GlassFish V3
&
OSGi

Brainstorming Session -I
04 June 2008, SCA

Sahoo@Sun.COM
Sun Microsystems

Agenda

- State of GlassFish V3 and OSGi
- Work to do
  - Short term
  - Long term

State of the GFv3 and OSGi

- All our modules have been bundlized
- They are able to run on Felix, Equinox, Knopflerfish, as well as HK2

State of the GFv3 and OSGi - contd

- We are NOT a good citizen in modular world – we are not even close to being good
- Too many exported packages (790)
- We have split packages
- We rely on Require-Bundle
- We have too many inter-dependencies (1364)
- Not able to replace parts by alternative implementations

How to improve the situation?

- Come up with manageable number of modules
- Define boundaries of modules
  - Identifies what you offer (APIs)
  - Identify how you are extended (SPIs)
  - Use package names to reflect these notions
  - Organise source code accordingly
    - http://wiki.glassfish.java.net/Wiki.jsp?page=V3Adm
    - This allows us to identify owners, separate functional areas into their own projects

Goals

- Parts of the system can be updated at runtime without having to restart
- New capabilities can be added at runtime
  - Basic fw already exists
  - Needs integration with OBR
- Mix & Match (more on next slide)
- Support New Application Model

Mix & Match - Examples

1. Jetty or Tomcat as the Web container, yet http connector is Grizzly
2. Mina or something else as the NIO framework instead of Grizzly
3. GlassFish web container in “other” OSGi enabled application servers.

New Application Model

1. OSGi web application
   1. OSGi applications making use of OSGi HTTP Service
2. Hybrid Web Applications
   1. Use a javax.servlet web container for Web facilities
   2. Use OSGi facilities for dependency management, life cycle, services
      1. Improved code sharing (extension mechanism is a failure, not because it is bad, but the way it is implemented)

Shot Term (v3 lite) Deliverables

1. Switch to Import-Package
2. Defining boundaries of modules
3. Export-Package to use proper versions
4. Remove dependency on class loader hierarchy
   1. Provide an option to switch to use library directory for backward compatibility reasons
5. Embeddable in other OSGi containers (e.g. Eclipse IDE)

Shot Term (v3 lite) Deliverables

1. Add support for direct deployment of OSGi bundles
   - They could be libraries, applications, etc.
2. Library sharing by applications
3. Administration of OSGi bundles via admin console, cli, etc.

Long Term Deliverables

1. Support new application model
2. Keep track of OSGi EEG and implement relevant stuff.