



Hibernate: State of the Union

Emmanuel Bernard

JBoss, a Division of Red Hat

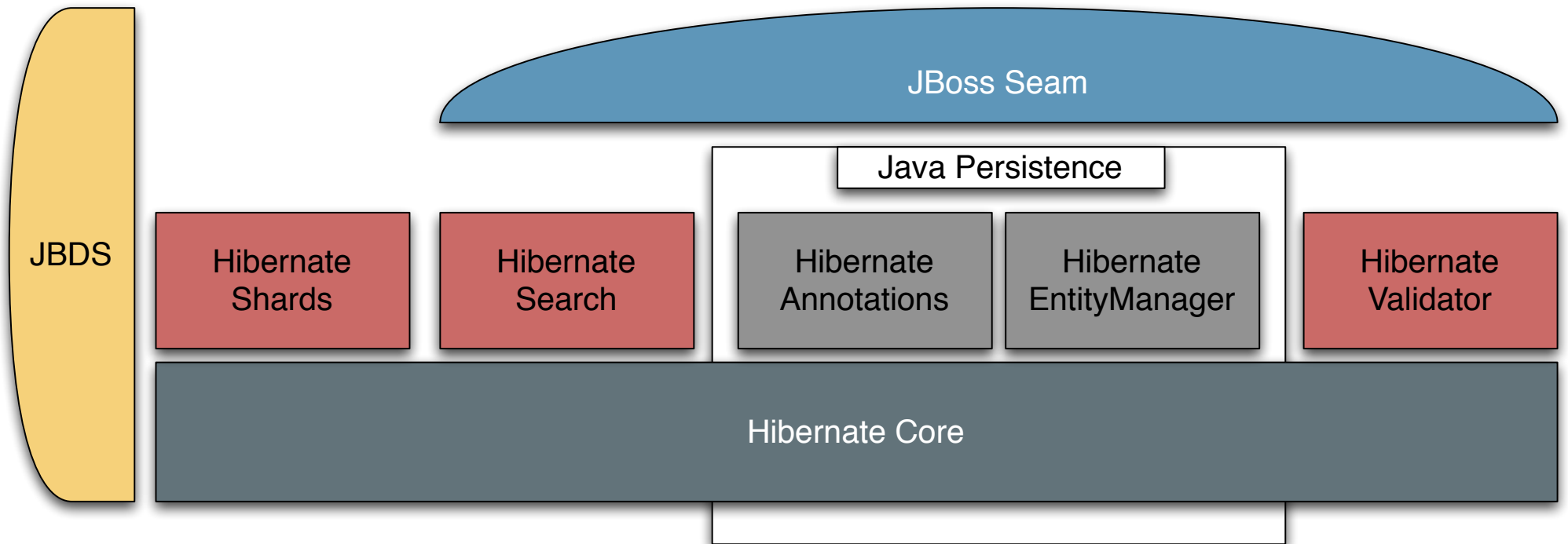
Doer



What is Hibernate

- Object Relational Mapper
- History
 - Founded in 2001 by Gavin King
 - Gain popularity (most popular ORM ever)
 - Hibernate joins the JBoss portfolio
 - Heavily influences EJB 3.0 and Java Persistence 1.0
 - Implements Java Persistence 1.0
 - Work in progress Java Persistence 2.0
- ORM? Really?

Hibernate platform



Agenda

- Hibernate Core
 - some less known Hibernate core features
 - short and long term evolution
- Java Persistence
 - Hibernate Annotations
 - Hibernate EntityManager
 - Java Persistence 2.0
- Hibernate Tools
- Hibernate Validator
- Hibernate Search
- Hibernate Shards

Hibernate Core

Some features you might not know about

- Custom id generator
- Generated properties
- Replication
- ResultTransformer

Hibernate Core - short term

- Performance
 - Second level cache redesign
 - context and data awareness
 - unflushed change replication
- Flexibility
 - JDBC Work API
 - Maven build
- Optimizations
 - stateless API

Hibernate Core - long term

- Ease of dev
 - targets Java 5
 - Merge Java Persistence projects into core
 - faster startup in dev mode
- Simpler configuration
 - less, less string based
 - mutable at runtime
- More mapping flexibilities
 - polymorphic components
 - entity modes

Hibernate Core - long term

- Optimizations
 - Fetching optimizations (under the hood)
- Query
 - UNION / MINUS
 - HQL Delete cascade
 - Align Criteria and HQL
- Functional
 - temporal and versioned data

Java Persistence

- Standardization of lightweight ORM
 - API, semantic, metadata, packaging
- Domain model centric
- Annotation based metadata
- Handle the Persistence Context life cycle
- Work in SE and EE

- Heavy involvement of JBoss
- Very good adoption

Hibernate Annotations

- Annotation based metadata
 - based 100% on Java Persistence annotations
 - extend the spec
- 95+ % Hibernate Core features covered
 - flexible and exotic mapping
 - optimization (lazy, caching, ...)
- Integrates with Hibernate Search and Validator transparently

- BTW, why annotations?

```
@Entity
class Essay {

    @Id @GeneratedValue
    Long id

    @Column(name="fld_name", nullable=false)
    String name

    @ManyToOne
    Author author
}
```

Hibernate EntityManager

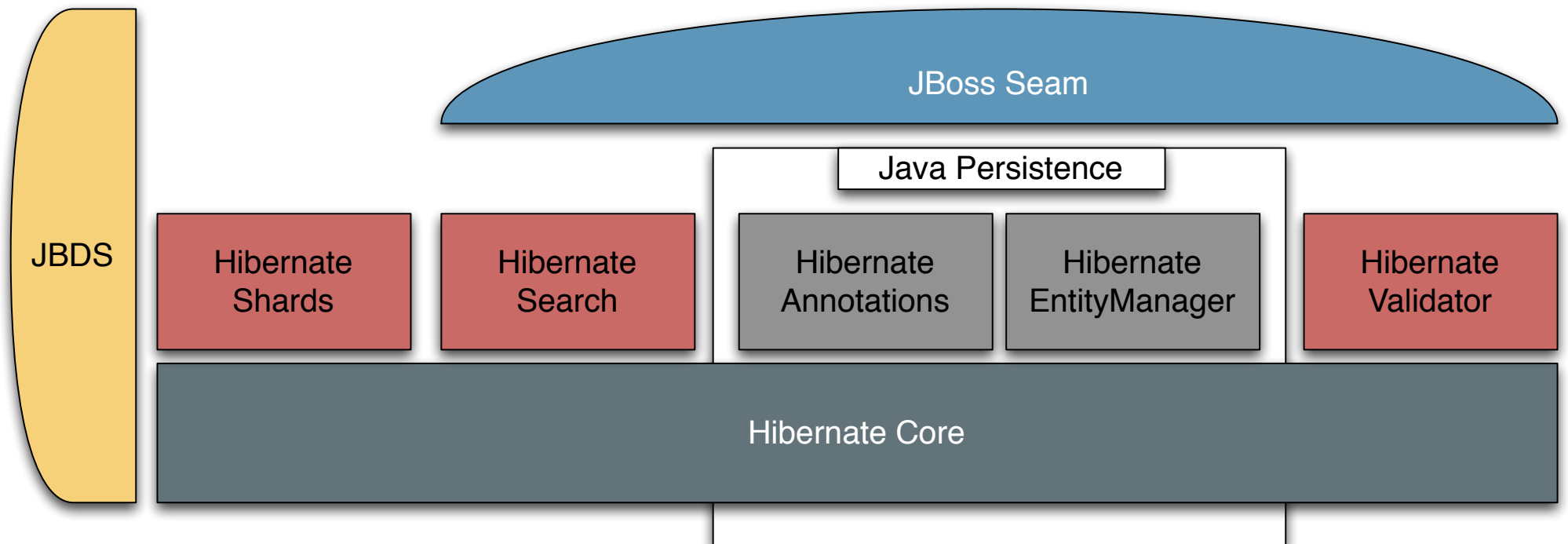
- Thin wrapper on top of Hibernate Core
- Implements the Java Persistence API
 - API
 - Configuration
 - packaging
- Entity auto discovery
- Environment
 - Work in Java EE 5 app servers
 - Work in J2SE app servers (JDK 5)
 - Work in SE

Java Persistence 2.0

- Part of Java EE 6
- Incorporate feedbacks

- Criteria API
- More advanced mappings
- Conversation
- Validation

Hibernate platform

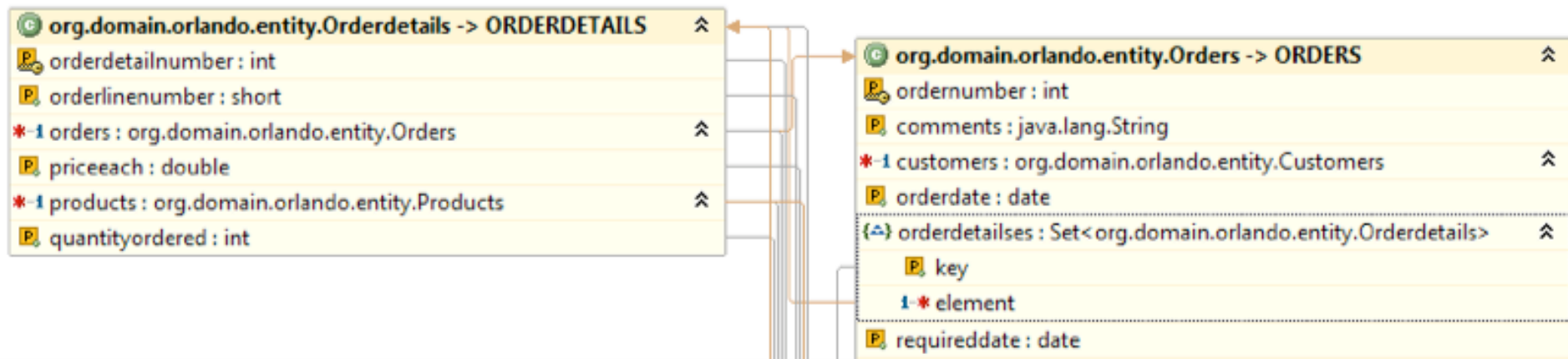


Hibernate Tools

- <http://tools.hibernate.org>
- Ant tasks
- Support hbm and JPA
- schema export / schema update
- Flexible reverse engineering and code generation
 - database schema
 - mappings
- hbmlint (experimental)
 - "lint was the name originally given to a particular program that flagged suspicious and non-portable constructs (likely to be bugs)"

Hibernate tools in Eclipse

- Prototype queries
 - HQL/JPA-QL/Criteria
 - code inline autocompletion and validation
- Mapping visualization
- Flexible Reverse engineering and code generation
- Bundled in JBoss Developer Studio



```

*OfficesList.java
@Override
public String getEjbql() {
    return "select officeces from Offices offices where offices.";
}
    
```

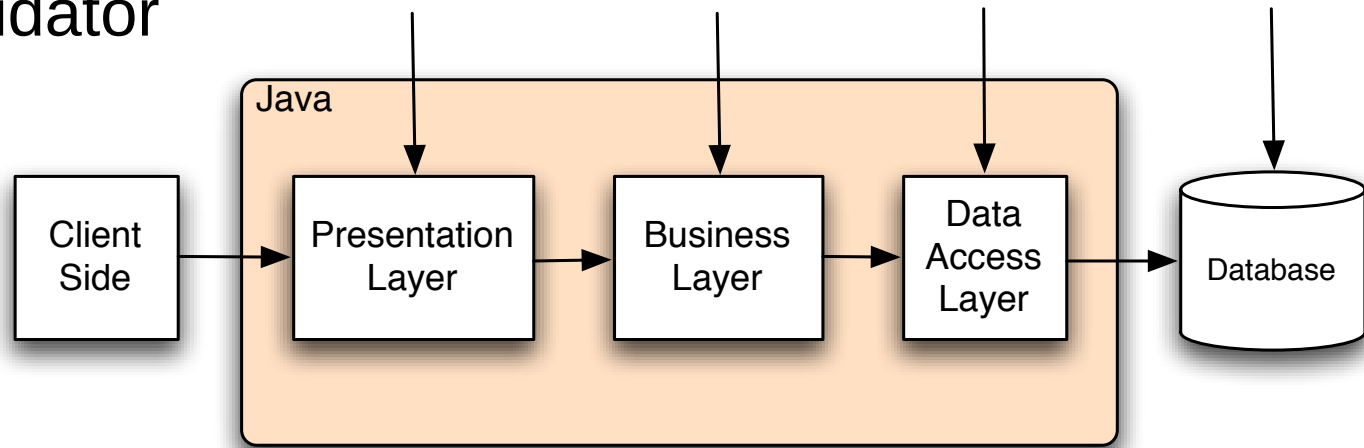
```

0: org.domain.orlando.entity.Offices
-----
select
offices0_.OFFICECODE as OFFICECODE31_
offices0_.ADDRESSLINE1 as ADDRESSL2_31_
offices0_.ADDRESSLINE2 as ADDRESSL3_31_
offices0_.CITY as CITY31_
offices0_.COUNTRY as COUNTRY31_
offices0_.PHONE as PHONE31_
offices0_.POSTALCODE as POSTALCODE31_
offices0_.STATE as STATE31_
offices0_.TERRITORY as TERRITORY31_
from
PUBLIC.OFFICES offices0_
    
```

- addressline1 - Offices
 - addressline2 - Offices
 - city - Offices
 - country - Offices
 - officecode - Offices
 - phone - Offices
 - postalcode - Offices
 - state - Offices
 - territory - Offices
- Press 'Ctrl+Space' to show JPA F

Hibernate Validator

- Annotation driven constraints
- Declare once, check anywhere
- Custom validator
- i18n



- JSR 303: Bean Validation

```
@Entity
class Essay {

    @Id @GeneratedValue
    Long id

    @NotNull @Length(30)
    @Column(name="fld_name", nullable=false)
    String name

    @ManyToOne
    @NotNull(message="essay must have an author")
    Author author
}
```

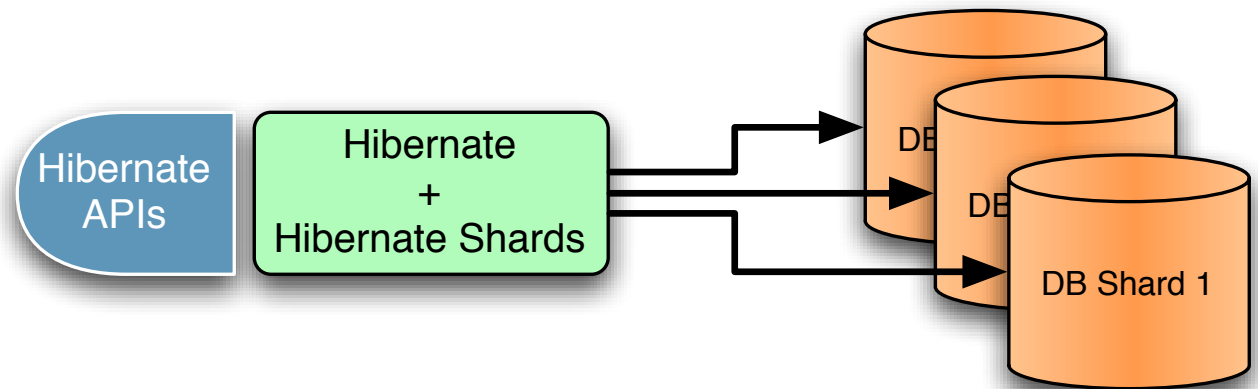
Hibernate Search

- Google your domain model
 - relevance, approximation, wildcard, words
- Solves the objects / index mismatches
 - db / index synchronization
 - object model / string conversion
 - no query paradigm shift

- Ease of use
- Clustering

Hibernate Shards

- Too much data for one database
 - horizontal partitioning
- Reuse Hibernate
 - API
 - mapping
 - configuration
- Sharding strategy
 - how to split data
 - how to query data
 - where to retrieve data





Q&A



More infos

- <http://www.hibernate.org>
- <http://in.relation.to>

- <http://www.seamframework.org>

- Parleys.com
 - <http://parleys.com>