Server Side Java: What's Next?

Larry Cable, BEA Systems Inc.
(a wholly owned subsidiary of Oracle Corporation)

TS-7866

ORACLE | bea

Java

Sun microsystems

Discover some of the emerging technologies and trends that may influence the composition of “Server-Side” Java™ Platform over the next 2-3+ years.

GOAL

What this …

- *IS* …
  - Is a somewhat (albeit informed) subjective look at the “state of the art” in server-side technologies, the challenges facing server-side users, and how the platform may evolve using these technologies to address those challenges over the next few years (2-3+)…
- *IS NOT* …
  - It’s not a thinly veiled product roadmap (honestly)
  - It’s not a delivery commitment from either BEAS or ORCL

How Does Technology Evolve?

"(Increments in) technology are an evolution of what came previously… in an attempt to address the challenges ahead"

- Larry Cable, "stating the obvious", JavaOne℠ 2008 Conference.

Agenda

- The Past: how did we get here?
- The Present: where are we?
- The Future: where might we go?

Agenda

- The Past: how did we get here?
- The Present: where are we?
- The Future: where might we go?

What “begat” Java 2 Platform, Enterprise Edition (J2EE)?

- 2.5 tier web applications; CGI-bin, ISAPI, NSAPI, Apache modules, database access…
- CORBA, Distributed Objects, (n=3) tier distributed systems…
- Microsoft:
  - DCOM Transaction Coordinator (Viper)
  - ActiveServerPages & Controls
  - …
- The emergence of the concept of an “application server”
  - And the fragmentation of the industry

What influenced the design of Java EE Software?

- The Java Language unchanged as it was in 1.2:
  - No Generics
  - No Annotations
  - …
- Containers/Component "models" were "state of the art"
  - POJOs had not been discovered yet! ☺
    - Strong typing thru class inheritance &| interface implementation
  - Although JavaBeans sets the stage for Dependency Injection (later), Components typically "lookup" their dependencies
  - "Boilerplate" while onerous, was not eschewed!
- The emergence of XML
- Declarative vs imperative Programming
- Deployment and Packaging model

Java 2 Enterprise Edition “lineage”

- J2EE 1.2: 1999(2000)
- J2EE 1.3: 2001
- J2EE 1.4: 2003
- Java EE 5: 2006
- Java EE 6: 2008(?)
- Java EE 7: 20{0|1}?(?)
- ???

Java 2 Enterprise Edition (1.2)

- EJB™ (1.1) software
- Servlet (2.2)
- JavaServer™ Pages (JSP) (1.1) software
- Java DataBase Connectivity (JDBC) (2.0) software
- JTA (1.0.1)
- JTS (0.95)
- Java Message Service (JMS) (1.0.2) API
- Java Naming and Directory Interface (JNDI) (1.2) API
- JavaIDL*
- JavaMail (1.1)
- Java Activation Framework (1.0)

(*) Actually just the necessary/optional IDL(IIOP)<->Java/EJB mappings

Java 2 Enterprise Edition (1.3)

- EJB (2.0) software
- Servlet (2.3)
- JSP (1.2) software
- JDBC (2.1) software
- JTA (1.0.1)
- JTS (1.0)
- JMS (1.0.2) API
- JNDI (1.2) API
- JavaIDL*
- JavaMail (1.1)
- Java Activation Framework (1.0)
- Connectors (1.0)
- Java API for XML Processing (JAXP) (1.0) software
- Java Authentication and Authorization Service (JAAS) (1.0) software

(*) Actually just the necessary/optional IDL(IIOP)<->Java/EJB mappings

Java 2 Enterprise Edition (1.4)

- EJB (2.1) software
- Servlet (2.4)
- JSP (2.0) software
- JDBC (3.0) software
- JTA (1.0.1B)
- JTS (1.0)
- JMS (1.1) API
- JNDI (1.2) API
- JavaIDL
- JavaMail (1.2)
- Java Activation Framework (1.0)
- Connectors (1.5)
- JAXP (1.2) software
- JAAS (1.0) software
- JACC (1.0)
- JAX-RPC (1.0)
- JSR 109 -WS for Java EE (1.0)
- Java API for XML Registries (JAXR) (1.0) software
- Java Management Extensions (JMX) (1.1)
- JSR 77 – management (1.0)
- JSR 88 – deployment (1.0)

Agenda

- The Past: how did we get here?
- The Present: where are we?
- The Future: where might we go?

Java Enterprise Edition 5.0

- EJB (3.0)
- JPA (1.0)
- JDO (2.0)
- Servlet (2.5)
- JSP (2.1)
- JSTL (1.2)
- JSF(1.2)
- JAX-WS (2.0)
- JSR 181 - WS metadata (1.0)
- SAAJ (1.3)
- JDBC (3.0)
- JTA (1.1)
- JTS (1.0)
- JMS (1.1)
- JNDI (1.2)
- JavaIDL*
- JavaMail (1.4)
- Java Activation Framework (1.1)
- Connectors (1.5)
- JAXP (1.3)
- StAX (1.0)
- JAXB (2.0)
- JAAS (1.0)
- JACC (1.1)
- JAX-RPC (1.1)
- JSR 109 – WS for J EE (1.2)
- JAXR (1.0)
- JMX (1.2)
- JSR 77 – management (1.1)
- JSR 88 – deployment (1.1)

What got us here…

- Industry:
  - *POJOs* "discovered":
    - JDO
    - Spring Framework
    - Hibernate
    - …
  - *Web Services/WS-I*
    - MSFT/IBM
  - AJAX/RIA
  - …
- Language/Platform:
  - Generics
  - *Annotations*

But Java EE is not the only "game in town…"

- The Java Standard Edition Platform (and The Java Enterprise Edition Platform) are probably the best thing that ever happened to encourage "Open Source"
  - As a platform and foundation for innovation…
  - As a source of inspiration (and frustration)…
- Thus the explosion of a multitude of open source frameworks etc:
  - Spring
  - Hibernate
  - Struts/Shale
  - Beehive, XMLBeans
  - Velocity & Turbine, Cocoon, Wicket, *YAFJBWTF …*
  - SEAM, ADF, …
  - …

Java Enterprise Edition 6.0 (proposed)

- EJB (3.1)
- JPA (2.0)
- JDO (2.0)
- Servlet (3.0)
- JSP (2.2)
- JSTL (1.2)
- JSF(1.2)
- JAX-WS (2.1)
- JAX-RS (1.0)
- *JSR 299* - "Web Beans"
- *JSR 236* - Timer for Application Servers
- *JSR 237* - Work Manager " " "
- JSR 181 - WS metadata (1.0)
- SAAJ (1.3)
- JDBC (4.0)
- JTA (1.1)
- JTS (1.0)
- JMS (1.1)
- JNDI (1.2)
- JavaIDL*
- JavaMail (1.4)
- Java Activation Framework (1.1)
- Connectors (1.6)
- JAXP (1.3)
- StAX (1.1)
- JAXB (2.1)
- JAAS (1.0)
- JACC (1.1)
- JAX-RPC (1.1)
- JSR 109 - WS for J EE (1.2)
- JAXR (1.0)
- JMX (1.2)
- JSR 250 - Common Annotations
- JSR 45 - Debugging for other languages
- JSR 77 - management (1.1)
- JSR 88 - deployment (1.1)
- *JSR 196* - Authentication SPI

Java EE 6.0 Profile "A" (proposed)

- EJB (3.1)
- JPA (2.0)
- JDO (2.0)
- Servlet (3.0)
- JSP (2.2)
- JSTL (1.2)
- JSF(1.2)
- JAX-WS (2.1)
- JAX-RS (1.0)
- *JSR 299* - "Web Beans"
- *JSR 236* - Timer for Application Servers
- *JSR 237* - Work Manager " " "
- JSR 181 - WS metadata (1.0)
- SAAJ (1.3)
- JDBC (4.0)
- JTA (1.1)
- JTS (1.0)
- JMS (1.1)
- JNDI (1.2)
- JavaIDL*
- JavaMail (1.4)
- Java Activation Framework (1.1)
- Connectors (1.6)
- JAXP (1.3)
- StAX (1.1)
- JAXB (2.1)
- JAAS (1.0)
- JACC (1.1)
- JAX-RPC (1.1)
- JSR 109 - WS for J EE (1.2)
- JAXR (1.0)
- JMX (1.2)
- JSR 250 - Common Annotations
- JSR 45 - Debugging for other languages
- JSR 77 - management (1.1)
- JSR 88 - deployment (1.1)
- *JSR 196* - Authentication SPI

Java EE 6.0 Profile "B" (proposed)

- EJB (3.1) Lite (subset)
- JPA (2.0)
- JDO (2.0)
- Servlet (3.0)
- JSP (2.2)
- JSTL (1.2)
- JSF(1.2) *
- JAX-WS (2.1)
- JAX-RS (1.0)
- *JSR 299* - "Web Beans" *
- *JSR 236* - Timer for Application Servers
- *JSR 237* - Work Manager " " "
- JSR 181 - WS metadata (1.0)
- SAAJ (1.3)
- JDBC (4.0)
- JTA (1.1)
- JTS (1.0)
- JMS (1.1)
- JNDI (1.2)
- JavaIDL*
- JavaMail (1.4)
- Java Activation Framework (1.1)
- Connectors (1.6)
- JAXP (1.3)
- StAX (1.1)
- JAXB (2.1)
- JAAS (1.0)
- JACC (1.1)
- JAX-RPC (1.1)
- JSR 109 - WS for J EE (1.2)
- JAXR (1.0)
- JMX (1.2)
- JSR 250 - Common Annotations
- JSR 45 - Debugging for other languages
- JSR 77 - management (1.1)
- JSR 88 - deployment (1.1)
- *JSR 196* - Authentication SPI

Agenda

- The Past: how did we get here?
- The Present: where are we?
- The Future: where might we go?

Server Side Java: What's in the Box?

Programming Model Featureset

- Application (Programming) Model(s)
  - Platform for layered technologies and containers (i.e portal, ESB, BPEL, scripting languages, etc)
- (Developer) Tools Integration
- EAI (Integration & Interop) Solutions
- Emerging technologies/"value add"

Mission Critical "runtime" Featureset

- Enterprise '*ilities*':
  - Reliability
  - Availability
  - Scalability
  - Performance
  - Security
  - Virtualization
- Operational Administration and Management

Application Architecture Trends

- Enterprise SOA
  - Migration from vertical apps to SOA Enterprise wide
- SaaS
  - Internal: from IT Ops to LOB
  - External:...
- XTP (source: Gartner):

  "An application style aimed at supporting the design, development, management and maintenance of distributed TP applications characterized by exceptionally demanding performance, scalability, security, manageability and dependability requirements"
  - Online gaming
  - Social computing sites
  - ...

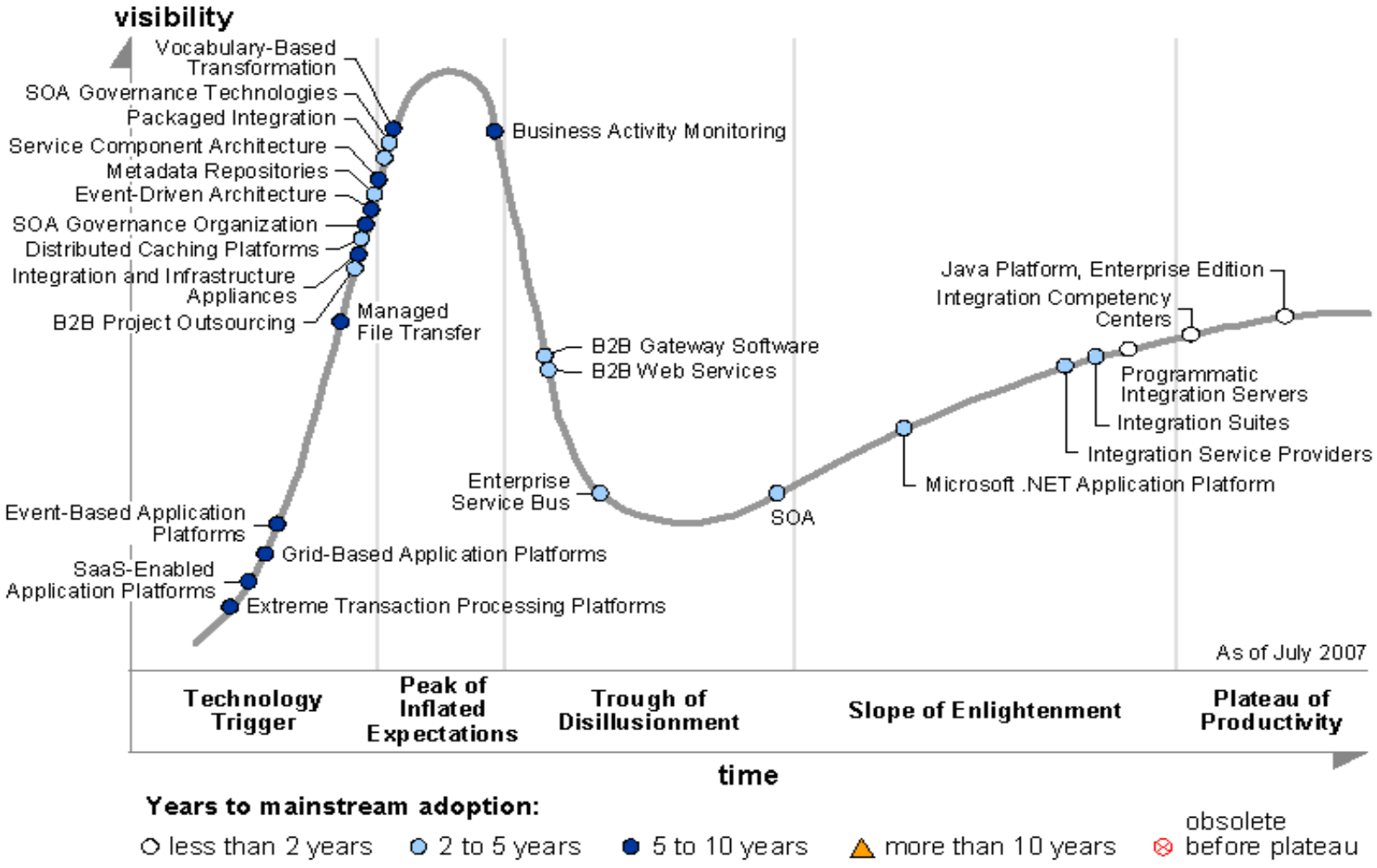
Programming Model Trends

- Java platform essential to the Enterprise:
  - Proven for enterprise applications
  - Rich ecosystem
  - Multivendor
  - Standards-based
  - Open Source
- Growing diversity of programming models/containers
  - Java EE
  - Web Services, REST, POX/HTTP …
  - Spring, SpringDM
  - OSGi
  - SCA
  - Web 2.0 RIA: (AJAX, COMET, Silverlight, AIR (FLEX, Flash)
  - Scripting Languages (Ruby (Rails), PHP, Groovy, Python…)
  - BPM, ESB, Event-driven….

IT Operations Trends

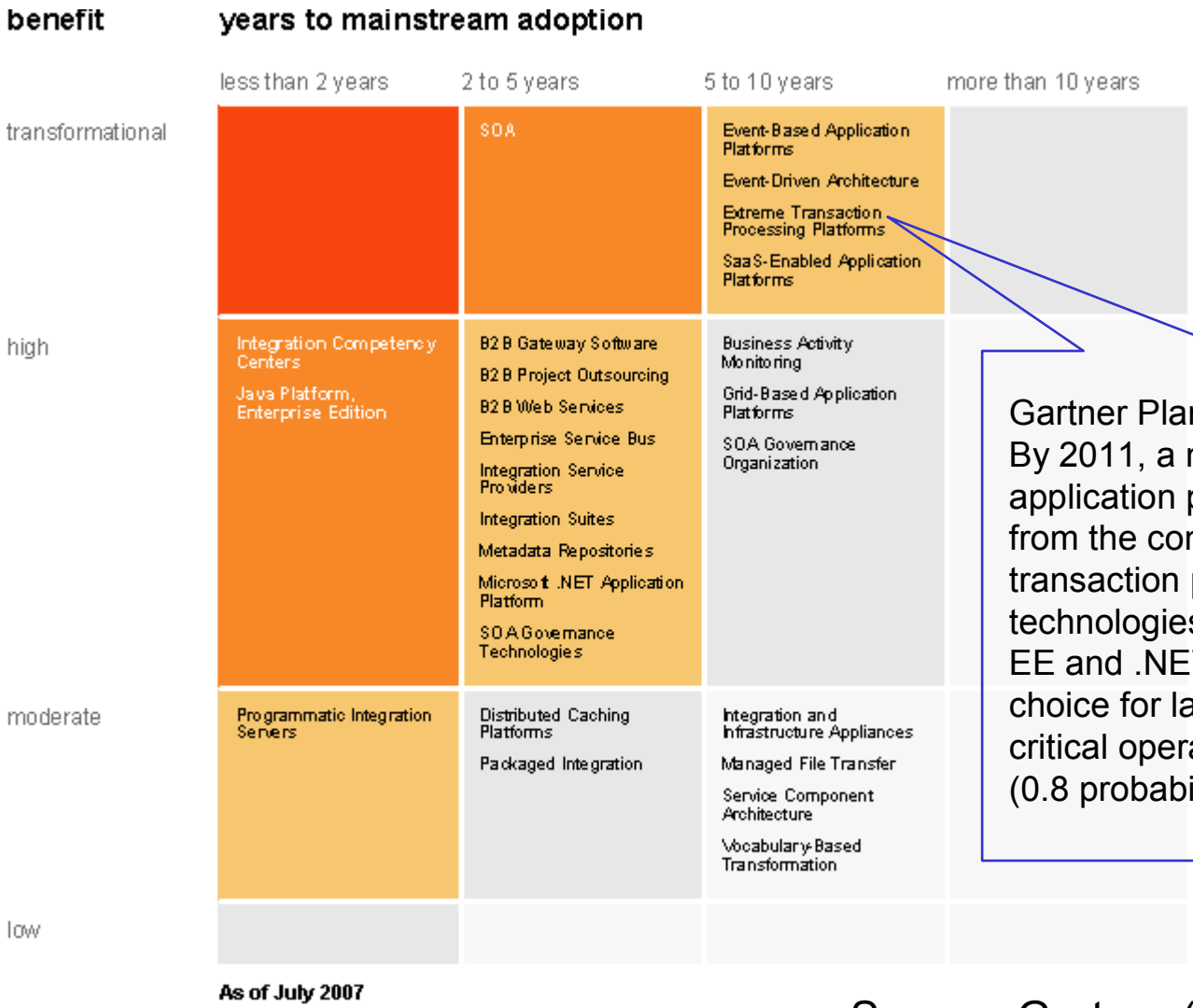
- Data Center Consolidation:
  - Many smaller to fewer larger data centers, integrated “round the clock”, w/”disaster recovery” integral, operated 7/24/365…
- Large Server Farms
  - From larger SMP H/W to massively scaled commodity H/W
  - Shared pool of compute resources
- Server Virtualization:
  - Optimize horizontally scaled server utilization dynamically
- SaaS/Business vs. cost center
  - Audit costs/usage
  - Charge back

Market Hype Cycle



Source: Gartner (February 2007)

Market Adoption



Gartner Planning Assumption:
By 2011, a new generation of application platforms stemming from the convergence of extreme transaction processing (XTP) technologies will supersede Java EE and .NET as the platform of choice for large-scale, business-critical operational applications (0.8 probability).

Source: Gartner (July 2007)

Application Infrastructure Challenges

- Convergence placing new demands on application infrastructure
  - More flexible (and demanding) application architectures
  - Diverse programming models and containers
  - Global, massively scaled, mission critical server farms
- Challenges
  - How to deliver the production support developers demand?
  - How to provision and scale this environment?
  - How to describe and package composite applications?
  - How to deploy the composite applications?
  - How to monitor usage/SLAs across composites?
  - How to adapt to meet unpredictable demands?
  - How to maintain, upgrade, manage?
  - Cost, time to market, 24x7, ….

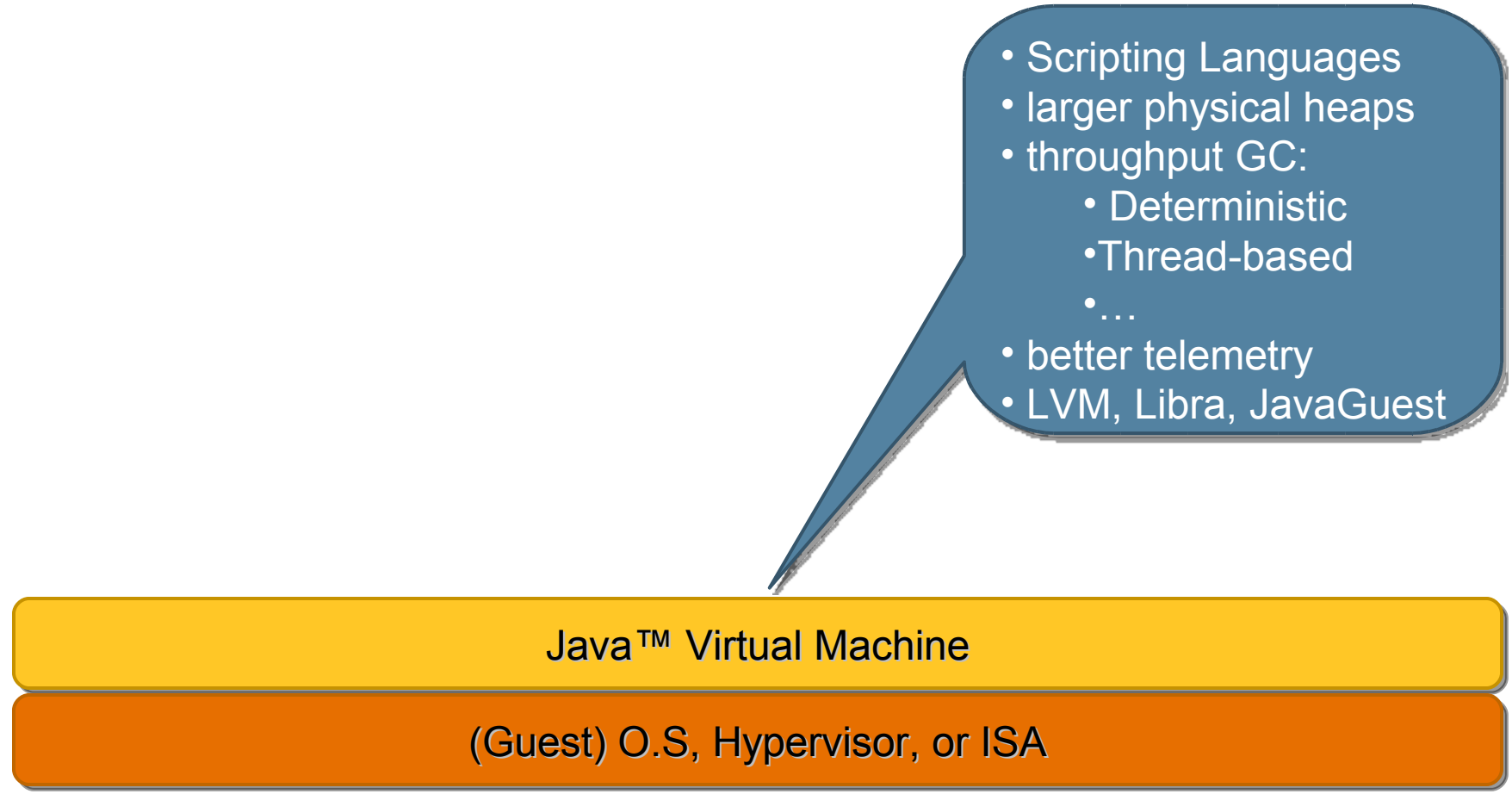
What Technologies and Challenges will effect the Programming Model?

- Challenges:
  - SOA
  - RIA
  - .Net/WCF
  - REST
  - …
- Technologies:
  - ***POJOs***
  - ***Annotations***
  - Packaging:
    - JSR 291 (OSGi)
    - JSR 277
    - JSR 294
  - "***closures***":
    http://www.javac.info/consensus-closures-jsr.html
  - Scripting:
    - JSR 292

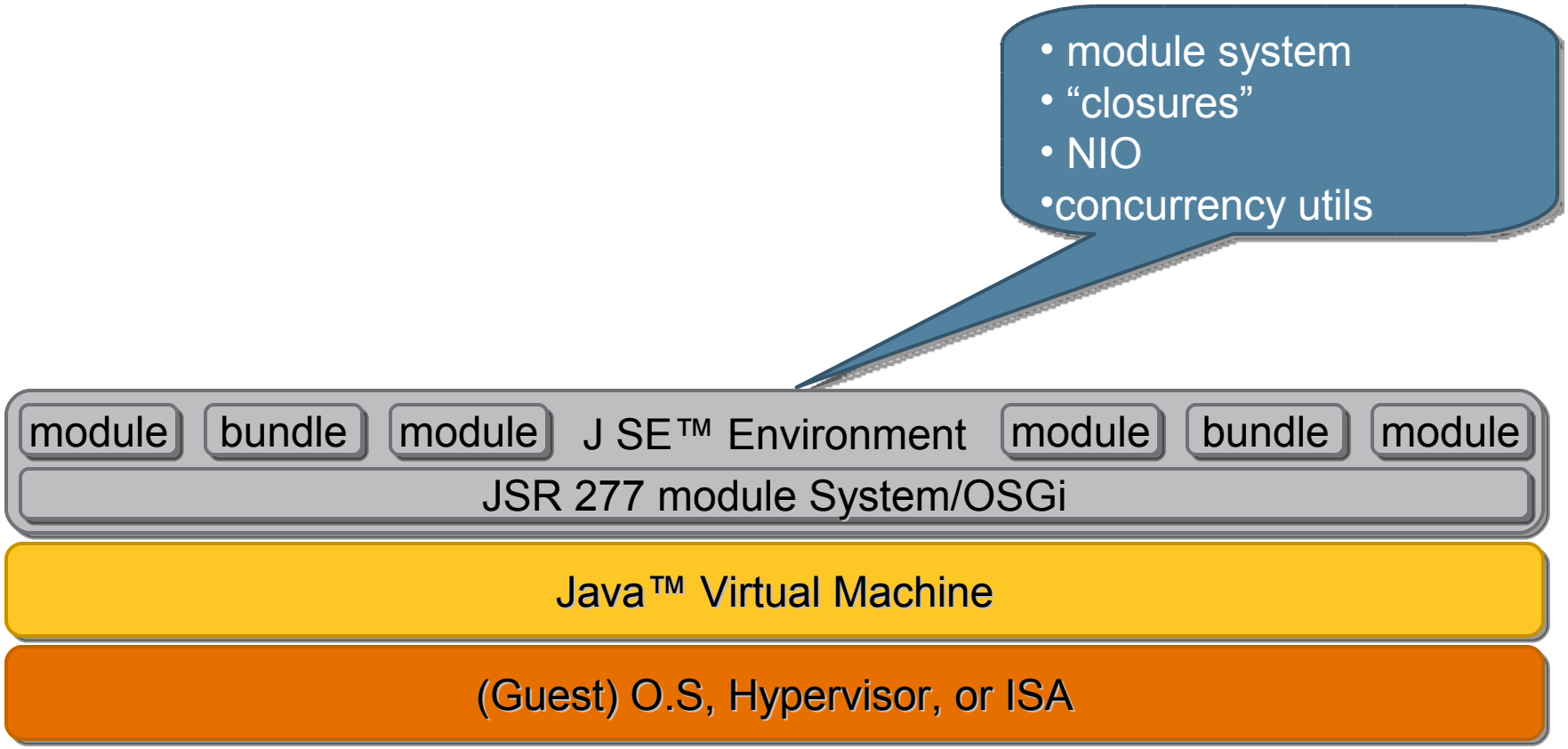
What Technologies will effect the RASP and OA&M environments?

- Challenges:
  - Virtualization
  - “Cloud” computing
  - …
- Technologies:
  - ”cloud” OSGi
  - SCA
  - …

JVM™ Software



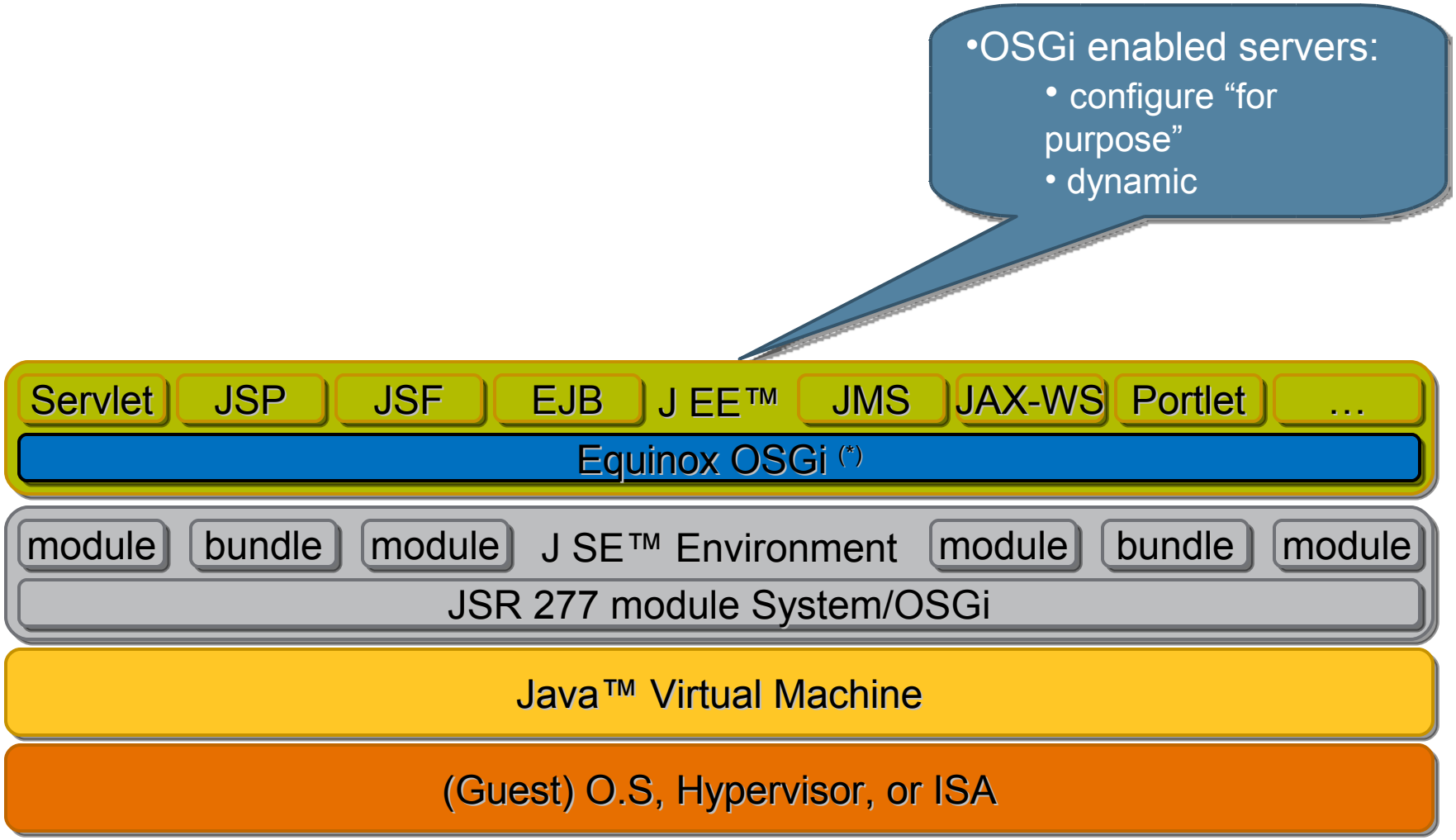
Java Platform, Standard Edition (Java SE): The Language, and the libraries…



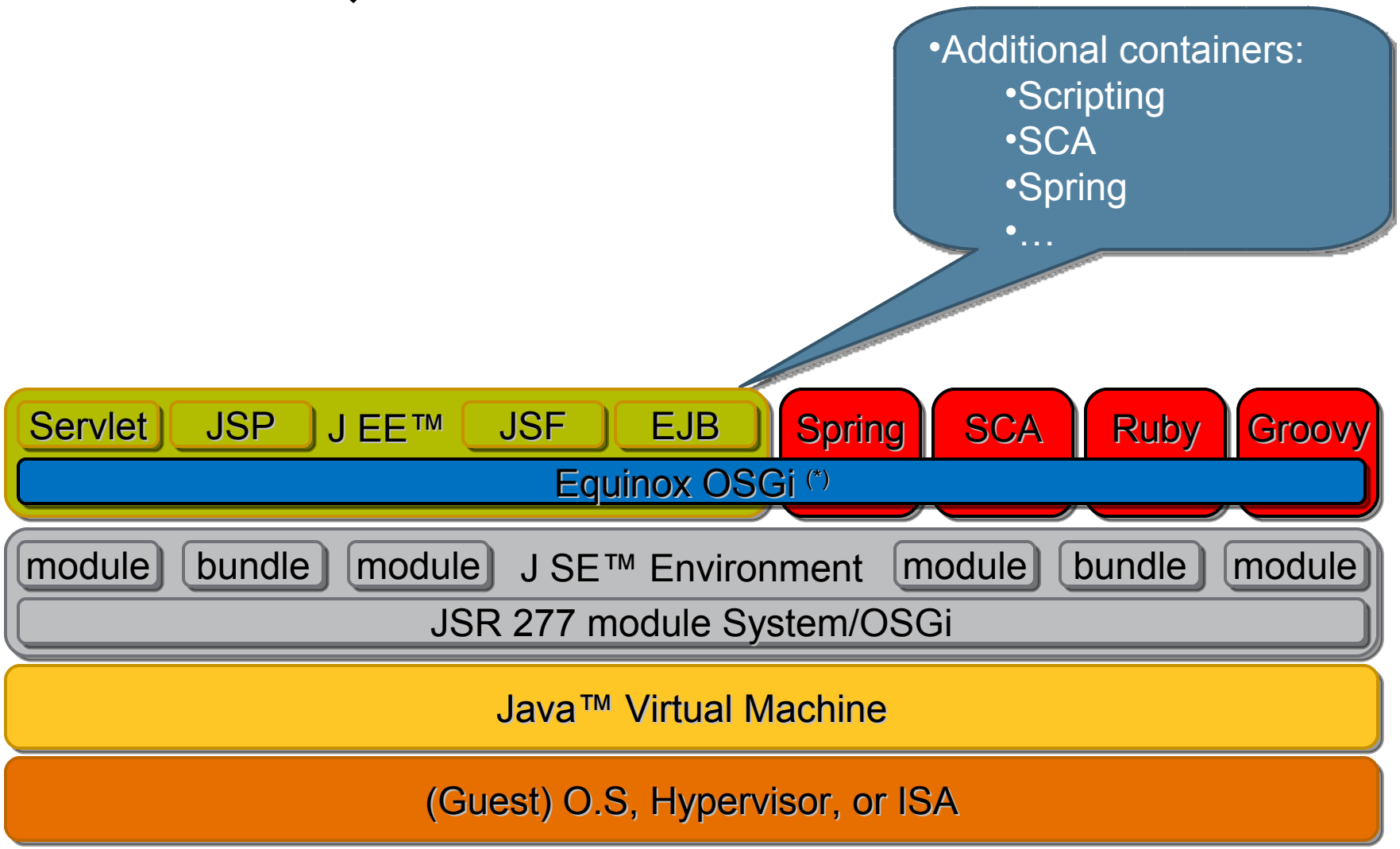
Modularity: why is it important?

- Better Packaging
- Better Interface definitions
- Package/version level dependency management
- Improved memory management/footprint control
- Issue: OSGi (JSR 294) “vs” Java Module System (JSR 277)

Java EE: The Platform …



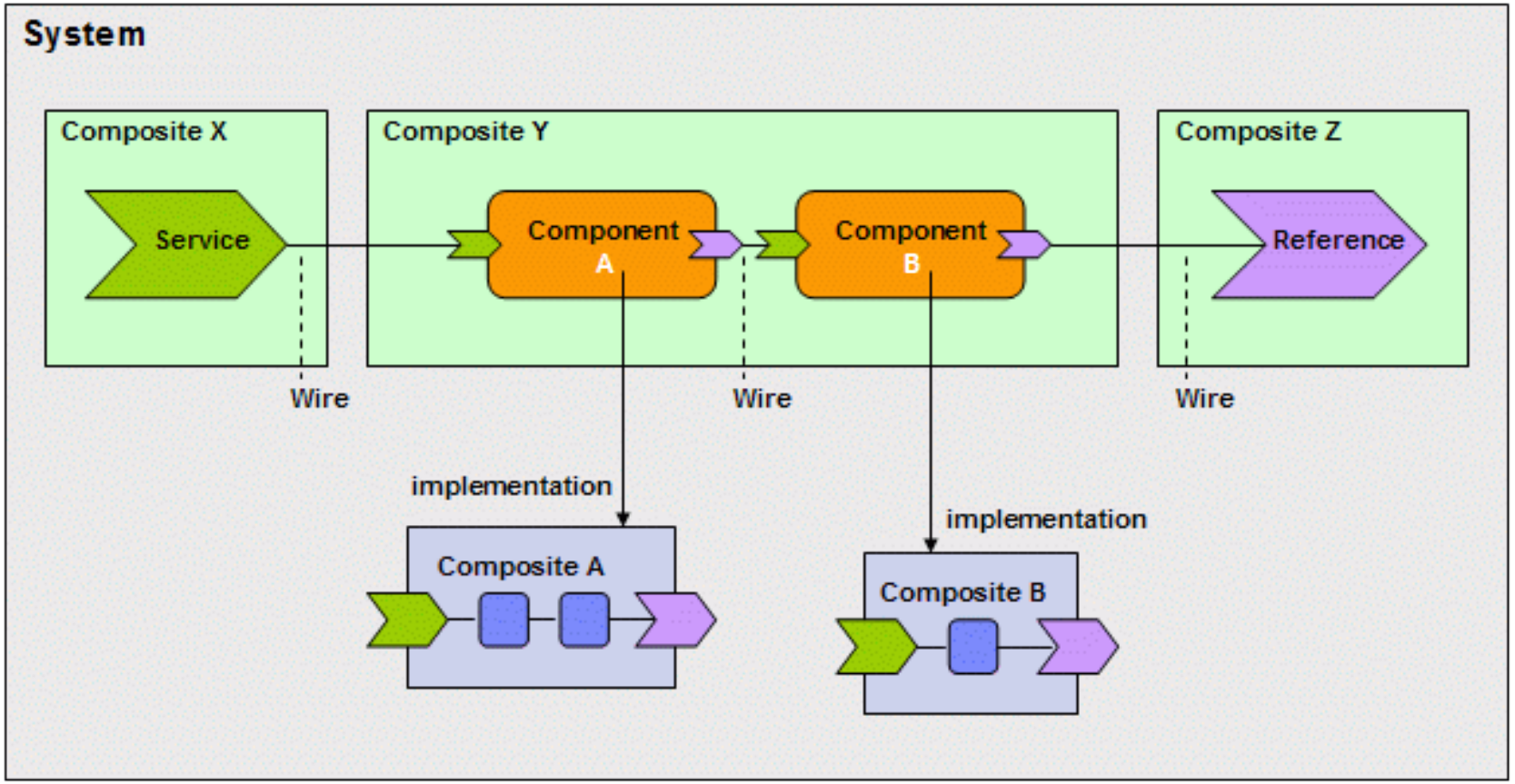
Server-Side Java Platform: …



Scripting…

- It's arguable that the JVM is the best execution platform for scripting languages
  - Ubiquity
  - R&D investment in performance, debugging, telemetry …
  - Interoperability between hosted languages (JVM lingua franca)
  - …
- Scripting Languages bring added value to the server-side:
  - Rails
  - PHP's rich libraries
- But do such environments want to be "polluted" by Java platform?
  - Certainly the intent of Groovy
  - JRuby & Rails?
- Will they thrive?

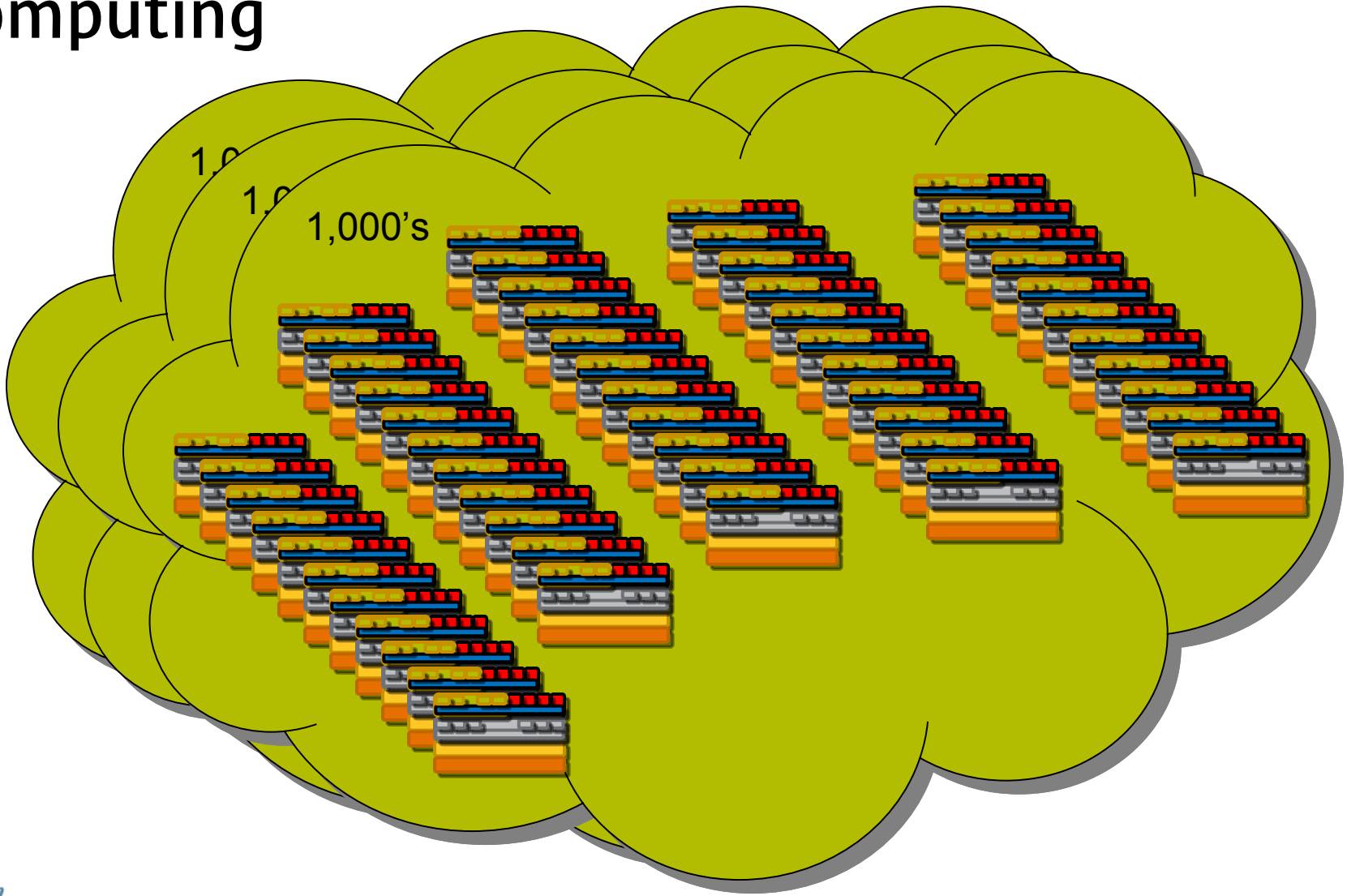
SCA


System
Composite X
Service
Wire
Composite Y
Component A
Component B
Wire
Composite Z
Reference
Wire
implementation
Composite A
implementation
Composite B

Whither SCA?

- SCA is a set of specifications that:
  - define Java platform, C++, BPEL component model(s)
  - define an assembly model enabling protocol independent “wiring”
  - JMS, EJB software and Web Service bindings
  - An extensible policy model
  - …
- Response to .NET/WCF
- But does server side Java platform need another component/container model?
- For more details goto: http://www.osoa.org

Server-Side Java Platform meets Cloud Computing



Cloud Computing? … Enterprise Java Platform Fabric!

- Amazon, Google, IBM, SalesForce, MSFT SaaS platforms
- But what about the Enterprise?
  - Multiple data centers (global)
  - 10,000's of machines - virtualized
  - 1,000's of applications – virtualization enables applicances
  - Enterprise SOA
- The real opportunity for "cloud" is inside the Enterprise!
  - Example: Infiniflow™ Service Fabric (Newton open source) from http://www.paremus.com and http://newton.codecauldron.org

THANK YOU

Larry Cable, BEA Systems Inc.

TS-7866