



Free your code.

JBoss
Community

The background of the slide is a rich, deep red color with a subtle texture, resembling heavy velvet or silk curtains. The curtains are gathered at the top and bottom, creating a classic stage-like appearance. The lighting is slightly darker at the top and bottom edges, making the central area where the text is located stand out.

Hibernate Search

Hardy Ferentschik, Red Hat



The toolbox



The toolbox

Build tool	Ant/Maven
------------	-----------



The toolbox

Build tool	Ant/Maven
Container	Tomcat/JBoss



The toolbox

Build tool	Ant/Maven
Container	Tomcat/JBoss
MVC	Struts/Seam



The toolbox

Build tool	Ant/Maven
Container	Tomcat/JBoss
MVC	Struts/Seam
Domain model	JPA/Hibernate



The toolbox

Build tool	Ant/Maven
Container	Tomcat/JBoss
MVC	Struts/Seam
Domain model	JPA/Hibernate
Search	?

“LIKE” queries

```
title = (title == null) ? "%" : "%" + title.toLowerCase() + "%";
actor = (actor == null) ? "%" : "%" + actor.toLowerCase() + "%";

em.createQuery(
    "select distinct p from Product p JOIN p.actors a " +
    "where lower(p.title) like :title " +
    "and lower(a.name) LIKE :actor order by p.title")
    .setParameter("title", title)
    .setParameter("actor", actor));
```

SQL shortcomings

SQL shortcomings

Wildcard / word search	%hibernate%
---------------------------	-------------

SQL shortcomings

Wildcard / word search	%hibernate%
Approximation	hybernate

SQL shortcomings

Wildcard / word search	%hibernate%
Approximation	hybernate
Proximity	'Java' close to 'Persistence'

SQL shortcomings

Wildcard / word search	%hibernate%
Approximation	hybernate
Proximity	'Java' close to 'Persistence'
Result scoring	

SQL shortcomings

Wildcard / word search	%hibernate%
Approximation	hybernate
Proximity	'java' close to 'Persistence'
Result scoring	
Multi-"column" search	

Lucene

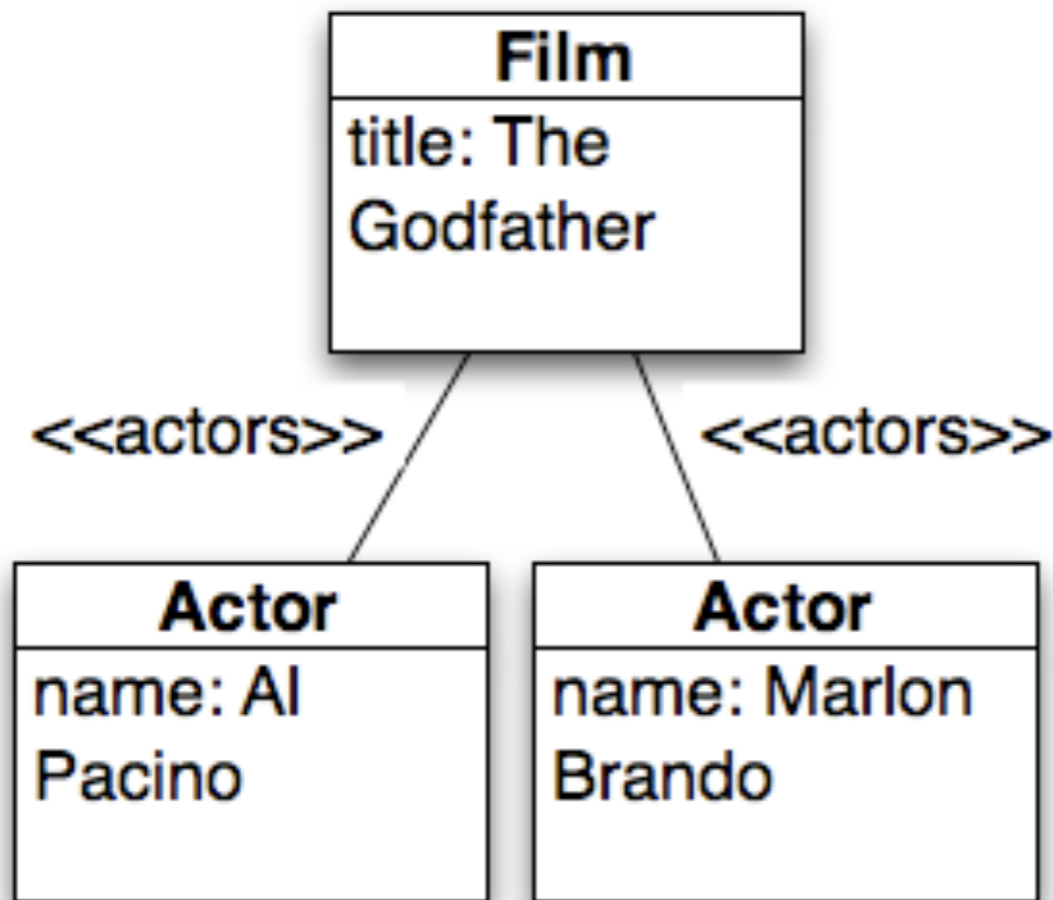
- ❑ Powerful fulltext search engine
- ❑ Open Source
- ❑ In the TOP 10 of downloaded Apache projects

Lucene DIY

- ❑ Structural mismatch
- ❑ Retrieval mismatch
- ❑ Synchronization mismatch

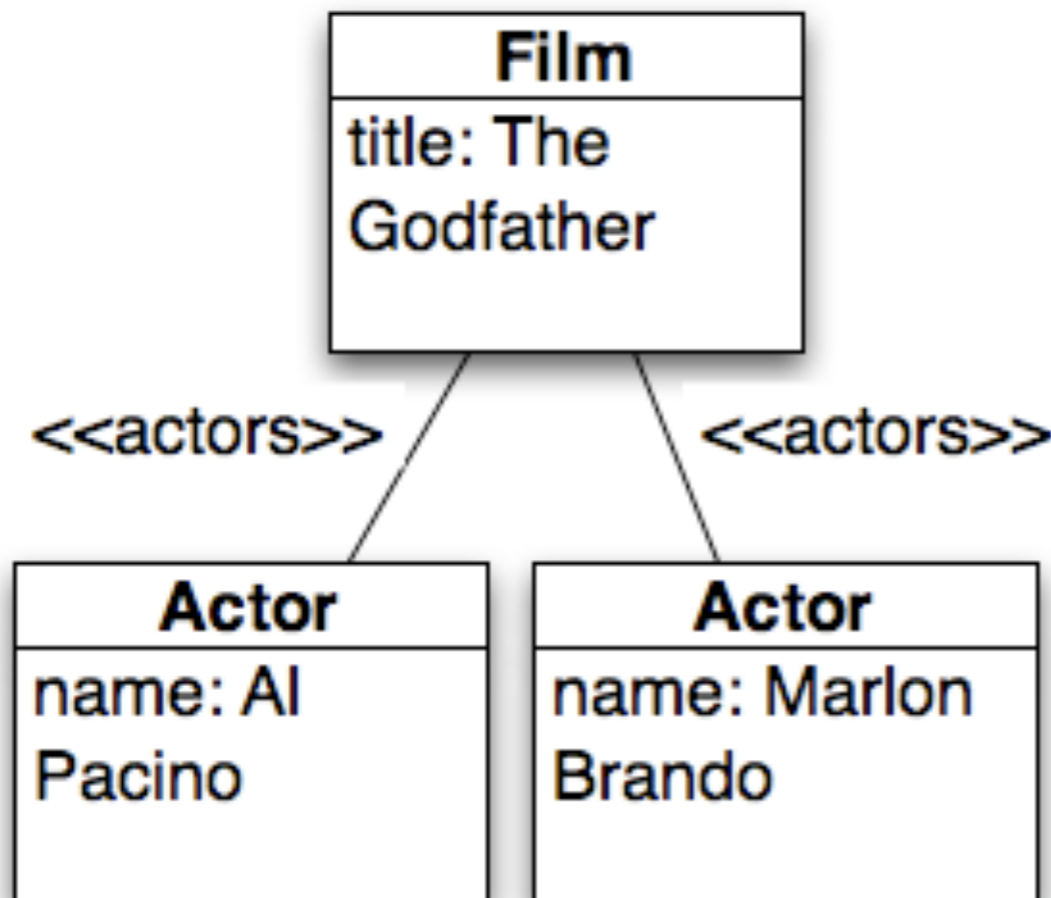
Structural Mismatch

Object world



Structural Mismatch

Object world



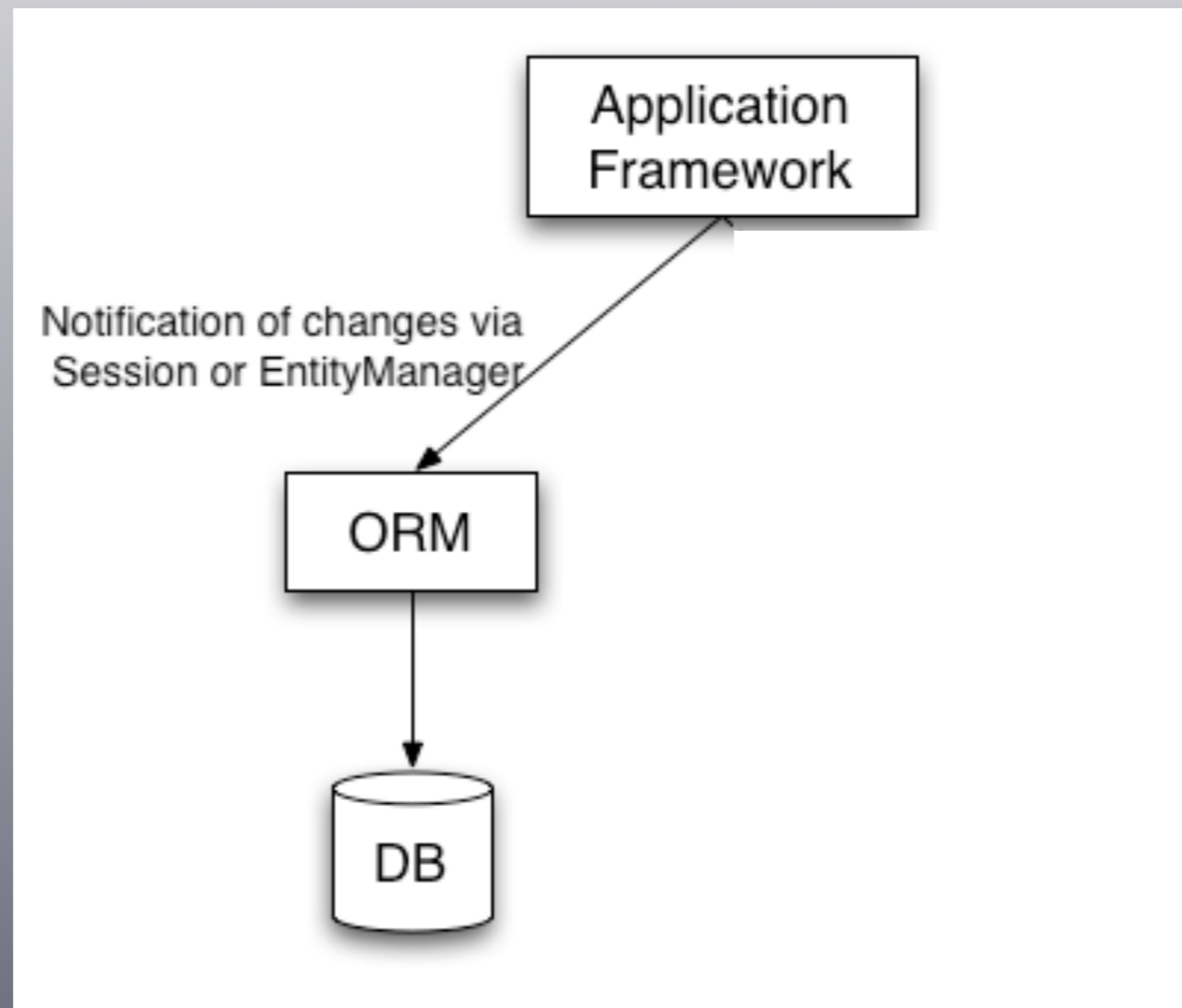
Index world

Film Document	
title	godfather
film.actor	al pacino
film.actor	marlon brando

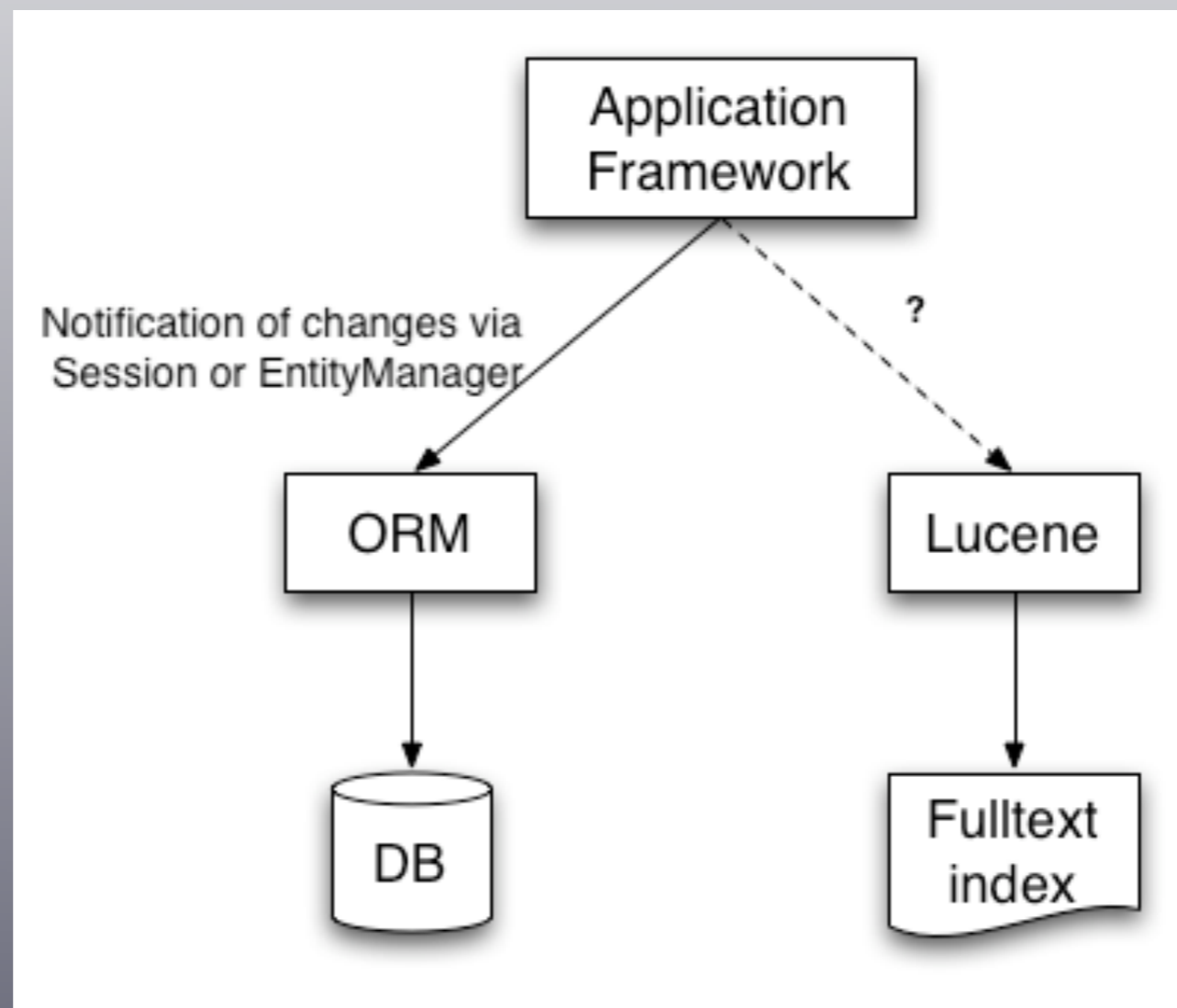
Retrieval Mismatch

- Index contains Documents not Objects
- Even if you re-hydrate you don't have managed objects

Synchronization mismatch



Synchronization mismatch





 HIBERNATE

Success

Configure

- Enable Search via event listeners
- Add Backend options

Configure

- Enable Search via event listeners
- Add Backend options

```
<property name="hibernate.search.default.indexBase"  
value="/var/lucene/indexes" />
```

Annotate

```
@Entity
public class Essay {
    ...
    @Id
    public Long getId() { return id; }

    public String getSummary() { return summary; }

    @Lob
    public String getText() { return text; }

    @ManyToOne
    public Author getAuthor() { return author; }
}
```

Annotate

```
@Entity @Indexed
public class Essay {
    ...
    @Id
    public Long getId() { return id; }

    public String getSummary() { return summary; }

    @Lob
    public String getText() { return text; }

    @ManyToOne
    public Author getAuthor() { return author; }
}
```

Annotate

```
@Entity @Indexed
public class Essay {
    ...
    @Id
    public Long getId() { return id; }

    @Field
    public String getSummary() { return summary; }

    @Lob
    public String getText() { return text; }

    @ManyToOne
    public Author getAuthor() { return author; }
}
```

Annotate

```
@Entity @Indexed
public class Essay {
    ...
    @Id
    public Long getId() { return id; }

    @Field
    public String getSummary() { return summary; }

    @Lob @Field
    public String getText() { return text; }

    @ManyToOne
    public Author getAuthor() { return author; }
}
```

Annotate

```
@Entity @Indexed
public class Essay {
    ...
    @Id
    public Long getId() { return id; }

    @Field
    public String getSummary() { return summary; }

    @Lob @Field
    public String getText() { return text; }

    @ManyToOne @IndexedEmbedded
    public Author getAuthor() { return author; }
}
```

Search

```
String searchQuery = "Hibernate Search";
String[] productFields = {"summary", "author.name"};

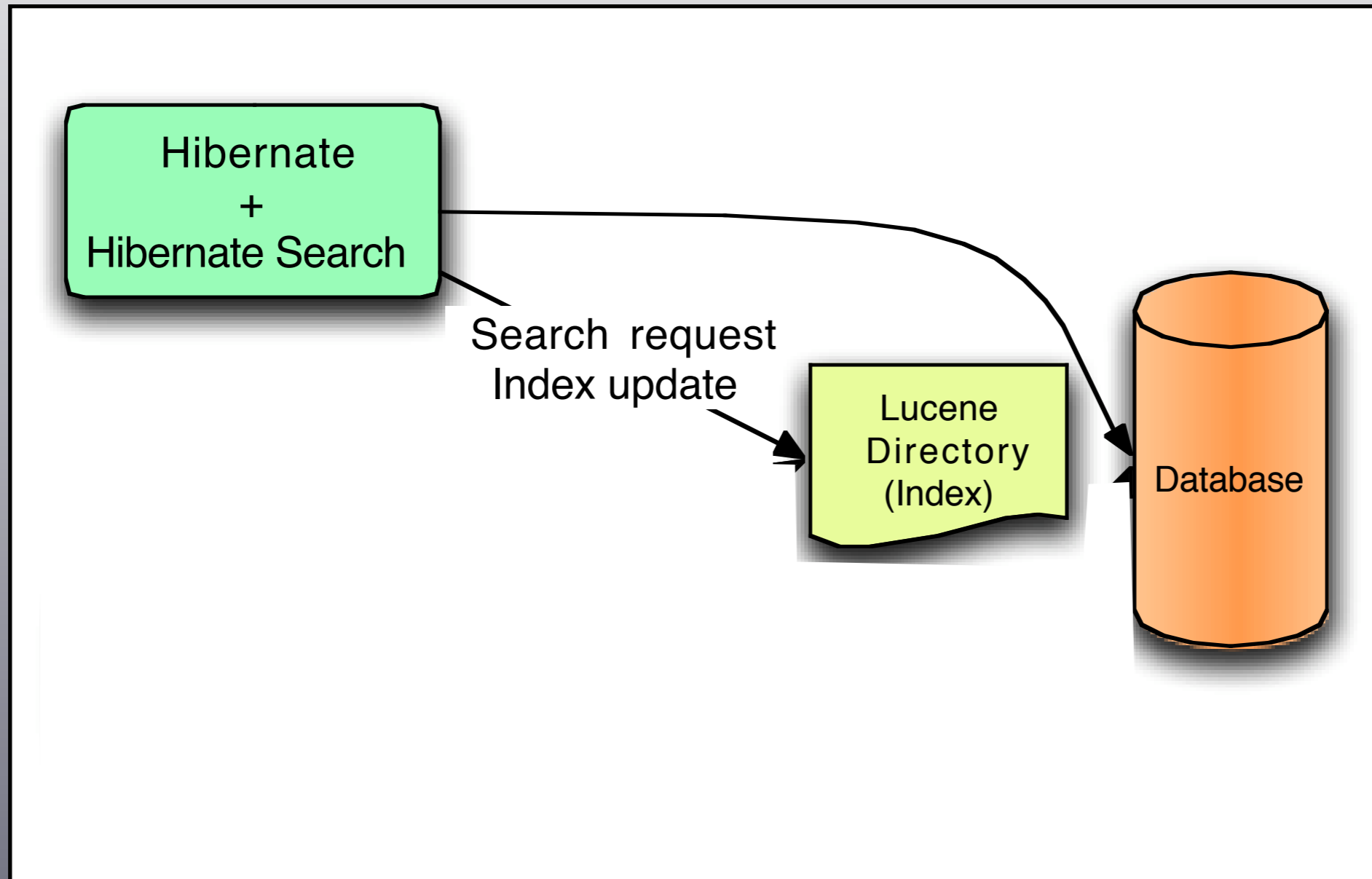
// Lucene
QueryParser parser = new MultiFieldQueryParser(productFields,
                                                new StandardAnalyzer());
Query luceneQuery = parser.parse(searchQuery);

// Hibernate Search
FullTextEntityManager ftEm = Search.
    getFullTextEntityManager((EntityManager)em);

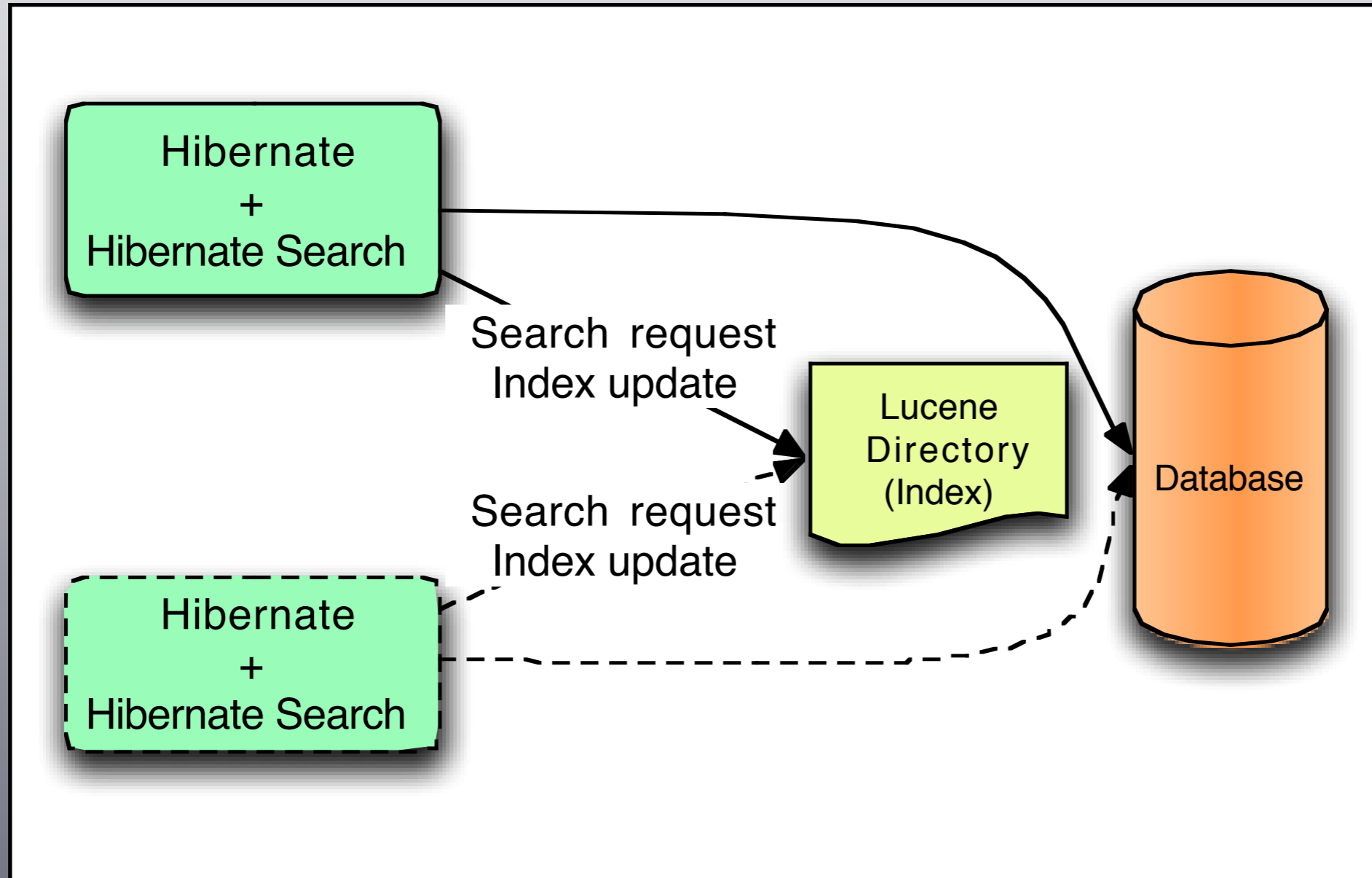
FullTextQuery query =
    ftEm.createFullTextQuery( luceneQuery, Essay.class );

List<Essay> items = query.getResultList();
```

Architecture



Architecture



A background of red curtains with a central opening. The curtains are gathered at the top and bottom, creating a frame for the text.

The goodies

Analyzers

- Take text as an input, **chunk** it into individual words and optionally applying a chain of **filter operations** on the tokens.



Welcome to JA00 2009



Document	
title	

to JA00 2009



Document	
title	welcom

JAOO 2009



Document	
title	welcom

2009



Document	
title	welcom jaoo



2009



Document	
title	welcom jaoo java



Document	
title	welcom jaoo java 2009



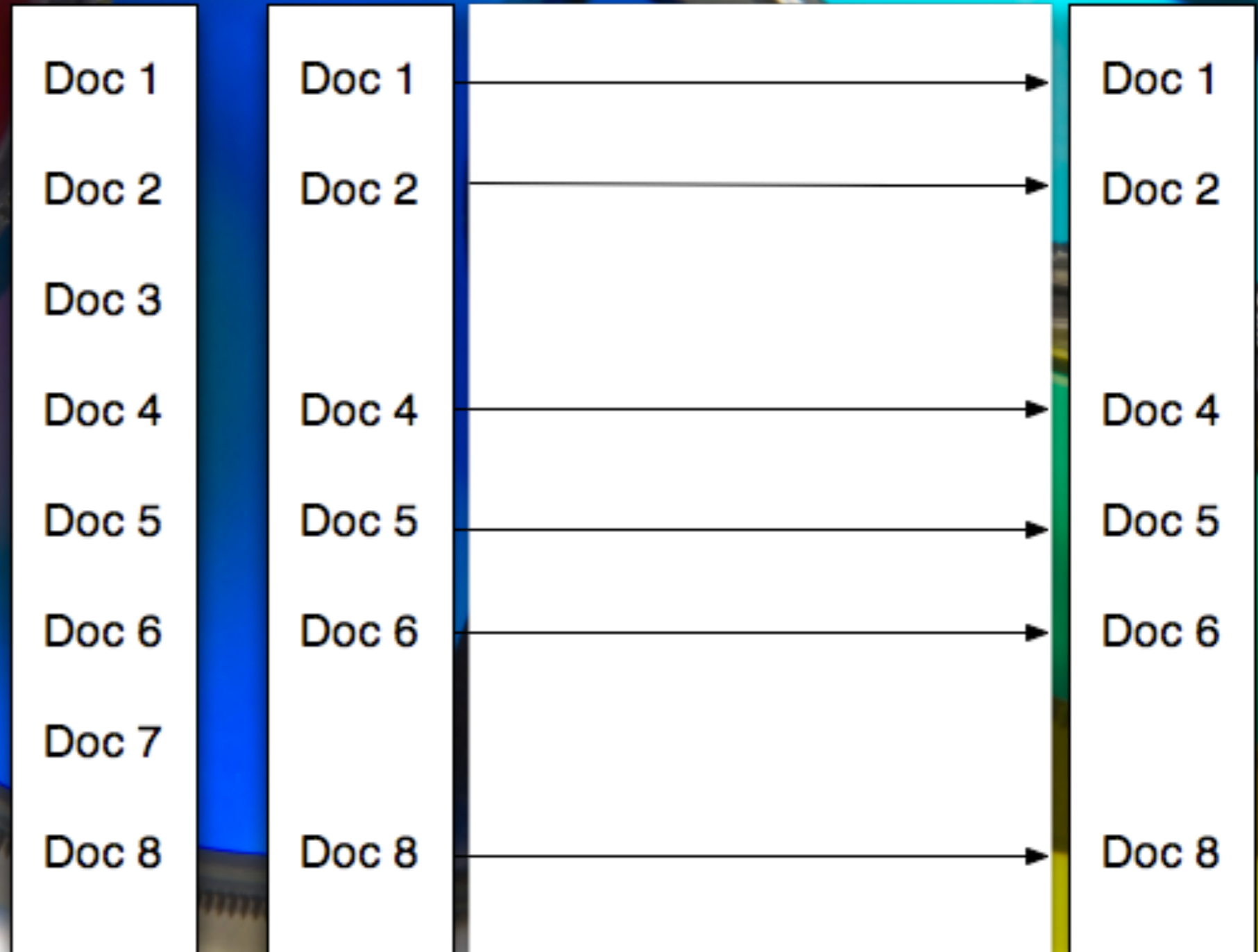
Analyzer Example

```
@Entity
@Indexed
@AnalyzerDef(name = "customanalyzer",
  tokenizer = @TokenizerDef(factory=StandardTokenizerFactory.class),
  filters = {
    @TokenFilterDef(factory = LowerCaseFilterFactory.class),
    @TokenFilterDef(factory = SnowballPorterFilterFactory.class,
      params = {@Parameter(name = "language", value = "English")})
  })
public class Book {
  ...
  @Field(index=Index.TOKENIZED, store=Store.NO)
  @Analyzer(definition = "customanalyzer")
  private String title;
  ...
}
```

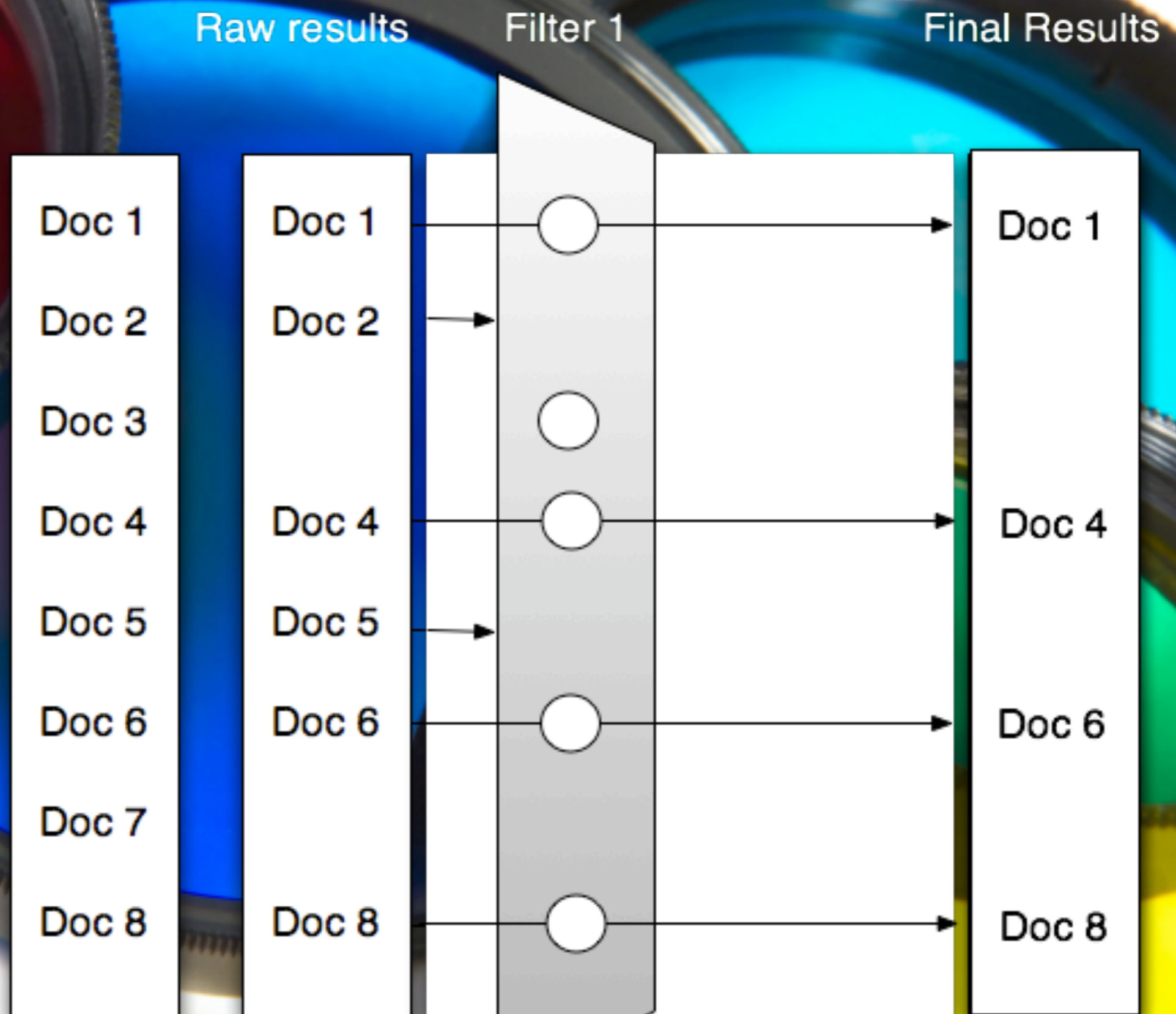
Filters

Raw results

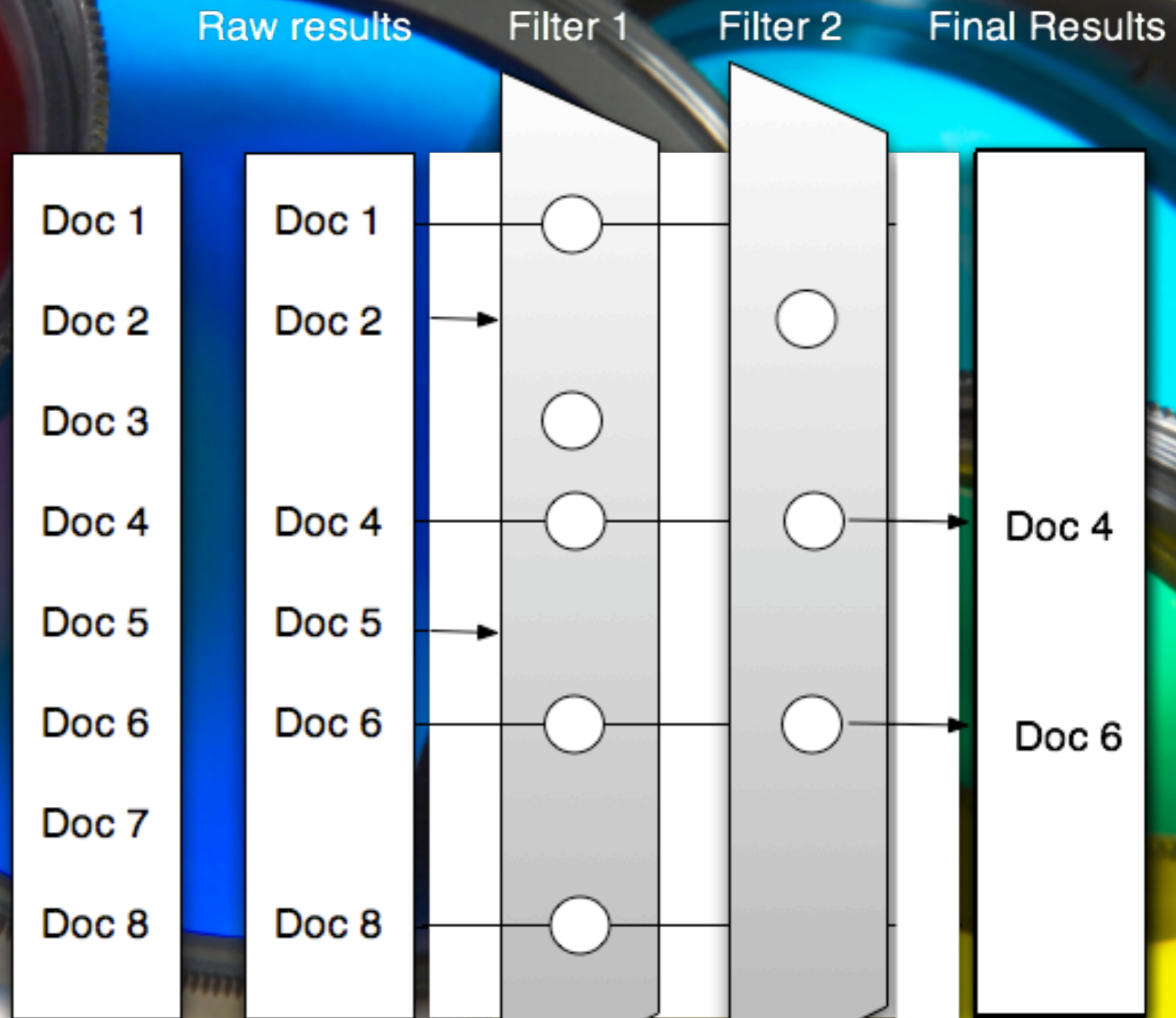
Final Results



Filters



Filters



Filter Example

```
@Entity
@Indexed
@FullTextFilterDefs ( {
    @FullTextFilterDef (name="bestDriver",
                        impl=BestDriversFilter.class) ,
    @FullTextFilterDef (name="security",
                        impl=SecurityFilterFactory.class)
})
public class Driver { ... }
```

```
...
fullTextQuery = s.createFullTextQuery( query, Driver.class );
fullTextQuery.enableFullTextFilter("bestDriver");
fullTextQuery.enableFullTextFilter("security").
    setParameter( "login", "andre" );
fullTextQuery.list();
...
```

Filter Example

```
@Entity
@Indexed
@FullTextFilterDefs ( {
    @FullTextFilterDef (name="bestDriver",
                        impl=BestDriversFilter.class) ,
    @FullTextFilterDef (name="security",
                        impl=SecurityFilterFactory.class)
})
public class Driver { ... }
```

```
...
fullTextQuery = s.createFullTextQuery( query, Driver.class );
fullTextQuery.enableFullTextFilter("bestDriver"); ←
fullTextQuery.enableFullTextFilter("security").
    setParameter( "login", "andre" );
fullTextQuery.list();
...
```

Filter Example

```
@Entity
@Indexed
@FullTextFilterDefs ( {
    @FullTextFilterDef (name="bestDriver",
                        impl=BestDriversFilter.class) ,
    @FullTextFilterDef (name="security",
                        impl=SecurityFilterFactory.class)
})
public class Driver { ... }
```

```
...
fullTextQuery = s.createFullTextQuery( query, Driver.class );
fullTextQuery.enableFullTextFilter("bestDriver"); ←
fullTextQuery.enableFullTextFilter("security"). ←
    setParameter( "login", "andre" );
fullTextQuery.list();
...
```


Projection

- ❑ Projection on metadata (SCORE, BOOST, ID, ...)
- ❑ No DB access

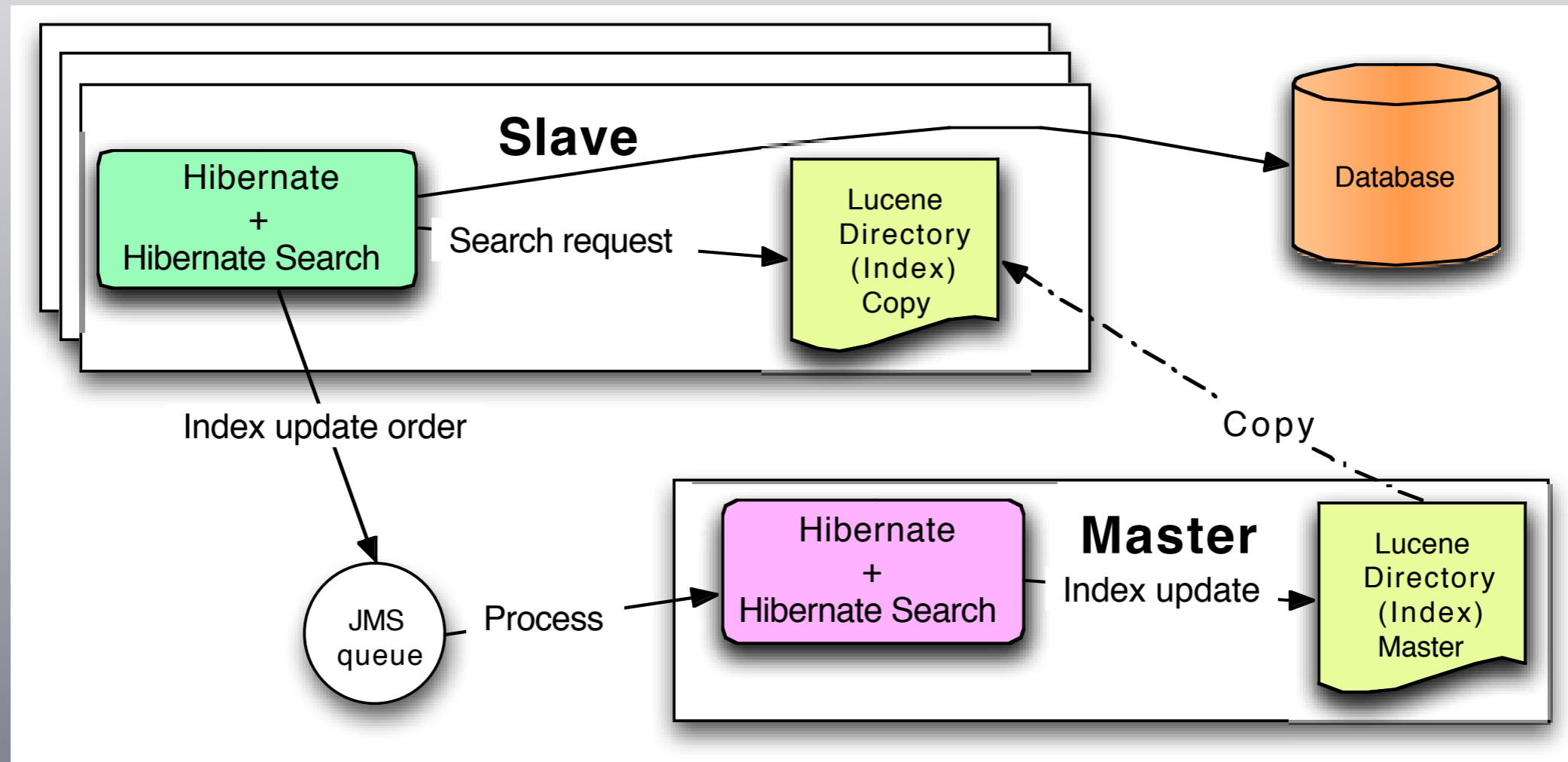
Projection Example

```
FullTextQuery query = s.createFullTextQuery(luceneQuery, Book.class );
query.setProjection( "id", "summary", "body", "mainAuthor.name" );

List results = query.list();
Object[] firstResult = (Object[]) results.get(0);

Integer id = firstResult[0];
String summary = firstResult[1];
String body = firstResult[2];
String authorName = firstResult[3];
```

Clustering



And even more

- ❑ Index sharding
- ❑ Automatic index optimization
- ❑ Manual indexing and purging
- ❑ Shared Lucene resources
- ❑ Access to native Lucene

Hibernate Search

Fulltext search without
the hassle!

A background of rich red, vertically pleated curtains. The curtains are pulled back at the top corners, creating a dark, shadowed opening in the center. The lighting is soft, highlighting the texture and folds of the fabric.

Q&A

More Info

- Hibernate Search
 - <http://search.hibernate.org>
 - Hibernate Search in Action
- Apache Lucene
 - <http://lucene.apache.org>
 - Lucene In Action
- <http://in.relation.to>
- <http://forum.hibernate.org/viewforum.php?f=9>
- hardy.ferentschik@redhat.com

