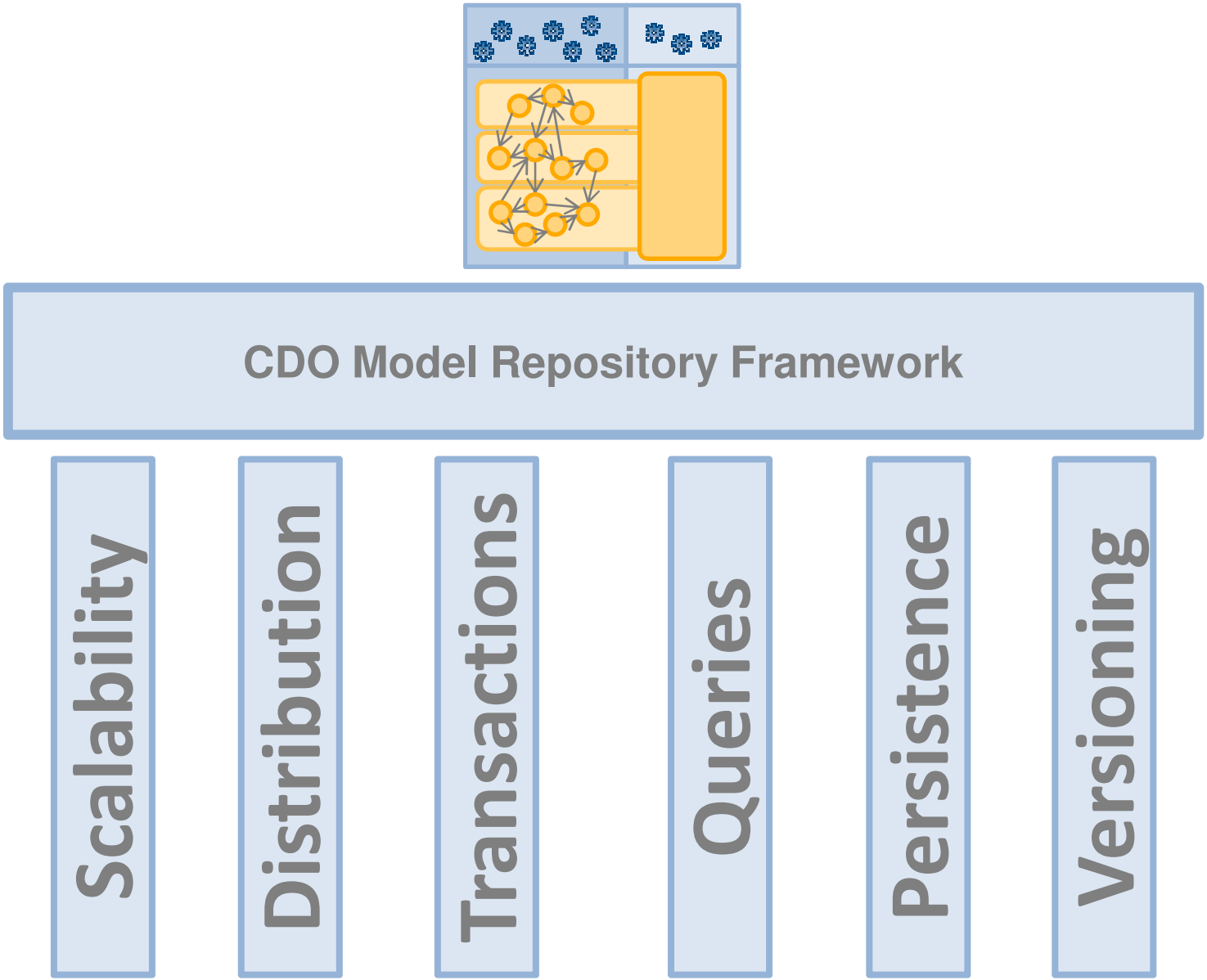
Tuesday, 11:10, 50 minutes | Room 206

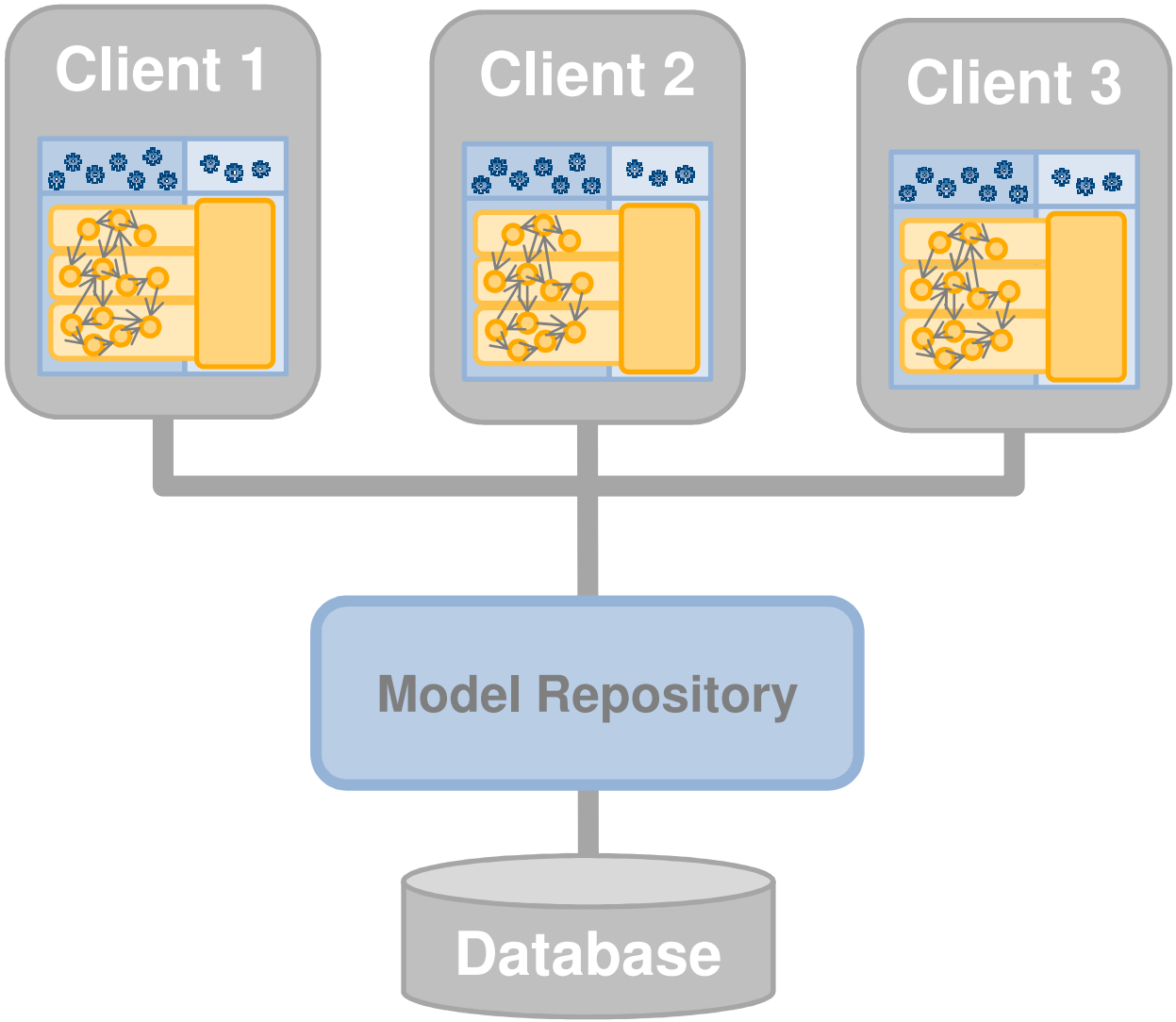
7 · 8 · 9 · 10 · 11 · 12 · 13 · 14 · 15 · 16 · 17 · 18

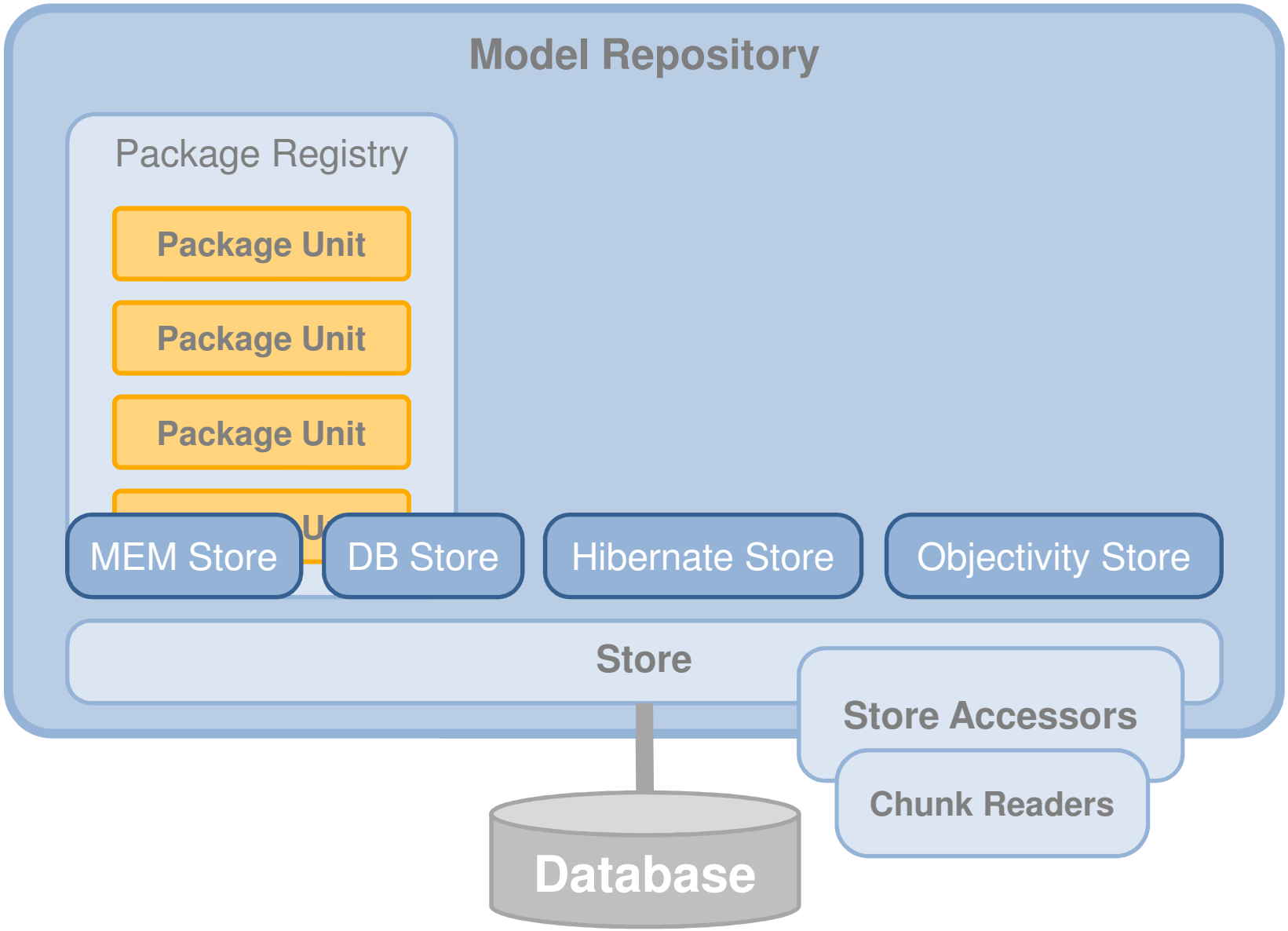


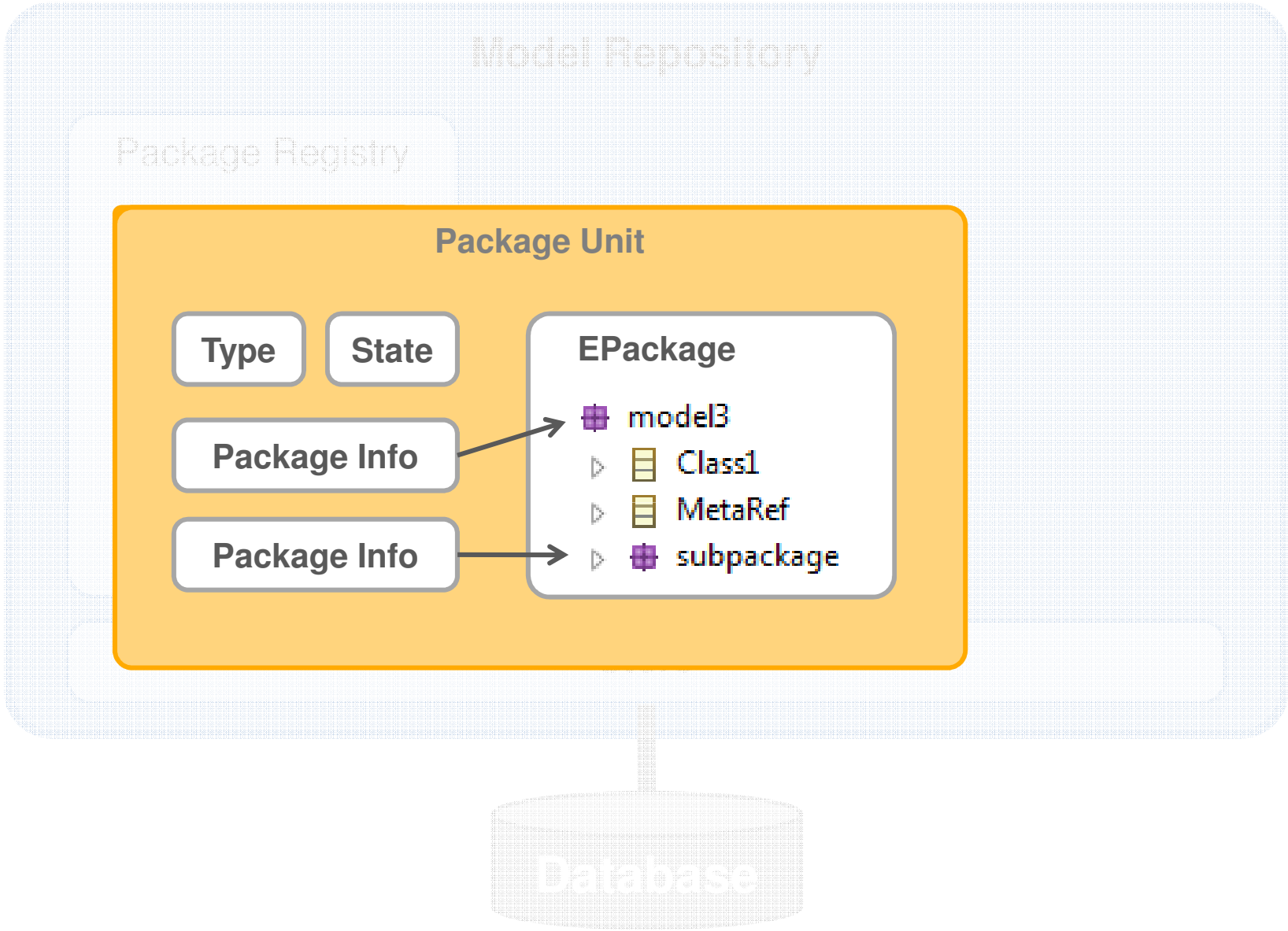


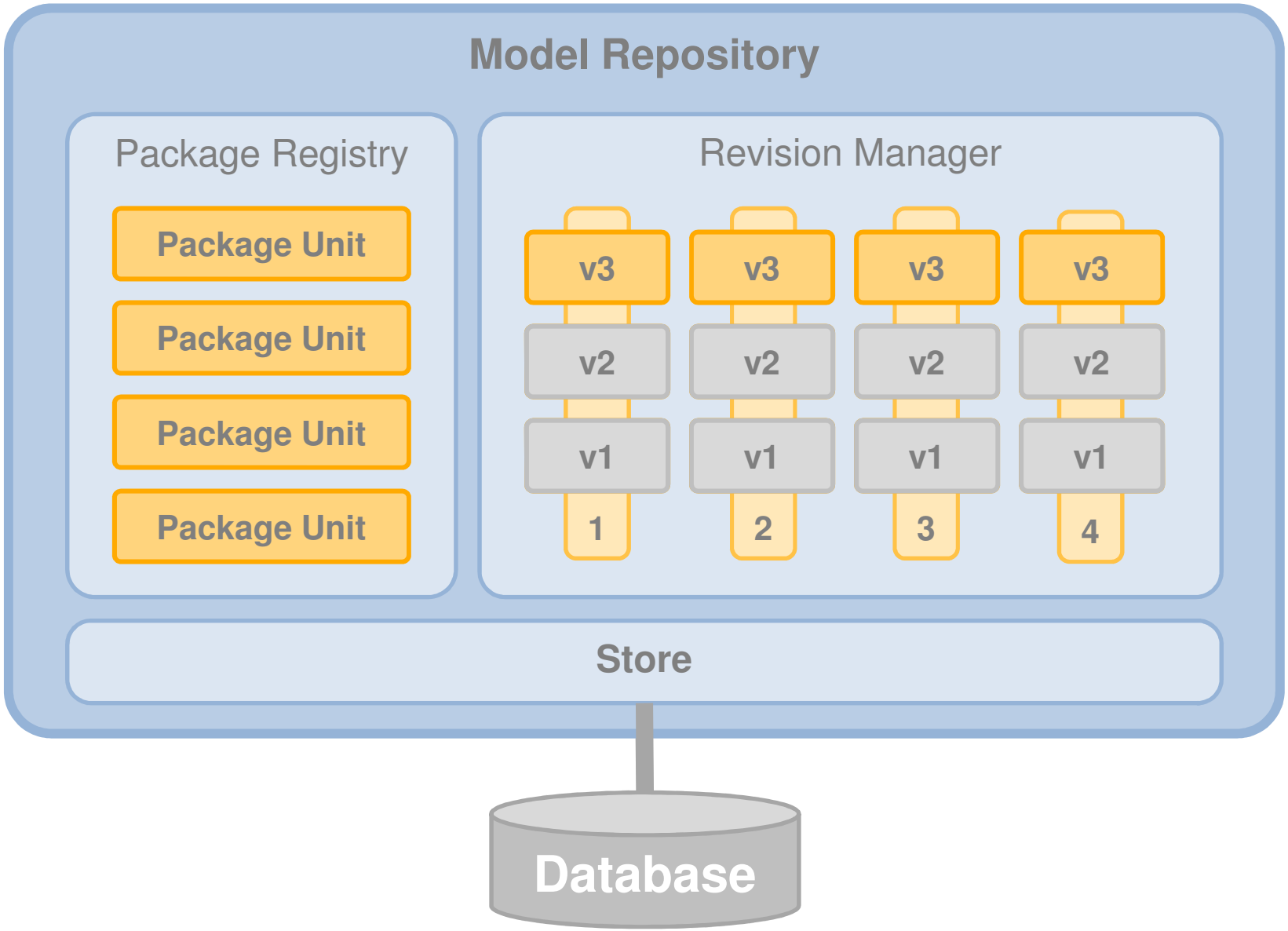


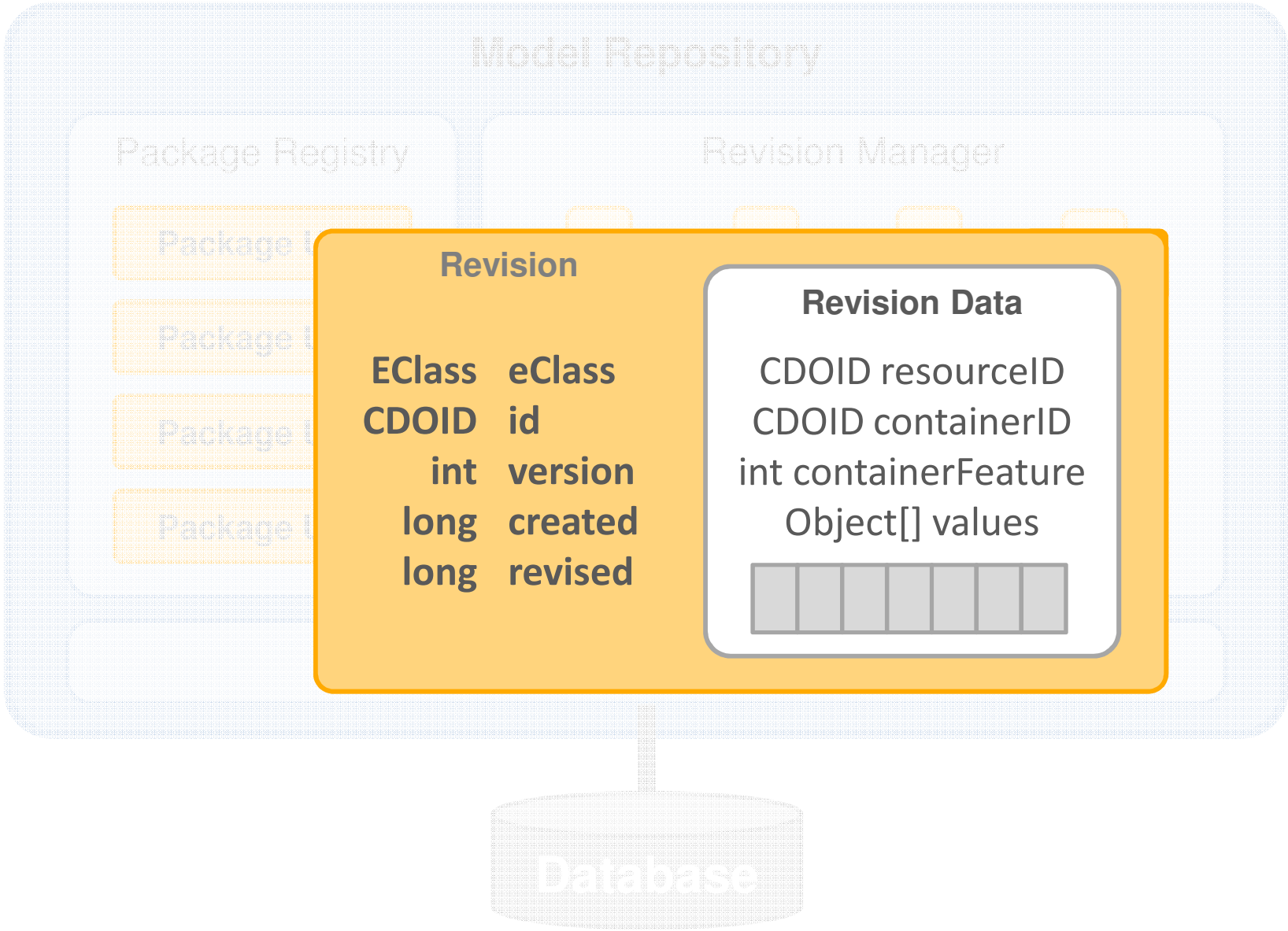

CDO Model Repository Framework
Scalability
Distribution
Transactions
Queries
Persistence
Versioning

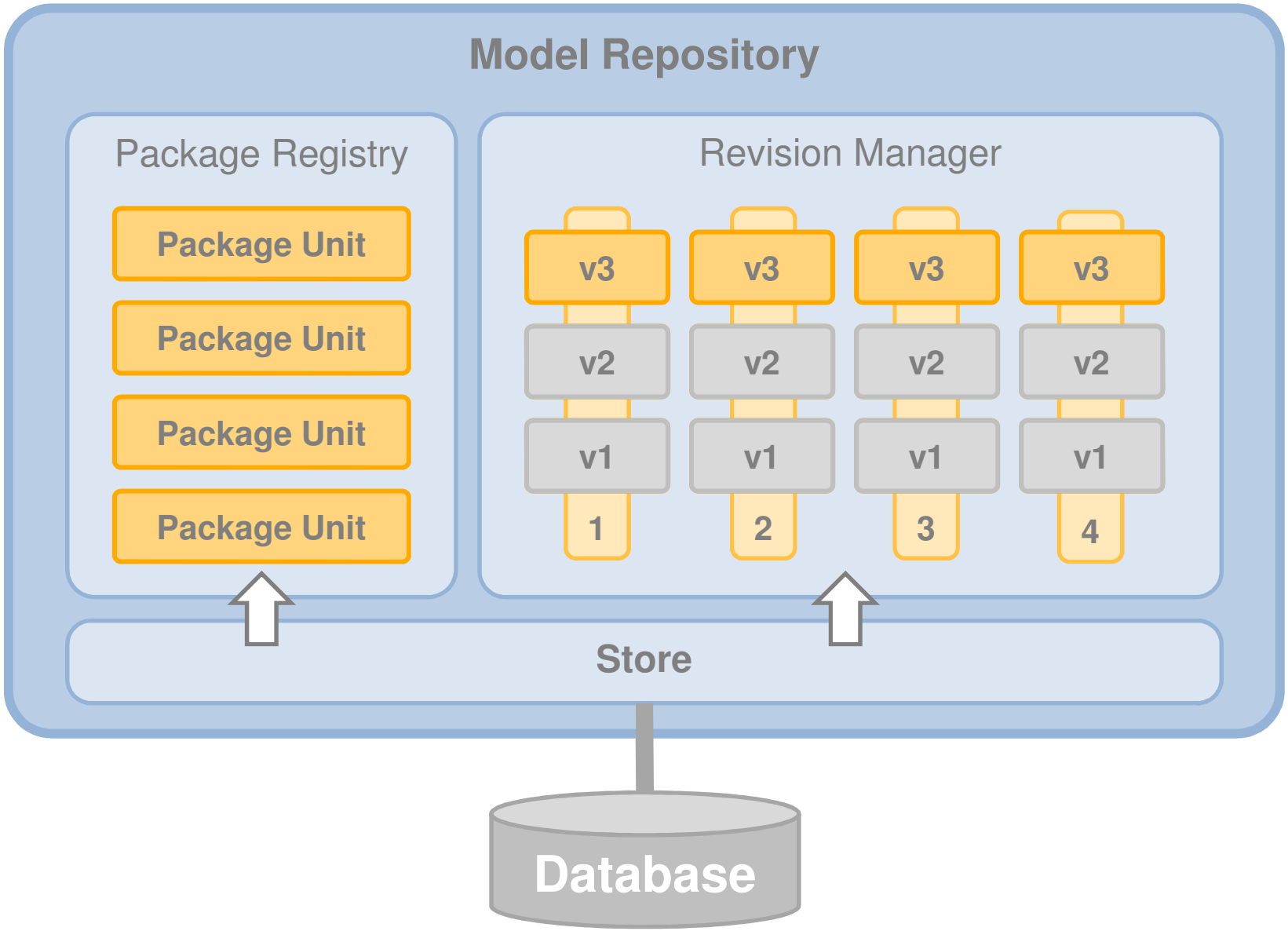


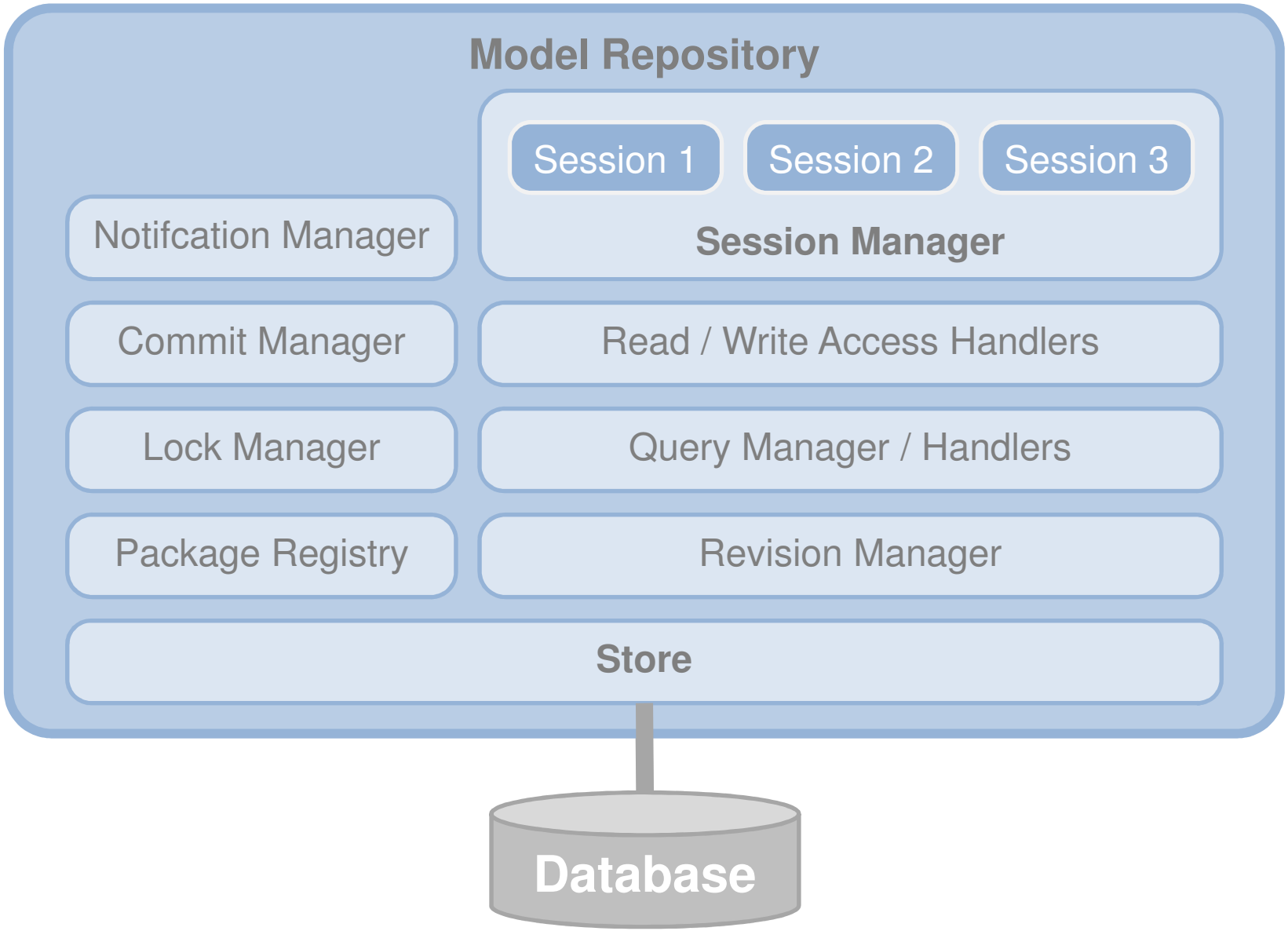






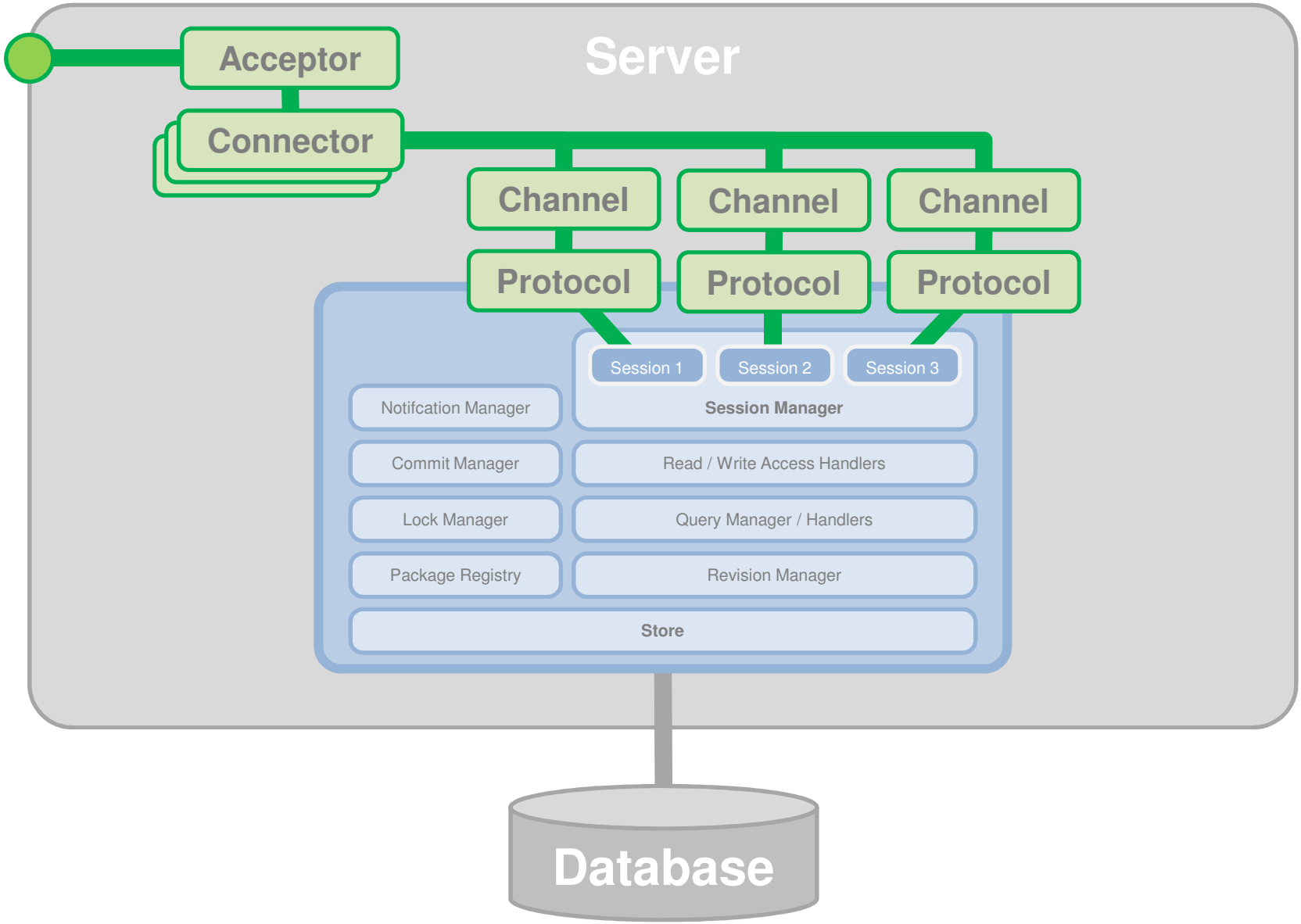






TestServer.java

```
IMappingStrategy mappingStrategy = CDODBUtil.createHorizontalMappingStrategy();
IDBAdapter dbAdapter = new EmbeddedDerbyAdapter();

EmbeddedDataSource dataSource = new EmbeddedDataSource();
dataSource.setUser("sa");
dataSource.setDatabaseName("cdo");
dataSource.setCreateDatabase("create");
IDBConnectionProvider dbConnectionProvider = DBUtil.createConnectionProvider(dataSource);

IStore store = CDODBUtil.createStore(mappingStrategy, dbAdapter, dbConnectionProvider);

Map<String, String> props = new HashMap<String, String>();
props.put(IRepository.Props.CURRENT_LRU_CAPACITY, "100000");

IRepository repository = CDOServerUtil.createRepository(REPOSITORY_NAME, store, props);
```

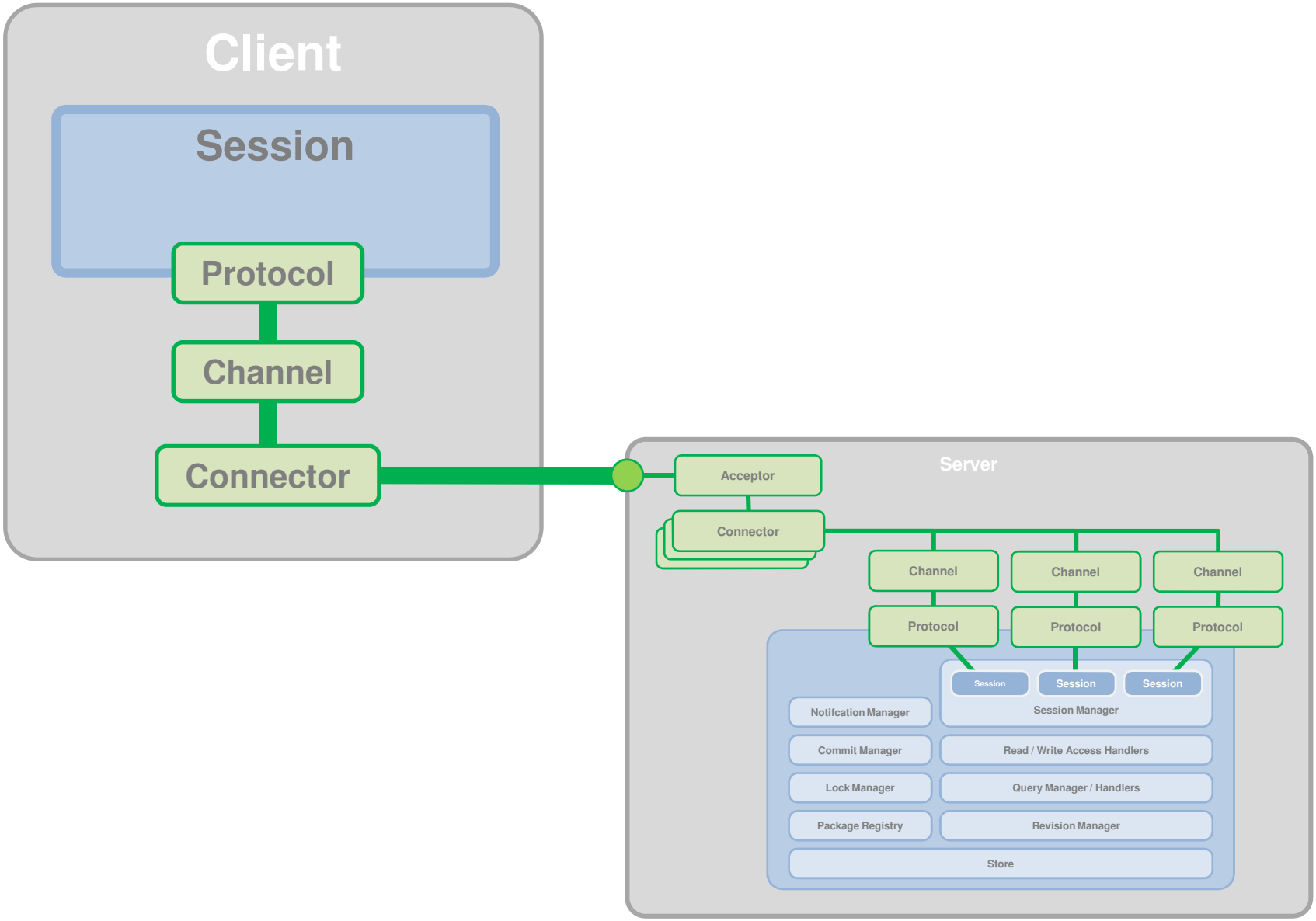


TestServer.java

```
IManagedContainer container = ContainerUtil.createContainer();
Net4jUtil.prepareContainer(container);
TCPUtil.prepareContainer(container);
CDOServerUtil.prepareContainer(container);
CDOServerUtil.addRepository(container, repository);

IAcceptor acceptor = (IAcceptor)container.getElement("org.eclipse.net4j.acceptors", "tcp", null);

System.out.println("Press any key to shutdown");
while (System.in.read() == -1)
{
  Thread.sleep(200);
}

container.deactivate();
```

cdo-server.xml

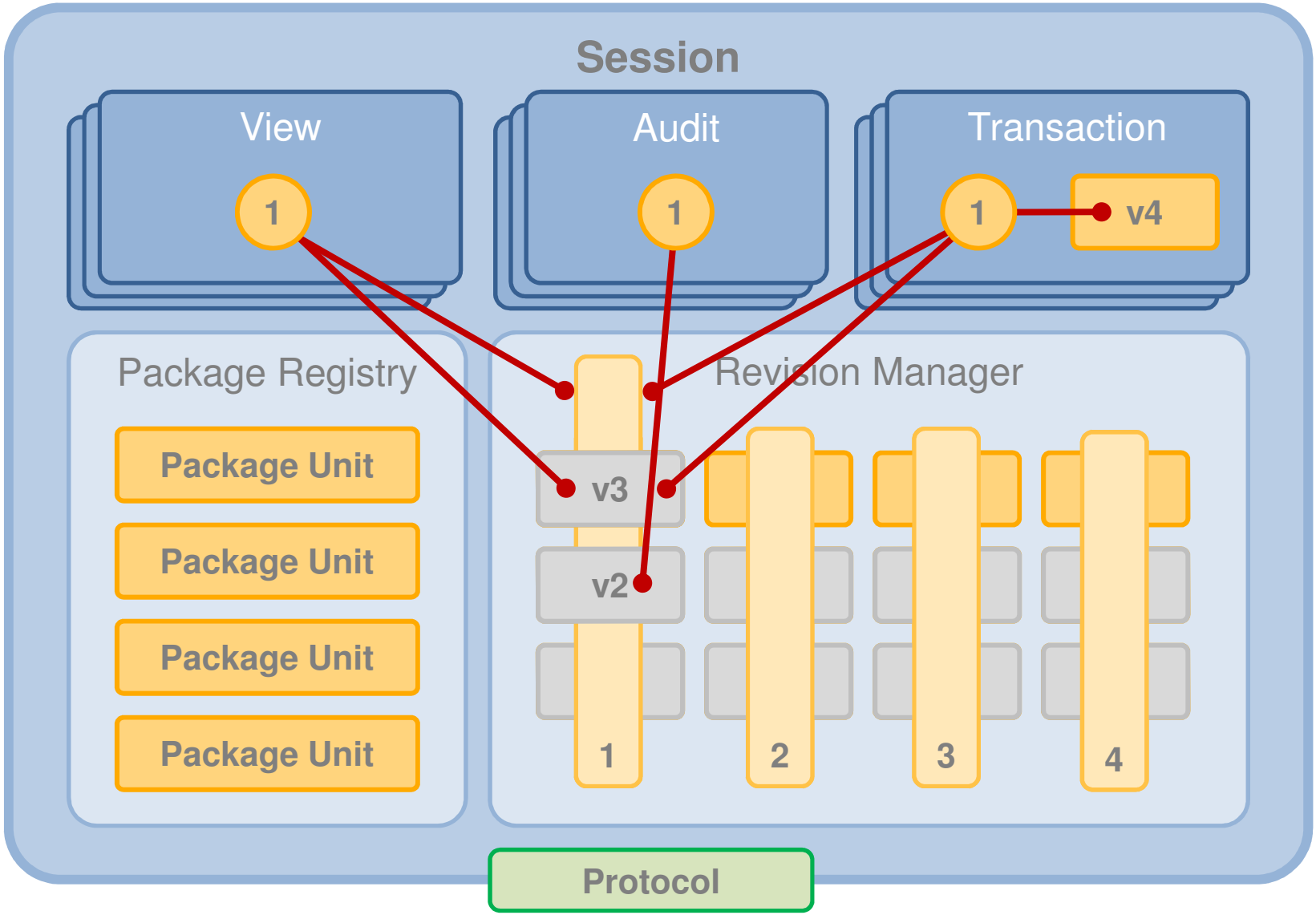
```
<?xml version="1.0" encoding="UTF-8"?>
<cdoServer>
  <repository name="repo1">
    <property name="currentLRUCapacity" value="100000"/>

    <store type="db">
      <mappingStrategy type="horizontal"/>
      <jdbcDelegate type="preparedStatement"/>
      <dbAdapter name="derby-embedded"/>
      <dataSource class="org.apache.derby.jdbc.EmbeddedDataSource"
        databaseName="/temp/cdodb1" createDatabase="create"/>
    </store>
  </repository>

  <acceptor type="tcp" listenAddr="0.0.0.0" port="2036"/>
</cdoServer>
```

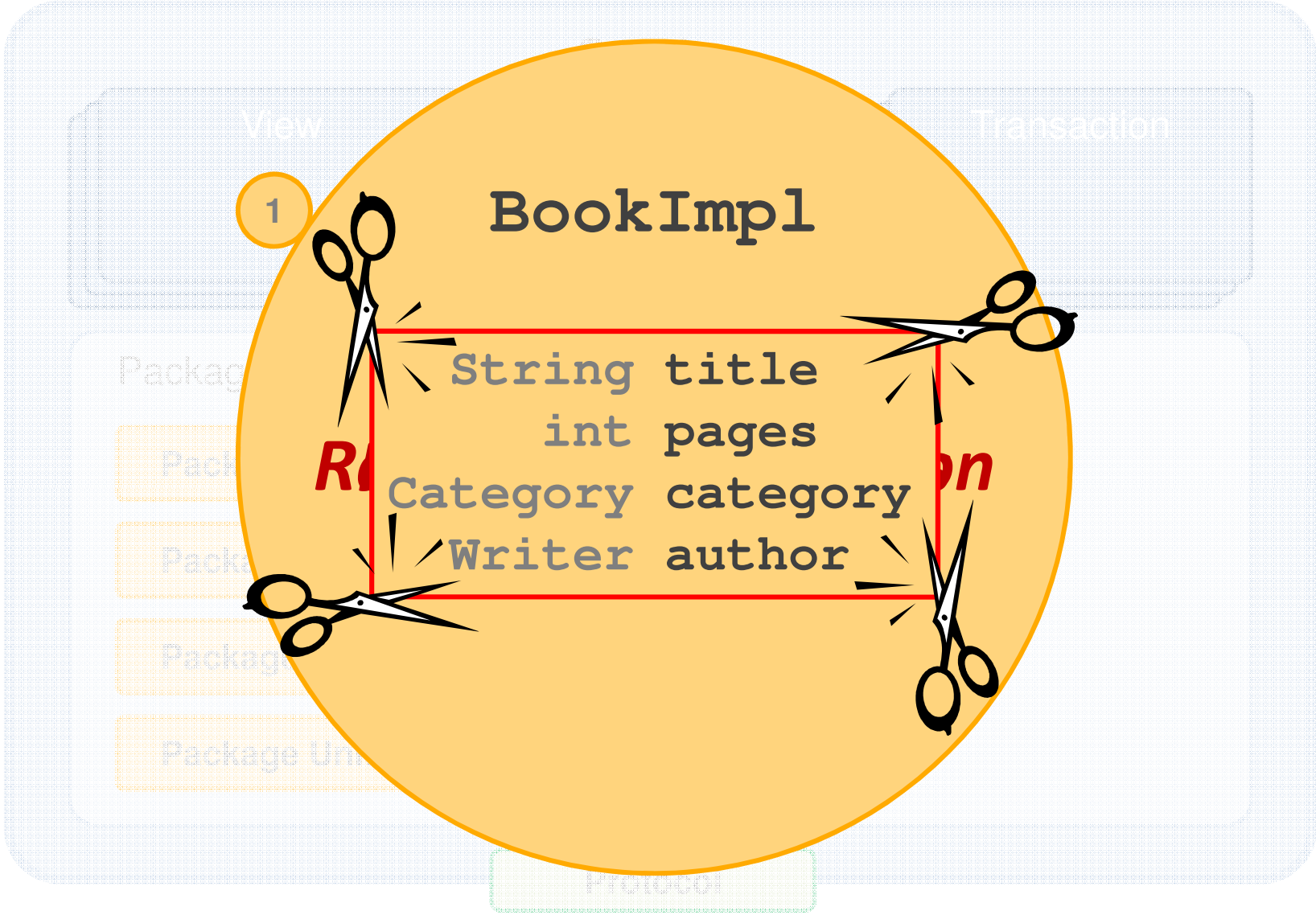


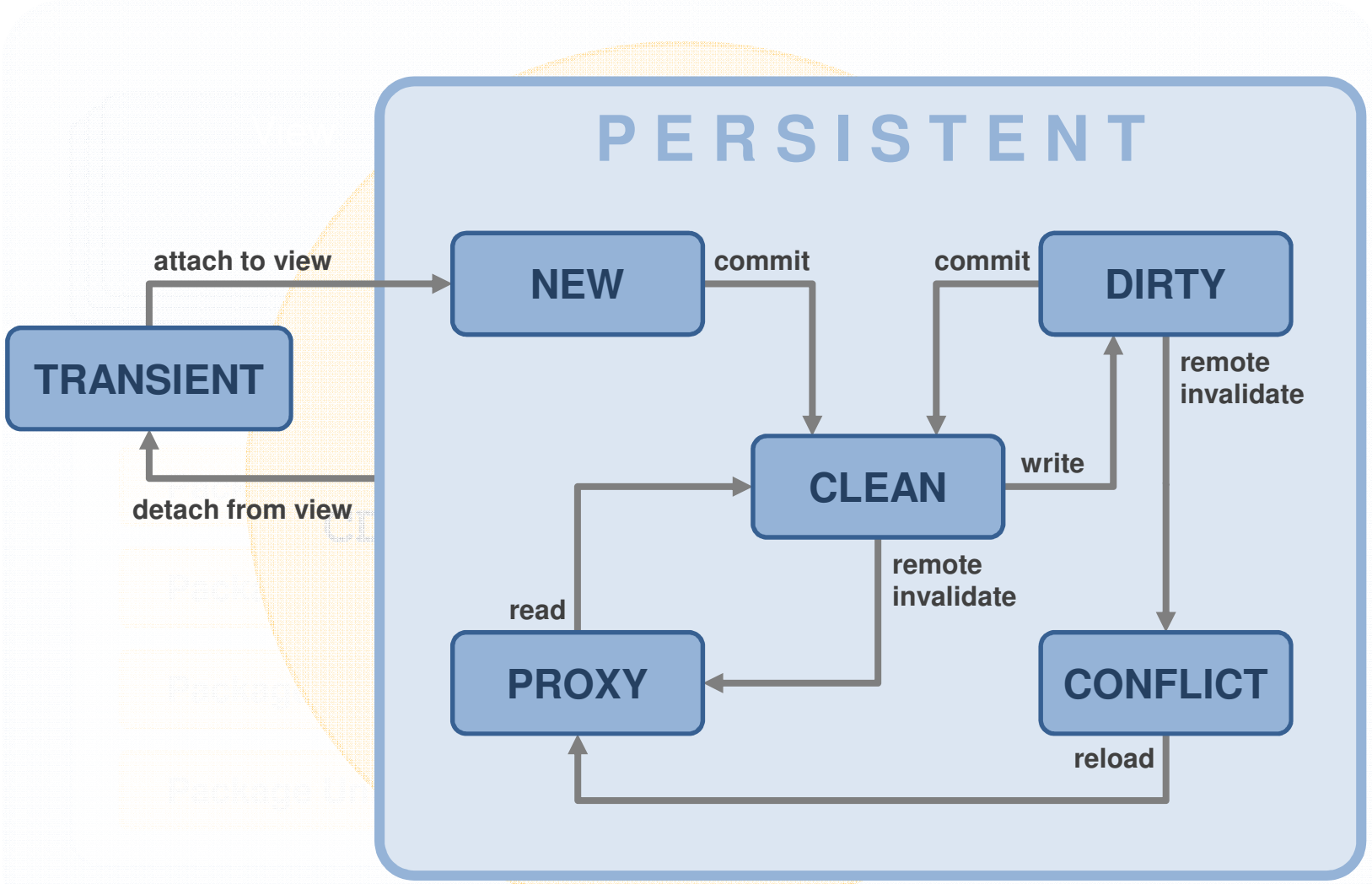
Client

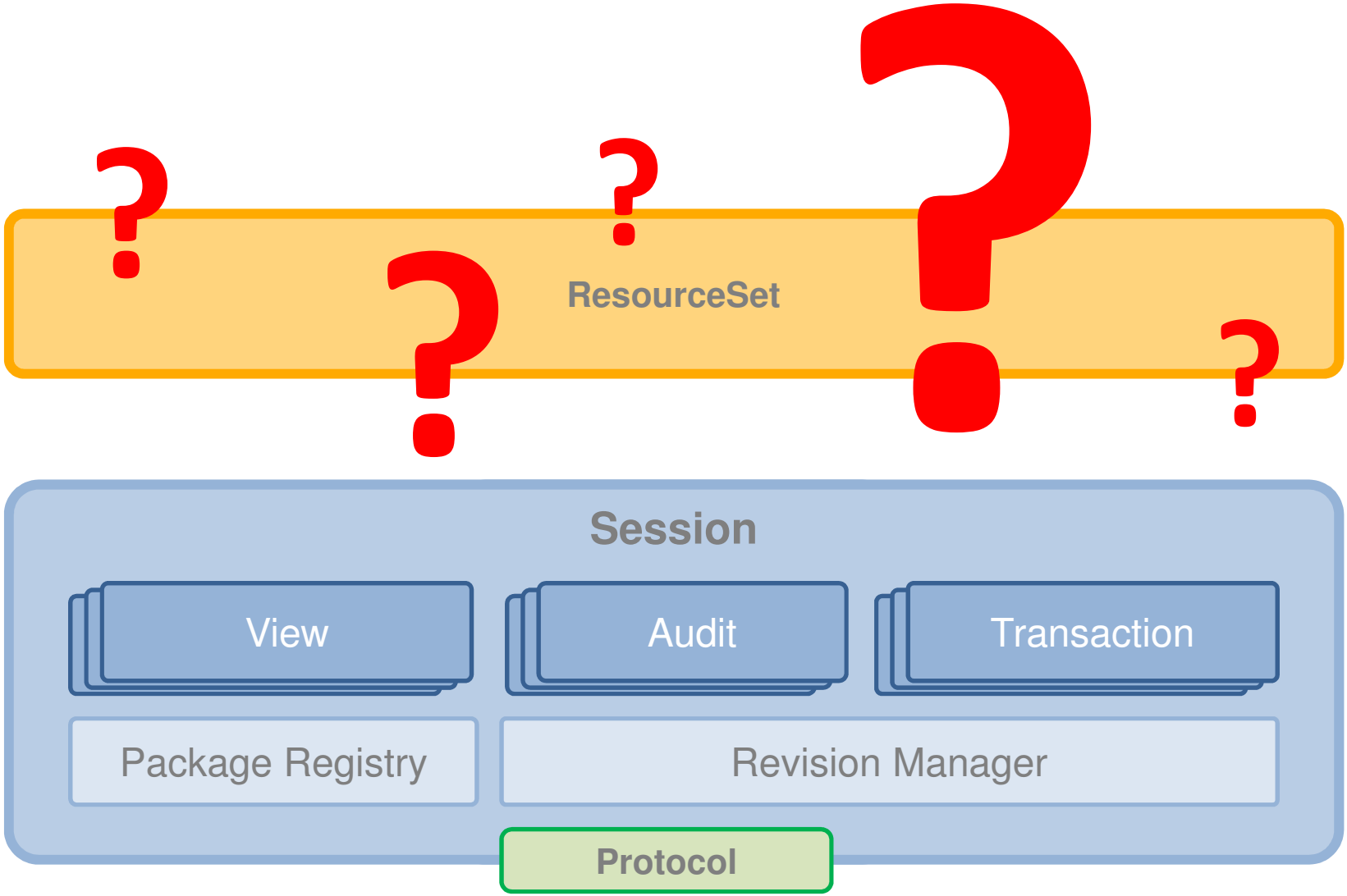
TestClient.java

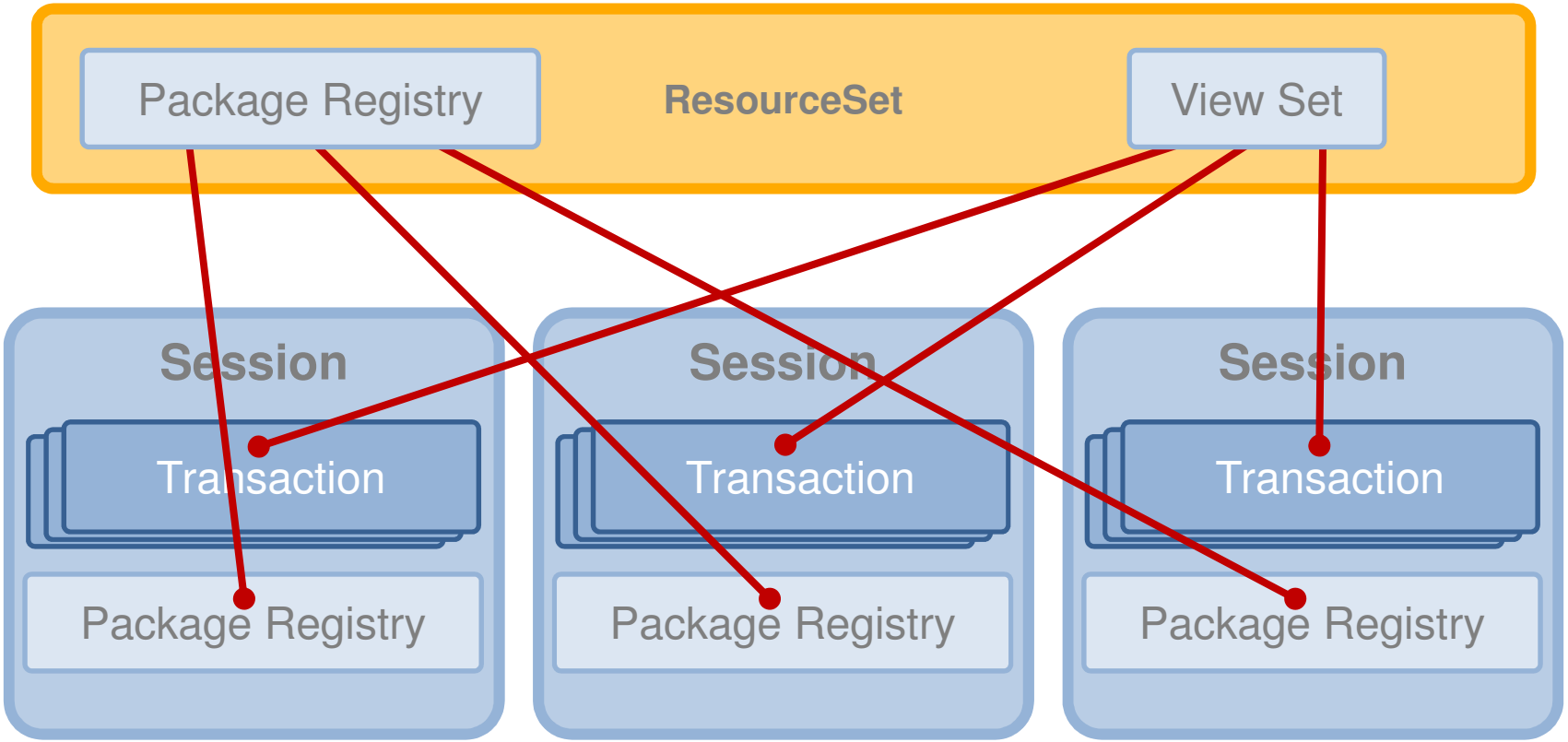
```
IManagedContainer container = ContainerUtil.createContainer();
Net4jUtil.prepareContainer(container);
TCPUtil.prepareContainer(container);

IConnector connector = TCPUtil.getConnector(container, "localhost:2036");

CDOSessionConfiguration configuration = CDONet4jUtil.createSessionConfiguration();
configuration.setConnector(connector);
configuration.setRepositoryName("repo1");

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

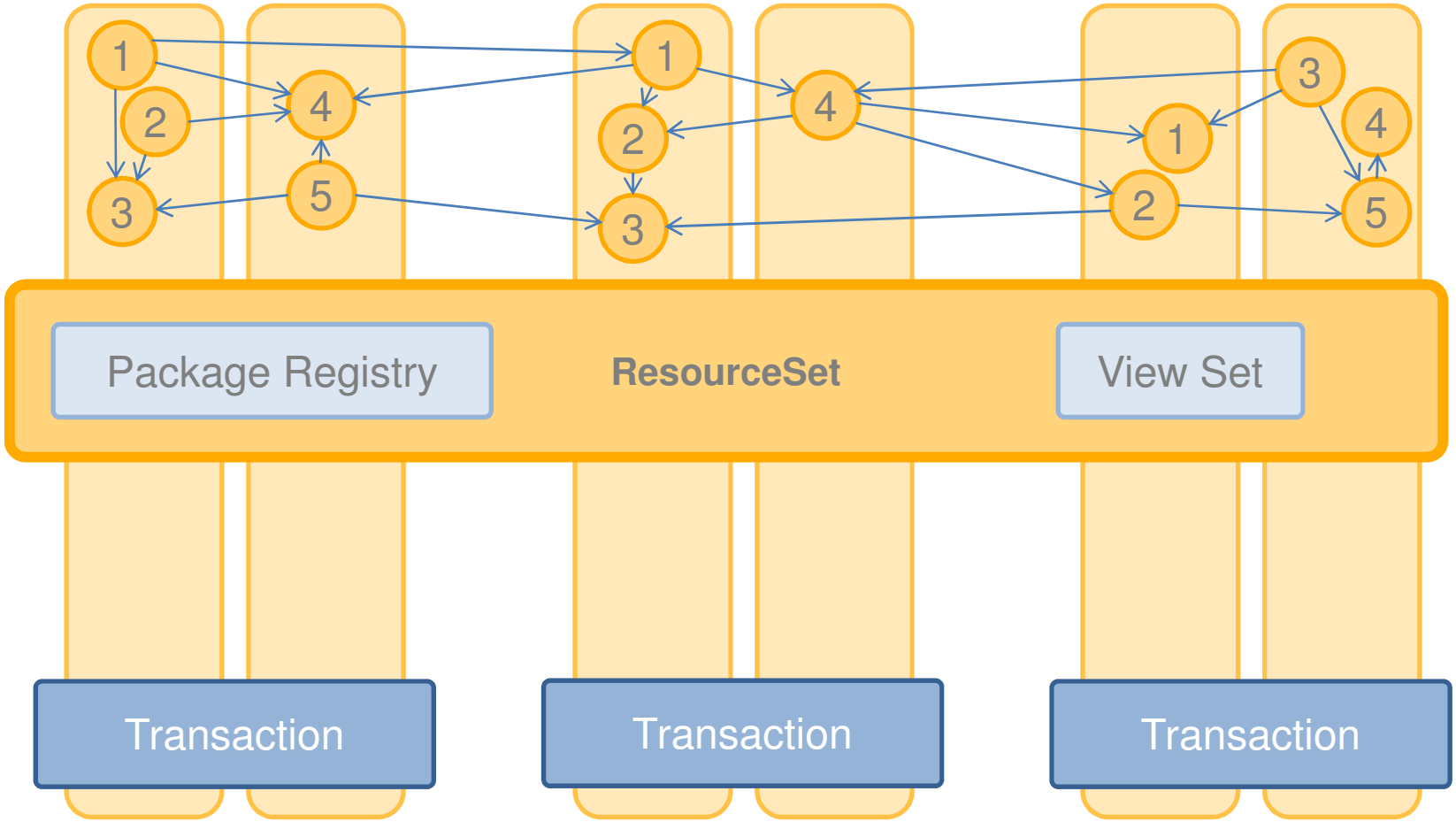


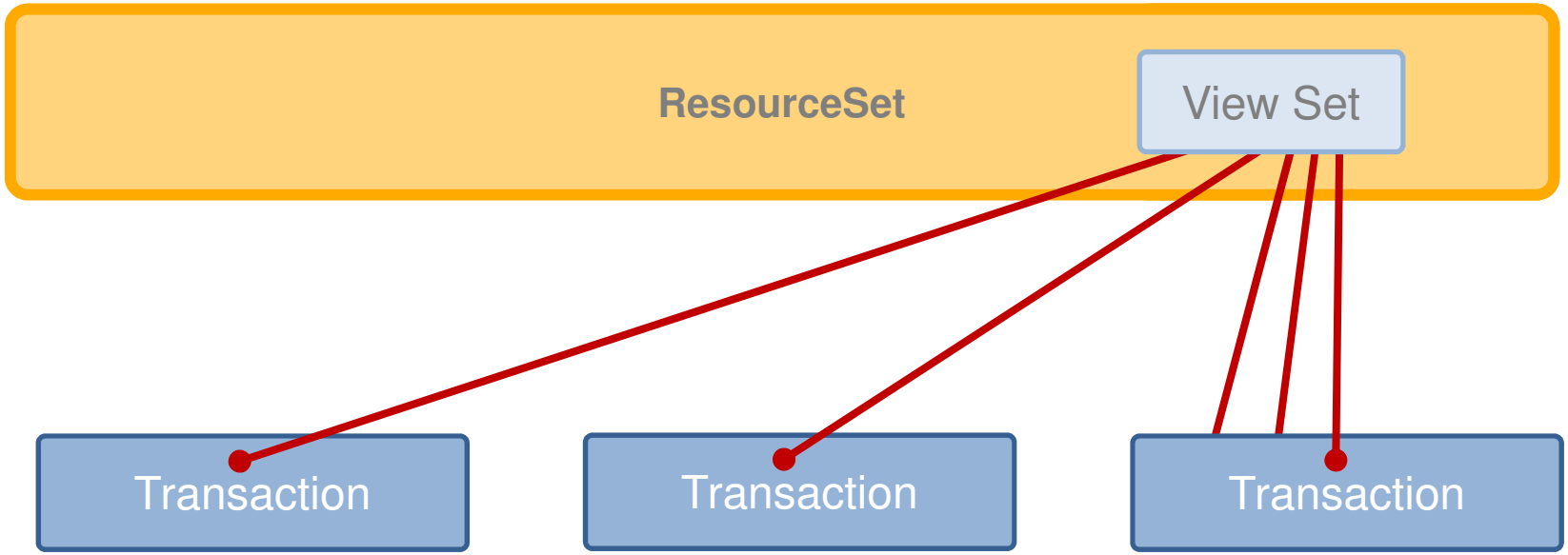


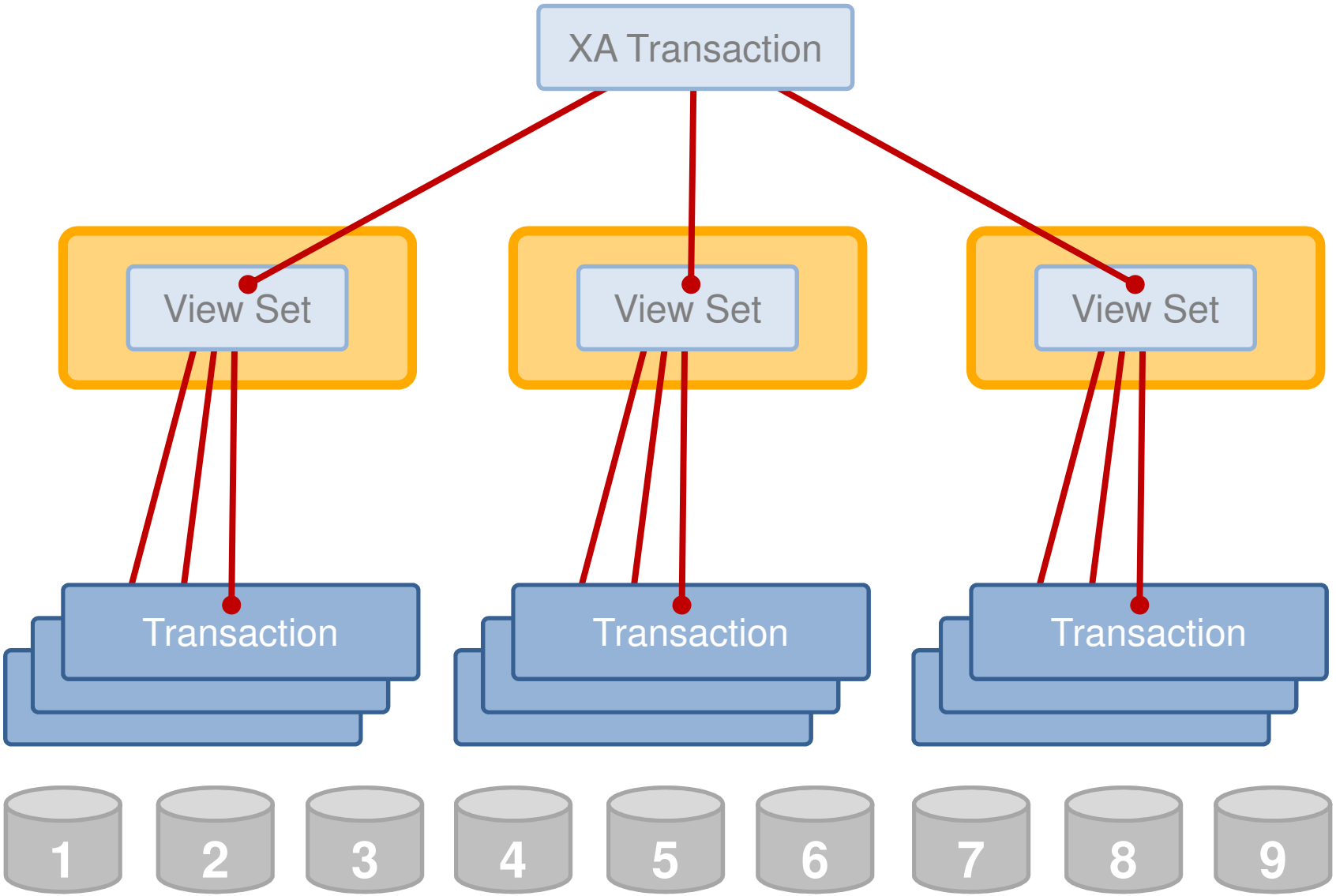












http://www.eclipse.org/cdo

NEXT STEPS

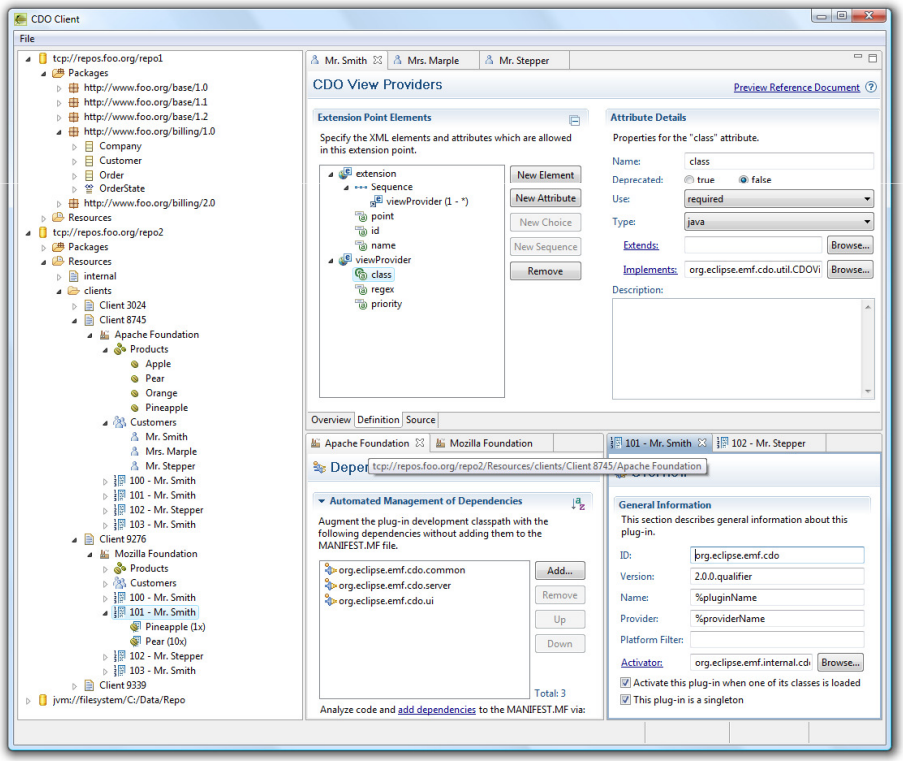
Offline mode
- Checkout
- Re-sync

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)

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



Opening Views, Audits and Transactions

```
// 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();
```