

Compass



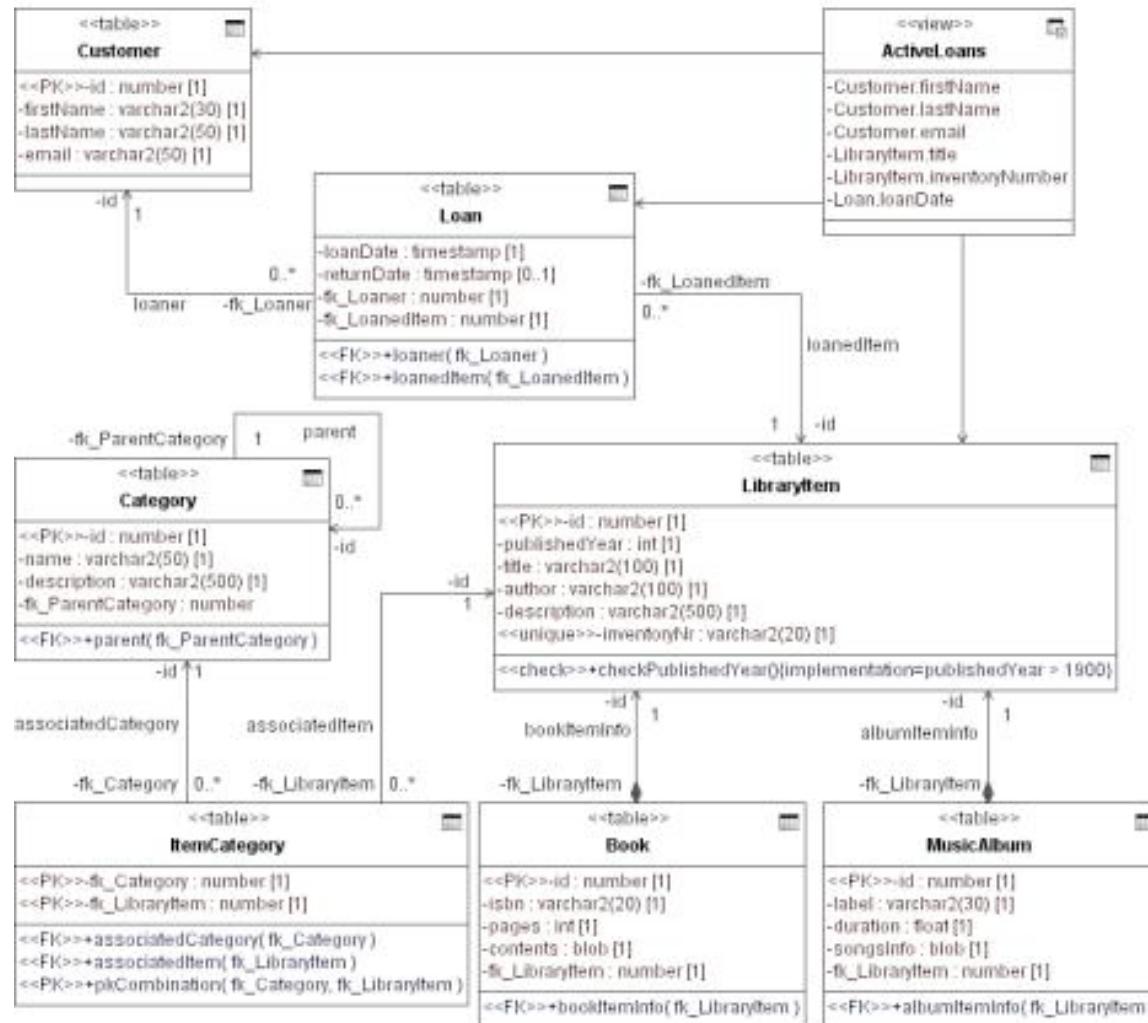
By : Shay Banon, Compass Founder



Introduction

Relational Model

- ▶ Search is more than “select ... like “%test%”



I want Google

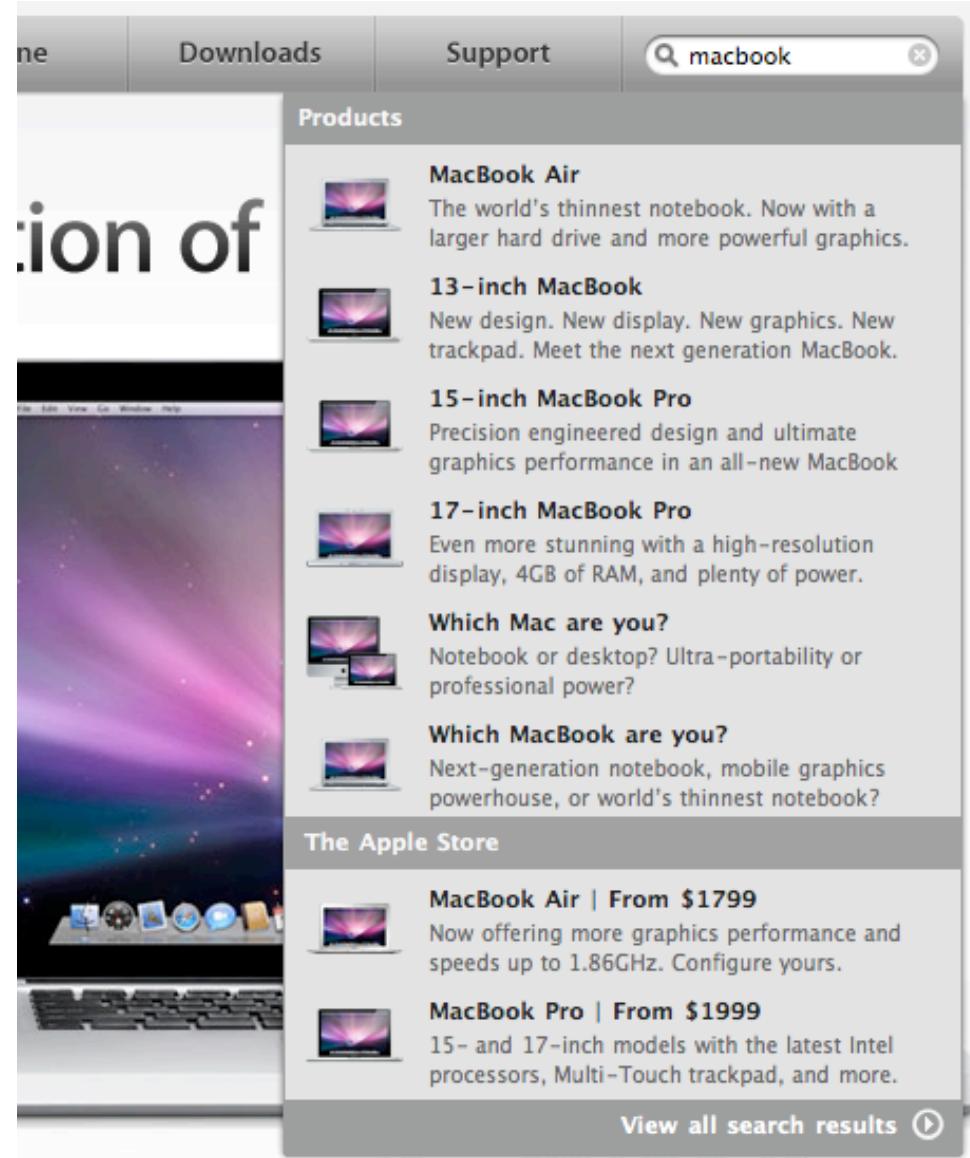
- ▶ Users want any application to be google like searchable



More than Google?

▶ We know more
than Google
when it comes to
our application.

▶ Shh, don't tell
Google



The screenshot shows a web browser window with the URL 'apple.com' in the address bar. A search bar at the top contains the text 'macbook'. Below the search bar, there are navigation links for 'Home', 'Downloads', 'Support', and a search icon. The main content area features a large image of a MacBook screen displaying a purple and pink nebula wallpaper. To the right of the image, the word 'Products' is displayed above a list of Mac models. The list includes:

- MacBook Air**: The world's thinnest notebook. Now with a larger hard drive and more powerful graphics.
- 13-inch MacBook**: New design. New display. New graphics. New trackpad. Meet the next generation MacBook.
- 15-inch MacBook Pro**: Precision engineered design and ultimate graphics performance in an all-new MacBook.
- 17-inch MacBook Pro**: Even more stunning with a high-resolution display, 4GB of RAM, and plenty of power.
- Which Mac are you?**: Notebook or desktop? Ultra-portability or professional power?
- Which MacBook are you?**: Next-generation notebook, mobile graphics powerhouse, or world's thinnest notebook?

Below this section, there is a heading 'The Apple Store' followed by two promotional boxes:

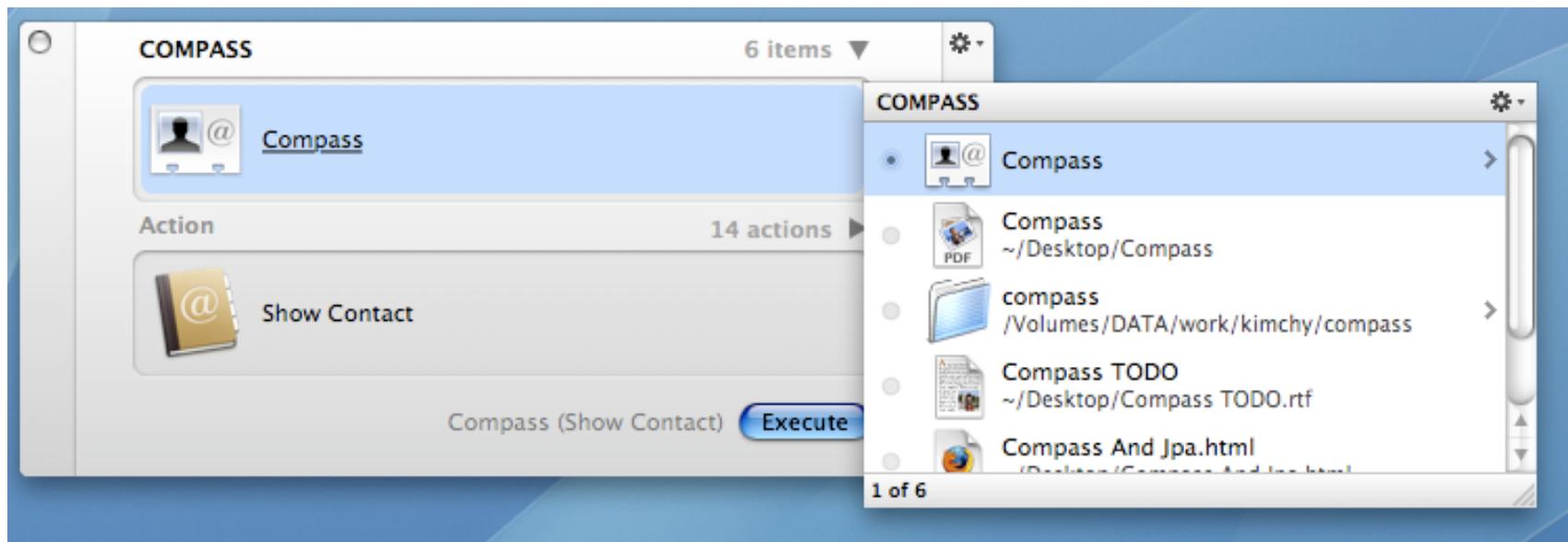
- MacBook Air | From \$1799**: Now offering more graphics performance and speeds up to 1.86GHz. Configure yours.
- MacBook Pro | From \$1999**: 15- and 17-inch models with the latest Intel processors, Multi-Touch trackpad, and more.

A 'View all search results' button is located at the bottom right of the search results area.

The new 15-inch MacBook Pro

Navigation Model

- ▶ Actions based on types
- ▶ Search and Rank based navigation



Integrate Search

- ▶ Used to be very complex
- ▶ http://www.kimchy.org/the_birth_of_compass/





- ▶ The ubiquitous IR library for search enabling application
- ▶ Very powerful
- ▶ Very low level
- ▶ Still hard to integrate





Enter Compass

Introducing Compass

- ▶ Simplify Search
- ▶ Open Source (Apache Licensed)
- ▶ 3.5 years old (started at version 0.4)
- ▶ 2.1 GA released 1-NOV-2008
- ▶ 2.2 M1 very close





Compass Core

RSEM – Basic

```
<resource alias="a">  
    <resource-id name="id"/>  
</resource>
```

```
Compass compass = getCompass();  
Resource resource = compass.getResourceFactory().createResource("a");  
resource.addProperty("id", 1);  
resource.addProperty("name", "some name");  
resource.addProperty("age", 45);  
session.save(resource);
```

RSEM – Converters

```
<resource alias="a">
    <resource-id name="id"/>
    <resource-property name="age" converter="int" format="000" />
</resource>
```

```
Compass compass = getCompass();
Resource resource = compass.getResourceFactory().createResource("a");
resource.addProperty("id", 1);
resource.addProperty("name", "some name");
resource.addProperty("age", 45);
session.save(resource);
```

RSEM – Index Settings

```
<resource alias="a">
  <resource-id name="id"/>
  <resource-property name="name" index="not_analyzed" store="compress" />
  <resource-property name="age" converter="int" format="000" />
</resource>
```

```
Compass compass = getCompass();
Resource resource = compass.getResourceFactory().createResource("a");
resource.addProperty("id", 1);
resource.addProperty("name", "some name");
resource.addProperty("age", 45);
session.save(resource);
```

RSEM – Search and Load

```
Resource resource = session.loadResource("a", 1);
System.out.println("Name is: " + resource.getObject("name"));
System.out.println("Age is: " + resource.getObject("age"));
```

```
CompassHits hits = session.find("name:some");
System.out.println("Resource with id: " + hits.resource(0).getId());
hits = session.find("some"); // all support
System.out.println("Resource with id: " + hits.resource(0).getId());
hits = session.find("a.name:some"); // dot path support
System.out.println("Resource with id: " + hits.resource(0).getId());
```

OSEM – Basic

```
@Searchable  
public class Author {  
  
    @SearchableId  
    int id;  
  
    @SearchableProperty  
    String name;
```

```
Author author = new Author();  
author.id = 1;  
author.name = "some name";  
session.save(author);
```



OSEM Basic – Search & Load

```
Author author = session.load(Author.class, 1);
System.out.println("Author name: " + author.name);
Resource resource = session.loadResource(Author.class, 1);
System.out.println("Author name: " + resource.getObject("name"));

CompassHits hits = session.find("name:some");
author = (Author) hits.data(0);
System.out.println("Author name: " + author.name);

hits = session.find("some"); // all support
hits = session.find("author.name:some"); // dot path support
```

OSEM – Component

```
@Searchable
public class Author {

    @SearchableId
    int id;

    @SearchableComponent
    Name name;

}

@Searchable(root = false)
public class Name {

    @SearchableProperty
    String firstName;

    @SearchableProperty
    String lastName;
}
```

```
Author author = new Author();
author.id = 1;
author.name = new Name("some", "name");
session.save(author);
```

OSEM Component – Search & Load

```
author = session.load(Author.class, 1);
System.out.println("Author first name: " + author.name.firstName);
Resource resource = session.loadResource(Author.class, 1);
System.out.println("Author first name: " + resource.getObject("firstName"));

hits = session.find("firstName:some");
hits = session.find("some"); // all support
hits = session.find("author.name.firstName:some"); // dot path support
```

OSEM Basic – Searchable Property

```
@Searchable
public class Author {

    @SearchableId
    int id;

    @SearchableProperty(format = "000")
    int age;

    @SearchableProperty(index = Index.NOT_ANALYZED)
    String isbn;

    @SearchableComponent
    Name name;

}
```

OSEM – Collections

```
@Searchable
public class Author {

    @SearchableId
    int id;

    @SearchableProperty
    List<String> booksIsbns;

    @SearchableComponent
    List<Name> names;
}
```



More Mappings

- ▶ More RSEM options
- ▶ More OSEM options
- ▶ XSEM – XML to Search Engine Mapping
- ▶ JSEM – JSON to Search Engine Mapping



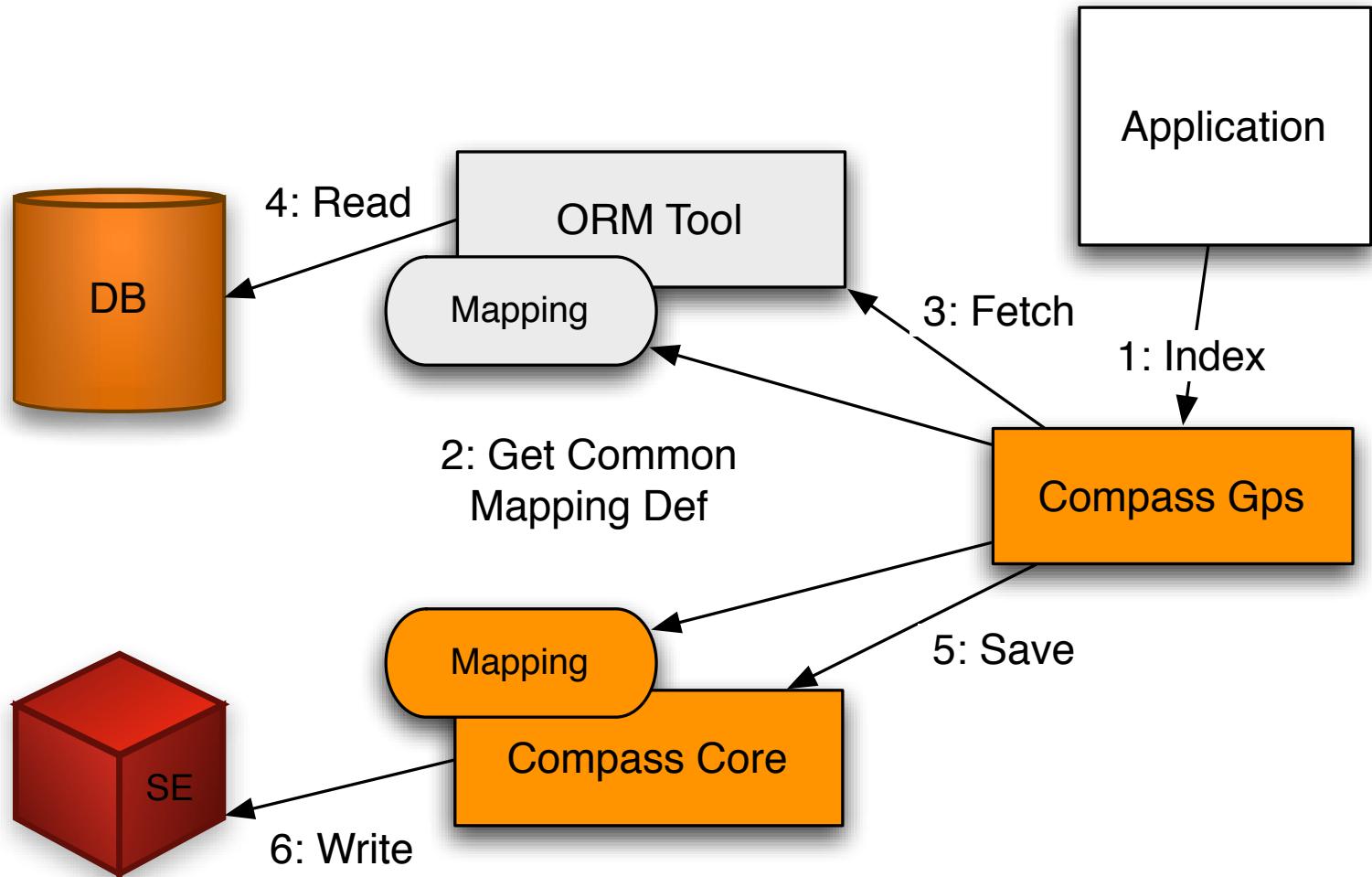


Compass Gps

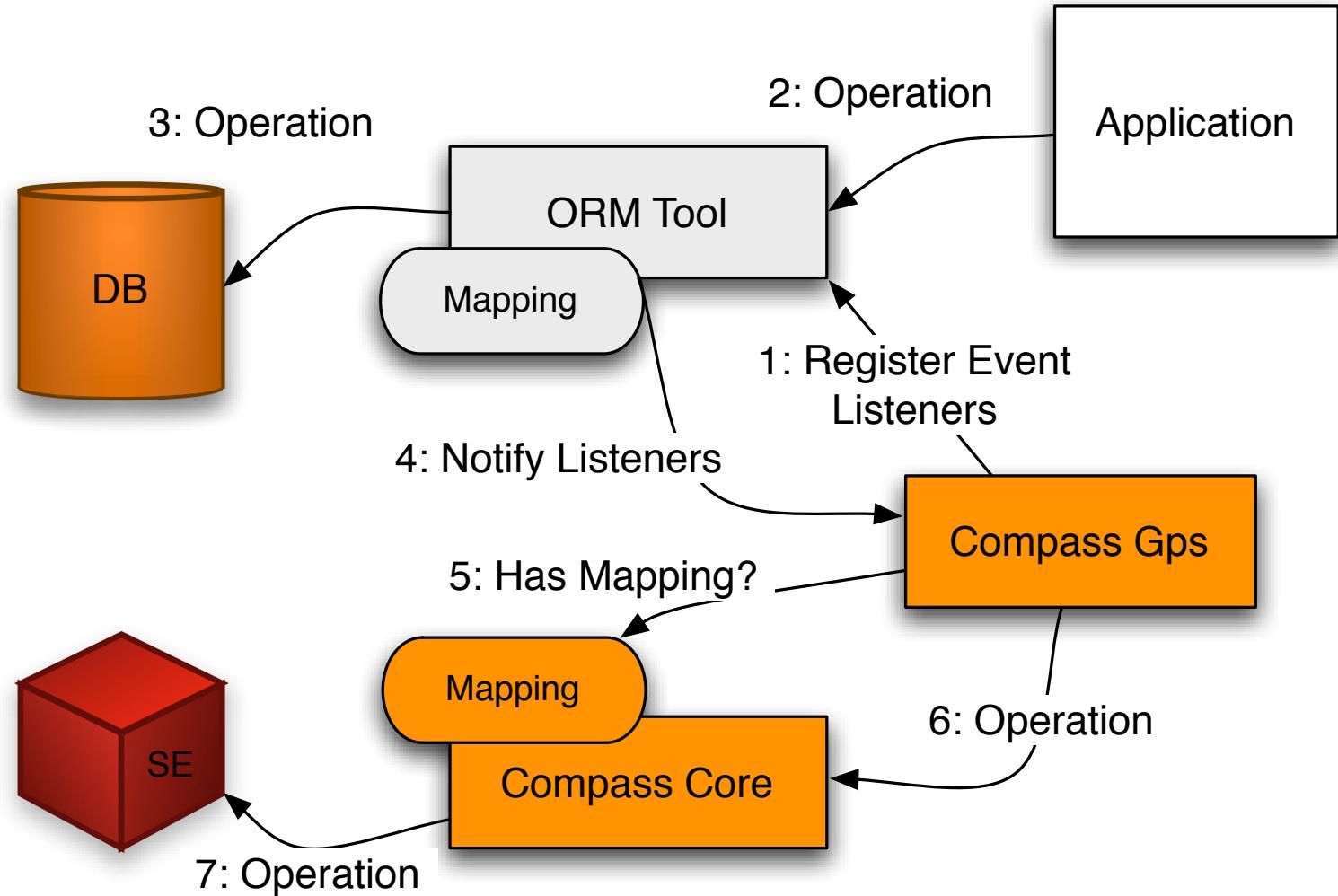
Overview

- ▶ Integration Module with other “data driven” frameworks
- ▶ Mainly integration with ORMs
 - » Embedded
 - » Through JPA
 - » External

Compass Gps Index



Compass Gps Mirror



Embedded Hibernate

```
<hibernate-configuration>
    <session-factory>

        <property name="hibernate.dialect">org.hibernate.dialect.HSQLDialect</property>
        <property name="hibernate.connection.driver_class">org.hsqldb.jdbcDriver</property>
        <property name="hibernate.connection.url">jdbc:hsqldb:mem:test</property>
        <property name="hibernate.connection.username">sa</property>
        <property name="hibernate.connection.password"></property>

        <property name="compass.engine.connection">target/test-index</property>

        <mapping resource="org/compass/gps/device/hibernate/simple/Simple.hbm.xml"/>
        <mapping resource="org/compass/gps/device/hibernate/simple/SimpleEx.hbm.xml"/>

        <event type="post-update">
            <listener class="org.compass.gps.device.hibernate.embedded.CompassEventListener"/>
        </event>
        <event type="post-insert">
            <listener class="org.compass.gps.device.hibernate.embedded.CompassEventListener"/>
        </event>
        <event type="post-delete">
            <listener class="org.compass.gps.device.hibernate.embedded.CompassEventListener"/>
        </event>
```

Embedded Hibernate

```
SessionFactory sessionFactory = null;
Session hibernateSession = null;

Compass compass = HibernateHelper.getCompass(sessionFactory);
CompassSession session = compass.openSession();
// work with the session, mainly do search

// use CompassGps to (re)index the whole data
CompassGps compassGps = HibernateHelper.getCompassGps(sessionFactory);
compassGps.index();
```

Embedded JPA – EclipseLink

```
<persistence-unit name="embeddeclipselink" transaction-type="RESOURCE_LOCAL">
    <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
    <properties>
        <!-- Other properties -->
        <property name="eclipselink.session.customizer"
            value="org.compass.gps.device.jpa.embedded.eclipselink.CompassSessionCust
                <property name="compass.engine.connection" value="target/test-index"/>
        </properties>
    </persistence-unit>
```

Embedded JPA – Hibernate

```
<persistence-unit name="embeddedhibernate" transaction-type="RESOURCE_LOCAL">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <properties>
        <!-- Other properties -->
        <property name="compass.engine.connection" value="target/test-index"/>
    </properties>
</persistence-unit>
```

Embedded JPA – OpenJPA

```
<persistence-unit name="embeddedopenjpa" transaction-type="RESOURCE_LOCAL">
    <provider>org.apache.openjpa.persistence.PersistenceProviderImpl</provider>
    <properties>
        <!-- Other properties -->

        <property name="compass.engine.connection" value="target/test-index"/>
    </properties>
</persistence-unit>
```

Embedded JPA – TopLink

```
<persistence-unit name="embeddedtoplink" transaction-type="RESOURCE_LOCAL">
    <provider>oracle.toplink.essentials.PersistenceProvider</provider>
    <properties>
        <!-- Other properties -->

        <property name="toplink.session.customizer"
            value="org.compass.gps.device.jpa.embedded.toplink.CompassSessionCustomizer" />

        <property name="compass.engine.connection" value="target/test-index"/>
    </properties>
</persistence-unit>
```

Embedded JPA – One API

```
EntityManagerFactory entityManagerFactory = null;

GenericJpaHelper jpaHelper = new GenericJpaHelper(entityManagerFactory);
Compass compass = jpaHelper.getCompass();
CompassSession session = compass.openSession();
// work with the session, mainly do search

// use CompassGps to (re)index the whole data
JpaCompassGps compassGps = jpaHelper.getJpaCompassGps();
compassGps.index();
```

External JPA

```
Compass compass = new CompassConfiguration()
    .setConnection("target/test-index")
    .addScan("com.mycompany")
    .buildCompass();

JpaCompassGps compassGps = new DefaultJpaCompassGps(compass);
// automatic detection of Hibernate / TopLink / EclipseLink / OpenJPA
compassGps.addGpsDevice(new JpaGpsDevice("jpa", entityManagerFactory));
compassGps.start();

// we can now index
compassGps.index();
```



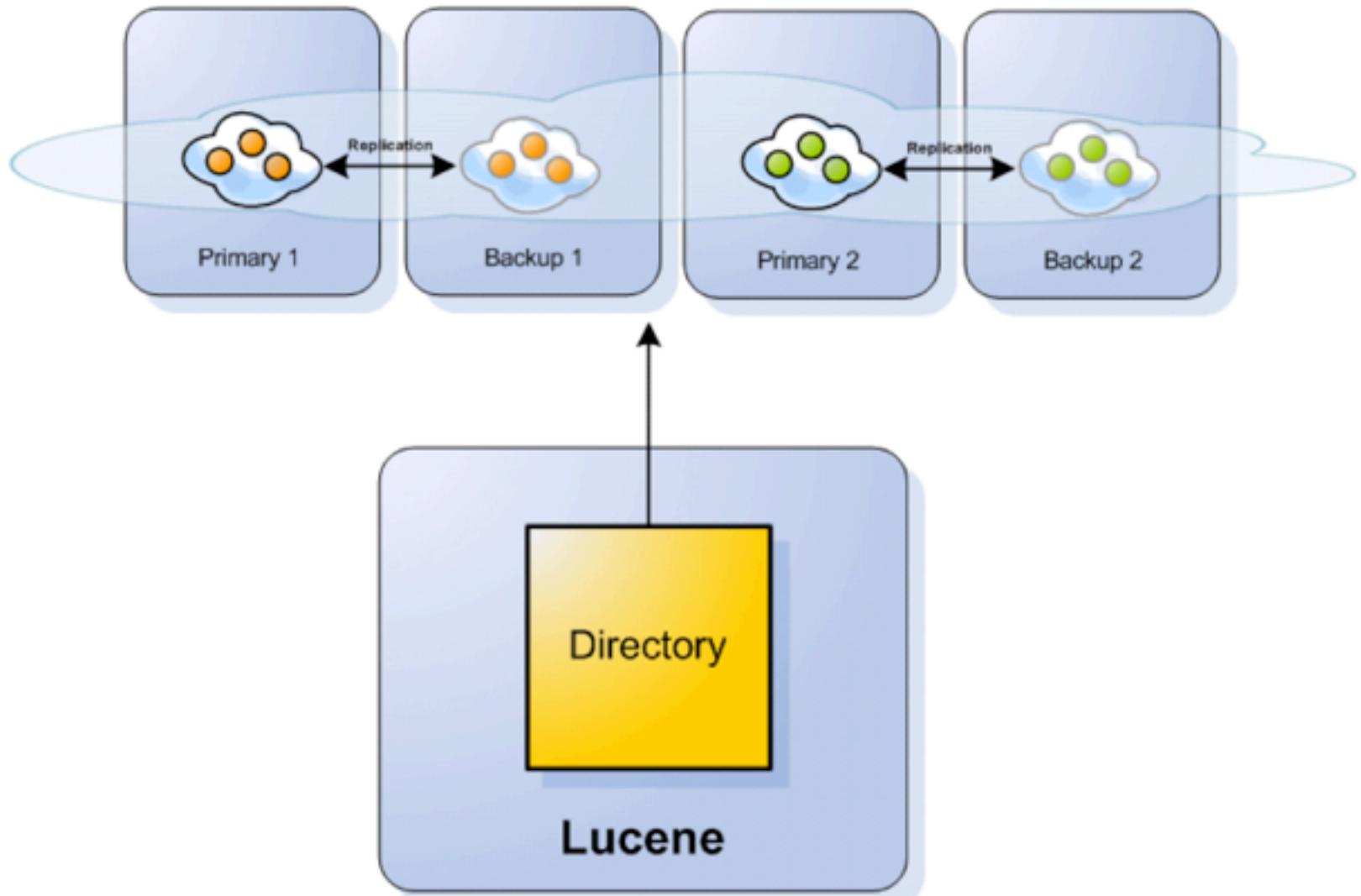
Compass Needle

Overview

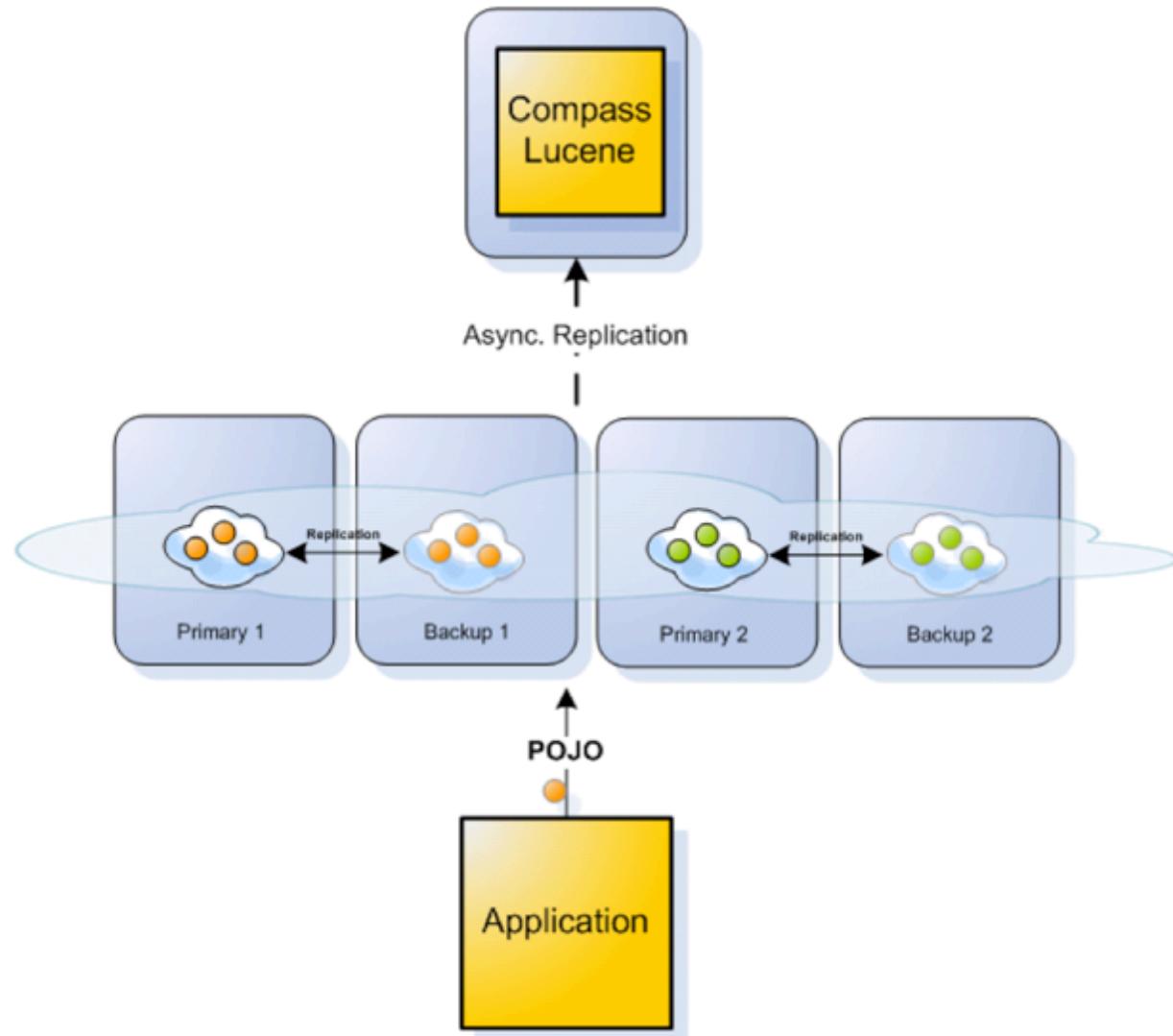
- ▶ Integration with distributed products / libraries.
- ▶ Store the index on a data grid: GigaSpaces, Terracotta, Coherence.
- ▶ Automatically index data grid content such as GigaSpaces and Coherence.



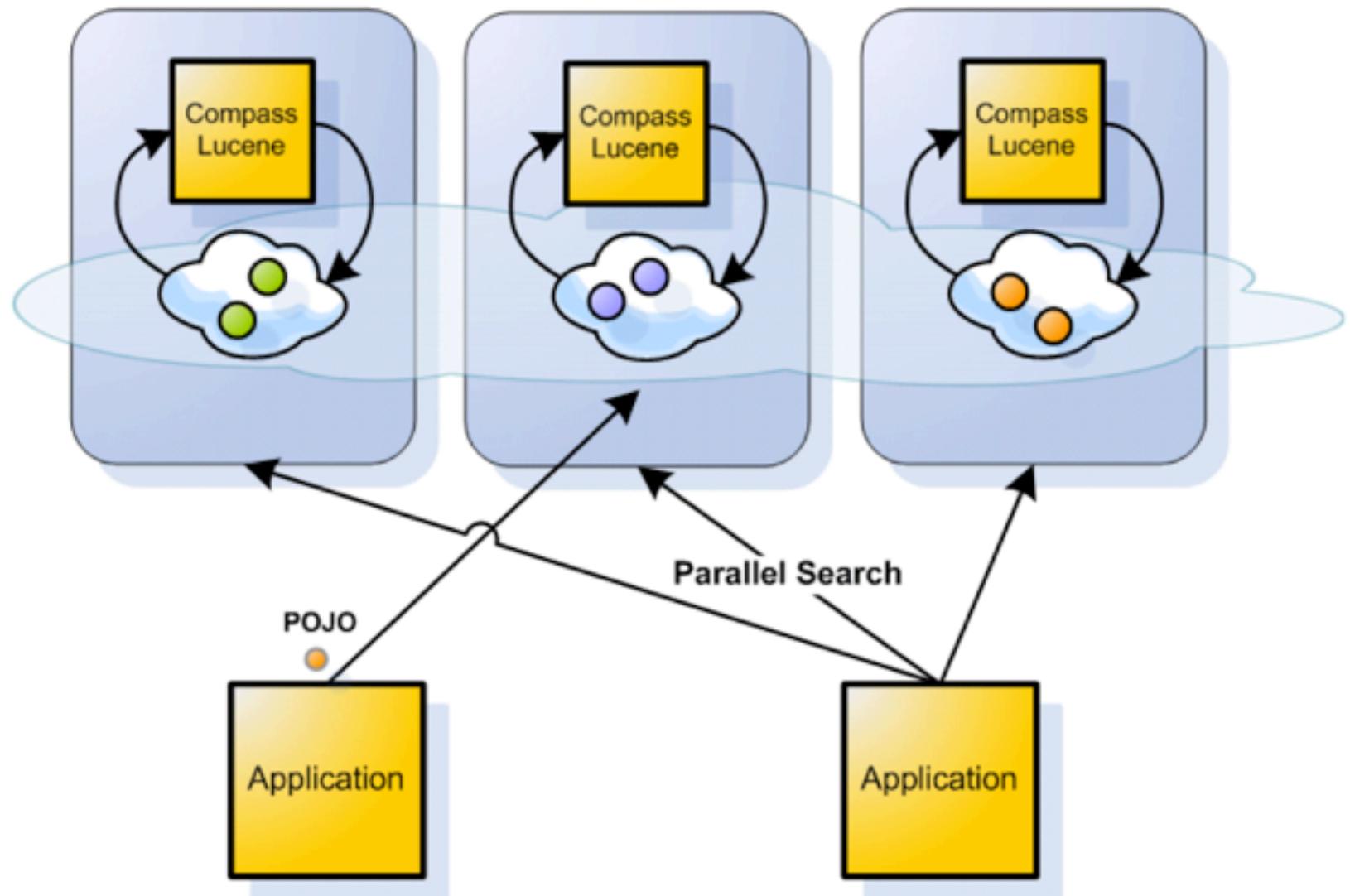
Index Stored in a Data Grid



Index the Data Grid



Collocated Indexing and Search





Compass Future

Open Discussion