

Open Software Integrators, LLC



Practical EJB3



Introduction

- Andrew C. Oliver (“Andy” if you're not selling me V14GR4).
- What this talk is/isn't
- “Market Research” (aka lies)
- In a nutshell (EJB3/Spring/Guice)



What this talk is/isn't

- This talk is
 - A brief introduction to the most important stuff in a typical EJB3 application.
 - Practitioner to practitioner (no crackpot theory)
 - Focused on the most important stuff
- This talk is not
 - A complete introduction to every feature, nook and gee-whiz cranny.
 - An in-depth look.

- Spr
- M
- 92
- J2
- Sim
- the

Google Fight : Make this fight with googleFight spring +java VS EJB3 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.googlefight.com/index.php?lang=en_GB&w...

Latest BBC Headlines NASA Breaking News The Kernel Virtual ... NC HB 1587 c-training OpenDS

market statist... Vinny Carpent... InfoQ: Spring ... Find Jobs: Sea... Google Figh...

Googlefight

The classics
Funny fights
Fight of the month
Last 20 fights

Results on Google :

spring java 822,000 results	EJB3 731,000 results
---------------------------------------	--------------------------------

spring +java
EJB3
Make a fight

Dogfight Flight
Amazing Flying Adventure For The Air Combat Enthusiast. Buy Online!
www.XperienceDays.com

Watch Pacquiao vs Marquez
Watch Manny Pacquiao vs. Juan Manuel Marquez Live online
Watch-Sports-Live.com/Boxing

Stand Against HIV/AIDS
Join Magic Johnson as he fights HIV/AIDS in the black community.
www.IStandWithMagic.com

Brioude Internet raynette ABONDANCE

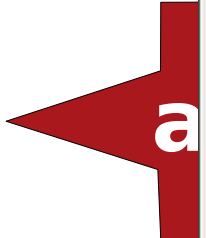
REFERENCEMENT 2.0
Olivier Andrieu
Le livre sur le référencement de sites web

mozbot! Search **spring +java** and **EJB3** on the web

This site is not affiliated with or sponsored by Google

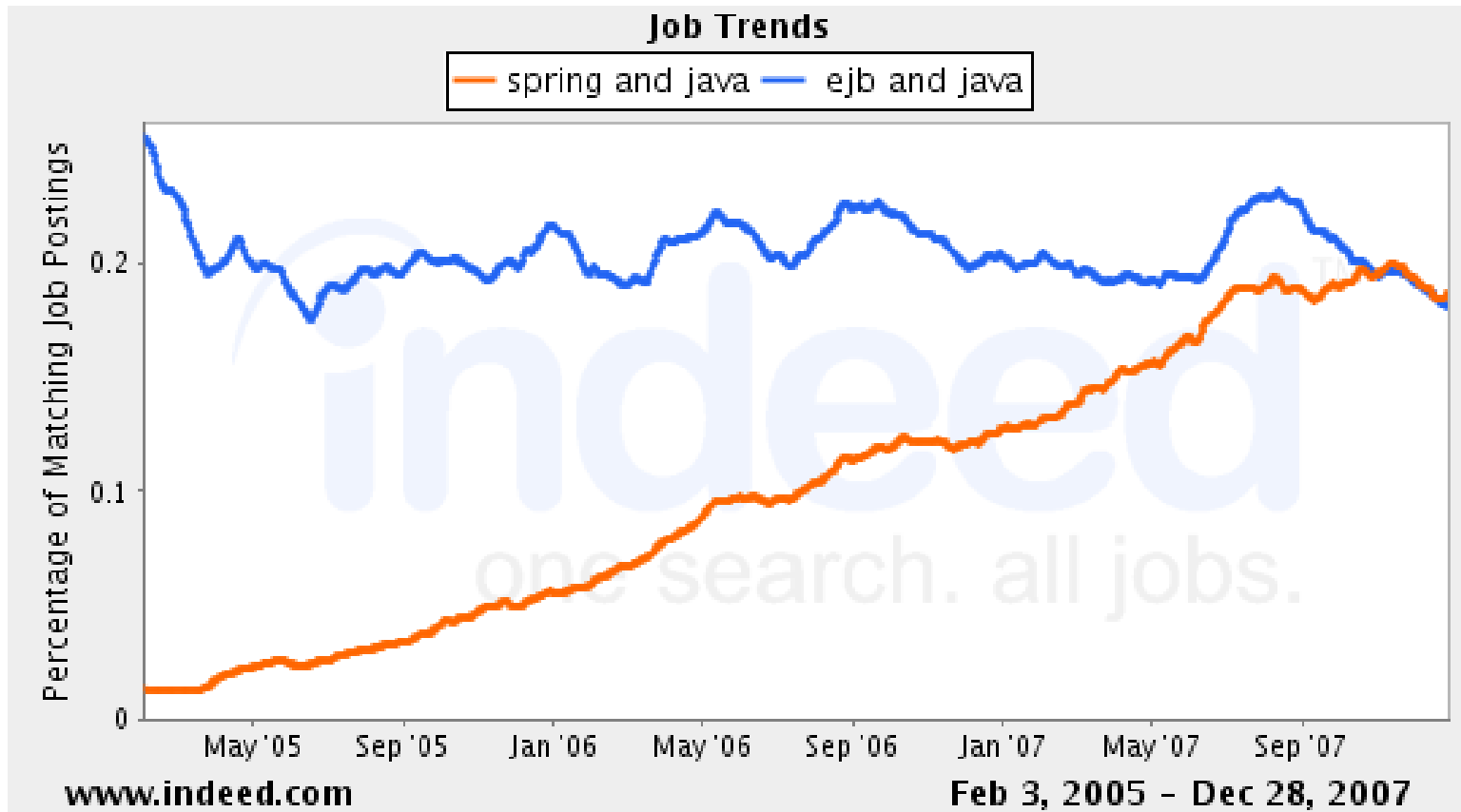
SE-Tools Network | SE-Inspector | SE-Keywords | SE-Check | SE-Rank | SE-Bid | Se-Spider | Se-Flash | Se-Fight | Meceoo.com

Done



Marketing...but...

- The indeed.com that the Spring guys used last year





Why compare

- As a “gun for hire” of last resort, I deal with everything. “If it don't make dollars then it don't make sense!”
- The market doesn't decide the best solution, but does decide what pays the best.
- If you have only one approach to solving a problem, you probably don't have the best approach much of the time.
- EJB is not an open or free specification all implementations are encumbered.



EJB3/Spring/Guice

EJB3

- annotation driven
- simple
- JCP “standard”
- multiple-vendor
- versioned with J2EE
- simple purpose

Spring

- annotation support
- full-featured
- open source
- single vendor*
- self versioned
- multi-purpose



Guice

- Framework rather than a framework+library
- The most verbose
- lack of mature common used
- No “auto wiring”



A brief introduction to transactional computing

- When you interact with nearly any commercial database you have a transaction.
- Even in “auto-commit” mode.
- The trick is the granularity.



Classic Banking Example

- Deposit

```
public void deposit(Account acct, BigDecimal
amount) {
    tx.begin();
    acct.credit(amount);
    acctDAO.persist(acct);
    tx.commit();
}
```



Classic Banking Example

- Withdrawal

```
public void withdrawl(Account acct, BigDecimal amount)
{
    tx.begin();
    if (acct.getMaxAvailable().compareTo(amount) < 0) {
        tx.rollback();
        throw new NSFException("Overdraft of more than
"+acct.getMaxAvailable()+" is not allowed");
    }
    acct.debit(amount);
    acctDAO.persist(acct);
    tx.commit();
}
```



Classic Banking Example

- Transfer

```
public void transferFunds(Account from, Account to,  
BigDecimal amount) {  
    tx.begin();  
    withdrawl(from, amount);  
    deposit(to, amount);  
    tx.commit();  
}
```



Transactional Computing

- What about concurrency?
 - If two threads modify the same row what happens?
 - Row-locking?
 - Multi-Version Concurrency/Snapshot Isolation?
- What about long running transactions (aka “Conversations”)?



Database locking

- MySQL – row locks for writes, no locks for reads.
- SQL Server 2005 – snapshot isolation mode, no locks for reads/write. Default mode: repeatable read == read locks.



What EJB3 is

- A framework for transactional systems.
- The API for
 - a remoting framework.
 - a messaging framework.
 - a persistence framework.



EJB3 is appropriate for

- EJB is generally appropriate for:
 - Web Applications
 - Client-Server applications
 - Normal B2B applications
- EJB is generally inappropriate for
 - Reporting/Analytical Systems
 - ETL (Extract, Transform, Load)
 - Batch Processing

**Java, JDBC,
etc are not
generally
appropriate
here either.**



Our Example

Organization

Book

Placement

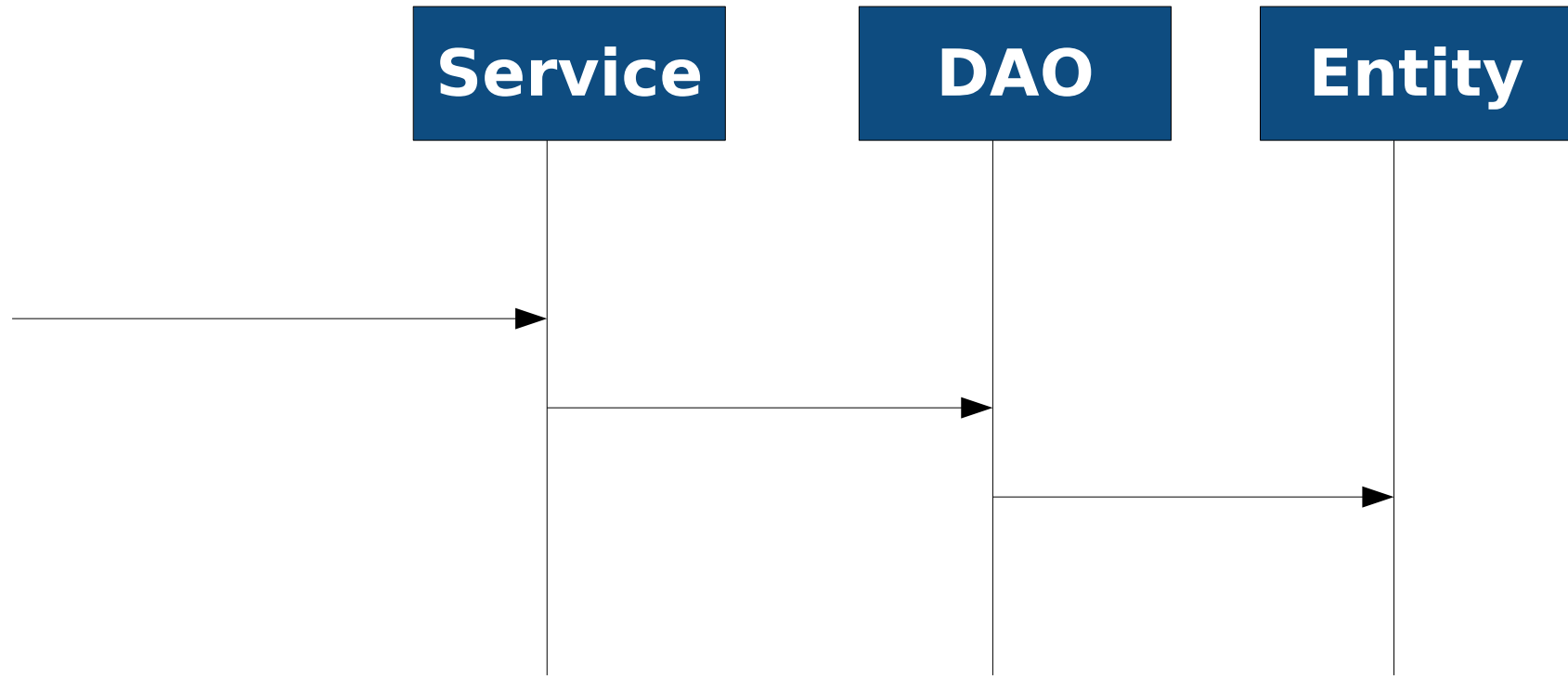
Reservation



Stateless Session Bean

- Transactional Demarcation
- Persistence Context management
- Dependency Injection

Basic Pattern





Service Bean

- start of the database transaction (demarcated with transaction attributes or default to REQUIRED)
- DAOs are injected here
- verbs like “transferMoney”



Service Bean

```
@TransactionAttribute(TransactionAttributeType.REQUIRES_NEW)
@TransactionAttribute(TransactionAttributeType.REQUIRED) //default
@EJB
BookBeanDAO bookDAO;
```



DAO

- Data Access Object
 - EntityManager is injected here
 - contains fetcher/finder methods which get exactly what you need
 - CRUD operations (Create Retrieve Update Delete) generally specific to managing one particular type of entity

DAO

```
@Stateless
public class BookBeanDAO implements
BookBeanDAOLocal {
    @PersistenceContext(unitName="publications")
    EntityManager session;

    ...

    public BookBean getBook(long bookId) {
        return (BookBean)
session.find(BookBean.class, bookId);
    }

    ...
}
```

DAO

```
public boolean bookExists(long orgId, String
type) {
    List count =
session.createNamedQuery("book.bookExists")
        .setParameter("org", orgId)
        .setParameter("type", type)
        .getResultList();
    return count != null && count.size() > 0 ?
((Integer)count.get(0)) > 0 : false;
}
```



Entity

- Annotated POJO
- CMP was inadequate
- Related to other Entities
- common types easy
- HQL is not harder than sql
- Named Parameters good
- version columns are great



Entity

```
@Entity @Table(name="rsrv_book")
public class BookBean implements
java.io.Serializable
{
    @Id @GeneratedValue(generator="uid_seq",
strategy= GenerationType.SEQUENCE)
    private long id;
    @ManyToOne
    @JoinColumn(name="org_id")
    @Cascade(value = { })
    private OrganizationBean organization;
    @Column(name="type")
    private String type;
```



Entity

```
@NamedQueries({  
...  
    @NamedQuery(name="book.bookExists",  
        query="select count(b) from BookBean b  
        where b.organization.id=:org and b.type=:type",  
        cacheable=true,  
        cacheMode=CacheModeType.NORMAL  
    )  
})
```



Generated Keys

- Preferred, business keys produce rigid schemas and are often imperfect.
- Do you need a sequential key?



UUID.java

```
public static UUID randomUUID()
{
    long lsb = r.nextLong();
    long msb = r.nextLong();

    lsb &= 0x3FFFFFFFFFFFFFFFFFL;
    lsb |= 0x8000000000000000L; // set top two
bits to variant 2
    msb &= 0xFFFFFFFFFFFF0FFF;
    msb |= 0x4000; // Version 4;
    return new UUID( msb, lsb );
}
```



UUIDGenerator.java

```
public Serializable
generate(SessionImplementor session, Object obj)
{
    return new StringBuffer(36)
        .append( format( getIP() ) ).append(sep)
        .append( format( getJVM() ) ).
append(sep)
        .append( format( getHiTime() ) ).
append(sep)
        .append( format( getLoTime() ) ).
append(sep)
        .append( format( getCount() ) )
        .toString();
}
```



UUIDGenerator.java

```
protected short getCount() {  
    synchronized(AbstractUUIDGenerator.class)  
    {  
        if (counter<0) counter=0;  
        return counter++;  
    }  
}
```



Why the Hibernate one is silly

$$p(n) \approx 1 - e^{-n^2/2 \cdot x}$$

n	probability
68,719,476,736 = 2^{36}	0.000000000000000004 (4×10^{-16})
2,199,023,255,552 = 2^{41}	0.000000000000000004 (4×10^{-13})
70,368,744,177,664 = 2^{46}	0.000000000004 (4×10^{-10})



Why the Hibernate one is silly

- One's annual risk of being hit by a meteorite is estimated to be one chance in 17 billion.
- In other words, only after generating 1 billion UUIDs every second for the next 100 years, the probability of creating just one duplicate would be about 50%.

-wikipedia



Why the Hibernate one is silly

- Windows clock resolution on Java is between 55 ms and 10ms (version of Windows, num processors).
- Clock calls are heavier
- but The IP probably protects from the worst of it so long as the network config is “normal”
- But this is synchronized when it doesn't really need to be.



Fetching

- Hibernate 3.x/JPA introduced HQL/JPA-QL fetch strategies
- `select XYZ from XYZ join fetch XYZ.child where ...`
- DAOs should do this on a use-case basis



MVCC/Optimistic Locking safely

- Database version columns
- update xyz set foo=bar, set version=origversion+1 where id=123 and version= origversion
- Difficult to do manually safely



Caching

- Design with this in mind, but do it last
- Consider cache invalidation
- Distributed Caches are probably not worth the trouble in Hibernate...

Cache...



Session



2nd Level



**Queries/
relationships**



Database



Retrofitting

- get your build/deploy right...
- Start with the entities, if you can map them to your schema then you can get moving.
- Next add DAOs, give them remote interfaces at first and do simple junit tests.
- Next move all the code from factories, tags, jsp pages, etc to methods in your service and remove the old methods...



Pain Points

- Try showing the SQL:
 - is it joined, subselect, fetched (no n+1)
 - explain plan, is it indexed?
 - can it outer join, should it?
- Try a thread dump
 - are we waiting on locks?
- Try the GC log, is memory full...
- Try caching
- Is your database locking (understand)?



EJB 3.1

use by
injection or
jndi lookups

- No Interface View

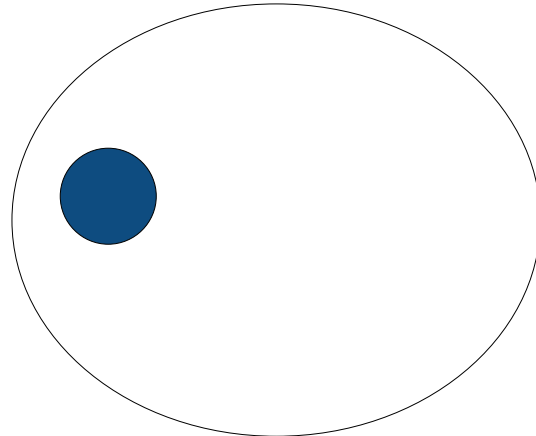
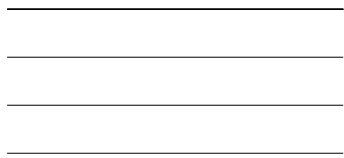
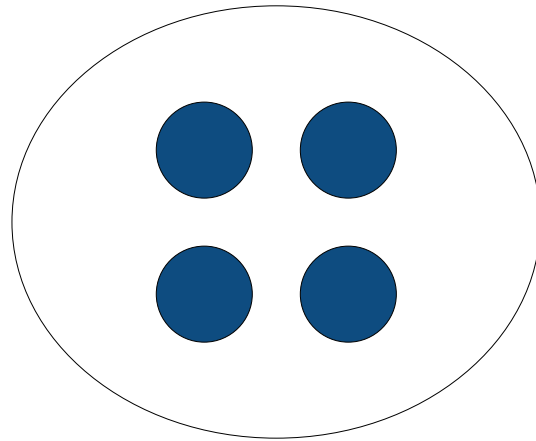
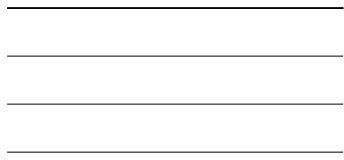
```
@Stateless
public class StockTrader {
    public void buy(Stock stock, int quantity) {...}
    public void sell(Stock stock, int quantity) {...}
}
```

```
@EJB StockTrader trade;
```

```
StockTrader trade =
(StockTrader)jndi.lookup("StockTrader/local");
```

EJB 3.1

- Singletons



- Stateless Session Beans: 4 threads == 4 beans
- Servlet thread model: 4 threads == 1 servlet



EJB 3.1

- Concurrency

```
@Singleton  
public class MyBean {  
    @ReadOnly public int getStuff() {...}  
    @Write public void setStuff(int i) {...}  
}
```

- Wrapper around `java.util.concurrent`



EJB 3.1

- Transactional Concurrency / Default modes

```
enum ConcurrencyMode {  
    SINGLE_THREADED,  
    MULTI_THREADED,  
    READ_WRITE,  
    WRITE_READ,  
    TRANSACTIONAL_READ_WRITE,  
    TRANSACTION_WRITE_READ  
}  
  
@Singleton(concurrency=WRITE_READ)  
public class myBean {...}
```



EJB 3.1

- Asynchronous

```
@Asynchronous  
public void initiateTask(...)
```



EJB 3.1

- WAR packaging of EJB3 beans
 - No more EAR/EJB-JAR if it is not desired
 - Shared naming environment
- Timer service enhancements
- Web Service enhancements (stateful endpoints)



JPA 2.0

- If you're using Hibernate, you're not missing much:
 - No public EDR is available
 - Ordered Lists – Supported by Hibernate
 - Unidirectional 1-M support w/o join table
 - Criteria Queries
 - Detached Objects (specified)
 - Validation
 - Embedded classes



Resources

- http://www.redhat.com/docs/manuals/jboss/jboss-eap-4.2/doc/Getting_Started/Sample_JSF_EJB3_Application.html
- <http://www.oracle.com/technology/tech/java/oc4j/ejb3>
- <http://www.oracle.com/technology/tech/java/ejb30.html>
- <http://hibernate.org>
- <http://www.jboss.com/docs/trailblazer> -when it works



Open Software Integrators

- Focus on
 - Enterprise scalability and performance.
 - Integration between Open Source and Proprietary platforms
- Specific competencies in:
 - Java, JBossAS, JBoss Portal, Hibernate, etc.
 - Linux, Apache, Postgresql, MySQL, etc.
 - Training, Consulting, Mentoring, Support...
- Hiring...



Email

- acoliver@gmail.com
- Temporary until we move our servers from ThePlanet.



Bibliography

- EJB 3.1:
 - <http://bill.burkecentral.com/category/ejb-31/>
 - <http://www.theserverside.com/tt/articles/article.tss?l=NewFeaturesinEJB3-1>
 - http://blogs.sun.com/kensaks/entry/ejb_3_1_presentation_at
 - <http://jcp.org/aboutJava/communityprocess/edr/jsr318/index.html>
- JPA 2.0:
 - http://weblogs.java.net/blog/davidvc/archive/2007/05/javaone_session.html



Bibliography

- SQL Server
 - <http://www.mssqlcity.com/Articles/Adm/SQL70Locks.htm>
 - <http://www.informit.com/articles/article.aspx?p=26890>
 - <http://msdn2.microsoft.com/en-us/library/ms345124.aspx>
- UUID
 - <http://en.wikipedia.org/wiki/UUID>
 - <http://repository.jboss.com/hibernate/3.1beta3jboss/src/hibernate-3.1-src.zip>