CHAIR FOR DATABASES AND
INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

Martin Braun - Bachelor Thesis presentation

UNIVERSITÄT BAYREUTH

THURSDAY, 24 September 2015

Integration of JPA-conform ORM-Implementations in Hibernate Search

CONTACT

Martin Braun
E-Mail martinbraun123@aol.com

DATENBANKEN UND INFORMATIONSSYSTEME

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

1. Introduction

- Abstraction/standardization is key in the software world (Java Enterprise)

- Database access is standardized
  - Object Relational Mappers (ORM)
  - Java Persistence API (JPA)
    - Hibernate ORM (Red Hat)
    - EclipseLink (Eclipse Foundation)
    - OpenJPA (Apache Foundation)

HIBERNATE

eclipselink

OpenJPA

- Special features of different JPA implementations
  - e.g. Hibernate Search

CONTACT

Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

1. Introduction

What is Hibernate Search?

- Lucene based fulltext search engine on top of Hibernate ORM
  - indexes and queries managed JPA objects
  - keeps index up-to-date

- Fulltext search in a regular RDBMS:

  SELECT book.id, book.name FROM book WHERE book.name LIKE %hobbit%;

- Hibernate Search: more complex queries and index options
  - fuzzy queries — *orignal hits original*
  - regular expression queries — *[Ll]ucene hits Lucene, lucene*
  - stemming (language specific) — *worker -> work*
  - comprehensive synonym support — *book -> essay, album, novel*

CONTACT

Martin Braun
E-Mail martinbraun123@aol.com

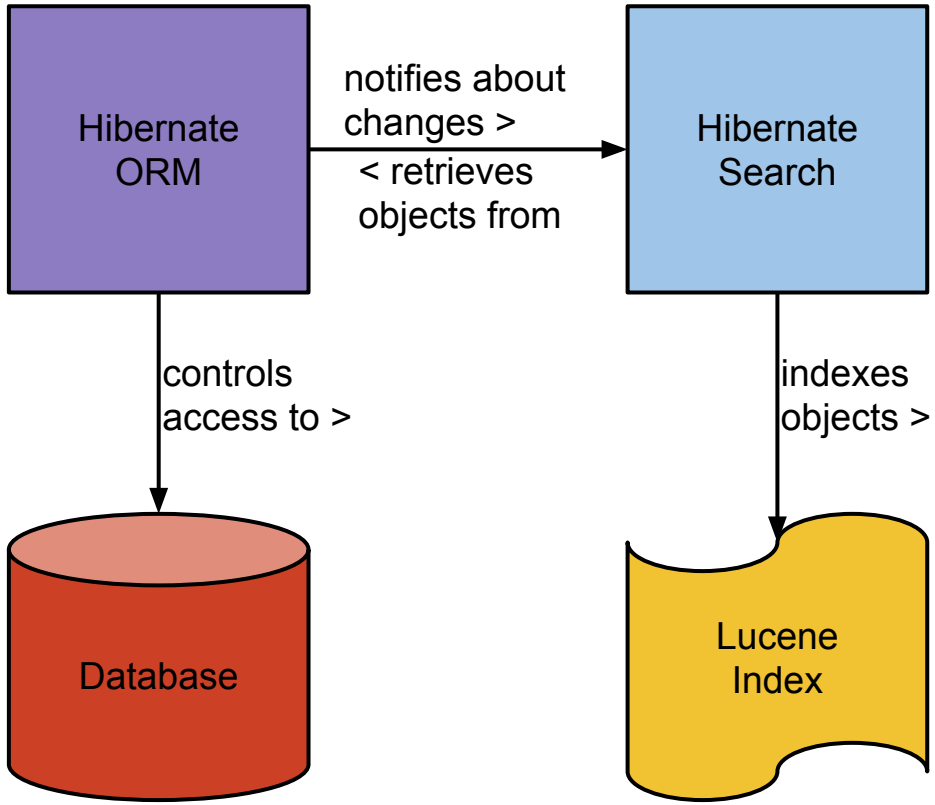
CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

1. Introduction

Hibernate Search with Hibernate ORM:



CONTACT

Martin Braun
E-Mail martinbraun123@aol.com

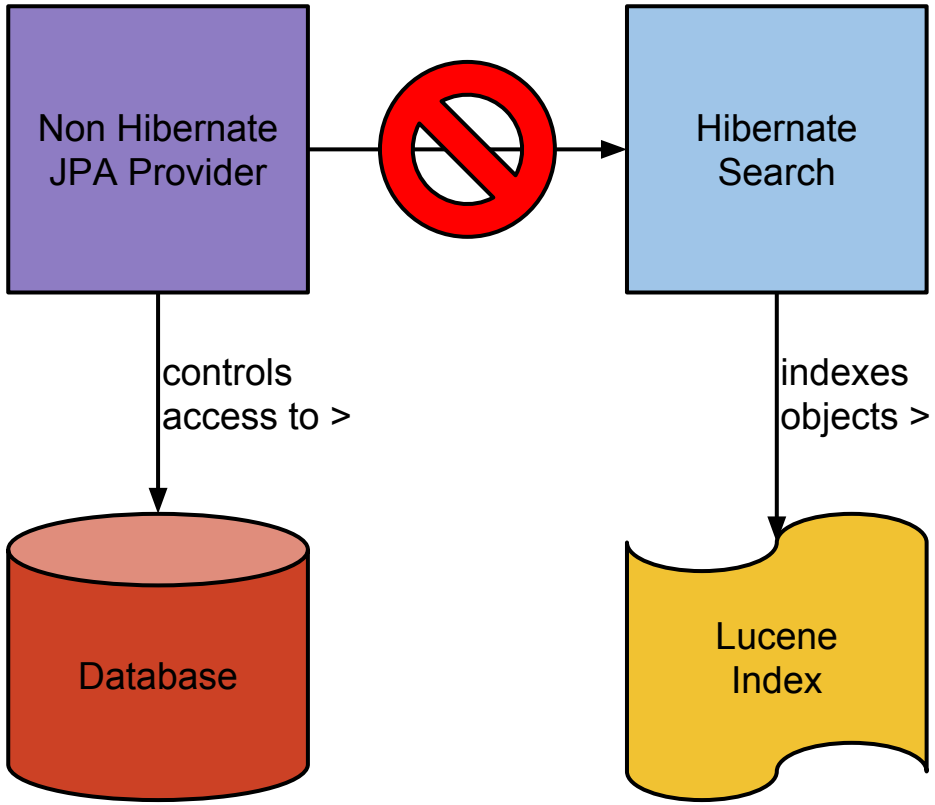
CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

1. Introduction

Hibernate Search with other JPA providers:



CONTACT

Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

1. Introduction

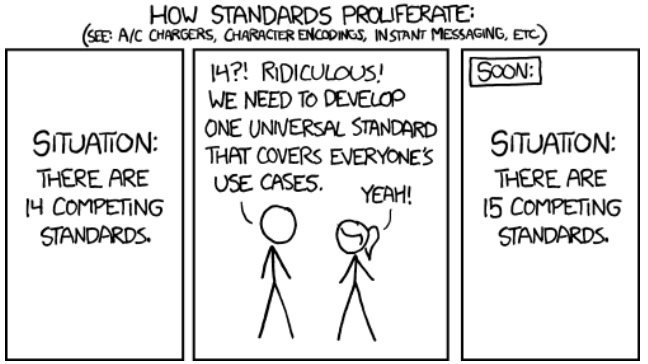
Why a generic Hibernate Search?

- Other search solutions are not integrated with JPA:
  - native Lucene
  - ElasticSearch / Solr

- lack of integration results in:
  - manual conversion into index objects required
  - no automatic index updating
  - no JPA objects as return values in queries

- No new generic alternative needed:
  - existing JPA based interfaces in Hibernate Search
  - backend (hibernate-search-engine) is integration agnostic

- This approach is backed by the Hibernate team



CONTACT

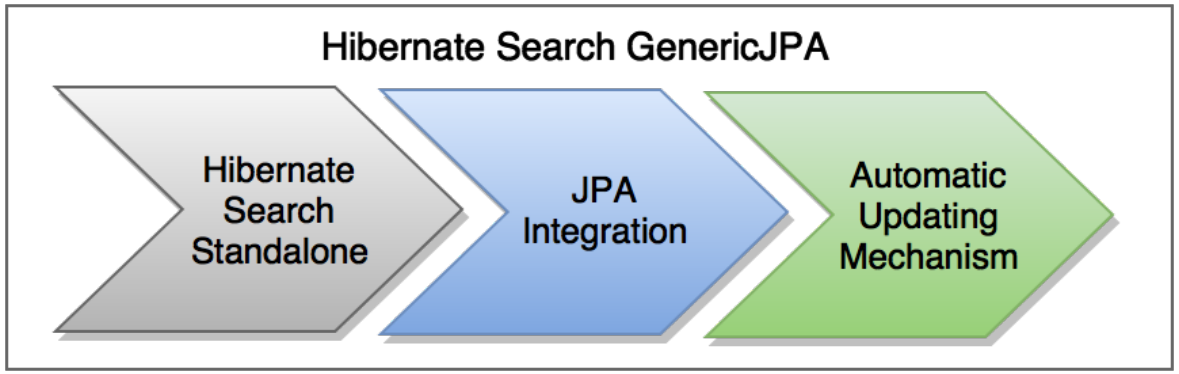
Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

2. Challenges



- Hibernate Search Standalone
  - abstraction layer on-top of the low-level API of Hibernate Search's engine

- JPA integration
  - integration of the standalone with generic JPA
  - re-use the interfaces from hibernate-search-orm

- Automatic Updating Mechanism

CONTACT

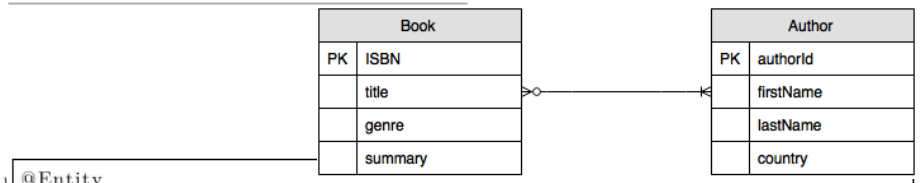
Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

3. Standalone Version of Hibernate Search




```java
@Entity
@Table(name = "Book")


public class Book {

        @Id
        @Column(name = "isbn")

        private String isbn;

        @Column(name = "title")

        private String title;

        @Column(name = "genre")

        private String genre;

        @Lob
        @Column(name = "summary")

        private String summary;

        @ManyToMany(mappedBy = "books", cascade = {
                CascadeType.MERGE,
                CascadeType.DETACH,
                CascadeType.PERSIST,
                CascadeType.REFRESH
        })

        private Set<Author> authors;

        //getters & setters ...
}
```

```java
@Entity
@Table(name = "Author")

public class Author {

        @Id
        @GeneratedValue(strategy = GenerationType.AUTO)
        @Column(name = "authorId")

        private Long authorId;

        @Column(name = "firstName")

        private String firstName;

        @Column(name = "lastName")

        private String lastName;

        @Column(name = "country")

        private String country;

        @ManyToMany(cascade = {
                CascadeType.MERGE,
                CascadeType.DETACH,
                CascadeType.PERSIST,
                CascadeType.REFRESH
        })
        @JoinTable(name = "Author_Book",
                joinColumns =
                        @JoinColumn(name = "authorFk",
                                referencedColumnName = "authorId"),
                inverseJoinColumns =
                        @JoinColumn(name = "bookFk",
                                referencedColumnName = "isbn"))

        private Set<Book> books;

        //getters & setters ...
}
```

THURSDAY, 24 SEPTEMBER 2015

CONTACT

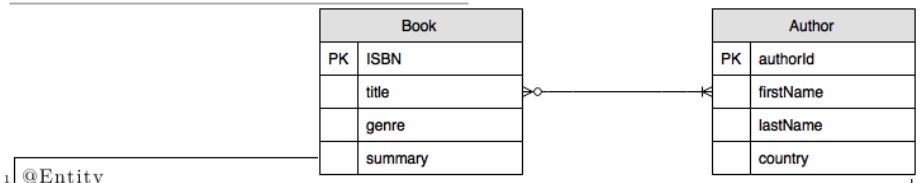
Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

3. Standalone Version of Hibernate Search




```java
@Entity
@Table(name = "Book")
@InIndex
@Indexed
public class Book {

        @Id
        @Column(name = "isbn")
        @DocumentId
        private String isbn;

        @Column(name = "title")
        @Field(store = Store.YES, index = Index.YES)
        private String title;

        @Column(name = "genre")
        @Field(store = Store.YES, index = Index.YES)
        private String genre;

        @Lob
        @Column(name = "summary")
        @Field(store = Store.NO, index = Index.YES)
        private String summary;

        @ManyToMany(mappedBy = "books", cascade = {
                CascadeType.MERGE,
                CascadeType.DETACH,
                CascadeType.PERSIST,
                CascadeType.REFRESH
        })
        @IndexedEmbedded(includeEmbeddedObjectId = true)
        private Set<Author> authors;

        //getters & setters ...
}
```

```java
@Entity
@Table(name = "Author")
@InIndex
public class Author {

        @Id
        @GeneratedValue(strategy = GenerationType.AUTO)
        @Column(name = "authorId")
        @DocumentId
        private Long authorId;

        @Column(name = "firstName")
        @Field(store = Store.YES, index = Index.YES)
        private String firstName;

        @Column(name = "lastName")
        @Field(store = Store.YES, index = Index.YES)
        private String lastName;

        @Column(name = "country")
        @Field(store = Store.YES, index = Index.YES)
        private String country;

        @ManyToMany(cascade = {
                CascadeType.MERGE,
                CascadeType.DETACH,
                CascadeType.PERSIST,
                CascadeType.REFRESH
        })
        @JoinTable(name = "Author_Book",
                joinColumns =
                        @JoinColumn(name = "authorFk",
                                referencedColumnName = "authorId"),
                inverseJoinColumns =
                        @JoinColumn(name = "bookFk",
                                referencedColumnName = "isbn"))
        @ContainedIn
        private Set<Book> books;

        //getters & setters ...
}
```

THURSDAY, 24 SEPTEMBER 2015

CONTACT

Martin Braun
E-Mail martinbraun123@aol.com

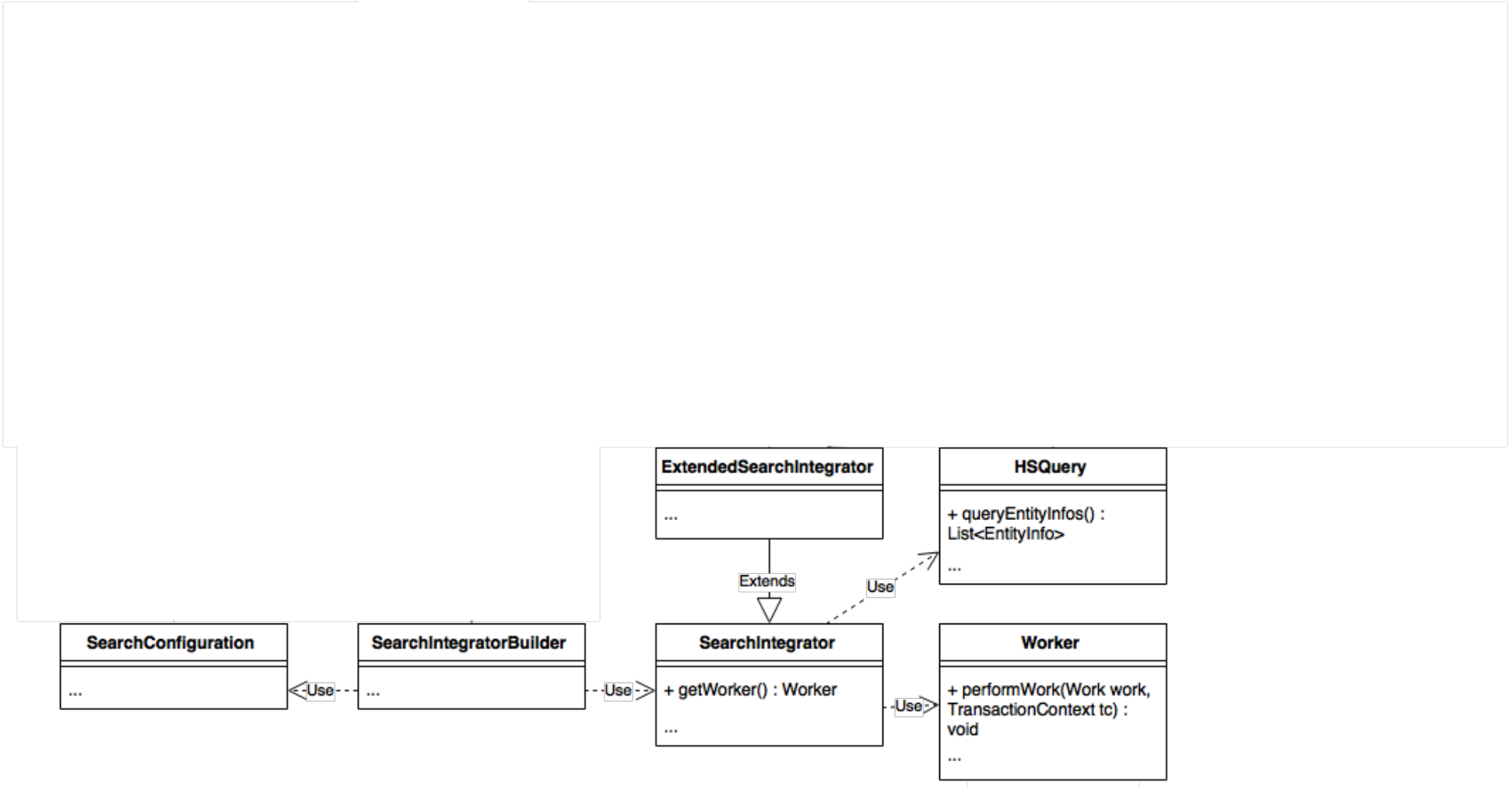
DATENBANKEN UND INFORMATIONSSYSTEME

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

3. Standalone Version of Hibernate Search



THURSDAY, 24 SEPTEMBER 2015

CONTACT

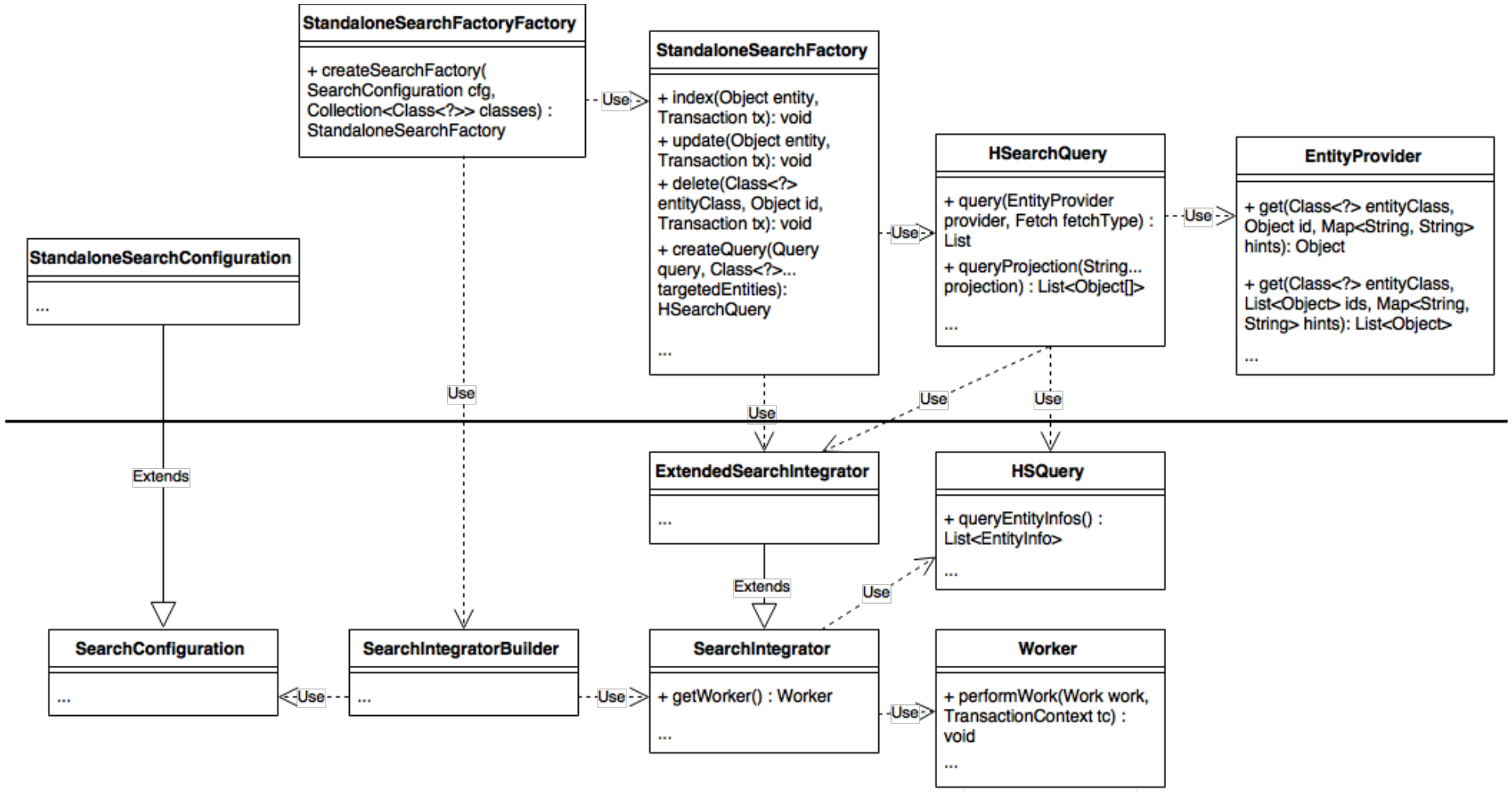
Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

3. Standalone Version of Hibernate Search



THURSDAY, 24 SEPTEMBER 2015

CONTACT

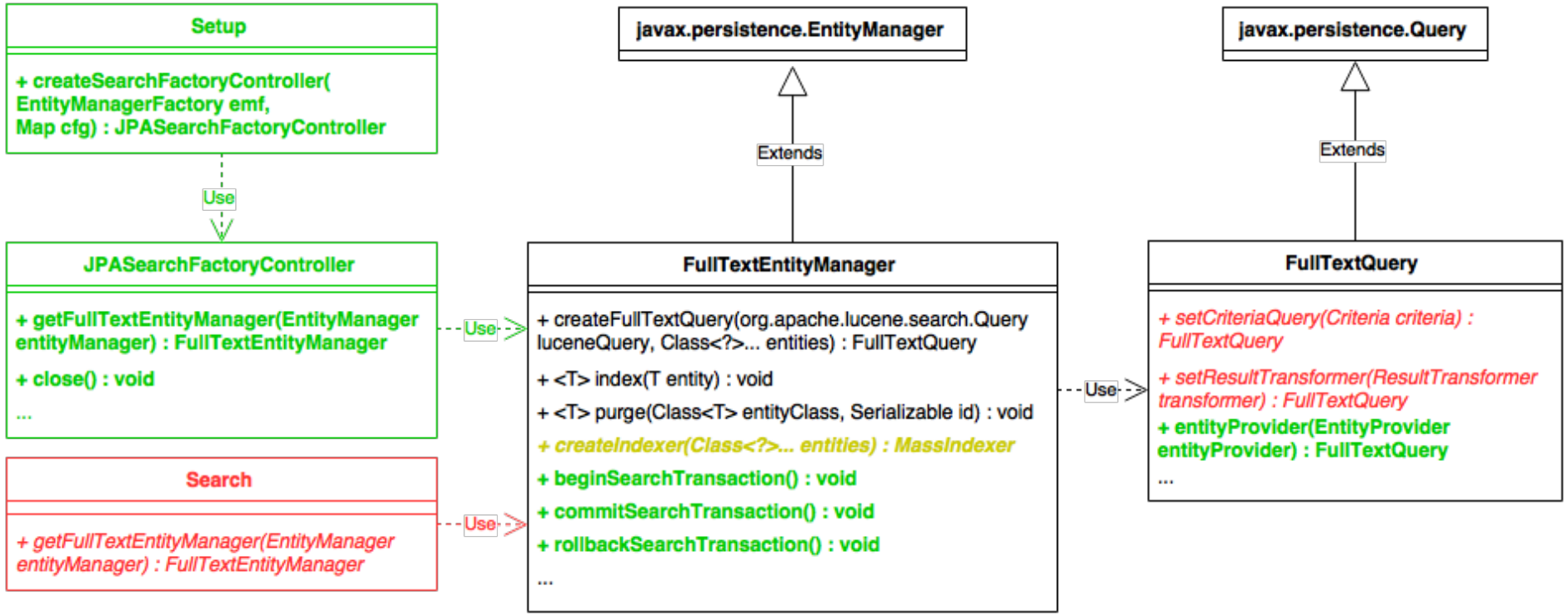
Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

4. JPA integration of the standalone version



THURSDAY, 24 SEPTEMBER 2015

CONTACT

Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

5. Automatic Index Updating

- Index must be kept up-to-date when database changes

- Manual index updating is hard to maintain

- Hibernate Search ORM:
  - integrates with Hibernate ORMs event model
  - does not work with direct database changes (SQL on DB, custom queries)

- Candidates for Hibernate Search GenericJPA:
  - 1. JPA event based model — synchronous
  - 2. Native event based model similar to Hibernate Search ORM — synchronous
  - 3. Trigger based approach — asynchronous

CONTACT

Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

5. Automatic Index Updating - JPA events

```java
public class EntityListener {

        @PostPersist
        public void persist(Object entity) {
                //handle the event
        }

        @PostUpdate
        public void update(Object entity) {
                //handle the event
        }

        @PostDelete
        public void delete(Object entity) {
                //handle the event
        }

}
```

```java
@EntityListeners( { EntityListener.class } )
public class Book {

        //...

}
```

Problems:

- JPA providers handle events differently

CONTACT

Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

5. Automatic Index Updating - JPA events

```
1  EntityManager em = ...;
2
3  em.getTransaction().begin();
4
5  Book book = em.find( Book.class, "someIsbn" );
6  book.setTitle( "someNewTitle" );
7
8  // flushes, so we retrieve the Book with the changes from above
9  // => event is triggered
10 List<Book> allBooks =
11         em.createQuery( "SELECT b FROM Book b" ).getResultList();
12
13 // we have no way to get this event to revert the wrong index change
14 em.getTransaction().rollback();
```

Problems:

- JPA providers handle events differently
- Events are triggered on flush

→ unusable for index updating

CONTACT

Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

5. Automatic Index Updating - Native Events

- All big JPA implementations (Hibernate ORM, EclipseLink, OpenJPA) have a native Listener mechanism

- Native update mechanisms in Hibernate Search GenericJPA:
  - Hibernate ORM
  - EclipseLink

- Same behaviour as Hibernate Search ORM's mechanism

  ⟶ do not work with direct database changes (SQL on DB, custom queries)

- Implementation is straight-forward, not part of thesis

CONTACT

Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

5. Automatic Index Updating - Triggers

- Triggers are supported by most RDBMSs

- Not standardized, code abstraction needed for generation

- Listen for changes directly in the database, write into auxiliary tables
  - changes with native SQL are recognized
  - support for legacy applications

CONTACT

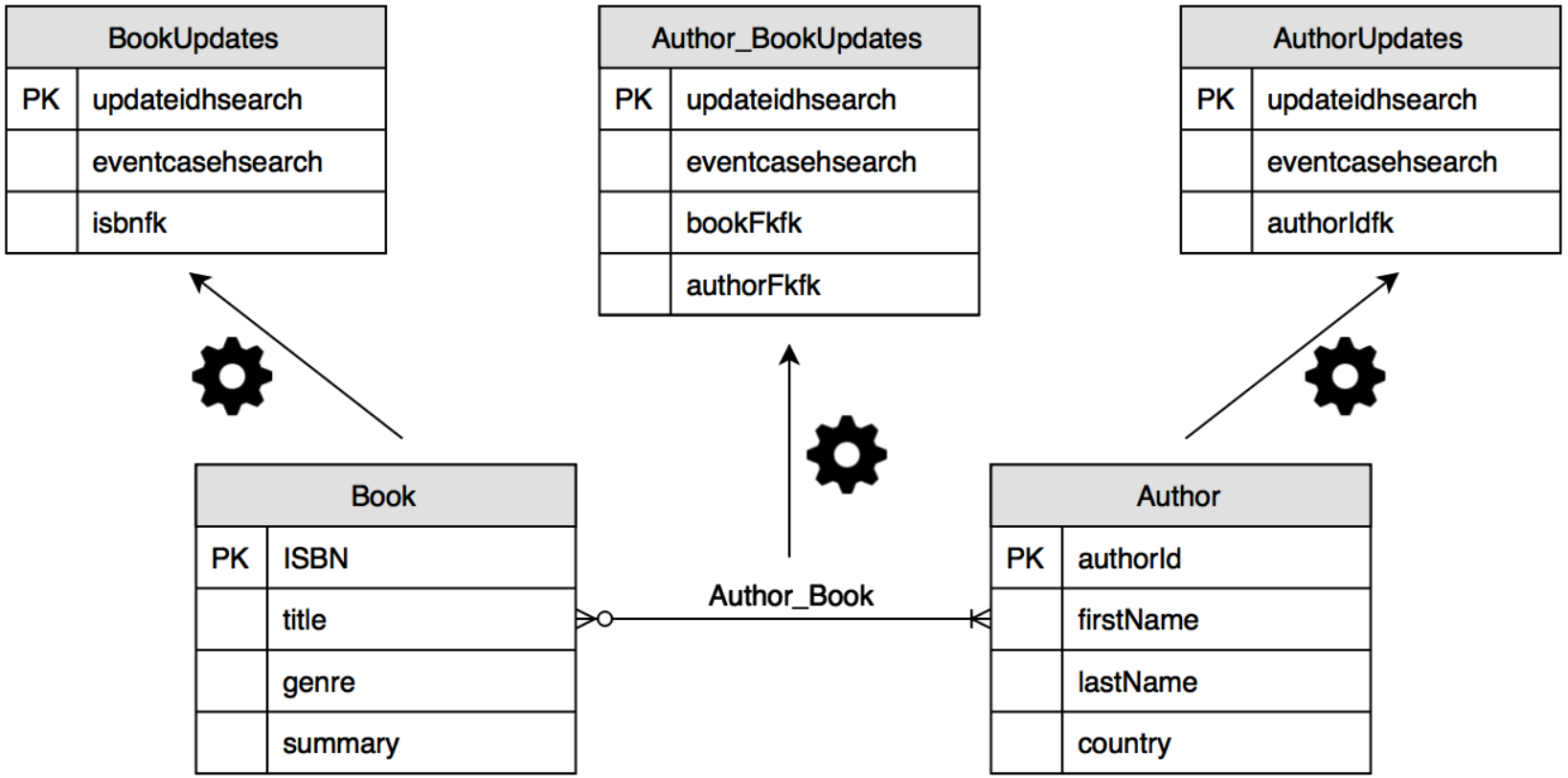
Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

5. Automatic Index Updating - Triggers



CONTACT

Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

5. Automatic Index Updating - Triggers

```java
@Entity
@InIndex
@Table(name = "Book")
@Indexed
@UpdateInfo(
        tableName = "Book",
        idInfos = @IdInfo(
                columns = @IdColumn(
                        column = "isbn",
                        columnType = ColumnType.STRING
                )
        )
)
public class Book {

        // ... unchanged.

        // mapping table events handled on Author side

        // getters & setters ...
}
```

```java
@Entity
@InIndex
@Table(name = "Author")
@UpdateInfo(
        tableName = "Author",
        idInfos = @IdInfo(
                columns = @IdColumn(
                        column = "authorId",
                        columnType = ColumnType.LONG
                )
        )
)
public class Author {

        // ... unchanged.

        @UpdateInfo(tableName = "Author_Book",
                idInfos = {
                @IdInfo(entity = Author.class,
                        columns = @IdColumn(
                                column = "authorFk",
                                columnType = ColumnType.LONG
                        )
                ),
                @IdInfo(entity = Book.class,
                        columns = @IdColumn(
                                column = "bookFk",
                                columnType = ColumnType.STRING
                        )
                )
        })
        private Set<Book> books;

        //getters & setters ...
}
```

THURSDAY, 24 SEPTEMBER 2015

CONTACT

Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

6. Usage Example

THURSDAY, 24 SEPTEMBER 2015

CONTACT

Martin Braun
E-Mail martinbraun123@aol.com

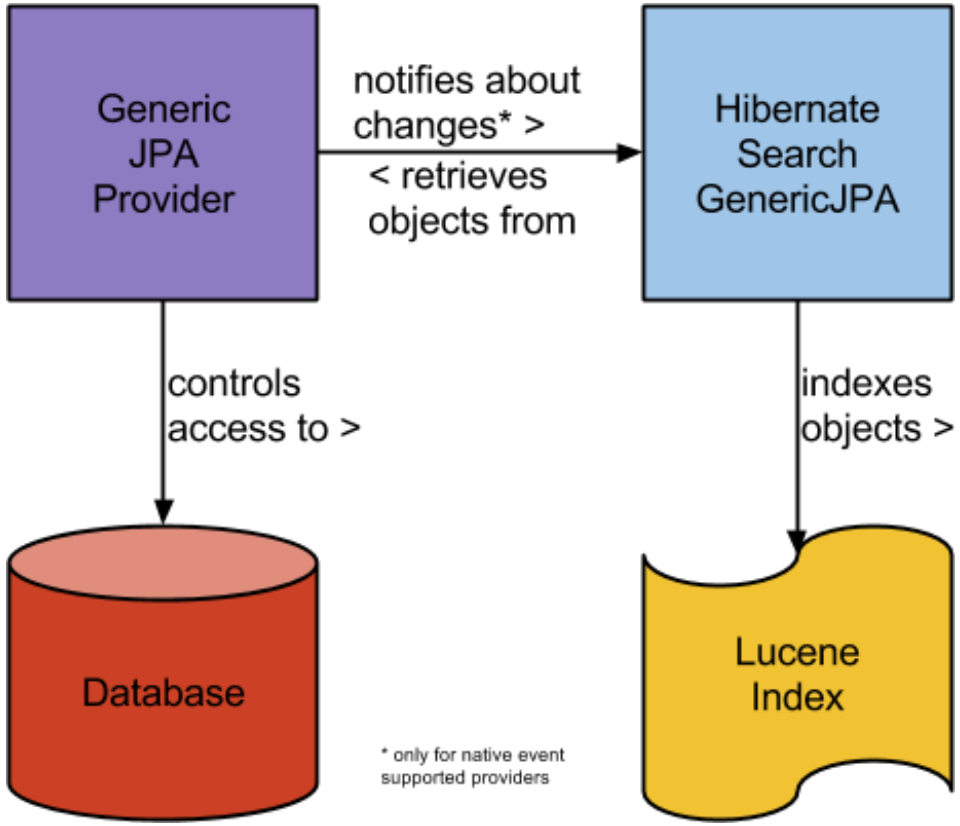
CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

7. Current situation

Hibernate Search without Hibernate ORM:



THURSDAY, 24 SEPTEMBER 2015

CONTACT

Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

8. Outlook

- Stable proof of concept
- Source-code on GitHub: https://github.com/Hotware/Hibernate-Search-GenericJPA
- Improvements in the trigger updating mechanism
- Merge with core Hibernate Search in November 2015

CONTACT

Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

THURSDAY, 24 SEPTEMBER 2015

CONTACT

Martin Braun
E-Mail martinbraun123@aol.com

CHAIR FOR DATABASES AND INFORMATION SYSTEMS

PROF. DR.-ING. STEFAN JABLONSKI
DR. BERNHARD VOLZ

UNIVERSITÄT BAYREUTH

Lucene Basics

Documents		
id	field1	field2
1	fulltext search lucene	search
2	lucene search	java
3	fulltext java	fulltext lucene

Inverted Index		
Term		Occurences
Field	Value	
field1	fulltext	1,3
field1	search	1,2
field1	lucene	1,2
field1	java	3
field2	search	1
field2	java	2
field2	fulltext	3
field2	lucene	3

CONTACT

Martin Braun
E-Mail martinbraun123@aol.com