



Java is a trademark of Sun Microsystems, Inc.



JavaOneSM

Full-Text Search: Human Heaven and Database Savior in the Cloud

Emmanuel Bernard

JBoss a Division of Red Hat

Aaron Walker

base2Services

Goals

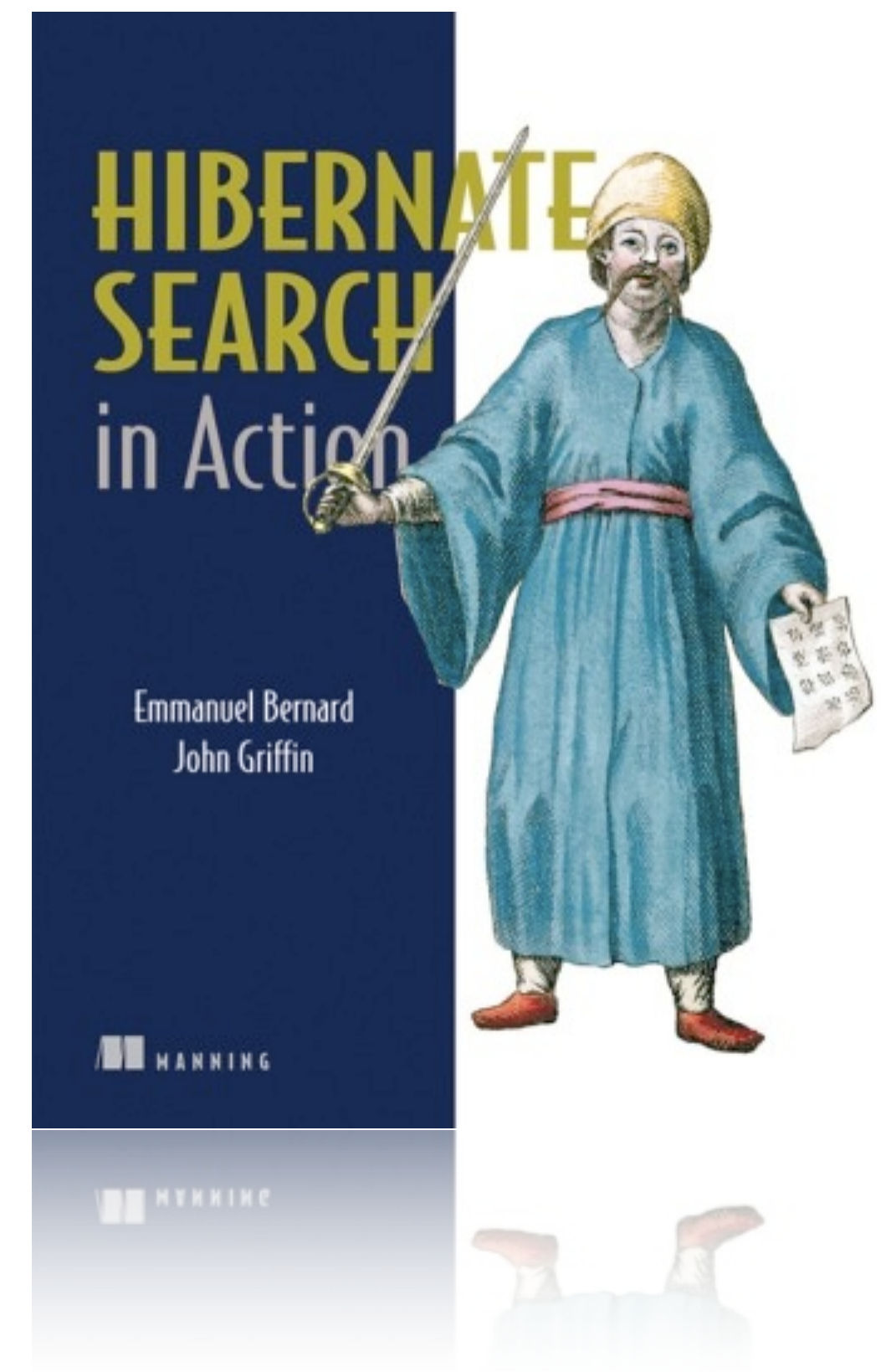
- > Happier users
- > Happier DBAs
- > Simplicity in the cloud

Emmanuel Bernard

 Hibernate Search in Action

 blog.emmanuelbernard.com

 twitter.com/emmanuelbernard



Aaron Walker



CTO base2Services



blog.base2services.com



twitter.com/aaronwalker





JavaOneSM

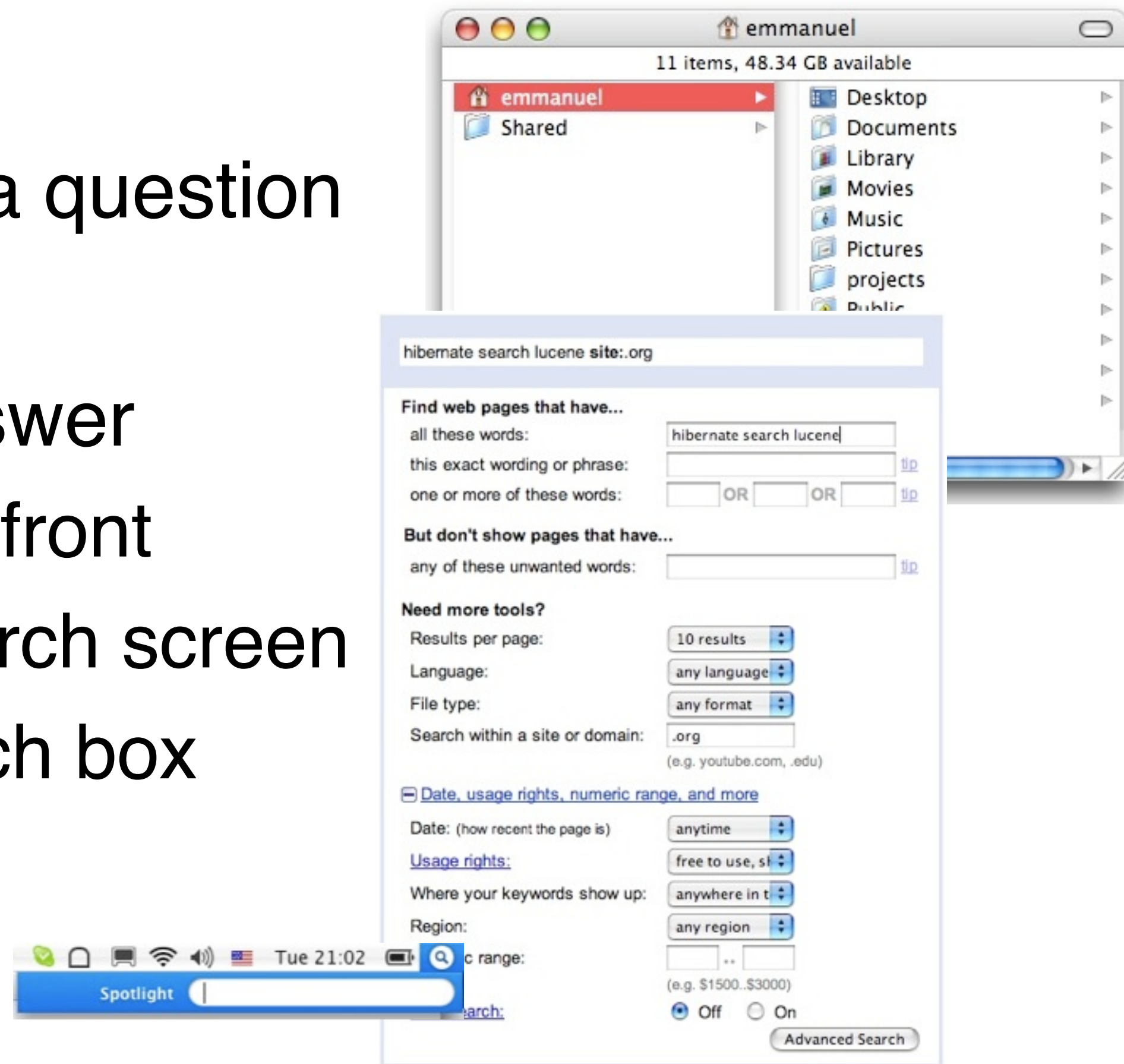
Thank You

Full-text Search and
Hibernate Search



What is searching?

- > Searching is asking a question
- > Different ways to answer
 - Categorize data up-front
 - Offer a detailed search screen
 - Offer a simple search box



SQL search limits

- > Wildcard / word search
 - ‘%hibernate%’
- > Approximation (or synonym)
 - ‘hybernate’
- > Proximity
 - ‘Java™’ close to ‘Persistence’
- > Relevance or (result scoring)
- > multi-“column” search

Full Text Search

- > Search information
 - by word
 - inverted indices (word frequency, position)

- > In RDBMS engines
 - portability (proprietary add-on on top of SQL)
 - flexibility
 - scalability

- > Standalone engine

Mismatches with a domain model

> Structural mismatch

- full text index are text only
- no reference/association between document

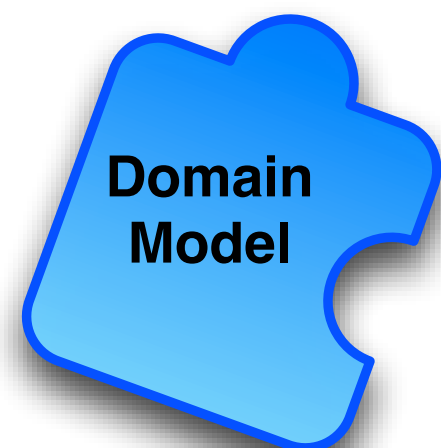


> Synchronization mismatch

- keeping index and database up to date

> Retrieval mismatch

- the index does not store objects
- certainly not managed objects



Hibernate Search

- > Transparent indexing through event system
 - PERSIST / UPDATE / DELETE
- > Convert the object structure into Index structure
 - metadata (annotations) driven
- > Expose full-text search as Hibernate queries
- > Uses Lucene under the hood
 - optimizations

Queries and indexing

> Query

- Managed objects
- extends Query APIs
- Minimal intrusion

> Indexing

- synchronous / asynchronous
- Plain Lucene / Clustered though JMS™

Mapping

@Entity @Indexed

```
public class Essay {
```

```
...
```

@Id @DocumentId

```
public Long getId() { return id; }
```

@Field(name="Abstract", index=Index.TOKENIZED, store=Store.YES)

```
public String getSummary() { return summary; }
```

@Lob @Field

```
public String getText() { return text; }
```

@ManyToOne @IndexedEmbedded

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

```
}
```


Query

```
FullTextEntityManager ftEm = Search.getFullTextEntityManager(em);  
  
FullTextSession ftSession = Search.getFullTextSession(session);  
  
org.hibernate.Query query = ftSession.createFullTextQuery(luceneQuery);  
List<?> results = query.setMaxResults(100).list();  
  
FullTextQuery query = ftSession.createFullTextQuery(luceneQuery, Author.class);  
@SuppressWarnings("unchecked")  
List<Author> results = query.setMaxResults(100).list();  
  
int totalNbrOfResults = query.getResultSize();
```



JavaOneSM

Thank You

Clustering search in a Java EE
environment without
compromising scalability

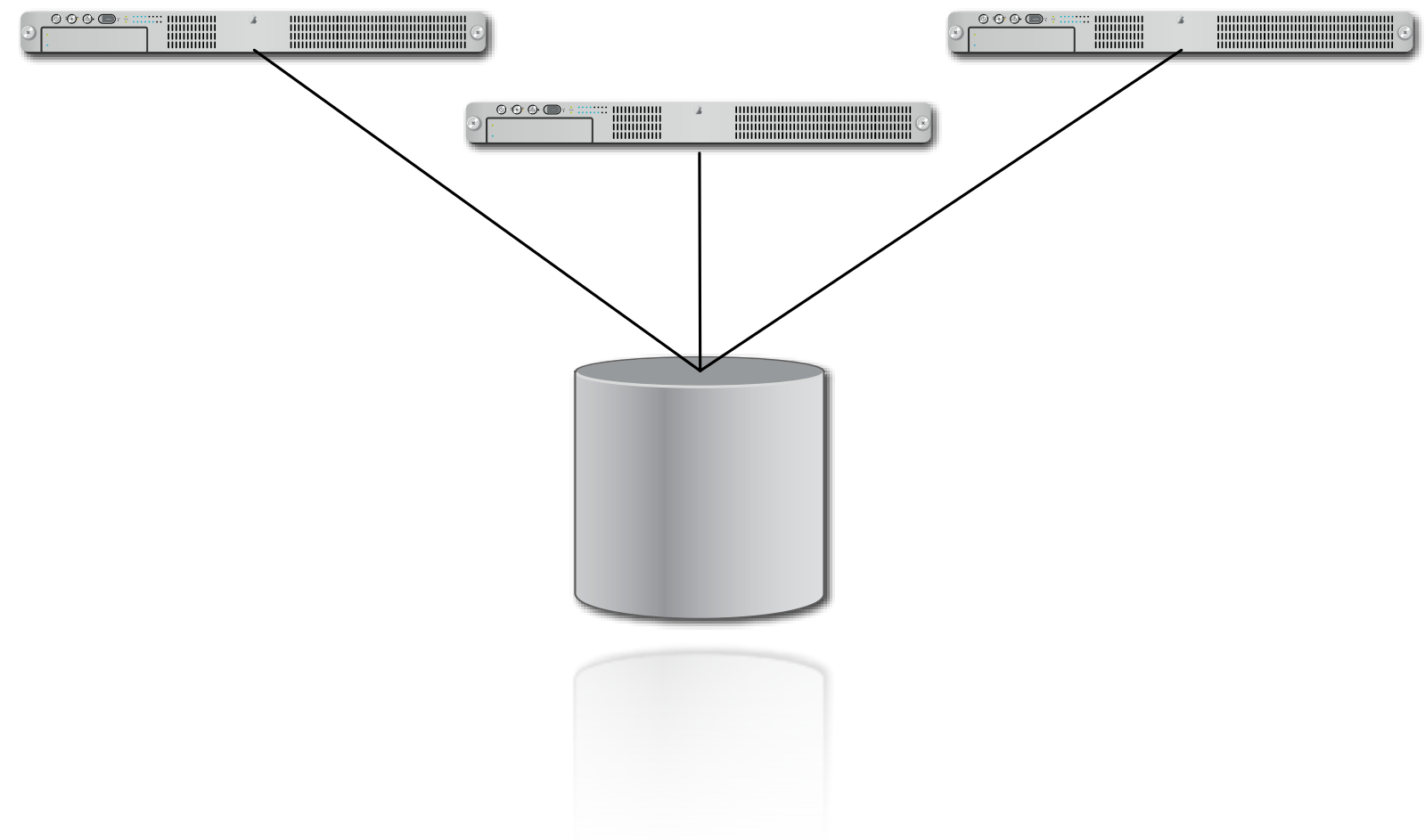


What are the problems we are trying to solve?

- > SQL limitations
 - proprietary full text search
- > performance bottlenecks
 - limited resources
 - non linear performance
- > scaling complexities
 - limited to scaling up
 - Vendor lock-in

```
MSSQL>  
SELECT * FROM articles  
WHERE CONTAINS((title, body), 'database');
```

```
MySQL>  
SELECT * FROM articles  
WHERE MATCH (title,body) AGAINST ('database');
```





JavaOneSM

Thank You

JUST MAGAZINES

Case study

JUST CARS JUST BIKES

JUST 4x4s JUST PARTS

JUST TRUCKS
HEAVY EQUIPMENT



Just Magazines

- > Australia's number 1 selling automotive magazine
- > Specializes in niche & customs vehicles
- > 525,000 readers across all magazines

Just Auto - Online automotive classifieds & communities

> Classifieds

- private & dealer ads

> Community features

- blogs
- projects
- clubs
- videos
- and more cool web 2.0 stuff :)

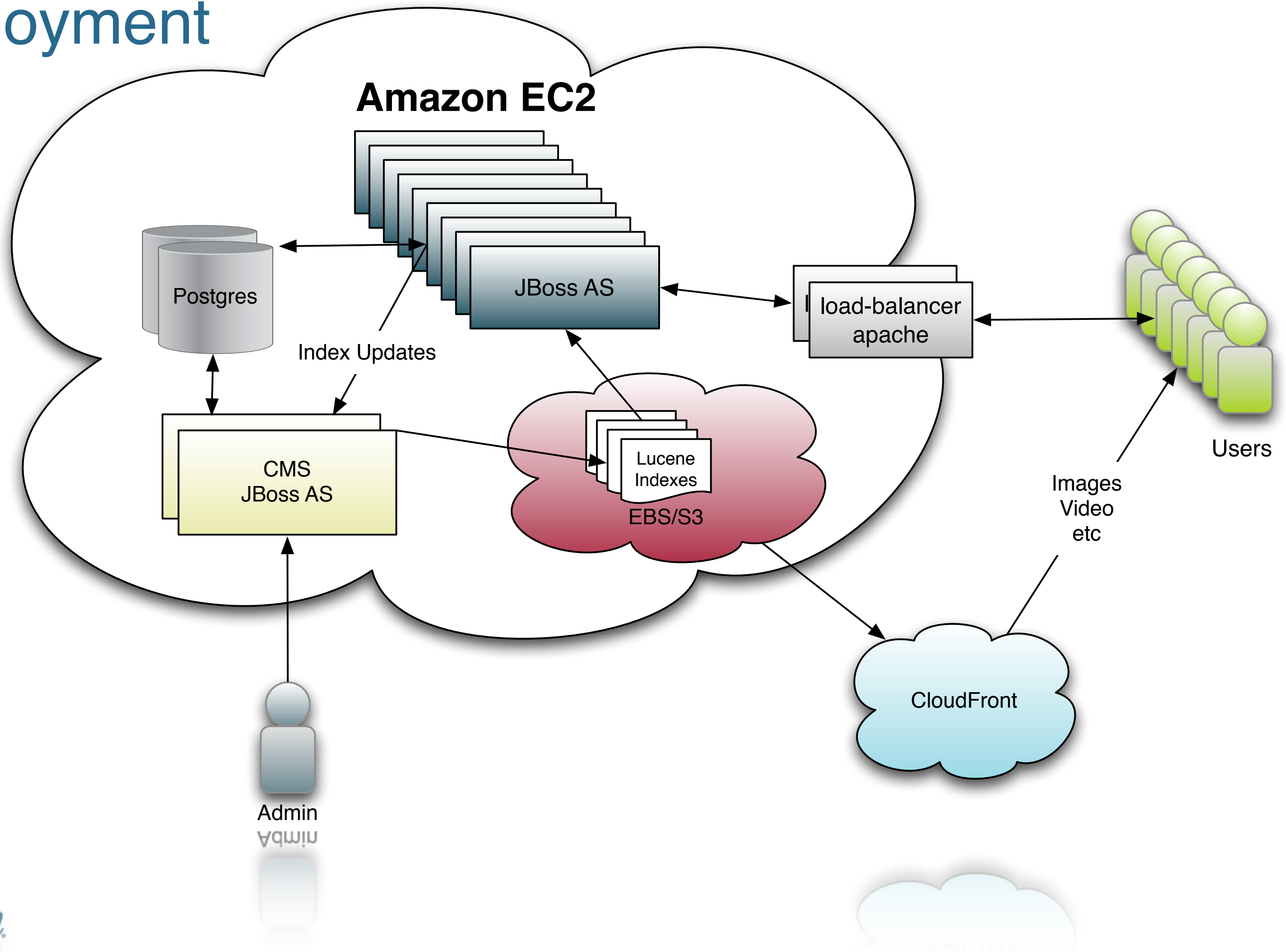
Technology Stack

- > Standard JEE APIs
 - primarily EJB 3.0, JPA & JAX-RS
- > Front-end
 - Freemarker templating engine
 - AJAX - mootools
- > **Hibernate Search**

Deployed in the Cloud

- > Amazon Web Services
 - EC2, EBS, S3 & CloudFront
- > JBoss AS on CentOS/RHEL
 - CMS Admin tool
 - Light-weight front-end (Stripped down JBoss AS)
 - JOPR - JBoss management console
- > Load-balancing
 - Apache httpd, mod_cluster + DNS round-robin

Deployment





JavaOneSM

Thank You

Techniques for building
highly scalable Web sites
and Web applications



Overview of using Hibernate Search query projection

- > Hibernate Search allows you to return a subset of properties directly from the Lucene index
- > **Avoids a database hit**
- > Requirements
 - the properties projected must be stored in the index
`@Field(store=Store.YES)`
 - only simple properties of the indexed entity or its embedded associations

Hibernate Search query projection - APIs

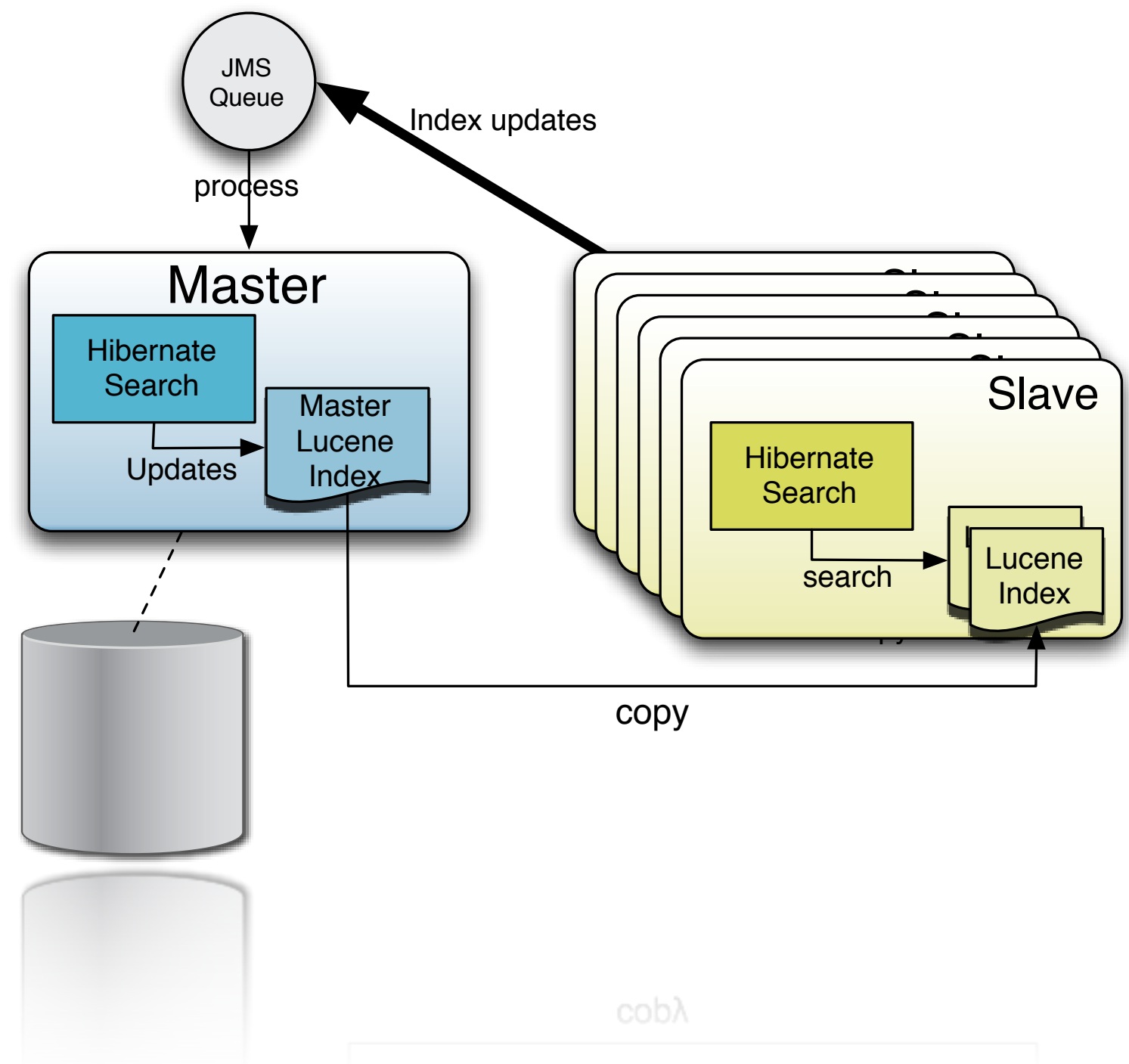
> Example - Result Transformer

```
org.hibernate.search.FullTextQuery query = s.createFullTextQuery( luceneQuery, Blog.class );  
  
query.setProjection( "title", "author.name" );  
  
query.setResultTransformer(  
    new StaticAliasToBeanResultTransformer( BlogView.class, "title", "author" )  
);  
  
List<BlogView> results = (List<BlogView>) query.list();  
for(BlogView view : results) {  
    log.info( "Blog: " + view.getTitle() + ", " + view.getAuthor() );  
}
```

- See `org.hibernate.transform.ResultTransformer` Interface for more details

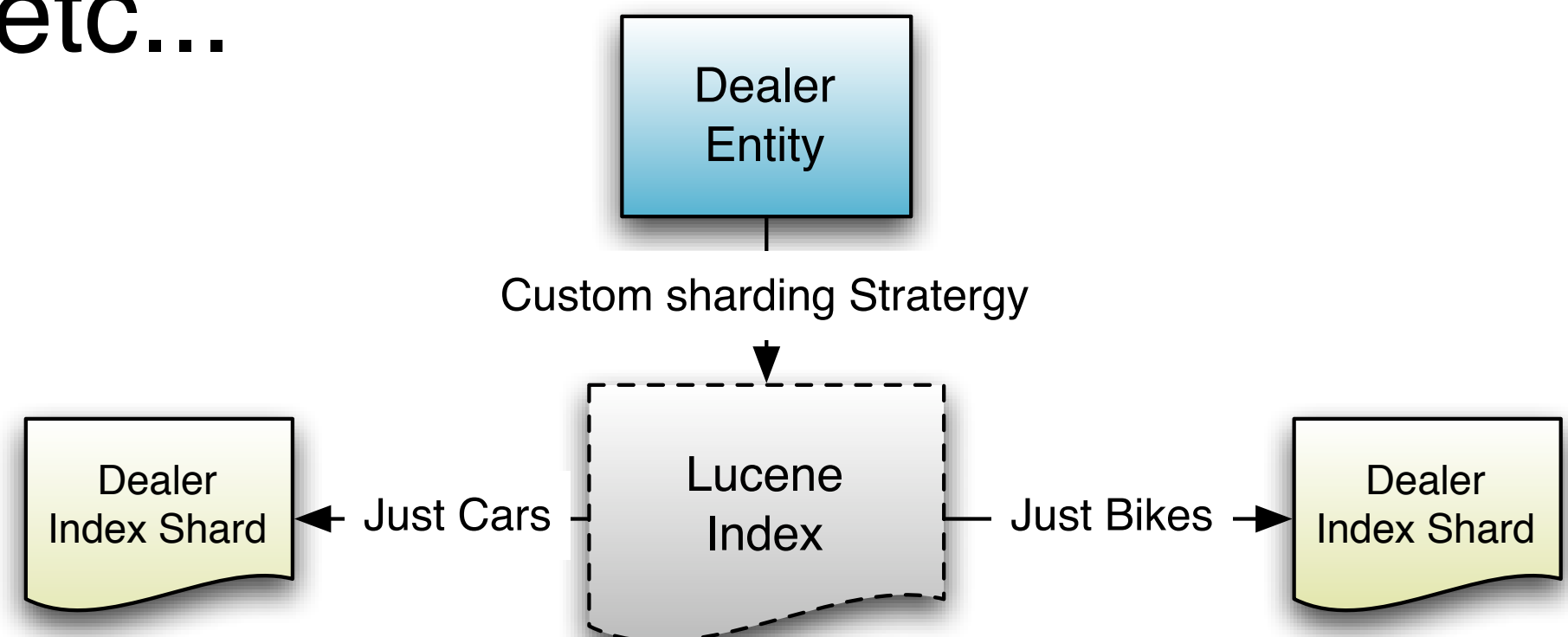
Overview of Hibernate Search index replication

- > Automatic replication
- > Local indexes
- > Updates delegated to a master
 - via JMS Queue
- > Can easily add more slaves



Overview of Hibernate Search index sharding

- > Allows you to index a given entity type into several sub indexes
 - default strategy uses hash of id field
- > Can Specify a custom sharding strategy
 - shard on a business field e.g geographic location, product category, etc...



Techniques for building applications that are cloud-ready

- > Break the architecture into small discrete pieces
 - separated CMS from content delivery
 - individual sites for Cars, Bikes etc...
 - JBoss micro-container
- > Independently deployable components
 - can deploy CMS across number of servers
 - mix and match site deployments

Take control of your cloud

> JOPR

- more than just a JBoss management console
- monitor OS, App Servers, Database and more
- pluggable agents with simple API

> EC2

- scriptable AMIs for rapid server configuration
- change an instances personality at runtime
- automate automate automate

So why Amazon Web Services?

> Flexibility

- easily add and remove instances
- scale on demand

> Play space

- can quick bring-up environments to experiment with
- production migration

> No lock-in

> Complete cloud offering

More Amazon Web Services

- > S3 - Simple Storage
- > Elastic Block Storage - EBS
 - fast persistence storage
 - mounted multiple volumes in RAID 0
 - snapshot backups to S3
- > CloudFront
 - content delivery network
 - used for static content images & video

Summary

> Hibernate Search

- unified programmatic model
- feels like Hibernate, search like Lucene

> Scalability

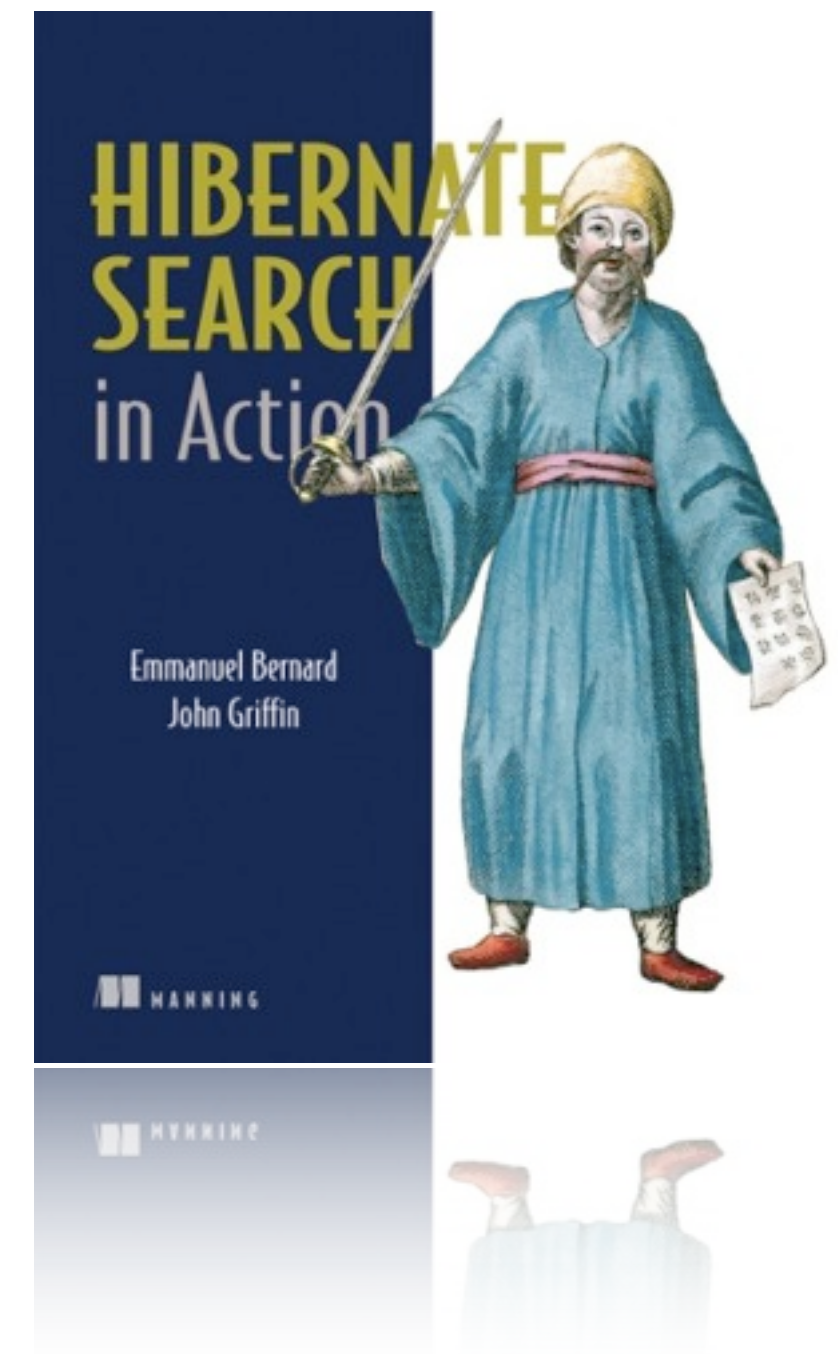
- avoid inessential database hits
- simple is better

> Simplicity in the Cloud

- design to scale out, not up!!!

Questions?

- > <http://search.hibernate.org>
- > Hibernate Search in Action (Manning)
- > <http://lucene.apache.org>
- > a.walker@base2services.com
- > emmanuel@hibernate.org





JavaOneSM

Thank You

Emmanuel Bernard

emmanuel@hibernate.org

Hibernate Search in Action - Manning

<http://search.hibernate.org>

<http://in.relation.to/Bloggers/Emmanuel>

Aaron Walker

a.walker@base2services.com

<http://blog.base2services.com>

