Enabling the Enterprise with Next-Generation Mobile Architectures
Speaker and Agenda

• Speaker:
  – Mark VandenBrink (vmark@us.ibm.com)
  – IBM; Mobile device client architecture, Linux/Java
  – In a previous life, MIDP 1.0 EG lead (and well on my way to a complete recovery 😊)

• Agenda:
  – Enterprise applications mini-primer and use cases
  – Current mobile device clients (MIDP)
  – MEG: the future of mobile device clients
Enterprise Applications

• What is an “enterprise application”?  
  – Canonical example is PIM access (email, calendar, etc.)  
  – “Project existing enterprise services into various client channels”

• Client-side Requirements:  
  – Have low TCO  
  – Little-to-no deployment costs  
  – Be predictable, yet flexible  
  – Available and scalable  
  – Secure  
  – Capable of “off-line” processing
Enterprise Programming Model

- Dynamically managed
  - Platform
  - Applications
  - Richer experience
    - Alternate View/Control choices
    - Disconnected operation
• **Spacely Sprockets**
  – Provides a suite of e-Commerce web applications for business partners to create and submit orders for sprockets online
  – Catalog of available sprockets, current inventory, and prices change frequently

• **Spacely Sprockets Business Partners**
  – Would like to run the e-Commerce web application suite while at customer sites
  – Would like to use mobile phones as client tier
Scenario Requirements

• **Spacely Sprockets**
  - Leverage existing J2EE/portal-based application
  - Extend the application to support
    - Richer interaction on client tier
    - Alternative client devices
    - Disconnected use

• **Business Partners**
  - Use COTS equipment for client tier
Scenario: Browsing

Client-tier
- User Agent
  - Scripts
  - *ML

Client-tier
- View
  - View Logic
  - View Data

Middle-tier
- Controller
- Mediator
  - Business Logic
  - Business Data

Middle-tier
- Mediator
  - Transactional Messaging

Backend-tier
- Model
  - Business Logic
  - Business Data

© copyright 2004 by OSGi Alliance. All rights reserved.
Browsing Attributes

• Capability
  – Connected: render dynamic, network content
  – Disconnected: render (limited) cached content

• When to apply
  – Connected: When connected, subject to latency, bandwidth, and coverage concerns. R/W access
  – Disconnected: When not connected, R/O access

• Implications on Scenario
  – Connected: review pricing, place orders, etc.
  – Disconnected: review pricing only
Scenario: Offline Apps

Client-tier

User Agent
- Scripts
- *ML
- Markup

View
- View Logic
- View Data

Controller

Model
- Business Logic
- Business Data

Backend-tier

Model
- Business Logic
- Business Data

Transational Messaging

Mediator

© copyright 2004 by OSGi Alliance. All rights reserved.
Offline Apps
Attributes

- **Capability**
  - Provides local application programming model that interacts with external applications and services

- **When to apply**
  - When there is need for disconnected operation
  - When richer, alternate user interaction is needed (*i.e.* non-browser based)

- **Implications on scenario**
  - Spacely Sprockets business partners with little or no network connectivity can be fully serviced
Mobile Clients Today: MIDP

- Mobile Information Device Profile
- Most recent: Version 2.0
- Widespread adoption in mobile world
- Unit of deployment/packaging: MIDlet Suite
- Unit of execution: MIDlet
- Can act as client-tier via web services
MIDP for Enterprise Apps?

- Limited end-to-end model
  - Security, identity management, *etc*
  - Client/server tools
  - Weak (platform-level) support for disconnected operation (*e.g.* messaging, synchronization, *etc.*)
  - Unmanaged
- Relatively high cost of deployment for MIDlets
- No “shared component” support (by design)
- Limited interaction support (MIDlet only)
- Limitations can be overcome, but results in MOAM (Mother Of All MIDlets)
OSGi: Mobile Expert Group (MEG)

- Formed in Dec 2003
- Working on enabling a next-generation mobile phone architecture
- Based on V4 OSGi Framework
- Linked to JCP via JSR-232
MEG: Enterprise Attributes

• Allows for different user-interactions via “container” notion
• Allows for shared components via services
• Enterprise features designed in from start:
  – Security
  – Policy
  – Deployment
The End