BEA’s microService Architecture and OSGi: How Customers Benefit

Craig Blitz
Senior Product Manager, BEA Systems
Pre-SOA: Apps are Silos
SOA: Apps are Assemblies of Services

Service Infrastructure

App 1
Svc1
Other
Framework
Enterprise
Container
Virtual Machine
Platform 1

App 2
Svc2
Other
Container
Virtual Machine
Platform 2

App 3
Svc3
Other
Enterprise
Container
Virtual Machine
Platform 3

= non-service
= service
mSA: Platform Itself is Services

Service-Oriented Architecture

Service Infrastructure

App 1

App 2

App 3

Svc1

Svc2

Svc3

Platform 1

Platform 2

Platform 3

= non-service

= service
BEA’s microService Architecture (mSA)

• Create innovative products from existing components
• Naturally “blendable”
  • Products built from “best for purpose” modules
  • Proprietary BEA, Open Source, Third-Party
• Services plug into OSGi Backplane
  • Application Frameworks
  • Infrastructure Services
  • Activity Services
  • Presentation Services
mSA Design Principles

• Service-based infrastructure software
  • Lightweight, open, interoperable, embeddable
  • Naturally extensible for 3rd party development
  • Standards-based leveraging OSGi, SCA, etc.

• Based on the principles and philosophy of SOA
  • Service Network concepts
  • Modular and lightweight
  • Separation of concerns as opposed to point-to-point integration
OSGi: mSA’s Backplane

- OSGi key enabler to realizing mSA
- The Standard Java modularity environment
- Pluggable, service-based network
- Universal Middleware
  - Flexible, adaptable, embeddable
  - Allows mSA-based products to span broad variety execution environments
  - Natural affinity to virtualization solutions
OSGi/mSA Customer Benefits: Reduced Footprint

- Reduced footprint leads to improved IT efficiency
  - Reduce Operating Costs
  - Control Product Usage
- Pre-OSGi products
  - One-size fits all products
  - Add functionality to monolithic stack
- OSGi-based products
  - Tailored module sets to target problem at hand
  - Remove unused functionality
  - Package only needed functionality
  - Deploy only required services
  - Start/stop services on demand
OSGi/mSA Customer Benefits: Improved Serviceability & Availability

• Dynamic module management
  • install, start, stop, update and uninstall bundles
  • Extensive dependency management
  • Fully integrated with security architecture
• Enables zero down-time patch/upgrade
• Add functionality on-demand
• Remove services that are no longer needed
OSGi/mSA Customer Benefits: Improved Extensibility

- Controlled environment for adding modules
  - Class-loading model protects private resources
  - Replaces single class-path model or proprietary solutions
  - Built-in versioning
  - Run multiple versions of module in same application
  - Service registry to control dynamic interactions among modules
- Simplifies and standardizes third-party module integration
- More predictable interactions between modules via service registry
WebLogic Event Server: 100% mSA

the FIRST and ONLY Java container for High Volume, Real Time, Complex, Event-Driven applications

- **High Throughput**
  - >50,000 complex events per second
- **Complex Event Processing**
- **Extreme Low Latency -- Microseconds**
- **100% Latency Guarantee**
- **Purpose-built**
  - Specifically High Volume, Low Latency Complex Event Processing applications
  - 100% mSA – a BEA First

**WLEvS Conceptual Function**
WebLogic Event Server

Stream Processor

CEP Engine

connectivity

jetty

Real-Time Kernel

enterprise

security

JAXB

enterprise-base

http-base

spring

base

OSGi Framework

JRockit / WebLogic RealTime
BEA mSA and OSGi

- mSA is using principles of SOA to deliver new capabilities to its customers
- OSGi is a key, standards-based enabler of mSA
- New products continue to roll-out on mSA
- Resources
  - [http://www.bea.com/msa](http://www.bea.com/msa)
  - [http://www.bea.com/eventserver](http://www.bea.com/eventserver)
  - [http://osgi.org](http://osgi.org)