Enterprise Java goes OSGi:

The SpringSource dm Server

Adrian Colyer, CTO, SpringSource

Agenda

- What is the SpringSource dm Server?
  - and whatever happened to S2AP???
- Foundations: why OSGi?
- Why OSGi for *you*?
- Getting there from here
- The coming revolution
  - how do OSGi and the dm Server fit?

SpringSource dm Server

- It's an application server
  - but not as you know it
- It's open source
- It's "designed for Spring"
- It's built on OSGi
- It provides an OSGi-based deployment model for those that want it

http://www.springsource.com/products/suite/dmserver

SpringSource dm Server

- Deploy
  - Standard WAR files
  - Native OSGi bundles
  - Web modules (bundles)
  - “PAR” files
    - a group of bundles that together comprise an application
- Configure
  - dm Server is built out of configurable subsystems
    - use only what you need

Foundations: why OSGi?

"[IBM] has been shipping WebSphere Application Server built on OSGi since 2006. As a result, IBM clients benefit from a modular platform built with proven components and the ability to automatically use only the components required by their application."

Craig Hayman, VP, IBM WebSphere

http://www.osgi.org/wiki/uploads/News/2008_09_16_worldwide_market.pdf

“Oracle WebLogic Server is a great example of the customer benefits of modularization, with its reduced footprint, improved startup time, and flexible configuration options. OSGi technology provides the standards based foundation…”

Steven G. Harris, SVP Development, Oracle

http://www.osgi.org/wiki/uploads/News/2008_09_16_worldwide_market.pdf

“Running OSGi technology in JBoss Enterprise Middleware Solutions enables our customers to deliver safer services and applications in a more dynamic runtime environment.”

Sacha Labourey, VP Engineering, RedHat

http://www.osgi.org/wiki/uploads/News/2008_09_16_worldwide_market.pdf

“Sun has seen strong demand for OSGi technology within the GlassFish community. The GlassFish community will be able to take advantage of the modularity and dynamic extensibility implemented via an OSGi-technology based microkernel in the upcoming GlassFish v3 Prelude Release.”

Tom Kincaid, Executive Director, Application Platforms, Sun Microsystems

http://www.osgi.org/wiki/uploads/News/2008_09_16_worldwide_market.pdf

"Leading vendors using OSGi technology include IBM's WebSphere, Oracle's WebLogic, Paremus' Infiniflow Service Fabric, ProSyst's ModuleFusion, Red Hat's JBoss, SpringSource's SpringSource Application Platform and Sun Microsystems' GlassFish Enterprise Server.

Both Oracle and SAP AG have announced that they will use OSGi technology as the foundation for their next-generation application servers. "

OSGi Alliance Press Release, Sept. 16th 2008

http://www.osgi.org/wiki/uploads/News/2008_09_16_worldwide_market.pdf

Building on OSGi

- *Potential* end-user benefits:
  - modular platform
    - use only what you need
    - reduced footprint
    - improved start-up time
  - dynamic runtime environment
    - add and remove capabilities at runtime

Key questions

- *How modular is your server under the covers?*
  - determines your ability to match configuration to your requirements
- *To what extent are the dynamic capabilities of OSGi supported in your server?*
  - determines your ability to alter characteristics of a running system

spring source

Why OSGi for *you*?

Potential benefits

- Modularity
- Dependency management
- Deployment formats
- Runtime view of application wiring
- Dynamic add/remove/*update*
  - developer productivity
  - production change management

On Modularity

"I have been a fan of modular design even before I switched to my first Object Oriented Language. Although languages like Visual Basic, C++ and Java provide you with the means of building modular software, none ever enforce it. *No matter how strict I design modules in my application, they somehow always get tangled up in each other.*

That is until I started working with the beta of the SpringSource Application Platform (S2AP). *By dividing my application into bundles I can make a truly modular application.* Sure, this was already possible with OSGi runtimes, but that required a lot of plumbing and platform management."

- Jan-Hendrik Kuperus, blog, 23rd September 2008.

On deployment formats

"What I really enjoyed was the possibility to use bundles of Spring wired beans as first class citizen applications. Gone is the need for wrapping Spring in a web application to get a proper life cycle... POJOs rule!"

- David Dossot (blog) September 23rd 2008
http://ddossot.blogspot.com/2008/09/soon-serving-spring.html

On dynamic modules

To exercise this sweet platform beyond the obvious, I have built the prototype of a JMS-driven application. The architecture was pretty simple: a bundle using Message Driven POJOs to consume a Sun OpenMQ destination, which then informs clients of new messages through a collection of listener service references. *The idea was to allow the deployment of new clients or the hot replacement of a particular client at runtime. Verdict? It just works.* The SpringSource dm Server goes to great length to isolate you from bundles going up and down (depending on the cardinality of your references to OSGI services, you may still get a "service non available" exception).

- David Dossot (blog) September 23rd 2008
http://ddossot.blogspot.com/2008/09/soon-serving-spring.html

Potential blockers

- Simple programming and configuration model to exploit OSGi services
- Developer tool support for creating bundles
- Support for existing enterprise libraries
  - designed without OSGi in mind
- Availability of third-party libraries as bundles

Programming model

- Spring Dynamic Modules
- Standardization under OSGi R4.2: “Blueprint Service”

On the programming model

"The benefits of dynamic modules are well known, and mostly derived from the OSGI architecture. The goodness Spring adds on OSGI is the clean model for service declaration and referencing. This is priceless as it enables a true highly cohesive and loosely coupled application architecture within a single JVM. If you come from a world of EJBs and troubled class loaders, this is the holy grail."

- David Dossot (blog) September 23rd 2008
http://ddossot.blogspot.com/2008/09/soon-serving-spring.html

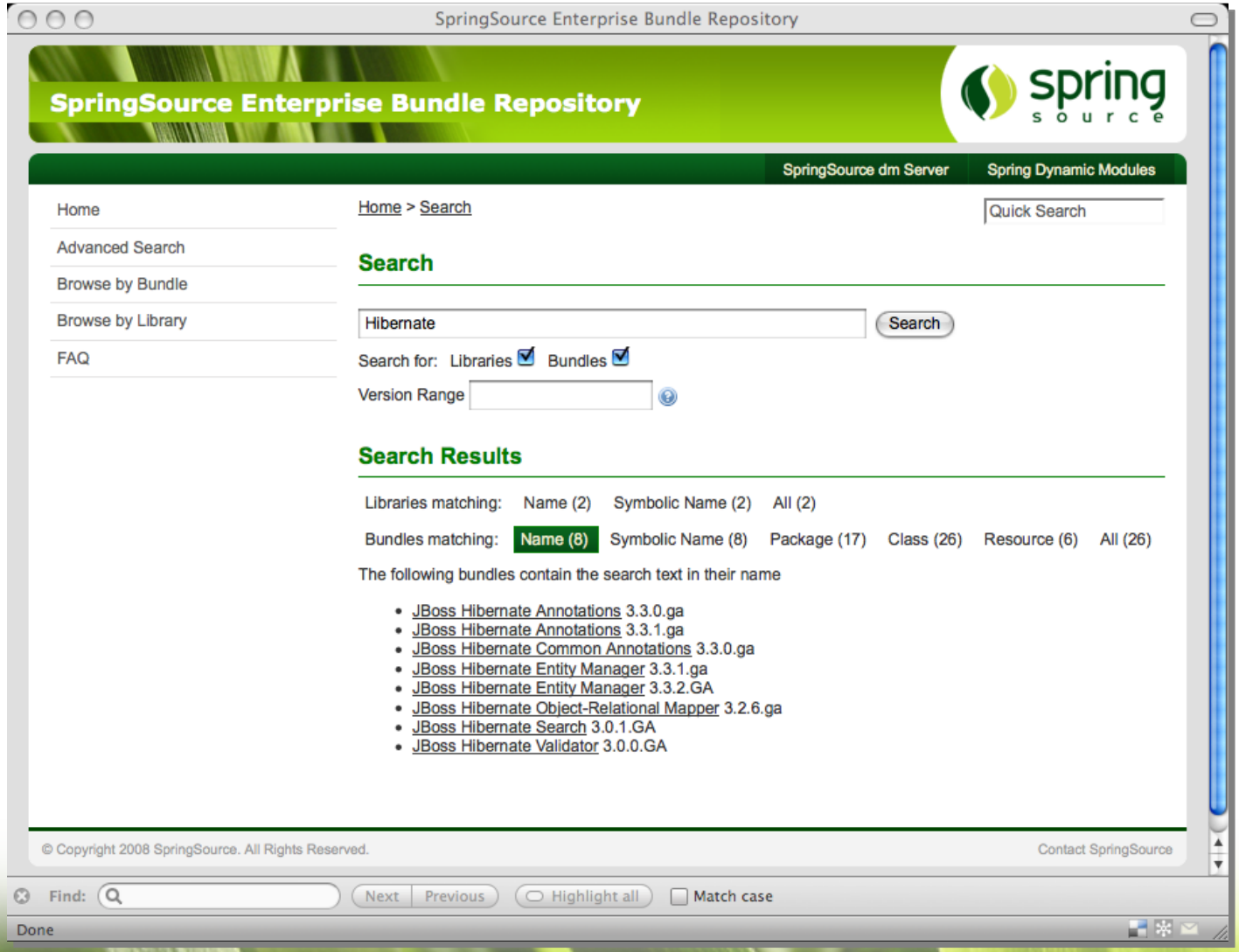
Developer Tools

- "dm Server" tools plugins
  - available standalone and as part of SpringSource Tool Suite
  - EPL

Existing libraries

- Extensive support inside dm Server
  - class path scanning
  - load-time weaving
  - resource loading
  - application context
  - ...

Enterprise Bundle Repository



Key questions

- Can I deploy my own applications as OSGi bundles?
- Do you offer a standards-based OSGi programming model?
- I need to use existing libraries in my application, how are they supported in your platform?
- Do you package your own offerings as OSGi bundles?
- Where can I easily find OSGi-ready versions of commonly used libraries?

Getting there from here

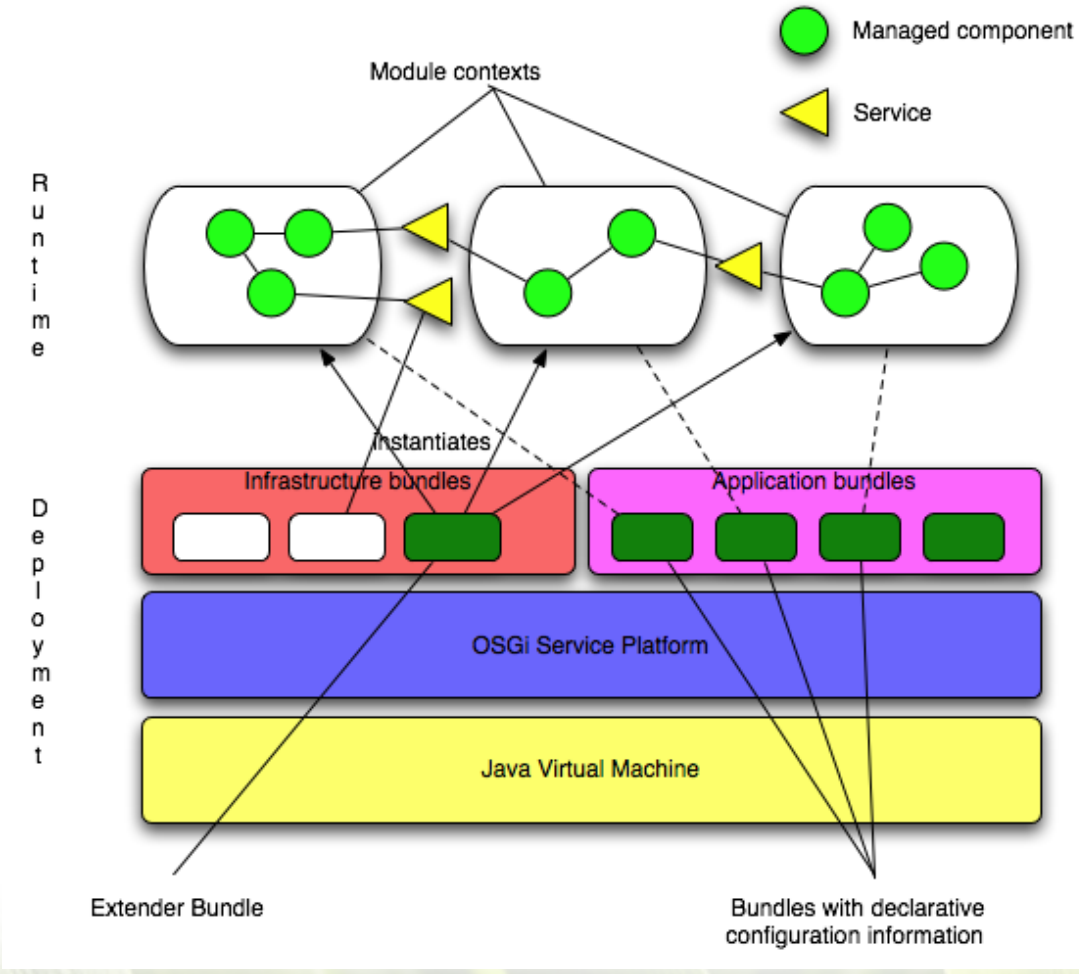
From war file to par file

- Deploy standard war files onto dm Server
  - Tomcat inside as the web container
- Incrementally take advantage of OSGi-based features
  - Dependency management (shared libraries)
  - Modularity & visibility (shared services)
  - Native deployment formats (module types and par file packaging)

Dependency Management

- Import-Package: org.somelib.foo;version="[1.0.0,1.1.0)"
- Import-Bundle: org.somelib;version="[1.0.0,1.1.0)"
- Import-Library: org.somelib;version="[1.0.0,1.1.0)"

Shared Services



Module Types

- Foundation for “personalities”
- Module-Type: xxx
  - Module-Type: Web
    - optimised directory layout
    - automatic creation of web.xml
    - optional module-type specific headers
      - Web-ContextPath, Web-DispatcherServletUrlPatterns, Web-FilterMappings
- More module types on the way

PAR files

- Application-SymbolicName
- Application-Version
- Application-Name
- Application-Description

- Unit of scoping, context class loading, management, deployment

The coming revolution?

Global class

- The case for Evolving Enterprise Platforms
  - Massimo Pezzini, VP Distinguished Analyst, Gartner
  - 22nd September 2008
- Notion of an application platform is transforming

Gartner recommendations

- If you invest in Spring today, it is your insurance policy for the future
- Invest in a platform that can deploy equally on premises and in the cloud
  - form of insurance, even if you don't need those capabilities today

dm Server in a virtual world

- Efficient use of virtual resources
  - makes multiplexing more effective
- Adaptation to changing workloads
  - both inter-vm (elastic) and intra-vm (adaptive)
- Foundation for provisioning
  - requires the ability to reason about dependencies and version compatibility

Summary

"The [dm Server] takes away all hassles and allows me to focus on designing and implementing a truly modular application. Adding a few OSGi-related configuration directives to my context xml enables almost all advanced features of OSGi. Combine that with the huge amount of pre-bundled enterprise libraries and the full Spring Framework at your disposal and you have the ultimate enterprise application platform!"

...

Summary

...

"I have been completely taken in by the ease of development and deployment with the SpringSource [dm Server]. If you ask me, the SpringSource dm Server is the future of Java EE and other application servers will have to follow in its direction."

- Jan-Hendrik Kuperus, blog, 23rd September 2008.

http://blojsom.jhkuperus.nl/blog/default/Java/SpringSource-Application-Platform-My-Holy-Grail?smm=y