MEG Overall Architecture
by Peter Kriens, CEO aQute, OSGi Fellow
History

- OSGi developed a generic Service Platform between 1998 and 2003
- The OSGi goals were deemed too ambitious for mobile phones in that time frame
  - MIDP 1.0 and 2.0
  - Simple and constrained
- Next generation of mobile phones will have enough power to run more ambitious applications than games
- Enterprise applications require a more ambitious platform architecture
- The requirements for such an architecture meshed nicely with the OSGi Service Platform
Goals

• The OSGi was missing
  – A simple application model that is similar to MIDLets
    • Simple
    • Fool-proof
  – A *standardized* Deployment and Device Management model
    • The OSGi standardized APIs to allow a diverse range of management practices
    • Mobile operators require standardized protocols
OSGi Service Platform

- Comprehensive standard for deploying Java applications to networked devices
- Key component is the bundle:
  - Java Archive
- Bundles are the foundation for other services
Application Model

- Generic Model for different types of applications
  - MEG Applications
  - MIDlets
  - Native Applications
- Standardized interaction between different application models and e.g. a desktop manager
MEG Application Container

- Available Apps
- Running Apps

- Application Manager
- Application
- Application Descriptor

- Application Container
- MEG Application Container
- MIDP Container
- Native Container

«Derive» «abstraction»
«interface»
«interface»

© copyright 2004 by OSGi Alliance. All rights reserved.
MEG Application Container

- Meglets are normal OSGi bundles
- The Meglet container detects meta information in the bundle and registers the Meglet with the Application Manager
- The Meglets can then be started by a “desktop” manager application
MEG Application Container

- Available Apps
- Running A...

«interface» Application

Application Manager

Application Container

«Derive» «abstraction»

MEG Application Container

MIDP Container

Native Container

«interface» Application Descriptor
Declarative Services

• Most complicating factory in OSGi programming is the dynamics
  – Services and bundles can come and go at any time
• These dynamics require very careful programming and testing
• Declarative services remove the dynamics from the view of the programmer
• A programmer never see services come and go (if so desired by that programmer)
• Result
  – Easier to program
  – More reliable

© copyright 2004 by OSGi Alliance. All rights reserved.
MIDP Container

- Provides a backward compatible environment with MIDP 1.0 and 2.0
- MIDLets can be used in the same way as Meglets or native applications
- Provides the opportunity to download JSRs that become available to the MIDLets
- MIDP containers are likely to be integrated with the native environment
Device Admin

- Device Admin provides the interface between the management center and the mobile device
- Device Admin is based on the OMA Device Management Tree
  - Becoming very popular
  - Unified management of diverse tasks
- Nodes in the DMT can be implemented as OSGi services
  - DMT plugins
- The model can be used with native aspects as well
Architectural Overview
Deployment Admin

- The Bundle Suite is a new artifact that contains bundles, configuration, and other resources
- Bundle Suites can be installed and uninstalled
- Deployment Admin
  - Manages dependencies (with versioning)
  - Maintains the correct state of the system
Conclusion

• The Mobile Expert Group significantly extends the OSGi Service Platform
  – Application model
    • Integration of native, midlets and meglets
    • An programmers model
  – Deployment Model
    • Bundle Suites
  – Device Management
The End