
Free your code.
JBoss Community
Designed by the JBoss Community Visual Design team... for everyone.

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
Business logic	JPA/Hibernate

The toolbox

Build tool	Ant/Maven
Container	Tomcat/JBoss
MVC	Struts/Seam
Business logic	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	hybernat

SQL shortcomings

Wildcard / word search	%hibernate%
Approximation	hybernat
Proximity	‘Java’ close to ‘Persistence’

SQL shortcomings

Wildcard / word search	%hibernate%
Approximation	hybernat
Proximity	‘Java’ close to ‘Persistence’
Result scoring	

SQL shortcomings

Wildcard / word search	%hibernate%
Approximation	hybernat
Proximity	‘Java’ close to ‘Persistence’
Result scoring	
Multi-”column” search	

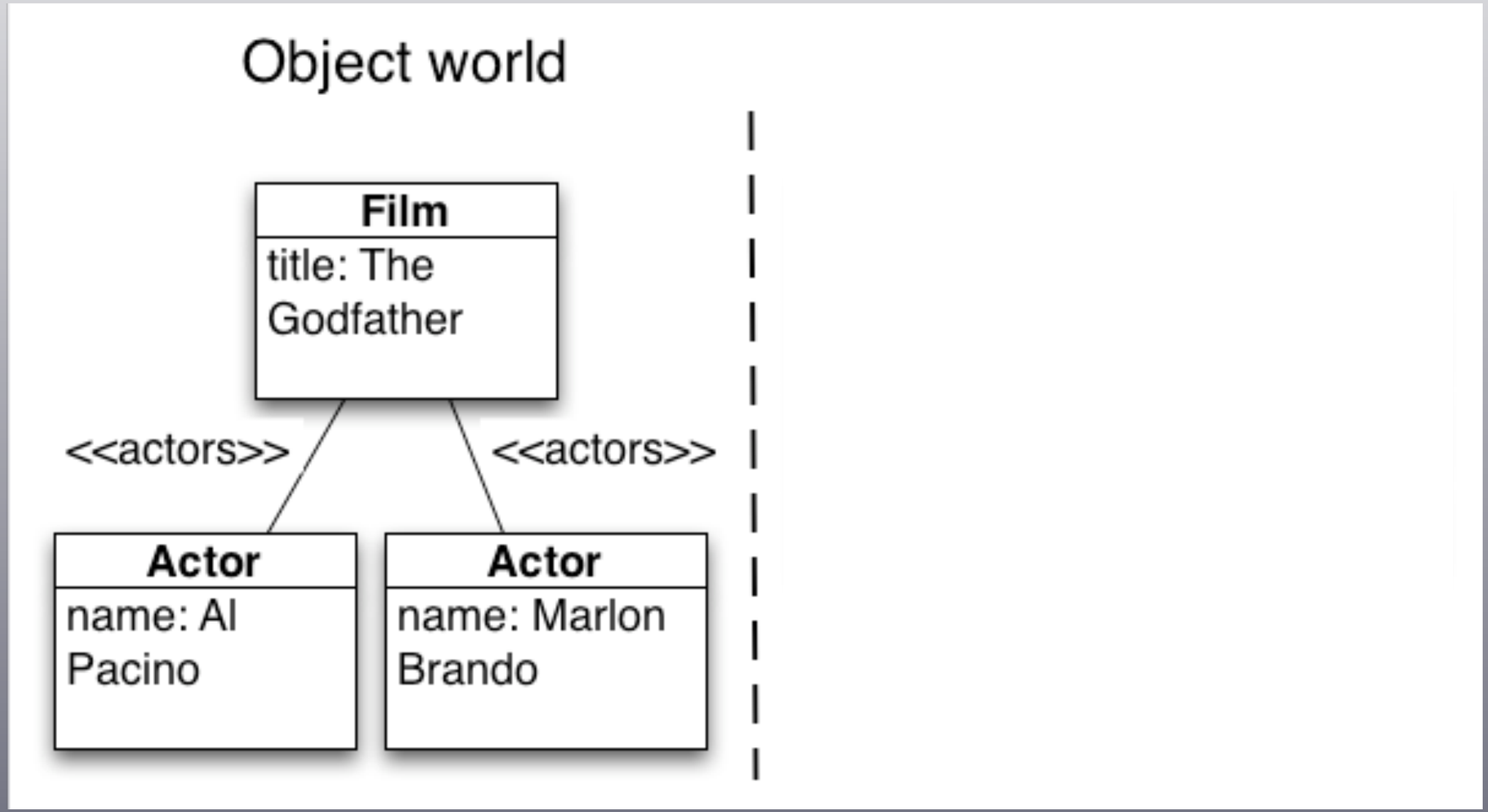
Lucene

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

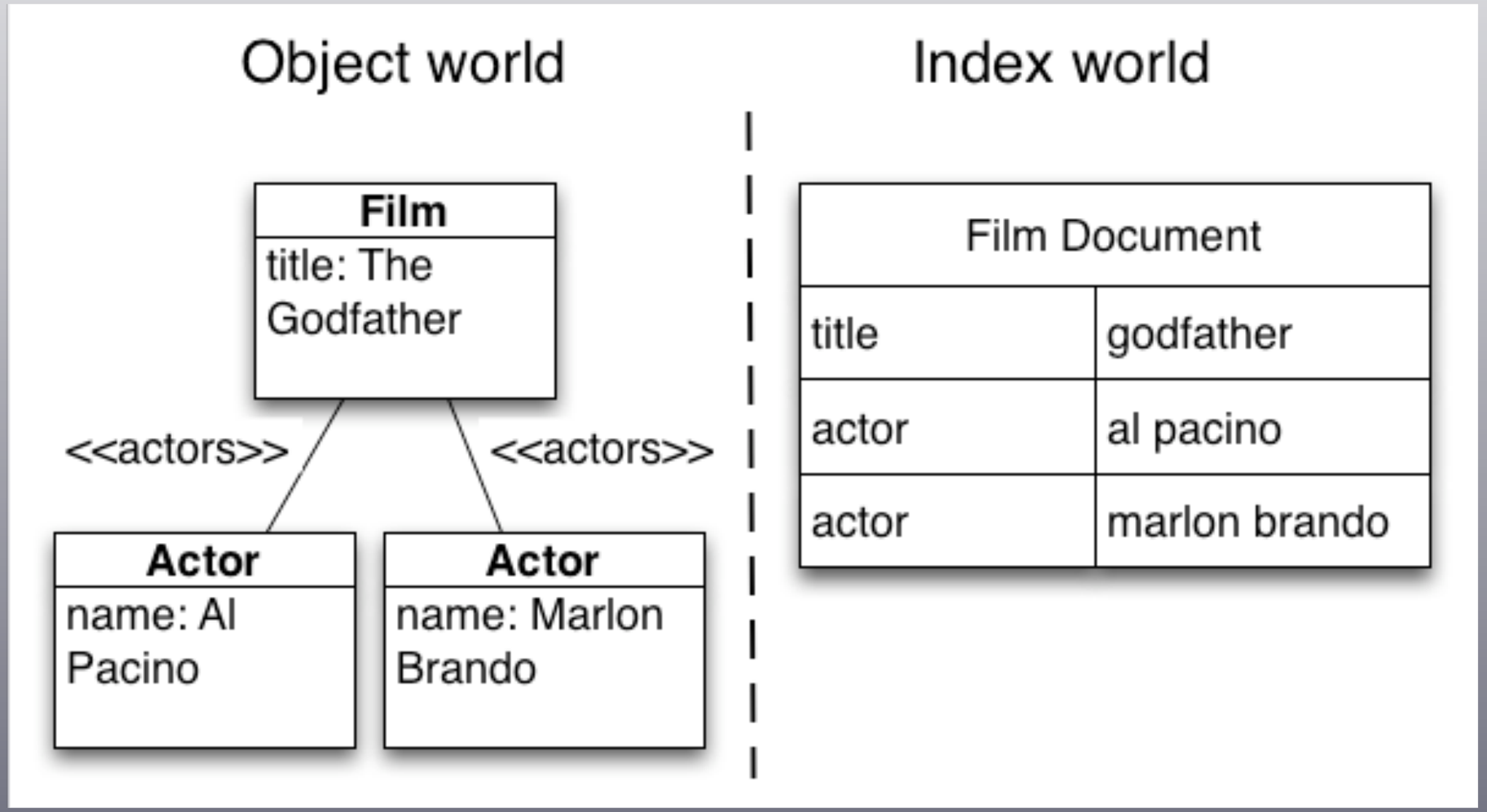
Lucene DIY

- Synchronization mismatch
- Structural mismatch
- Retrieval mismatch

Structural Mismatch



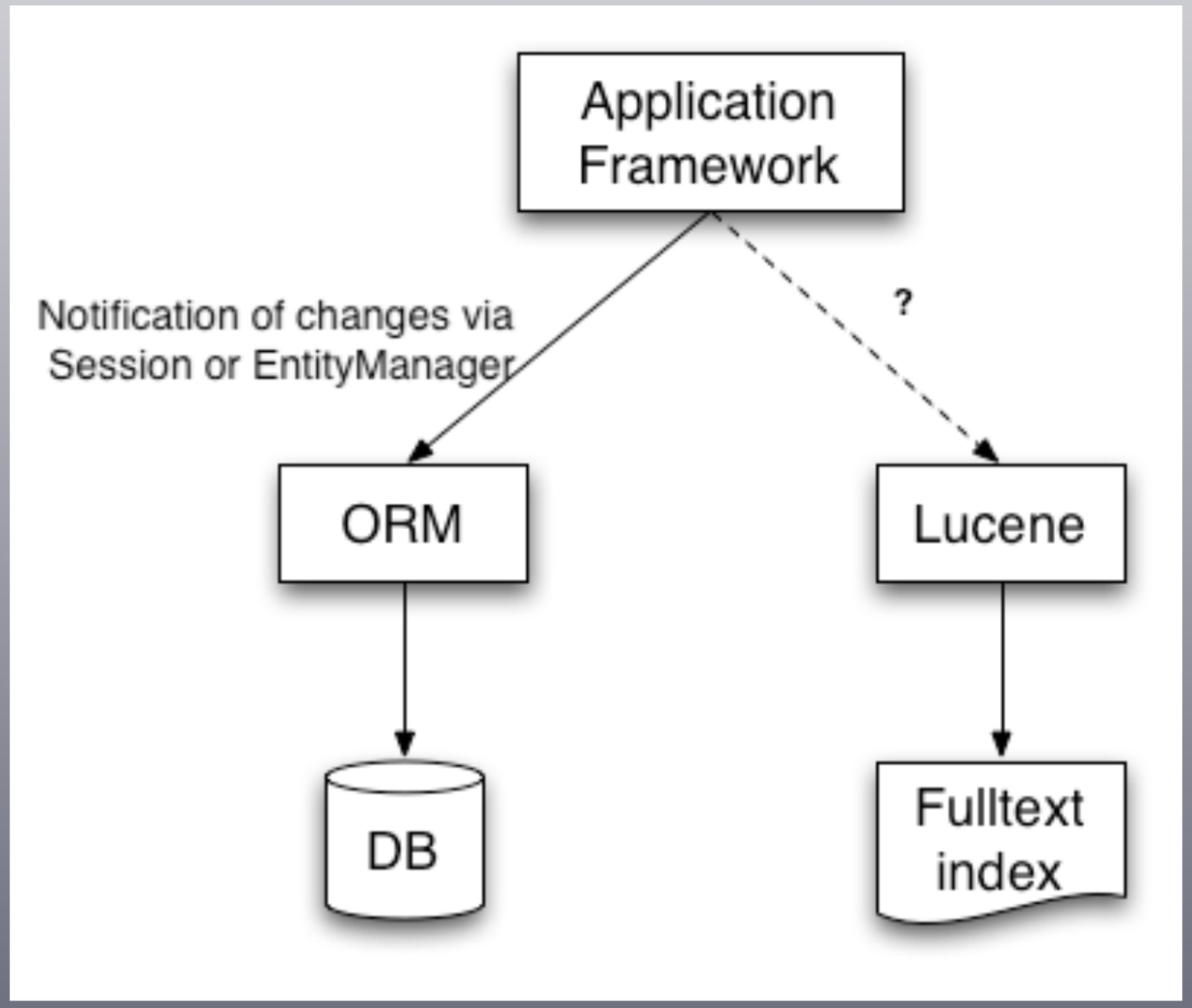
Structural Mismatch



Retrieval Mismatch

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

Synchronization mismatch



HIBERNATE

Lucene

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; }
}
```

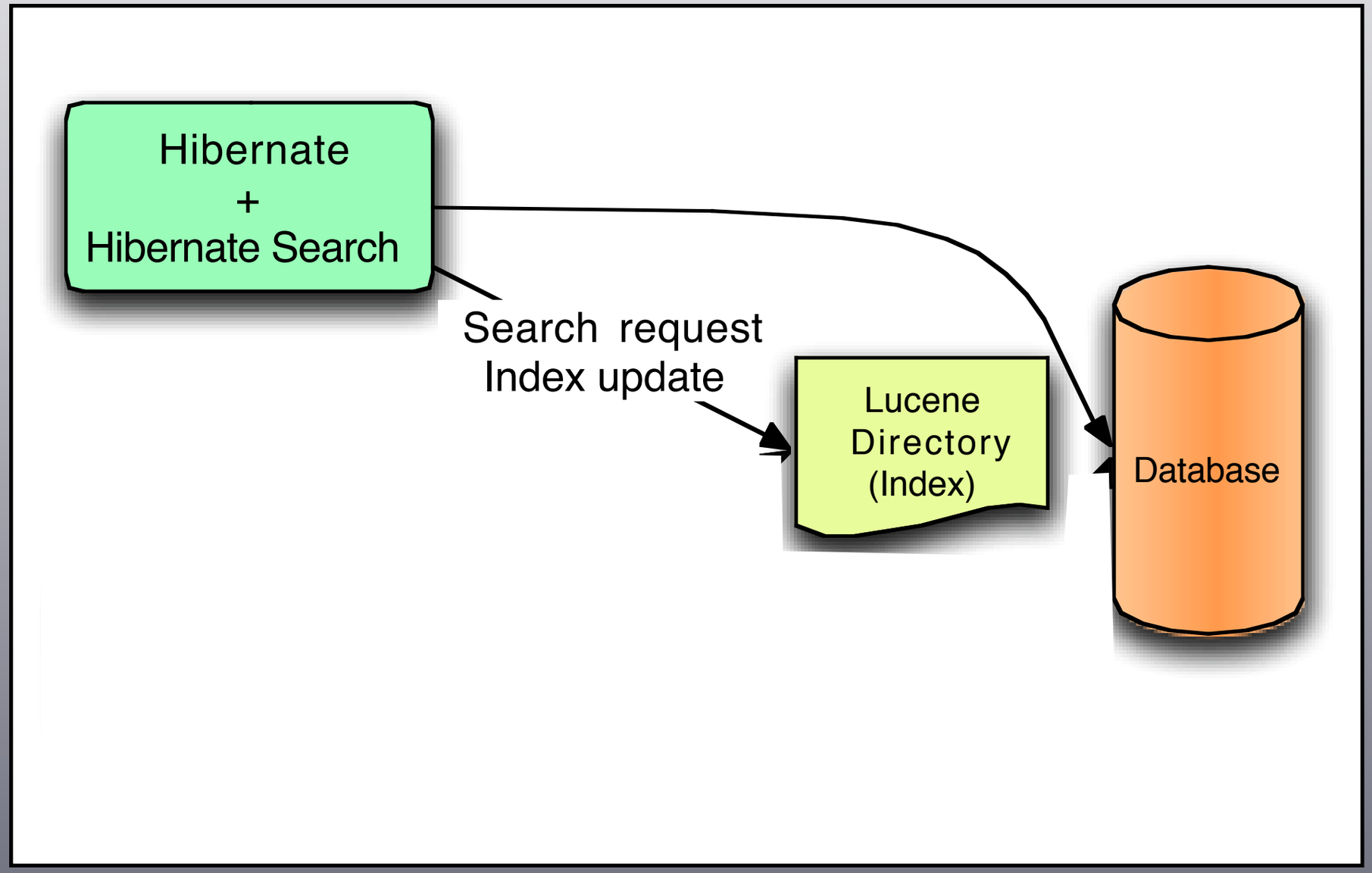
Annotate

```
@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; }
}
```

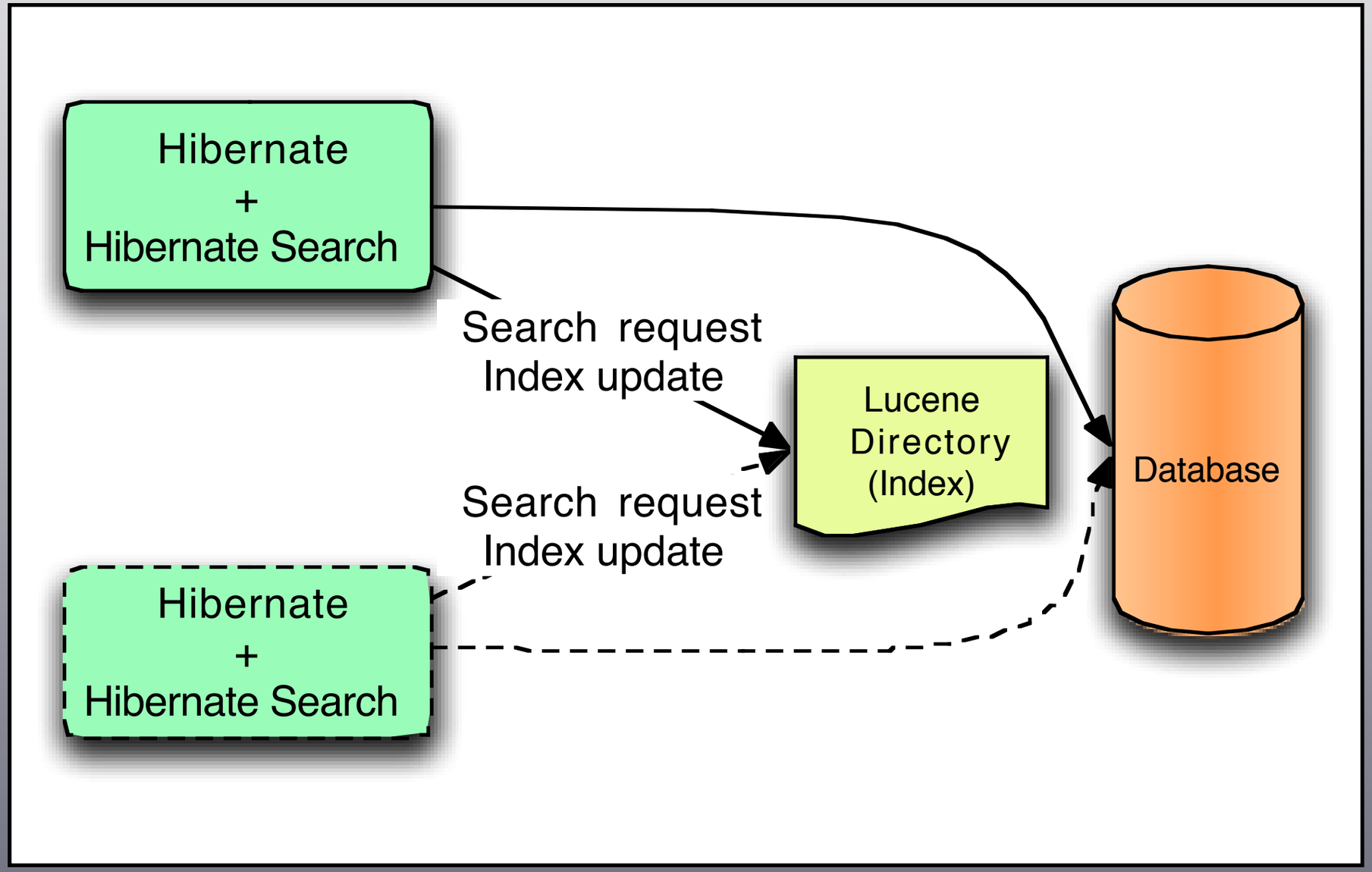
Search

```
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



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 Java Forum 2009

Document	
title	

to Java Forum 2009

Document	
title	welcom

Java Forum 2009

Document	
title	welcom

Forum 2009

Document	
title	welcom java

2009

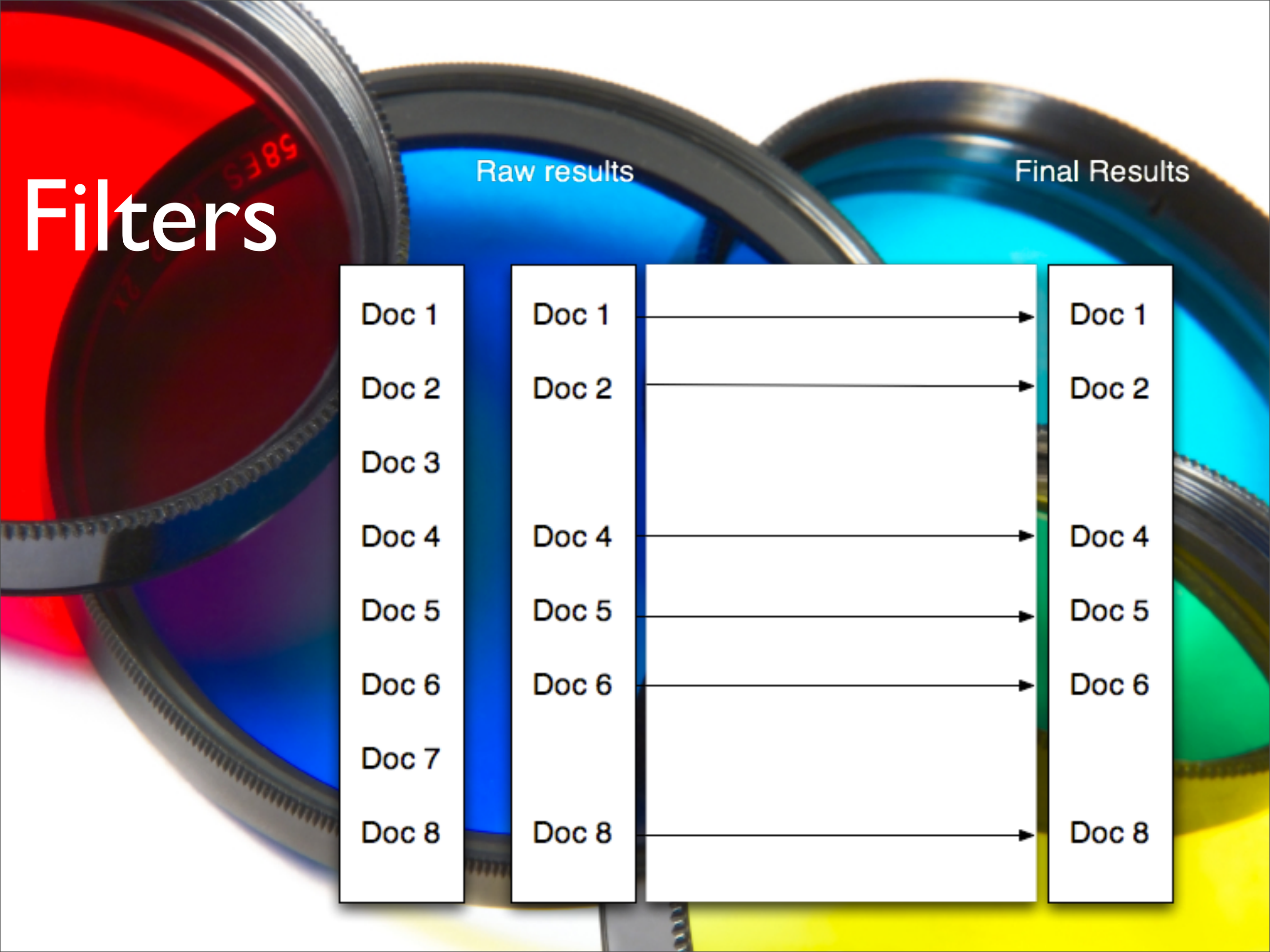
Document	
title	welcom java forum

Document	
title	welcom java forum 2009

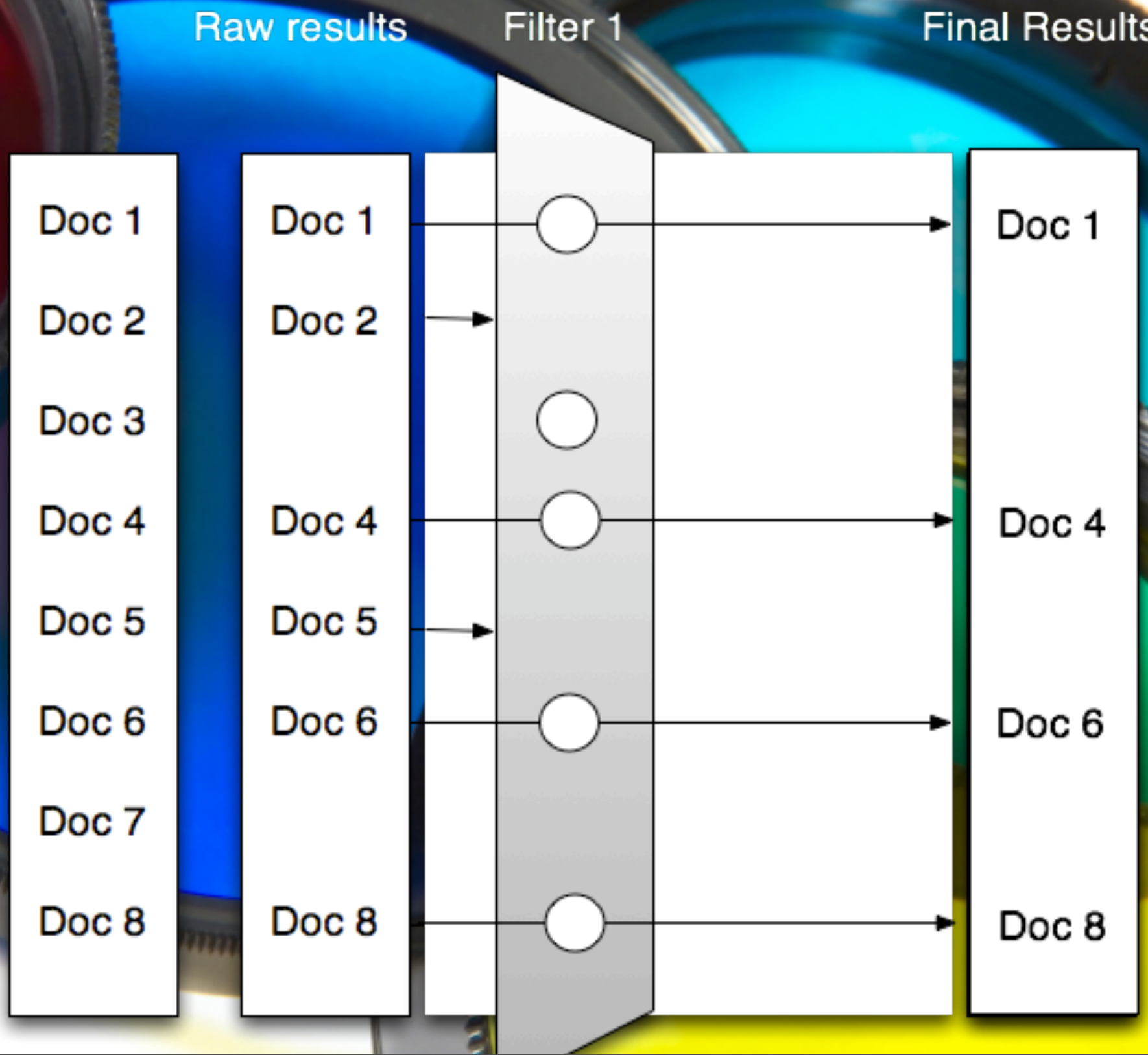
Analyzer Example

```
@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;
  ...
}
```

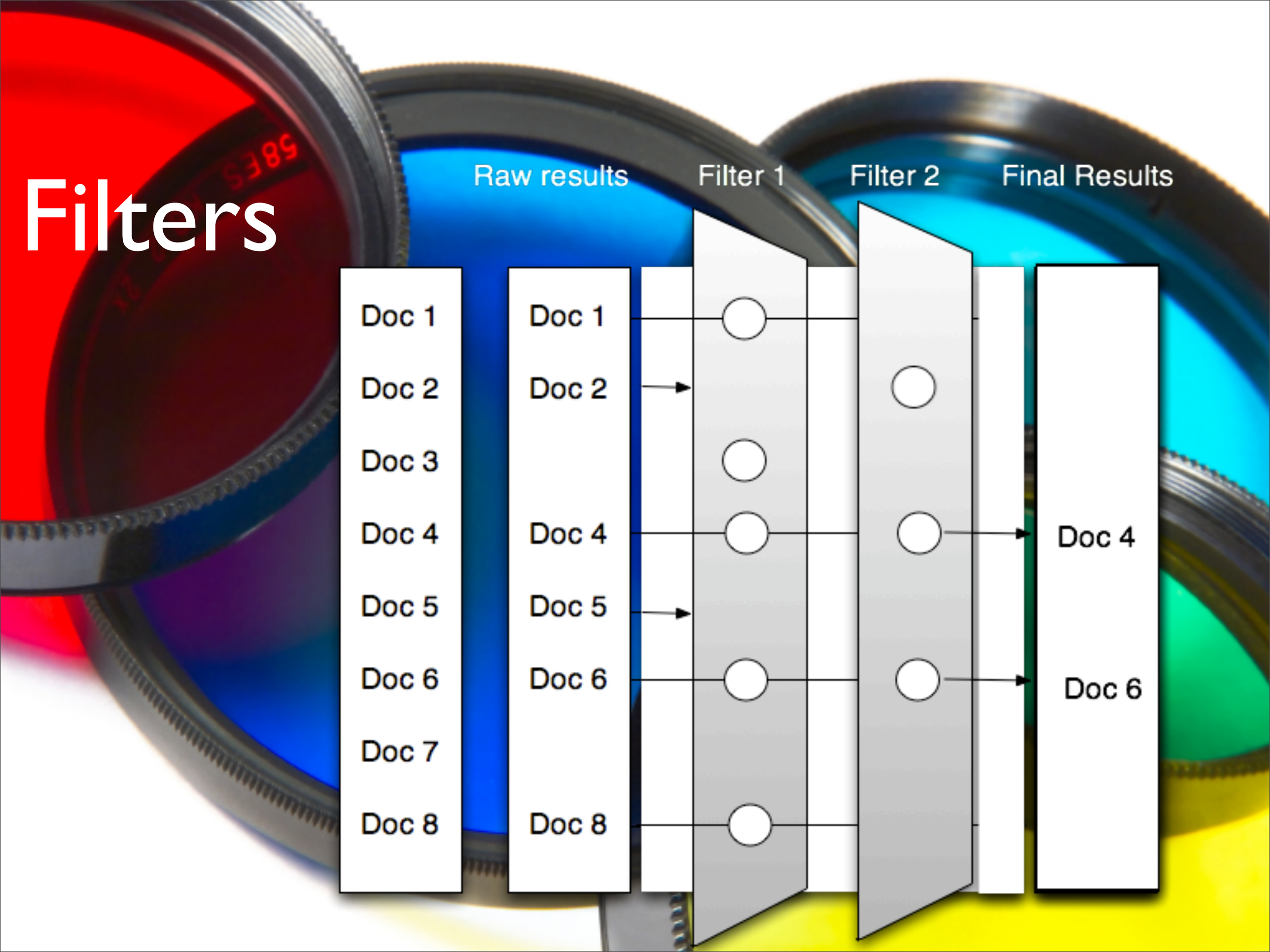
Filters


Raw results
Final Results
Doc 1
Doc 2
Doc 3
Doc 4
Doc 5
Doc 6
Doc 7
Doc 8
Doc 1
Doc 2
Doc 4
Doc 5
Doc 6
Doc 8
Doc 1
Doc 2
Doc 4
Doc 5
Doc 6
Doc 8

Filters


Raw results
Filter 1
Final Results
Doc 1
Doc 2
Doc 3
Doc 4
Doc 5
Doc 6
Doc 7
Doc 8
Doc 1
Doc 2
Doc 4
Doc 5
Doc 6
Doc 8
Doc 1
Doc 4
Doc 6
Doc 8

Filters


Raw results
Filter 1
Filter 2
Final Results
Doc 1
Doc 2
Doc 3
Doc 4
Doc 5
Doc 6
Doc 7
Doc 8
Doc 1
Doc 2
Doc 4
Doc 5
Doc 6
Doc 8
Doc 4
Doc 6

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();
...
```

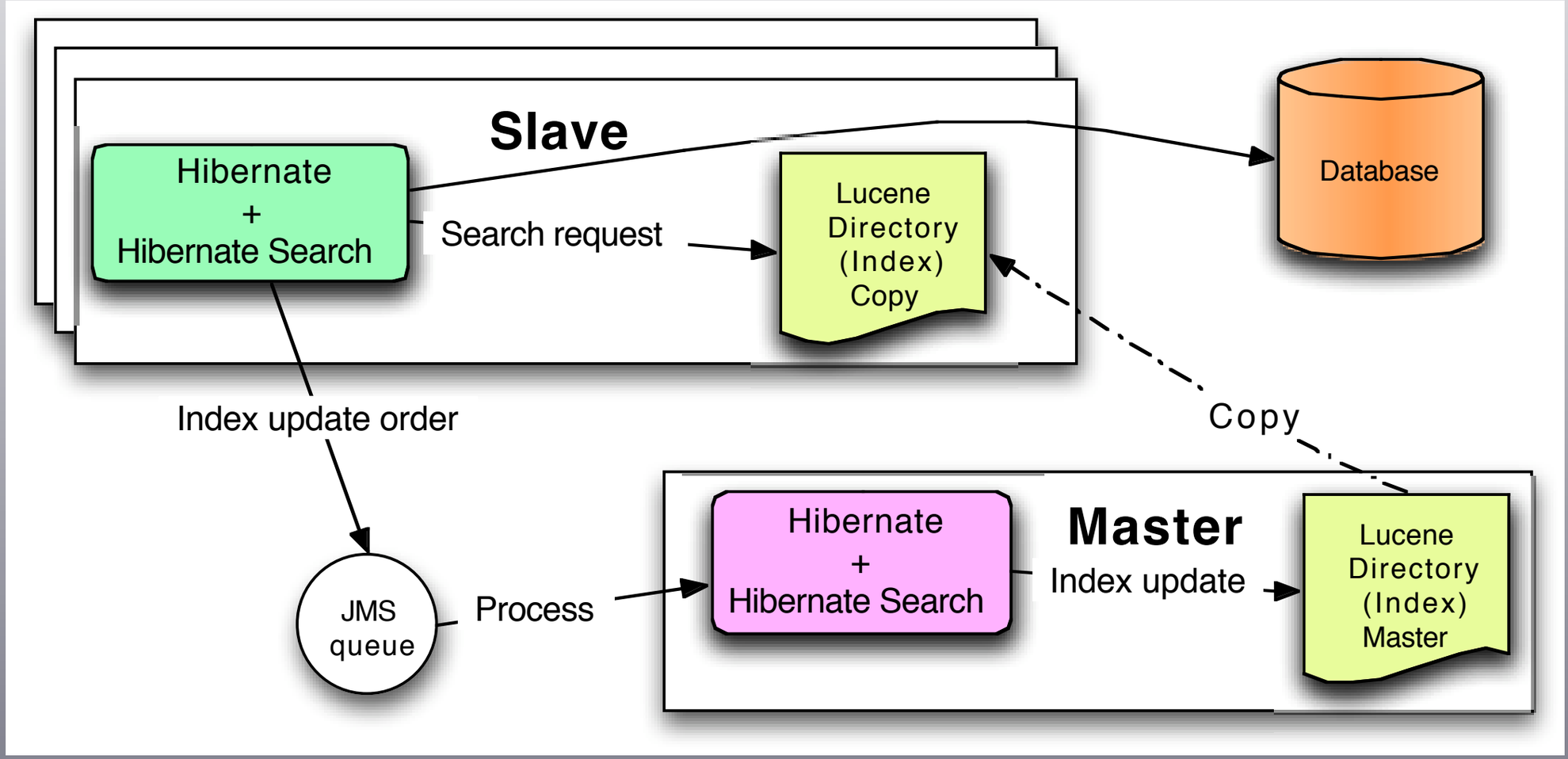
Projection

- 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


Slave
Hibernate
+
Hibernate Search
Search request
Lucene
Directory
(Index)
Copy
Database
Index update order
Copy
JMS
queue
Process
Hibernate
+
Hibernate Search
Master
Index update
Lucene
Directory
(Index)
Master

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!

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