



And the Winner is ...

Java User Group Switzerland, Bern, 23.05.2012

Simon Martinelli / @simas_ch

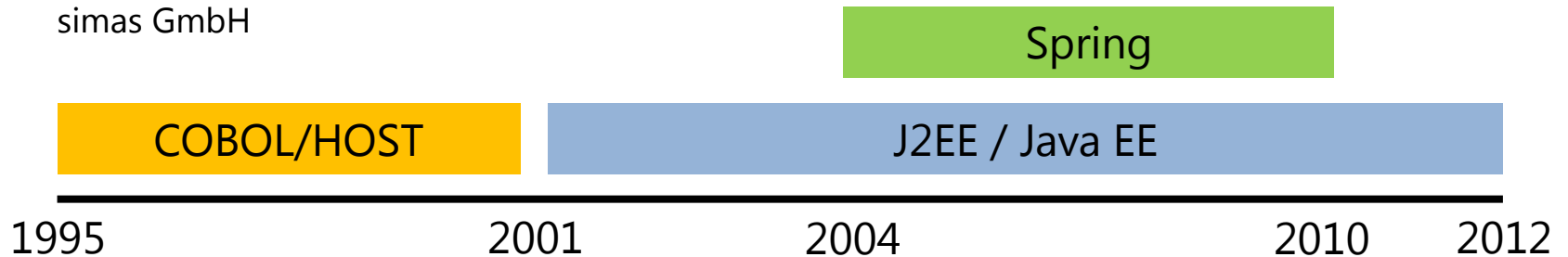
Agenda

- Introduction
- Dependency Injection
- Differences
- Conclusion

About me

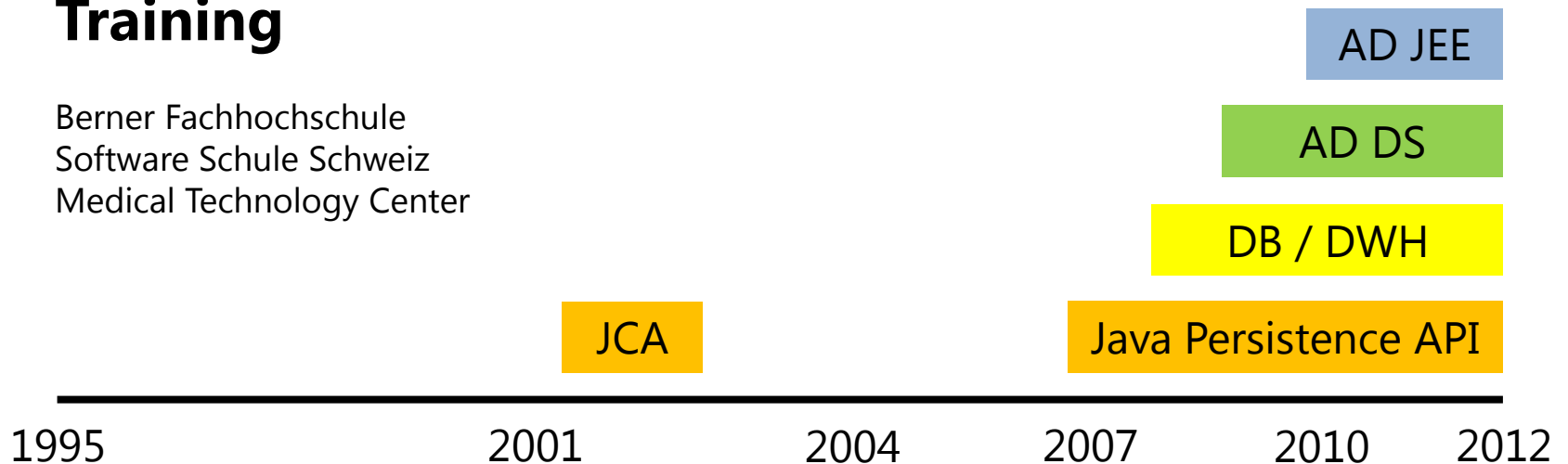
Development

SBB
ACS
simas GmbH



Training

Berner Fachhochschule
Software Schule Schweiz
Medical Technology Center



AD JEE = Architektur und Design Java EE / AD DS = Architektur und Design verteilter Systeme



Introduction



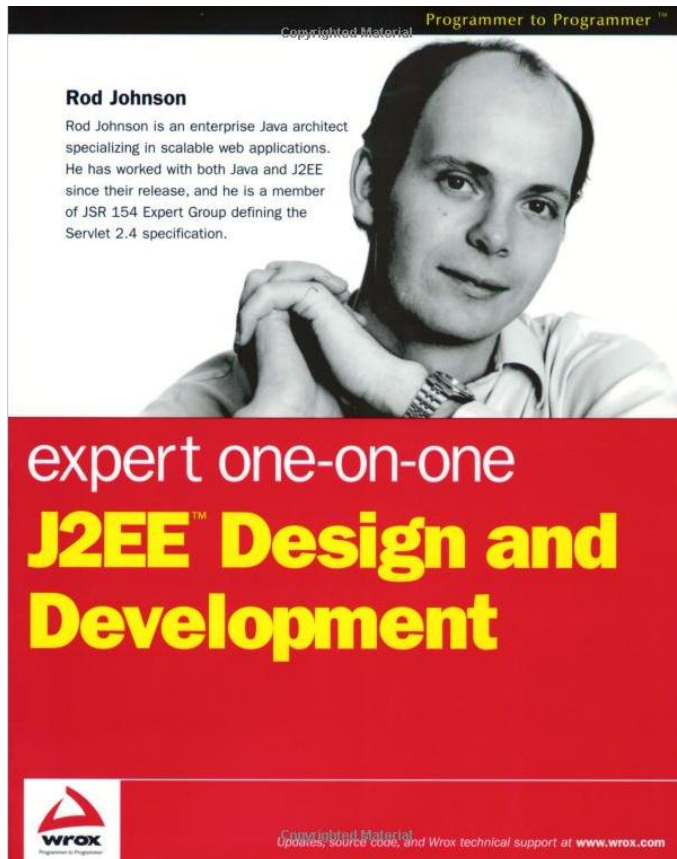
Poll on JAXenter in November 11

- Spring oder Java EE?
 - Ich bevorzuge Java EE (46%)
 - Ich bevorzuge Spring (34%)
 - Je nach Anwendungsfall das eine oder das andere (15%)
 - Es gibt bessere Alternativen (5%)

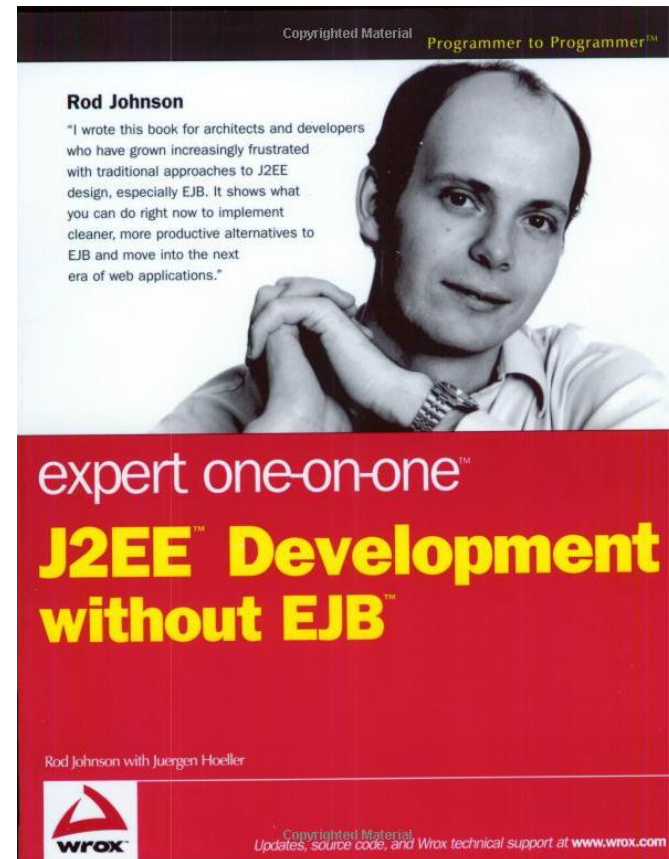
377 Teilnehmer

The Books

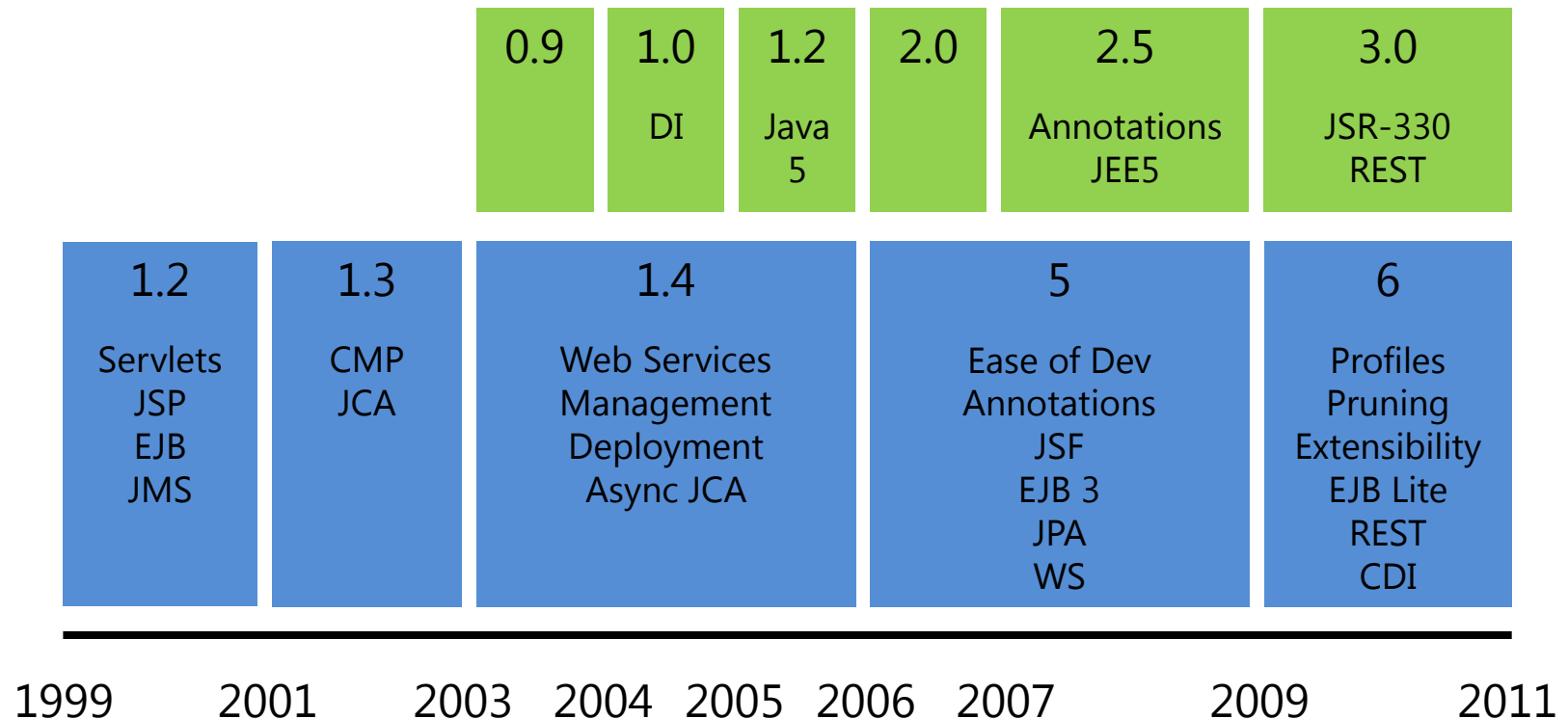
2002



2004



History



Spring Framework

J2EE / Java EE

Spring vs. J2EE

Java Enterprise Edition (JavaEE)

EJB

JCA

JPA

JSF

JSP

JNDI



ORACLE®



Spring Framework

POJO

POJI

IoC

DI

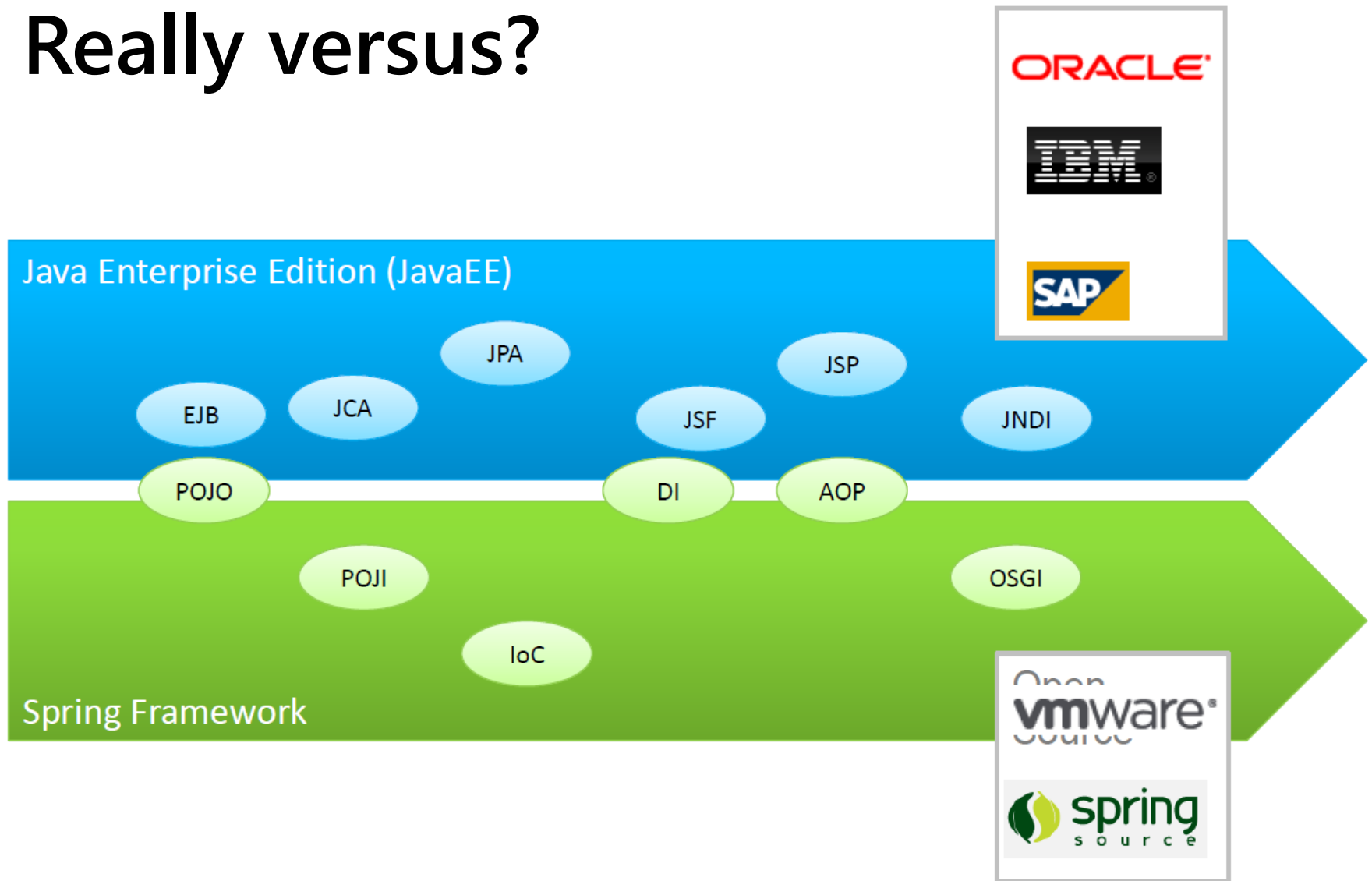
AOP

OSGI

Open
Source



Really versus?



Some Quotes

- «Frameworks like Spring are really just a bridge between the mistakes of the J2EE past and the success of Java EE future»

The age of frameworks is over, Cameron McKenzie

- «Due to Springs early success and adaption, Java EE is pushed to greatly simplify the Java EE programming model...»

Spring vs. Java EE and why I don't care, Eberhard Wolff



Dependency Injection

Java EE 6 DI



JSR 318: Enterprise JavaBeans 3.1

JSR 299: Contexts and Dependency Injection for the Java EE platform

JSR 330: Dependency Injection for Java

CDI Injection



The Bean

```
public class MyService {  
}
```

```
public class AnotherService {  
  
    @Inject  
    private MyService myService;  
}
```

The Injection Point
*Could also be a
constructor or a setter*

CDI Qualifiers



Two implementations of
the same interface

@Asynchronous

```
public class AsyncService  
    implements MyService {  
    ...  
}
```

@Synchronous

```
public class SyncService  
    implements MyService {  
    ...  
}
```

```
public class AnotherService {  
  
    @Inject @Synchronous  
    private MyService myService;  
}
```

Injection must be
qualified

CDI Producers



```
public class MyService {  
    public MyService(A a) {  
        // ...  
    }  
}
```

Producer Method to
create instance
A will be injected!

```
public class MyServiceProducer {  
  
    @Produces  
    public MyService createMyService(A a) {  
        return new MyService(a);  
    }  
}
```

```
public class AnotherService {  
  
    @Inject  
    private MyService myService;  
}
```

CDI Interceptors



The Interceptor

```
@Interceptor  
@Transactional  
public class TransactionInterceptor {  
    @AroundInvoke  
    public Object manageTransaction(InvocationContext ctx) {  
        // ...  
    }  
}
```

The Annotation

```
@InterceptorBinding  
public @interface Transactional {  
}
```

```
public class MyService {  
    @Transactional  
    public void foo() {  
        // ...  
    }  
}
```

A Transactional
Method

CDI Events



```
public class MyService {  
    @Inject  
    private Event<LoggedInEvent> loggedInEvent;  
  
    public void login(String username, String password) {  
        loggedInEvent.fire(new LoggedInEvent(username));  
        // ...  
    }  
}
```

The Event Object

Fire the Event

```
public class LoggedInObserver {  
  
    public void afterLogin(@Observers LoggedInEvent event) {  
        // ...  
    }  
}
```

The Listener
Called Synchronous!

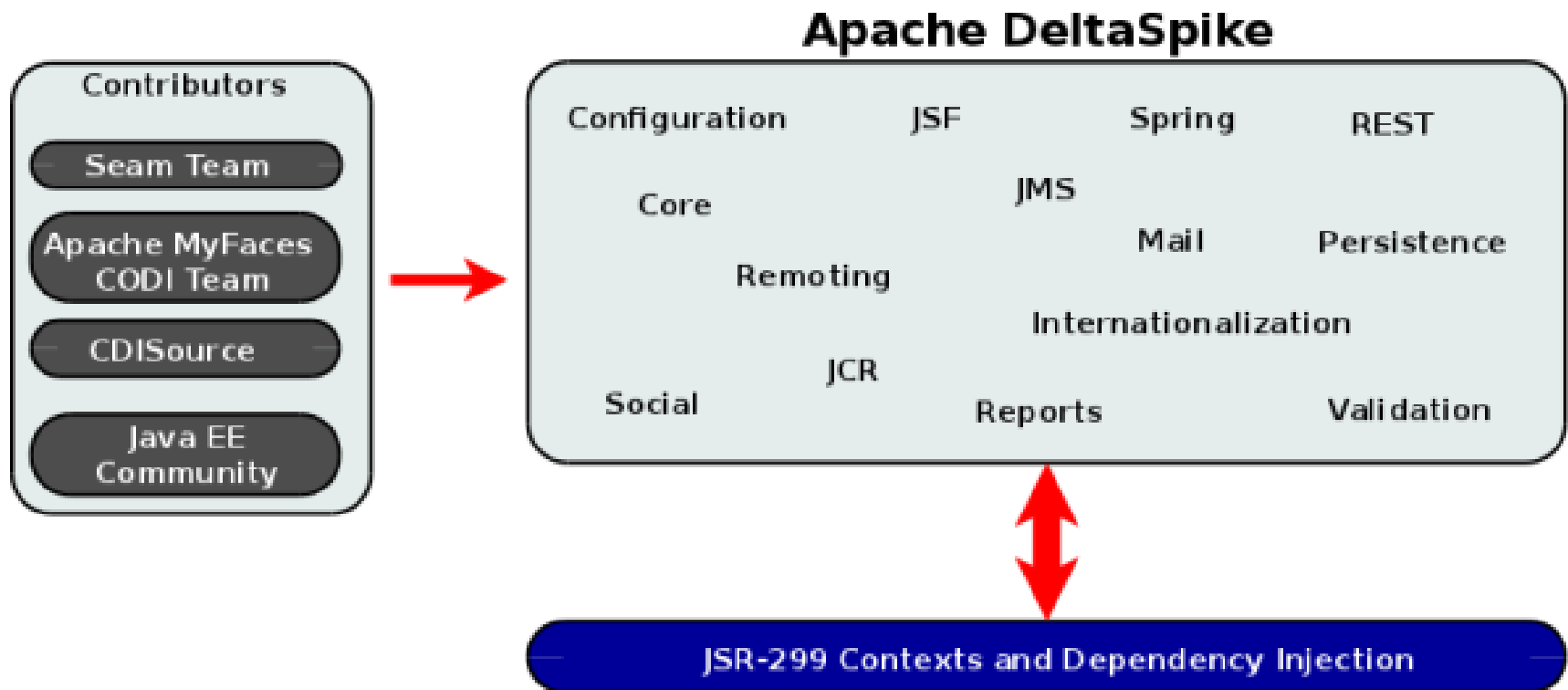
Asynchronous Events



```
@Stateless  
public class LoggedInObserver {  
  
    @Asynchronous  
    public void afterLogin(@Observers LoggedInEvent event) {  
        // ...  
    }  
}
```

The Listener as EJB
Called Asynchronous!

CDI Extensions



Spring DI: XML based

```
<beans>
  <bean id="myService" class="service.MyService"/>
  <bean id="anotherService" class="service.AnotherService">
    <property name="myService" ref="myService"/>
  </bean>
</beans>
```

```
public class AnotherService {

  private MyService myService;

  public void setMyService(MyService myService) {
    this.myService = myService;
  }
}
```

The Injection Point
*Could also be a
constructor*

Spring DI: Annotation based

```
<beans>
  <context:annotation-config/>
</beans>
```

Spring Annotations

```
public class AnotherService {
  @Autowired
  private MyService myService;
}
```

The Injection Point
*Could also be a
constructor or a setter*

JSR-330

```
public class AnotherService {
  @Inject
  private MyService myService;
}
```

JSR-250

```
public class AnotherService {
  @Resource
  private MyService myService;
}
```

Spring DI: Java based

@Configuration

```
public class AppConfig {
```

@Bean

```
    public MyService myService() {  
        return new MyService();  
    }
```

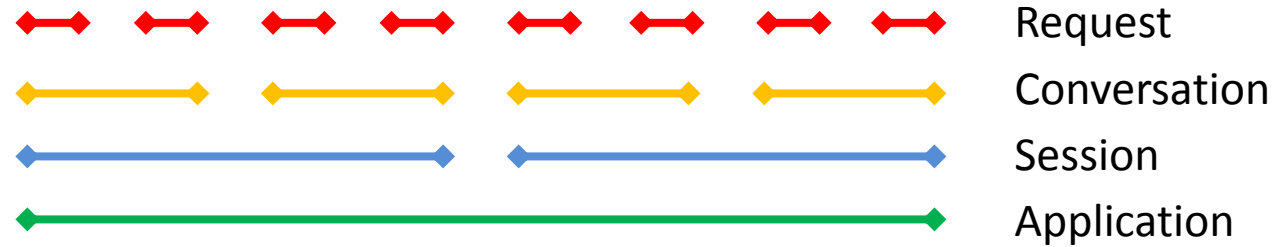


```
<beans>
```

```
    <bean id="myService" class="service.MyService"/>
```

```
</beans>
```

Scopes

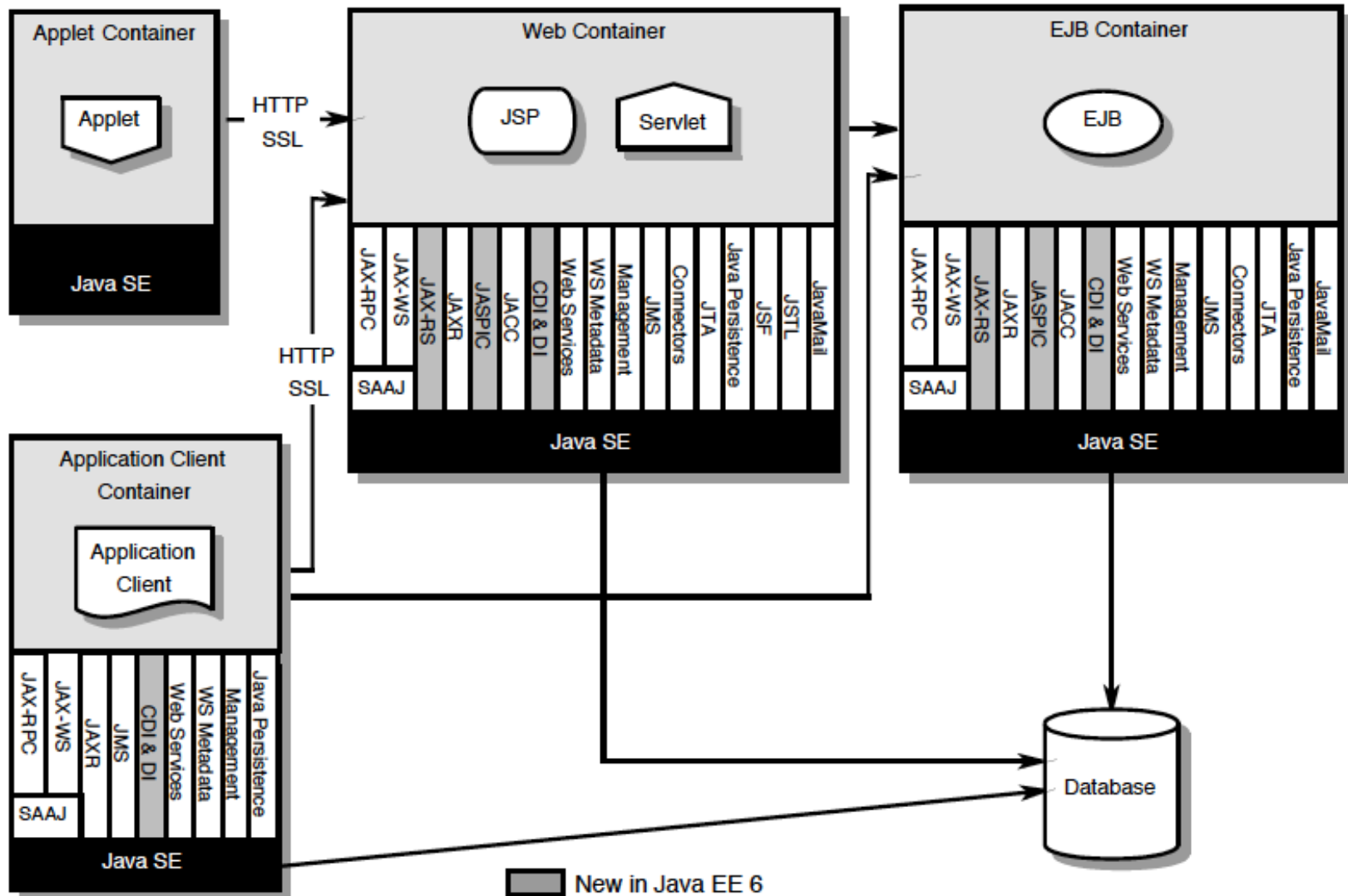


Java EE		Spring
EJB		General
Stateful	=	Prototype
Stateless	x	
Singleton	=	Singleton
CDI		Web
RequestScoped	=	Request
ConversationScoped	(=)	Spring Web Flow
SessionScoped	=	Session
ApplicationScoped	=	Singleton
Dependant	x	
	x	Global Session (only for Portlets)

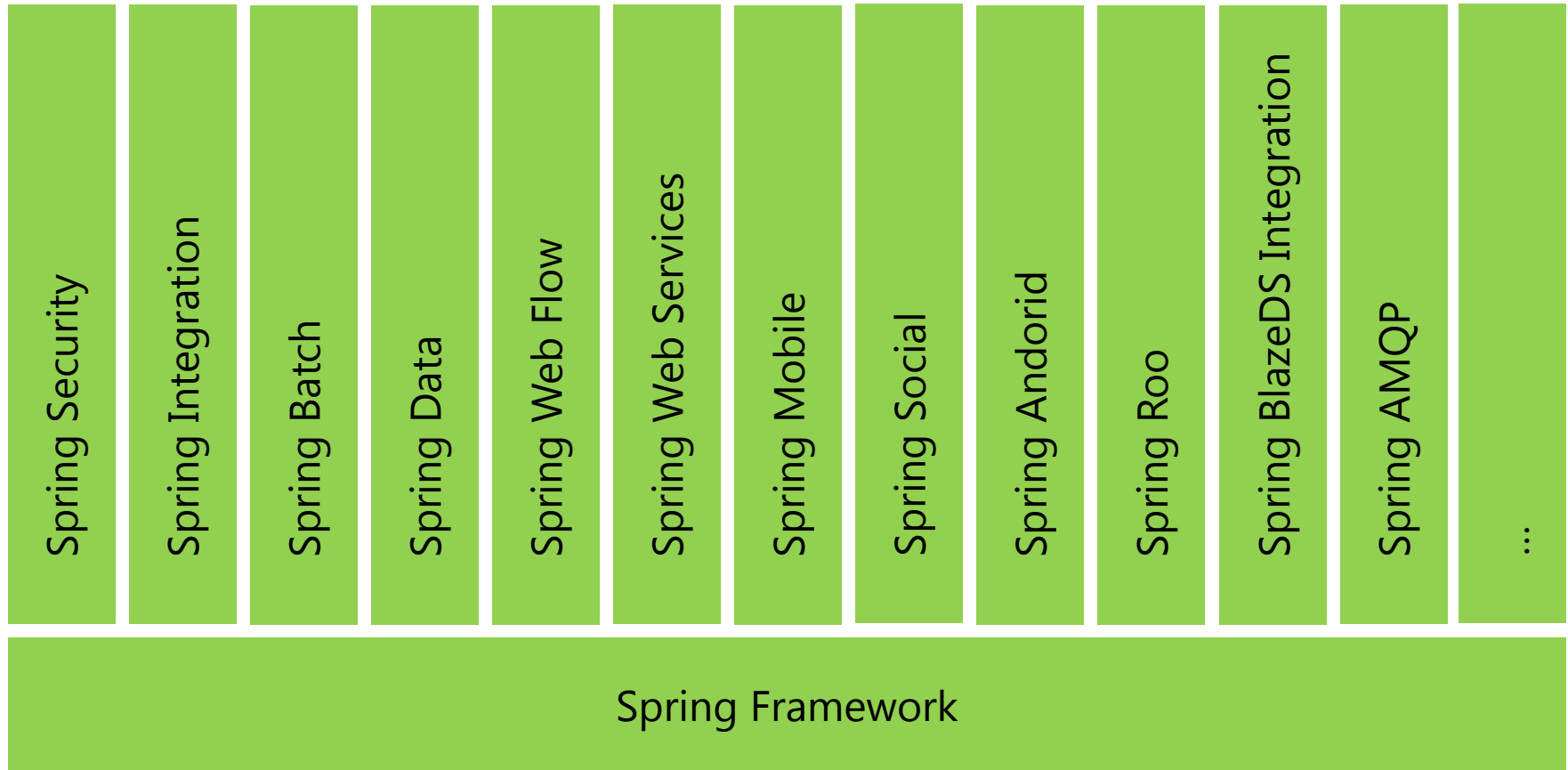


Differences

Java EE is a Specification



Spring is an Ecosystem



Spring uses Java EE

- Do you really need a wrapper around these APIs?

JPA

JCA

JMS

Mail

JMX

...

Defaults



```
@Stateless
public class MyService {

    public void foo() {
        ...
    }
}
```

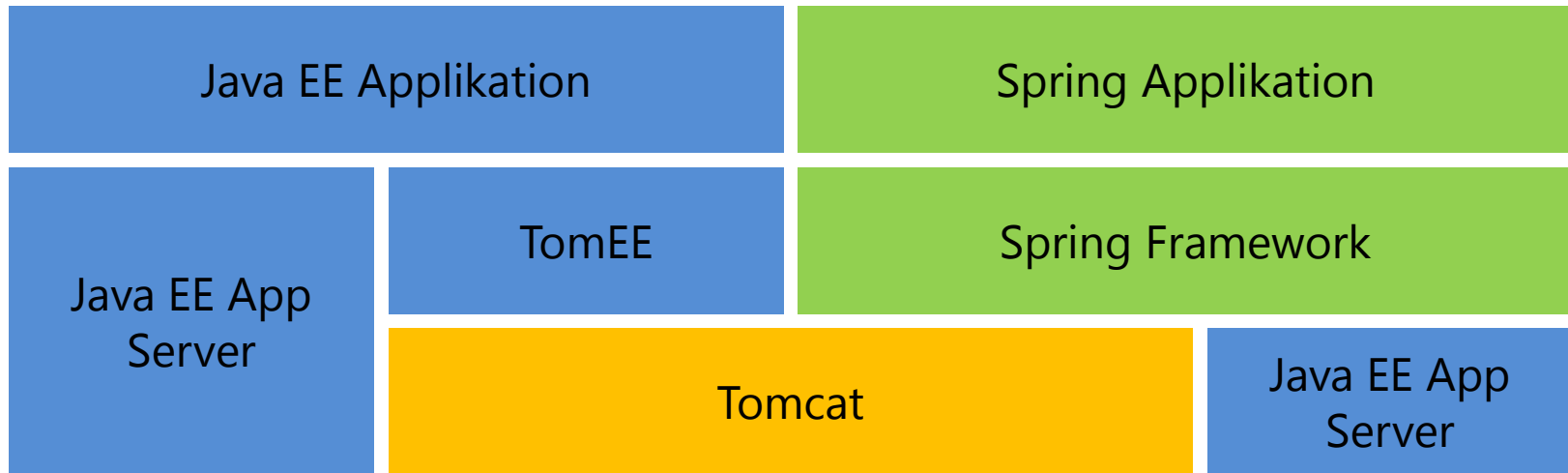


```
@Service
public class MyService {

    @Transactional
    public void foo() {
        ...
    }
}
```

Defaults to
`@TransactionAttribute(REQUIRED)`

Deployment



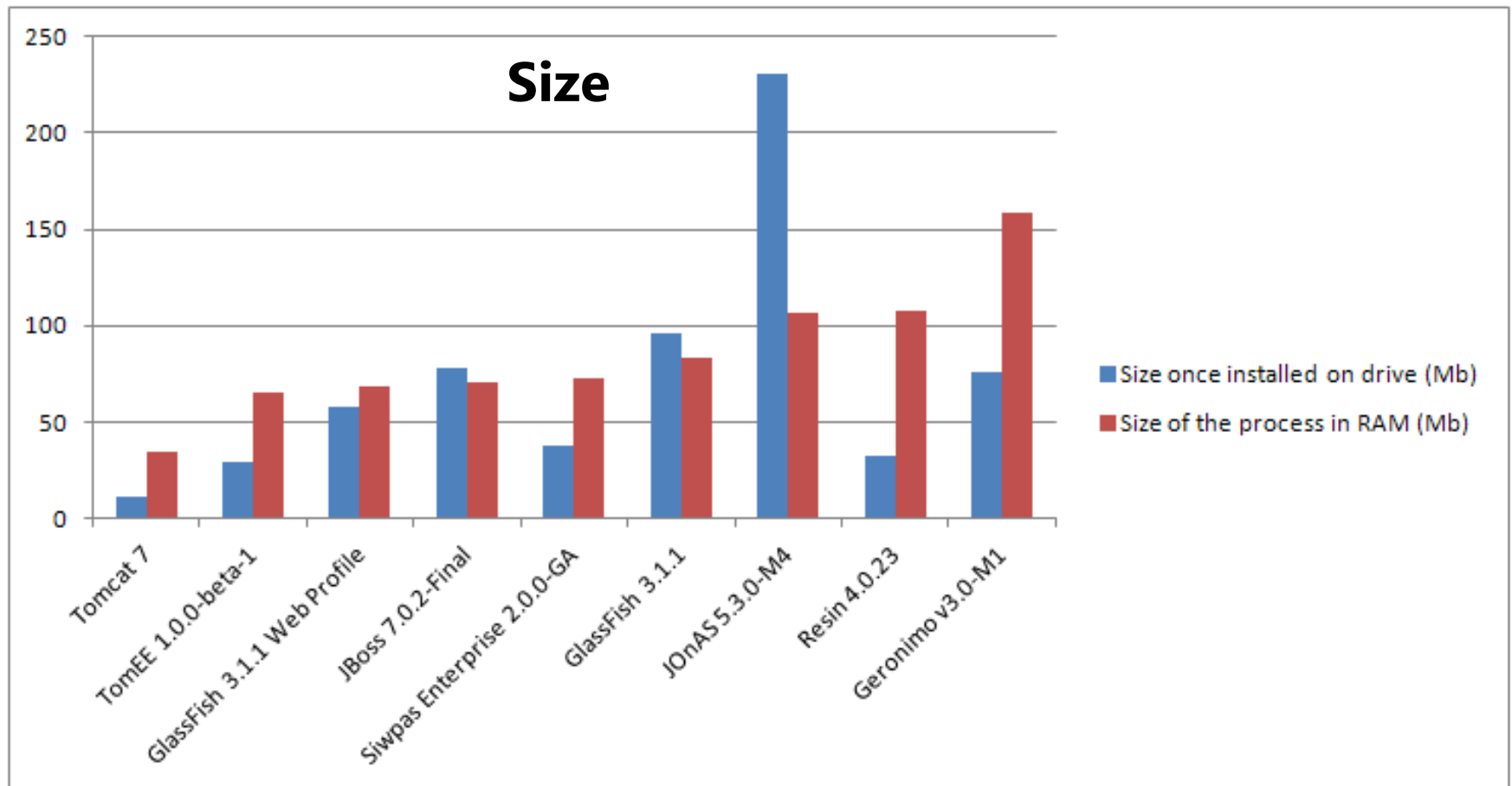
TomEE

CDI	Apache OpenWebBeans
EJB	Apache OpenEJB
Javamail	Apache Geronimo JavaMail
JPA	Apache OpenJPA
JSF	Apache MyFaces
JTA	Apache Geronimo Transaction

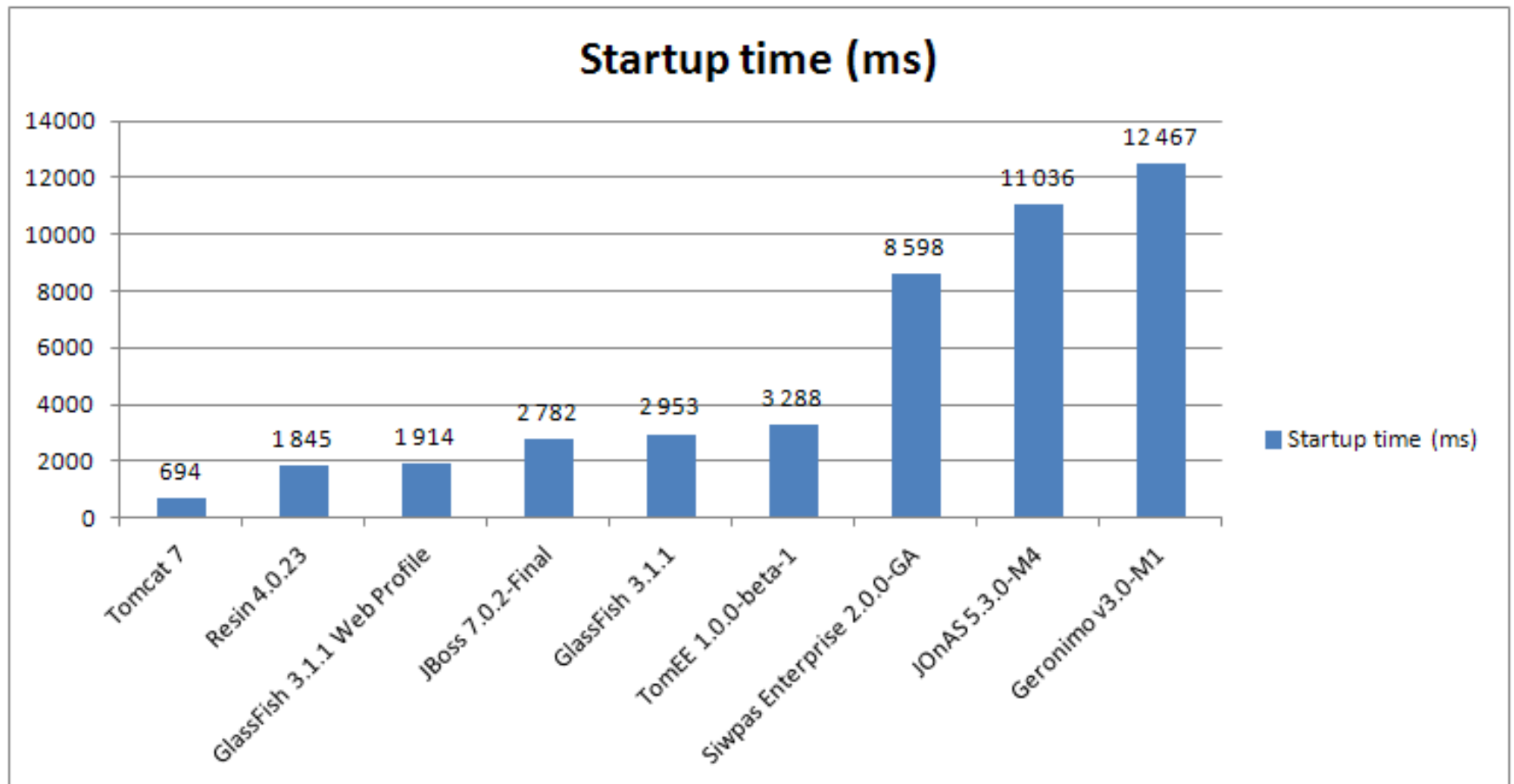
TomEE Plus

Connector	Apache Geronimo Connector
JMS	Apache ActiveMQ
Web Services	Apache CXF

Are App Servers still heavy?



Are App Servers still slow?

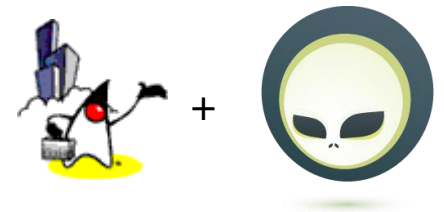


Interoperability

- Web
 - Spring Beans can be used with JSF
- CDI
 - Integration through Spring Bridge to CDI
- Metadata
 - Spring supports Java EE annotations
 - i.e. `@PersistenceContext`

Integration Testing

```
@RunWith (Arquillian.class)  
public class MyTest {  
    @EJB  
    private MyService myService;  
}
```



www.jboss.org/arquillian

```
@RunWith (SpringJUnit4ClassRunner.class)  
@ContextConfiguration  
public class MyTest {  
    @Autowired  
    private MyService myService;  
}
```



Summary



Topic	Java EE	Spring Framework
Framework	Specification based	Ecosystem
Defaults	Conventions over Configuration	No Default Behavior
Dependency Injection	CDI	Spring Container JSR 330 but not CDI!
Transactions	EJB	AOP / Annotations
Web Framework	JSF	Spring Web MVC
AOP	Interceptors	XML AspectJ
Integration Testing	Arquillian (non standard)	Spring Test Framework
Deployment	Part of the Platform	Part of the Application
Independency	Several Vendors	VMware



Conclusion

What is missing in Java EE?



ACL based Security

- Spring Security
- No Java EE support planned

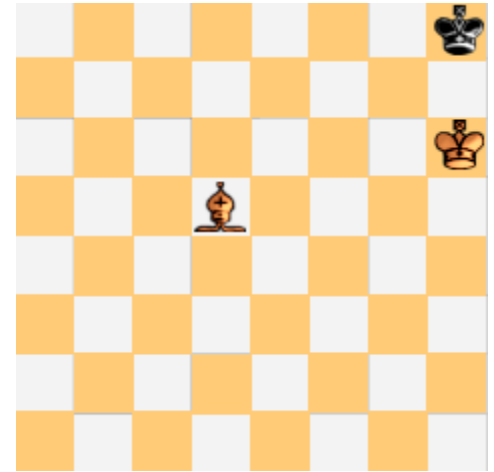
Batch

- Spring Batch
- JSR-352

NoSQL (Big Data)

- Spring Data
- No Java EE support planned

It's a Draw!



- Java EE 6 is a mature, easy to use framework inspired by Spring
- Unfortunately a lot of companies are still using application servers < Java EE 6
- Spring offers solutions for common problems like ACL based Security, Batch or NoSQL

The Future: Java EE 7



Clean Up

- JSF and CDI

Simplification

- JMS

Cloud

- Multitenancy

Enhancements

- CDI Java SE Bootstrap and JPA Integration

Batch

- JSR-352 Java Batch



Q&A