


TheServerSide.COM  
**JAVA in ACTION**  
An Enterprise Java Conference and Training Experience

# Trends and the Future of Enterprise Java

Floyd Marinescu  
 Founder, TheServerSide.com



---

---

---

---

---

---


---

---

TheServerSide.COM  
**JAVA in ACTION**  
An Enterprise Java Conference and Training Experience

Goals for this presentation

**Bring you up to date on important news and trends, both technical and cultural, that are affecting the future of Java, that developers should know about.**



---

---

---

---

---


---



---

---

TheServerSide.COM  
**JAVA in ACTION**  
An Enterprise Java Conference and Training Experience

About Floyd Marinescu

- wrote the book *EJB Design Patterns* (2002, Wiley)
- designed, implemented, and have been managing a team running TheServerSide.com J2EE Community since 2000
- I regularly report on new J2EE business and technical developments on TheServerSide
- I am 

---

---

---

---

---

---


---

---

TheServerSide.COM  
**JAVA in ACTION**  
An Enterprise Java Conference and Training Experience

### Presentation Overview

- **Technology Trends**
  - Foundations of tomorrow's application design
    - AOP, Dependency Injection, Annotations
  - Java as a platform for scripting languages
  - AJAX & Rich Clients
  - Web Services and SOA
  - JSR 170 – Content Repositories
  - The falling value proposition for EJB – can it come back?
- **Business / Political Trends**
  - Continued Dominance of Eclipse
  - Emerging Economies and Open Source
  - Apache Harmony
  - Open Source Business Models
  - Areas for commercial models?
- **J2EE to become the plumbing of the future?**



---

---

---

---

---

---


---

---

TheServerSide.COM  
**JAVA in ACTION**  
An Enterprise Java Conference and Training Experience

### AOP – Just beginning!

- **Refactor repeating (cross cutting) code into one place**
- **"apply some code that follows this generalized pattern to all of these places." - Bob Lee**
- **Two main functions:**
  - Intercepting method calls, field calls, constructor calls
  - Adding interfaces & implementation to existing Java classes



---

---

---

---

---

---


---

---

TheServerSide.COM  
**JAVA in ACTION**  
An Enterprise Java Conference and Training Experience

### AOP vs. OOP

- **OO = nouns and verbs**
- **AO is the adverbs and adjectives**
- **Identifying Aspects: look for adverbs and adjectives in your requirements**
  - a SecureBankAccount
  - or a *Secure* BankAccount



---

---

---

---

---

---


---

---

TheServerSide.COM  
JAVA in ACTION  
An Enterprise Java Conference and Training Experience

...AOP

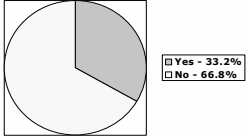
- **Practical applications of AOP in use today:**
  - Transactions
  - Security
  - Event handling
  - Caching
  - Exception Handling
  - Forcing a Singleton
  - Instrumentation / Metrics recording / Diagnostics
  - Testing your applications
  - Architecture Enforcement
  - Logging



TheServerSide.COM  
JAVA in ACTION  
An Enterprise Java Conference and Training Experience


AOP – Where are we today?

- **Survey of 184 most senior developers in the US:**
- **Have you used AOP in a production app or do you have definite plans to do so?**
- **Adoption: Slow, but so it always is!**



Response	Percentage
Yes	33.2%
No	66.8%

- **Getting started: Try Spring Framework, AspectWerkz, AspectJ**




TheServerSide.COM  
JAVA in ACTION  
An Enterprise Java Conference and Training Experience

Dependency Injection

- **Refactor configuration and dependency code out of business logic**
- **Inject dependencies into objects using plain java setter or constructors**
  - As opposed to explicit look up, or explicit call backs
- **A simple POJO with a setter for injection:**

```
public class MovieListener {  
    private MovieFinder _finder;  
    public void setMovieFinder(MovieFinder finder) { _finder  
        = finder; }  
    ...  
}
```



## ...Dependency Injection

### ● Benefits

- No need to lookup dependencies in business logic
- Easier to test
- Don't need to worry about class casts
- No dependency on container API's (reaction to EJB)
- Simpler, cleaner code

### ● Adoption

- Still in early phases
- Spring Framework, PicoContainer,...

### ● A must learn skill – will be mainstream soon



---

---

---

---

---

---

---

## JDK 5 Annotations – extend the language

### ● Used for expressing information about code (metadata), in the code

### ● Concept of code metadata not new, the *transient* Java modifier has always been in Java

### ● However, annotations allows incorporating new features in the language without having to wait for them to make it into the standard

```
@Deprecated public void obsoleteMethod()  
{ ... }
```



---

---

---

---

---

---

---

## AspectJ and Annotations Example

### Before Annotations:

```
public aspect Authorization {  
    pointcut restrictedOperation() : execution(* IRestrictedAccess+.*(..));  
    before() : restrictedOperation() {  
        // verify caller is authorized to access this  
        // operation ...  
    }  
}
```

### With Annotations:

```
@Aspect public class Authorization {  
    @Pointcut("execution(* IRestrictedAccess+.*(..))")  
    void restrictedOperation() {}  
    @Before("restrictedOperation()") public void verifyAccess()  
    {  
        // verify caller is authorized to access this  
        // operation ...  
    }  
}
```

Both Approaches are semantically THE SAME!



---

---

---

---

---

---

---

...JDK 5 Annotations – removes boiler plate

- **Removes boiler plate code**
  - EJB 2.0 requires remote interfaces
  - JAX-RPC web services requires an interface and implementation
- **Instead, simply “decorate” code with @remote or @webservice annotations**
- **Boiler plate code gets generated FOR YOU**



---

---

---

---

---

---

---

...Annotations – simplify configuration

- **Configuration today is VERY XML HEAVY**
- **Most open source frameworks & standards will offer configuration via annotations instead of in XML**

● **EJB 3 example:**

```
package example;

import static javax.ejb.TransactionAttributeType.SUPPORTS;
@javax.ejb.Stateless public class HelloBean implements Hello
{
    private String _greeting;
    @javax.ejb.Inject
    public void setGreeting(String greeting) { _greeting = greeting; }

    @javax.ejb.TransactionAttribute(SUPPORTS)
    public String hello() { return _greeting; }
}
```



---

---

---

---

---

---

---

...Annotations – industry effects

- **Annotations will be developed as needed for each tool you use**
- **Standardization efforts are also in effect (JSR 250):**
  - PropertySet, a generalized property set for J2SE
  - Generated, an indicator that the object or class was not hand-written
  - Resource and Resources, annotations to declare single and multiple resource references in J2EE, respectively
  - RunAs, RolesAllowed, Unchecked, and Exclude, annotations to control access to a given method based on J2EE security
  - RolesReferenced, an annotation to specify the roles required by a given class or method



---

---

---

---

---

---

---

## ...Annotations

### ● Adoption

- Early, but will be in widespread use by late 2006
- J2EE 5 to be released in 2006
- Open source frameworks currently working on annotations



---

---

---

---

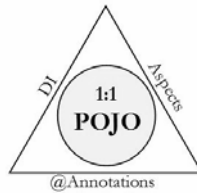
---

---

---

## Foundations of tomorrow's application design

- Community moving back to OO, POJO applications
- Dependency Injection, Aspects, and Annotations are all part of a modern application



---

---

---

---

---

---

---

## Java as a platform for scripting languages

- Dynamically typed Scripting languages such as Groovy becoming very popular
- Groovy combines features from languages like Python, Ruby, Smalltalk, and Perl in a very Java-like syntax
- Designed to be quick and easy to use:
  - Optional static typing
  - Regular expressions, lists and maps are first class citizens
  - Optional semi colons, parenthesis for method invocations, many more
- Groovy code compiles to Java .class files, and have access to the JDK



---

---

---

---

---


---

---

TheServerSide.COM  
**JAVA in ACTION**  
An Enterprise Java Conference and Training Experience

... Java as a platform for scripting languages

- **Developers can be more productive than in Java when:**
  - IO/Data manipulation
  - SQL scripting
  - Gui prototyping
  - Unit testing
  - batch/script processing – used on TSS as cron scripts
  - Implementing actions in MVC web apps
- **JSR 223: Scripting languages for the Java platform**
  - Standardizes hooks by which new languages can integrate into Java



---

---

---

---

---

---


---

---

TheServerSide.COM  
**JAVA in ACTION**  
An Enterprise Java Conference and Training Experience

... Java as a platform for scripting languages

- **The Java platform is no longer just Java**
- **Java = JVM + JDK = bytecode = Any language**
- **Java will be a host to other languages, such as BeanShell and Groovy**
- **Lack of standards have prevented mass Java-enabled scripting language adoption in the Java community**
  - Finalizing Groovy and Beanshell will change that
- **Having multiple languages in Java will attract a wider audience to Java**
- **Good strategy to increase mass Java adoption**



---

---

---

---

---

---


---

---

TheServerSide.COM  
**JAVA in ACTION**  
An Enterprise Java Conference and Training Experience

### AJAX and Rich Clients

- **Enterprise User Interface evolution**
  - Dumb terminal
  - Thick client / Client-Server / Fat Client
  - Today's standard: Thin client / HTML+HTTP / three tier
- **Benefits of thin client / HTML based UI's:**
  - Zero install
  - Standard platform for hosting/viewing UI's (the browser)
  - Real time updating of application (pages served from the server)
- **Problems with thin client / HTML UI's:**
  - Lack of rich features available in the desktop worlds
  - Stateless, request-response content-retrieval system
  - Server maintains state, all operations need to go to the server
  - Poor support for non-linear workflows



---

---

---

---

---

---

---

---

## ...AJAX and Rich Clients

- **Gap between desktop and web clients is huge**
- **Solution: Applications will have rich UI capabilities running on existing universally deployed platforms**
- **Platforms & rich client alternatives**
  - Macromedia Flash – Lasso, Macromedia Flex
  - Swing/Java – Swing on the client, Java webstart, JDNC
  - Browsers – AJAX
- **The hottest trend: AJAX – Asynchronous Java Script and XML**




---

---

---

---

---

---

---

---

## ...AJAX

- **Ajax is several technologies used together:**
  - XHTML and CSS
  - dynamic display and interaction using the Document Object Model
  - data interchange and manipulation using XML and XSLT
  - asynchronous data retrieval using XMLHttpRequest
  - and JavaScript binding everything together
- **XMLHttpRequest allows making asynchronous Javascript requests behind web pages**
- **Load of pages in partial fragments instead of all or nothing**




---

---

---

---

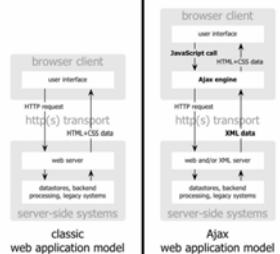
---

---

---

---

## AJAX



- Adaptive Path




---

---

---

---

---

---

---

---



TheServerSide.COM  
**JAVA in ACTION**  
An Enterprise Java Conference and Training Experience

## Industry is just beginning to use AJAX

- Partial page refreshes
- Piece meal real time validation
- Auto-completion
- Server-delivered pop-ups
- Asynchronous page refreshing
- Complex workflow in a single page

...

---

---

---

---

---

---

---

---

TheServerSide.COM  
**JAVA in ACTION**  
An Enterprise Java Conference and Training Experience

## AJAX and Rich clients

- Availability of mainstream AJAX implementations (Google GMail, Amazon.com, and more every day) will mean your manager will be soon demanding a rich client experience
- We are at the beginning of the adoption curve
- Java industry is tooling up
  - Open source frameworks to simplify AJAX & J2EE are emerging – DWR, JSON-RPC, SAJAX, Javascript templating
  - Web frameworks (Tapestry, JSF, etc) adding AJAX support
- Prepare for it now – learn AJAX

---

---

---

---

---

---

---

---

TheServerSide.COM  
**JAVA in ACTION**  
An Enterprise Java Conference and Training Experience

## Web services and SOA

- Web services being used pragmatically – far below the initial hype
- SOA – everyone talking about it – few realizing it's full potential
- What is SOA?
  - SOA is an architectural style that encourages the creation of loosely coupled business services
  - Loosely coupled services that are interoperable and technology-agnostic enable business flexibility
  - An SOA solution consists of a composite set of business services that realize an end-to-end business process
  - Each service provides an interface-based service description to support flexible and dynamically re-configurable processes

- John Reynolds

---

---

---

---

---

---

---

---

## Service Oriented Architectures

- **Important points about SOA:**
  - "Services" in SOA are business services... updating a loan application is a business service, updating a record in a database isn't
  - When services are "composed" into wider business processes, then you have an example of SOA
    - Business Process Engines exist to make service composition easier
  - Services are meant to be re-useable in new business processes, both internally and by business partners
  - SOA solutions favor flexibility over efficiency... machine cycles and network traffic are less important than being able to quickly implement and change business processes
- **What SOA is not:**
  - Services are not user interfaces. User interfaces invoke services.
  - Services can be implemented in any language, but all services must be invocable in a language agnostic way (eg: SOAP/HTTP)

- John Reynolds



---

---

---

---

---

---

---

---

## ...The truth about SOA

- **Few are actually doing SOA, most simply using web services:**
  1. We connect fat .Net clients to Java application servers (and use soap in between)
  2. We connect to some ASP (like Amazon, SF.com, etc.)
  3. We front ended some legacy system with services. - Bruce Schnier
- **Most applying the 3+N pattern**
  - Three tier application using web services to talk to N external services - this is not service composition
- **The standards are not fully ready for a complete SOA vision, but commercial and open source tools exist today that fill these gaps**
- **We are seeing early successes with SOA**
- **Prediction: SOA to be fairly well understood by end of 2005, but might not be called SOA**
- **Don Box 2005 prediction: "The term SOA will have been beaten to death and the software industry will invent or recycle some equally vague term to replace it."**



---

---

---

---

---

---

---

---

## JSR 170 Content Repository API for Java

- **Goal: abstract the details of application data storage and retrieval such that many different applications can use the same interface, for multiple purposes, without significant performance degradation**
- **Storage format complete hidden**
- **Specifies a hierarchical system of content organization**
- **Contains notions of workspaces (staging/production paradigm), versioning**
- **content querying/traversal capabilities**



---

---

---

---

---

---

---

---

## ...JSR 170 Content Repository API for Java

### ● Possibilities

- Unified API to content across the enterprise
- Standards applied across all portal/CMS products which have previously been highly proprietary
- End of vendor lock in with portal products (full import/export capabilities)
- Better, higher level of abstraction for projects that would have otherwise employed an object modeling + O/R approach

### ● Status

- Proposed Final Draft
- Support from all the major portal/CMS vendors
- Apache Jackrabbit project

- **Implication: Early adopter use by early 2006, mainstream use by 2007**



---

---

---

---

---

---

---

---

## The falling value proposition of old-style EJB

### ● 3 high-level categories of EJB benefits

- **Framework Benefits** – Security, Transactions, Pooling, packaging/deployment
- **Distribution Benefits** – It's the way to do distributed communication
- **Component Benefits**

*Quick Poll:  
How are  
people  
using  
EJB?*

### ● Where the community is going:

- **Framework** – POJO's via the three foundations: AOP + Dependency Injection + Annotations
- **Distribution** – Web Services over POJOs for distribution
- **Components** – Who cares? The Enterprise Component Market is dead

- **EJB 2.1 is still the best choice where a distributed object is required and an SOA approach is not feasible**



---

---

---

---

---

---

---

---

## ...EJB 3 to the rescue

- **Framework** – POJO's with annotations, dependency injection, and Interceptors (AOP) – surprised? ☺
- **Distribution** – can easily export both Web Services and RMI
- **Components** – Not components anymore – POJO's with annotations

@Stateless

```
@Interceptors({  
    "com.acme.AccountAudit",  
    "com.acme.Metrics",  
    "com.acme.CustomSecurity"  
})
```

```
public class AccountManagementBean implements AccountManagement {  
    public void createAccount(int accountNumber, AccountDetails  
        details) { ... }  
    public void deleteAccount(int accountNumber) { ... }  
    public void activateAccount(int accountNumber) { ... }  
    public void deactivateAccount(int accountNumber) { ... }  
    ...  
}
```



---

---

---

---

---

---


---

---

TheServerSide.COM  
JAVA in ACTION  
An Enterprise Java Conference and Training Experience

### EJB 3 Becoming cool again

- Good ideas from Spring and Hibernate in EJB 3
- Persistence spec has been separated from EJB core spec
  - Will allow tools to implement the spec in parts
- Status
  - Proposed Final Draft 2
  - Probably have a final final by end of 2005
  - JBoss, Oracle, Resin already have draft implementations to play with
- Implications
  - Gavin King – EJB 3 to be the next java beans
  - Nay sayers – Spring and Hibernate are already standards – who will need EJB 3?
  - Politics – what role will a vendor marketplace have when there are leading open source implementations?



---

---

---

---

---


---

---

TheServerSide.COM  
JAVA in ACTION  
An Enterprise Java Conference and Training Experience

### Presentation Overview

- Technology Trends
  - Foundations of tomorrow's application design
    - AOP, Dependency Injection, Annotations
  - Java as a platform for scripting languages
  - AJAX & Rich Clients
  - Web Services and SOA
  - JSR 170 – Content Repositories
  - The falling value proposition for EJB – can it come back?
- Business / Political Trends
  - Continued Dominance of Eclipse
  - Emerging Economies and Open Source
  - Apache Harmony
  - Open Source Business Models
  - Areas for commercial models?
  - J2EE to become the plumbing of the future?



---

---

---

---

---


---


---

TheServerSide.COM  
JAVA in ACTION  
An Enterprise Java Conference and Training Experience

### Continued Dominance of Eclipse

- Recent poll of 242 senior developers in the US at a TSS conference





---

---

---

---

---

---

---

### ...Continued Dominance of Eclipse

- **Evans Data on Eclipse as a primary IDE**
  - 70 - 90% growth world wide in 2004
- **Large ecosystem of plug-ins**
- **Industry giants contributing development resources to Eclipse**
- **A number of free tools available in the open source distribution**
- **May 31, 2005 – JBuilder announces Eclipse will be foundation of future versions of JBuilder IDE**
- **Implications**
  - Eclipse becomes the Java community's answer to Visual Studio.NET
  - IntelliJ IDEA, NetBeans, and Oracle JDeveloper to cater to niche markets
  - Eclipse to foster a market/ecosystem for development tools much like Visual Basic
  - Very important for Java's continued growth and adoption



---

---

---

---

---

---

---

### Emerging Economies and Open Source

- **Emerging economies will further open source**
- **Case study: Brazil**
  - Government mandate to use open source, and multi-platform software. Linux and Java seen as core to national interest.
  - Building their own open source JVM called JAVALI
  - Freedom from vendor lock-in important on a national scale



---

---

---

---

---

---

---

### Why does Brazil want open source Java?

- **Freedom to distribute JVM on any OS, (specifically Linux) not just Windows and Mac. Free systems require a free Java**
- **Want to build desktop apps with it on Linux, but can't if it's not part of the core distribution**
- **Want to expand Open Office with Java!**
- **Tax breaks for open source**
- **Port it to other systems, such as hand helds, voting machines, etc**
- **Guarantee that we have the rights to use it in the future**
  - In other words: protect from US trade sanctions



---

---

---

---

---

---

---

## ... Emerging Economies and Open Source

### ● Implications:

- China, Brazil, India, and emerging economies share the same interests
- Expect massive resources put behind open source in the next 5 years
- Open source Java on the way...



---

---

---

---

---

---

---

## Apache Harmony

### ● Open source JVM and JDK

#### ● Goals:

- create a compatible, independent implementation of J2SE 5 under the Apache License v2
- create a community-developed modular runtime (VM and class library) architecture to allow independent implementations to share runtime components, and allow independent innovation in runtime components

### ● Brought together some of the best minds on JVM's to contribute



---

---

---

---

---

---

---

## Apache Harmony – Why?

- An open source implementation can survive if Sun / vendors stop supporting Java
- Distribution rights – Linux and FreeBSD & more availability for Java
- Ability to ship custom JVM's with an application
  - Eg: JVM's with custom features, or ship apps with parts of JVM's (2meg exe file, or 100KB JAR that needs the JRE?)
- "enables collaboration in the parts of technology that are common to all"... letting organizations focus on things that add value on top of the common – Geir Magnusson
- Risks
  - A project of this scale is no easy matter
  - Other similar projects (Geronimo) proceeding slowly, so might this
  - Forking Java
  - Secret attempt by IBM to rule the world



---

---

---

---

---

---

---

### New open source business models

- **JBoss - pioneer**
- **Simula labs - VC focused on open source**
- **Interface21, numerous others**
- **Gluecode software - IBM buys Gluecode**
- **Implications**



---

---

---

---

---

---

---

### What's left for commercial models?

- **Open source software is increasingly standardizing more complex infrastructures**
- **First web servers, then appservers, now even portal servers and integration servers!**
- **Opportunities for commercial firms**
  - High end, large scale (WebSphere XD, BEA Tuxedo)
  - Niche tools - caching, finance, Jini
  - Anything ahead of the open source curve



---

---

---

---

---

---

---

### J2EE the plumbing of the future?

- **J2EE is increasingly becoming 'plumbing'**
- **Server vendors are focusing on building solutions on top of J2EE, such as portals & integration, will we need to use J2EE or abstractions on top of J2EE?**
- **J2EE 1.4 spec lead Mark Hapner...**
  - It's natural for the level you code at one point becomes plumbing in the future.
  - The challenge is for the community to continue evolving and creating better abstractions, such as Portlets, or JAX-RPC
  - "We will not continue to focus at one level, and sort of get stuck on that one level as the world moves forward. I think that's the challenge, and I think we're rising to that challenge now, and I think that you'll see new JSRs that are the product of that."



---

---

---

---


---

---

---

TheServerSide.COM  
JAVA in ACTION  
An Enterprise Java Conference and Training Experience

Questions



---

---

---

---

---

---

---