A RESTful Architecture for Adaptive and Multi-device Application Sharing

Vlad Stirbu
WS-REST2010
Raleigh, NC, 26.04.2010
Vlad Stirbu
- Senior Researcher
- Nokia Research Center, Tampere, Finland

3D Platforms Team
High Performance Mobile Platforms
- Ubiquitous interoperable architectures (extending the single device)
- Energy efficient computing architectures

Background
Application sharing

Simple sharing
- Render the UI as instructed by the application
  - Render screen captures
  - Drawing commands

Adaptive sharing
- Ability to adapt the UI to the local device look and feel
  - Descriptive elements
Interactive applications = model-view-controller
The remote model-view-controller

- Extension of classical MVC
- Replicate the View and part of the Controller or a user agent residing a remote device
- Keep the user agent and the application synchronized using an event-propagation mechanism
RESTifying the remote model-view-controller

• Resources
  • For view: http://app.example.com/ui/{uiElement}
  • For controller: http://app.example.com/ui/on_{uiElement}_{eventName}

• Unified access
  • The four HTTP verbs (e.g. GET, POST, PUT, DELETE) and WATCH

• Representations
  • Specific user interface description, e.g. XAML, Qt UI XML
Event propagation mechanism

- Orchestration
  - WADL manifest allows the user agent to interact with the relevant application resources
  - Performed during initialization
  - Subsequent errors trigger re-initialization
Implementation experience

- Extension of Qt application framework
- Implemented using Python bindings for Qt, e.g. PyQt or PySide
- Web backend provided by web.py
Conclusions

• User interfaces modeled using REST architectural style
  • View and controller explicit resources
  • HTTP content negotiation allows adaptations to user agent characteristics
• Leverage existing GUI toolkits development tools and developer communities
Questions?