08/12/2009

Paris JUG

www.parisjug.org

www.parisjug.org

Xebia zenika OBJET DIRECT NOVEDIA VISION SFEIR Fast Connect OXiane valtech

08/12/2009

Java EE 6 & Spring 3.0

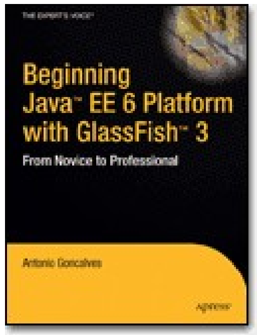
Antonio Goncalves

Michael Isvy

www.parisjug.org

Antonio Goncalves

- Freelance software architect
- Former BEA consultant
- Author (Java EE 5 and Java EE 6)
- JCP expert member
- Co-leader of the Paris JUG
- Les Cast Codeurs podcast
- Java Champion


Beginning Java EE 6 Platform with GlassFish 3
From Novice to Professional
Antonio Goncalves
Apress

Michael Isvy

- **Trainer and Consultant at SpringSource**
  - Trained 700+ people on Java related technologies
- **Open-source contributor**
  - Spring projects
- **Technical writer**
  - JournalDuNet, programmez!...

Get some information about the new release of Java EE and Spring and see where they fit together

Java EE 6 is out the 10th of December

Spring 3.0 in a few weeks

Disclaimer : this is not a technical presentation

First, let's talk about history

- **How did it all start?**
  - No EE standard

- **That was in 1999...**
  - No open source at the time
  - Home made framework
  - No logs, no transaction, no XML parser....

Era 1 : The Fat Boys

And then

- **Innovation came**
  - But not really from the Fat Boys

- **In 2002-2004...**
  - The open source rise

Era 2 : The amazones

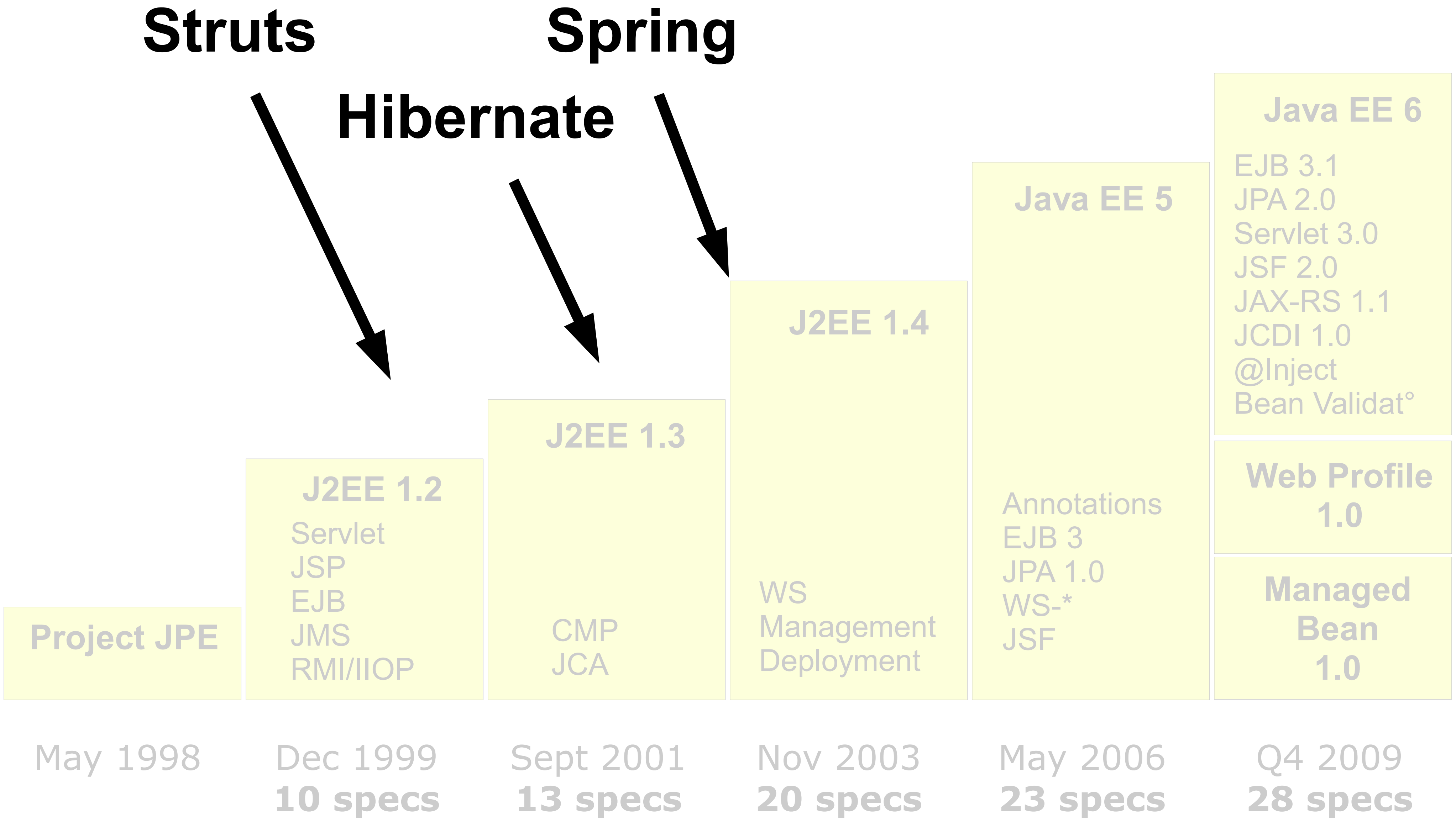
- Open-source projects

From J2EE to Java EE

You already know that

	Enterprise Application	Robust Scalable	Web Services	Ease of development	Ease of development (web)
Project JPE	**J2EE 1.2**	**J2EE 1.3**	**J2EE 1.4**	**Java EE 5**	**Java EE 6**
	Servlet JSP EJB JMS RMI/IIOP	CMP JCA	WS Management Deployment	Annotations EJB 3 JPA 1.0 WS-* JSF	EJB 3.1 JPA 2.0 Servlet 3.0 JSF 2.0 JAX-RS 1.1 JCDI 1.0 @Inject Bean Validat° **Web Profile 1.0** **Managed Bean 1.0**
May 1998	Dec 1999 **10 specs**	Sept 2001 **13 specs**	Nov 2003 **20 specs**	May 2006 **23 specs**	Q4 2009 **28 specs**

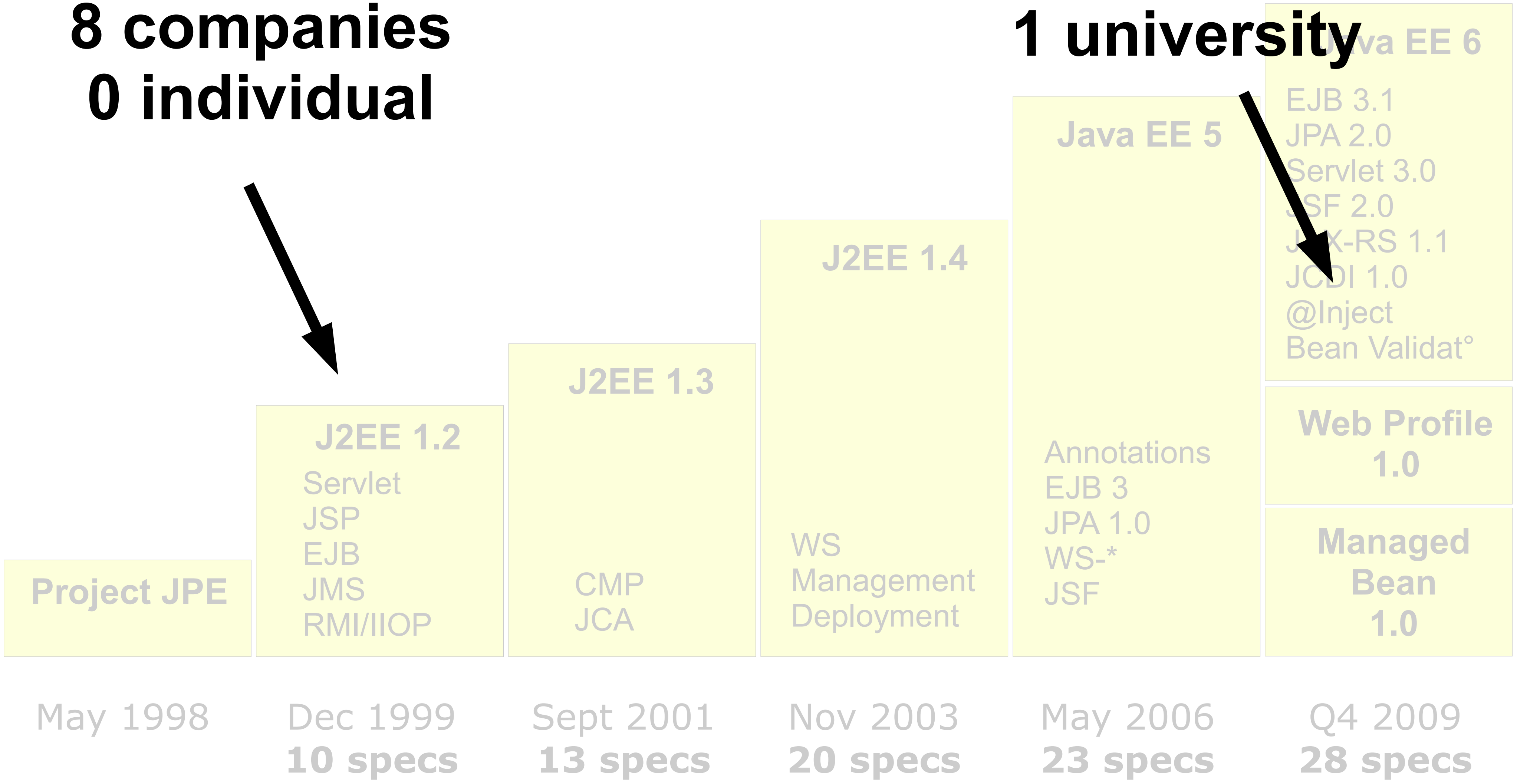
When did OS arrive ?



The expert groups

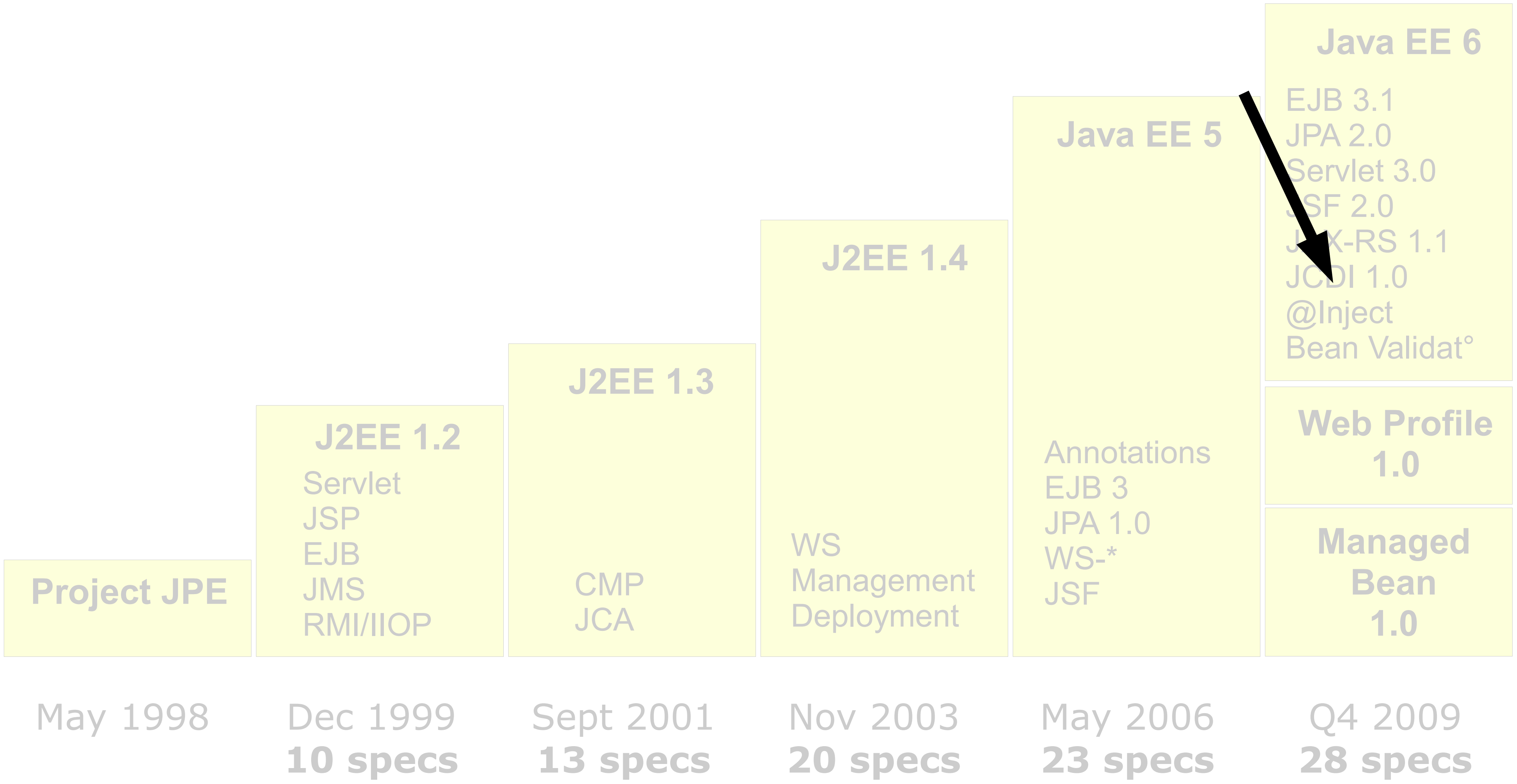
8 companies
0 individual

15 companies
10 individuals
1 university



And they all do OS

15 companies

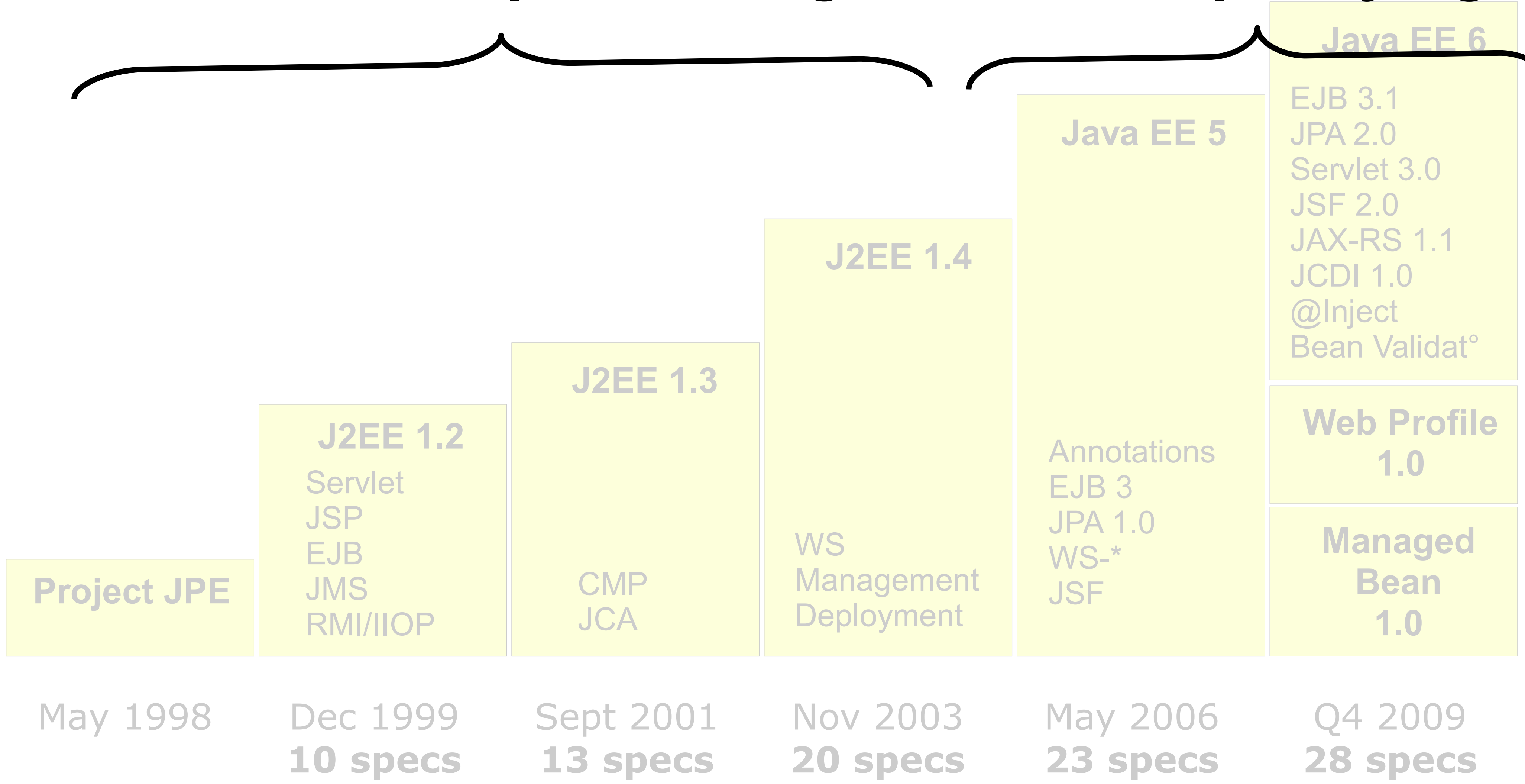


Project JPE	J2EE 1.2	J2EE 1.3	J2EE 1.4	Java EE 5	Java EE 6
	Servlet JSP EJB JMS RMI/IIOP	CMP JCA	WS Management Deployment	Annotations EJB 3 JPA 1.0 WS-* JSF	EJB 3.1 JPA 2.0 Servlet 3.0 JSF 2.0 JAX-RS 1.1 JCDI 1.0 @Inject Bean Validat° **Web Profile 1.0** **Managed Bean 1.0**
May 1998	Dec 1999 **10 specs**	Sept 2001 **13 specs**	Nov 2003 **20 specs**	May 2006 **23 specs**	Q4 2009 **28 specs**

From specifying to trying

Specifying first and then implementing

Implementing first then specifying



Project JPE	**J2EE 1.2**	**J2EE 1.3**	**J2EE 1.4**	**Java EE 5**	**Java EE 6**
	Servlet JSP EJB JMS RMI/IIOP	CMP JCA	WS Management Deployment	Annotations EJB 3 JPA 1.0 WS-* JSF	EJB 3.1 JPA 2.0 Servlet 3.0 JSF 2.0 JAX-RS 1.1 JCDI 1.0 @Inject Bean Validat° **Web Profile 1.0** **Managed Bean 1.0**
May 1998	Dec 1999	Sept 2001	Nov 2003	May 2006	Q4 2009
	10 specs	**13 specs**	**20 specs**	**23 specs**	**28 specs**

What was J2EE doing in 2003?

Home interface

```
public interface HelloHome extends EJBHome {
  Hello create() throws RemoteException,
                        CreateException;
}
```

Local interface

```
public interface Hello extends EJBObject {
  String sayHello() throws RemoteException;
  Date today() throws RemoteException;
}
```

EJB

```
public class HelloBean implements SessionBean {
  public HelloBean() { }
  public String sayHello() {return "Hello TSS Prague !";}
  public Date today() {return new Date(); }
  public void ejbCreate() throws CreateException { }
  public void setSessionContext(
    SessionContext sessionContext)
    throws EJBException { }
  public void ejbRemove() throws EJBException { }
  public void ejbActivate() throws EJBException { }
  public void ejbPassivate() throws EJBException { }
}
```

Deployment Descriptor

```
<ejb-jar>
  <enterprise-beans>
    <session>
      <display-name>HelloSB</display-name>
      <ejb-name>HelloBean</ejb-name>
      <home>HelloHome</home>
      <remote>Hello</remote>
      <ejb-class>HelloBean</ejb-class>
      <session-type>Stateless
      </session-type>
      <transaction-type>Container
      </transaction-type>
    </session>
    <assembly-descriptor>
      <container-transaction>
        <method>
          <ejb-name>HelloBean</ejb-name>
          <method-name>*</method-name>
        </method>
        <trans-attribute>Required
        </trans-attribute>
      </container-transaction>
    </assembly-descriptor>
  </enterprise-beans>
</ejb-jar>
```

What's happening today?

```
@Stateless
public class HelloBean {
  public String sayHello() {
    return "Hello World !";
  }
}
```

Java EE 6 highlights

New concepts

- **Prunning**
  - Soon less specs
- **Profiles**
  - Subset of the platform (Web Profile)
- **Portable JNDI names**
  - java:comp, java:module, java:app, java:global
- **EJB Lite**
  - Subset of EJB to run in a web container

New specifications

- **Managed Bean 1.0**
  - Container managed POJO
- **Interceptors 1.1**
  - Taken out from the EJB spec
- **Bean Validation 1.0**
  - Constrain once, validate anywhere
- **JAX-RS 1.1**
  - RESTful webservices
- **@Inject 1.0 & CDI 1.0**
  - Dependency injection

New features in existing specs

- **JPA 2.0**
  - Richer mapping and JPQL
  - Pessimist locking
  - Caching API
  - Type safe criteria API
- **EJB 3.1**
  - Optional interface
  - @Singleton
  - Asynchronous calls
  - Timer service
  - Embeddable container

New features in existing specs

- **Servlet 3.0**
  - Use annotation and optional web.xml
  - Extensibility with fragments
  - Asynchronous support
  - Security
- **JSF 2.0**
  - Use annotation and optional faces-config.xml
  - Composite components
  - Ajax support
  - Facelets
  - Ressource management

A community

An open source community

- **All RI are open source**
  - GlassFish V3 : EJB 3.1 and Servlet 3.0
  - EclipseLink : JPA 2.0
  - Jersey : JAX-RS 1.1
  - Metro : JAX-WS 2.2
  - Weld : CDI 1.0
  - Mojarra : JSF 2.0
  - Hibernate Validator : Bean Validation 1.0
  - Guice : @Inject 1.0

A business community

- **Some RI are used by other app servers**
  - JPA 2.0 (Sun) EclipseLink (Eclipse) used by GlassFish & Weblogic
  - Bean Validation (JBoss) used by GlassFish
  - Mojarra (Sun) used by JBoss
  - Weld (JBoss) used by GlassFish
  - Jersey (Sun) used by Weblogic
  - Metro (Sun) used by WebSphere, Weblogic & JBoss

Spring 3.0 highlights

Spring 3.0 highlights

- Rest support in Spring @MVC
- JavaConfig
- Spring EL
- Java 5 and above required
- ...

REST

Spring @MVC without REST

```
@RequestMapping(value = "/displayClient", method = GET)
public Client displayClient(@RequestParam("id") long id)
{
    return this.clientService.findClient(id);
}
```

Spring @MVC with REST

```
@RequestMapping(value = "/clients/{id}", method = GET)
public Client displayClient(@PathVariable("id") long id)
{
    return this.clientService.findClient(id);
}
```

JavaConfig

- **Top-notch dependency injection**

```
@Configuration
public class ApplicationConfig {
  @Autowired
  private DataSource dataSource;

  @Bean
  public ClientService clientService() {
   ClientService clientService = new ClientServiceImpl();
   clientService.setClientDAO(clientDAO());
   return clientService;
  }

  @Bean
  public ClientDAO clientDAO() {
   ClientDAO clientDAO = new ClientDAO( this.dataSource );
   return clientDao;
  }
}
```

Spring EL

Injecting more than a bean

```
<bean class="mycompany.RewardsTestDatabase">
    <property name="databaseName" value=""#{systemProperties.databaseName}"/>
    <property name="keyGenerator" value=""#{strategyBean.databaseKeyGenerator}"/>
</bean>
```

Advanced example with Spring Security

```
<http use-expressions="true">
<intercept-url pattern="/secure/**" access="hasRole('ROLE_SUPERVISOR')
               and hasIpAddress('192.168.1.0/24')" />
</http>
```

Java 5 required

- **For the first time, Spring will require Java 5**
  - From Spring 3.0
- **Non-Java 5 projects should remain on Spring 2.5**

A community

Pros & Cons

Java EE

- **Pros**
  - Standard
  - Relies on several companies (Sun, JBoss, IBM...)
  - A solid base to innovate (JTA, JMS, JPA...)
  - Works out of the box (all the specs are ready to use)
  - Several implementations
- **Cons**
  - Hard to extend (except with profiles)
  - Not all the JCP specs have public mailing list
  - Needs an implementation to run (Websphere 8?)

Spring

- **Pros**
  - Works on any kind of environment
    - Wide range of JVMs and App Servers
    - Already proven on large-scale applications
  - A « real » choice between annotations and XML
    - In many cases, XML is better than annotations :)
  - Community driven
- **Cons**
  - Relies on 4-5 smart people
  - Only one implementation

The future

- **Will Spring continue to influence Java EE?**
- **Plenty of features could be included into Java EE**
  - Spring AOP
  - Advanced Exception Management
  - Batch
  - Security
  - WebFlow

Integration

- **Injecting a Spring bean into an EJB 3.1**

```
@Stateless
@Interceptors(SpringBeanAutowiringInterceptor.class)
public class ClientServiceEjbImpl implements ClientService {

    // automatically injected with a matching Spring bean
    @Inject
    private AccountService accountService;
    // ...

}
```

@Inject is available since Spring 3.0-RC2. For older versions you can use @Autowired

The future

- **Will Java EE have more impact on Spring in the future?**
  - Will SpringSource provide a Java EE implementation?
  - Or just the Web Profile ?
  - One could use standard syntax with Spring extensions when needed

Q/A
Later on

www.parisjug.org

Sponsors

Xebia

zenika
ARCHITECTURE INFORMATIQUE

OBJET DIRECT
homsys Group

NOVEDIA
VISION

SFEIR

OXiane

Fast Connect

valtech

IBM®

PYXIS

Merci de votre attention!

www.parisjug.org

Licence

creative commons

Paternité-Pas d'Utilisation Commerciale-Partage des Conditions Initiales à l'Identique 2.0 France

- http://creativecommons.org/licenses/by-nc-sa/2.0/fr/