WSRF.NET Overview

Glenn Wasson
Marty Humphrey
University of Virginia
WSRF.NET is…

• An implementation of the Web Services Resource Framework (WSRF) and Web Services Notification (WSN) families of specs
  – WS-BaseNotification, WS-Topics, WS-BrokeredNotification
  – Using WS-Addressing 2004/03 (same as GT4)

• Version 2.0 now available!
WSRF.NET Details

• Provides a programming model, APIs and tools for building WS-Resources
  – Define stateful resources using .NET attributes
  – Integration with databases for back-end storage (e.g. Yukon, SqlExpress, Xindice)
  – Service aggregation tools

• WSRF.NET services are MS web services
  – Run on top standard MS infrastructure (e.g. IIS, ASP.NET and WSE 2.0)
WSRF.NET Features

- WS-Resources can be state or WIN32 process
- Notification infrastructure
  - Including light-weight client-side receiver
- Integration with Powerful DBs
  - But, you can write your own storage/retrieval code
- Attribute-based programming model
  - .NET supports many languages
- Many included services
  - Subscription manager, Notification Broker, etc.
Differences with OGSI.NET

• Largely, they are the differences between OGSI and WSRF
  – No GSHs, SDEs, gwsdl, etc. (mentioned in GT4 talk)
• WSRF.NET runs on “vanilla” MS infrastructure
  – OGSI.NET had its own container
• Persistence addressed more comprehensively
  – Services were stored in memory, resources can be stored in many ways
WSRF.NET Example

1. What resources are available for my application?
2. Does this user have an account in this VO?
3. Available Exec/Data Services
4. Reserve resources
5. Create new reservation under client’s DN
6. Create new data resource
7. Stage-in data
8. Start application
9. Claim reservation by lengthening resource’s lifetime
10a. Data input/output
10b. Launch job

Authorization based on DN, all messages X509 signed

WS-Resources are accounts
WS-Resources are allocatable resources
WS-Resources are reservations
WS-Resources are directories
WS-Resources are processes
More Information

• Project web site:  
  http://www.ws-rf.net

• Version 2.0 available!
• Documentation
• Papers
• Tutorial
PyGridWare
Overview

Keith R. Jackson
Joshua R. Boverhof
Lawrence Berkeley National Laboratory
Python WS Core

- Implementations of the Web Services Resource Framework (WSRF) and Web Service Notification (WSN) family of specifications
  - Based on June 2004 OASIS schemas with minor changes
    - Using 2004/03 version of WS-Addressing specification.
  - WS-BrokeredNotification not supported
- Compatible with the GT4 Java WS Core
A Closer Look

• Provides API and tools for building WS-Resources (Web services that operate on stateful resources)
• Provides Python binding generation from WSDL
  – Focus on WS-I BP-1.1 compliance
• Security
  – Transport layer (https)
  – Secure Conversation
  – Secure Messaging
A Closer Look (cont.)

• Implemented using standard Python software
  – ZSI for SOAP handling
    • Much of our work has been contributed back to ZSI
  – 4Suite and PyXML for XML
  – Twisted for the hosting environment and event loop
pyGridWare Features

- A lightweight standalone container
- Automatic service startup on container start
- Basic API for resource persistence and recovery
- Notification infrastructure:
  - Subscription Manager/Notification Producer
  - Lightweight Notification Consumer
pyGridWare Features

• Security infrastructure:
  – Pluggable handlers for WS-Security
  – Secure Conversation Service

• Support for wrapping legacy codes as Grid Services
More Information

• Project site
  – http://dsd.lbl.gov/gtg/projects/pyGridWare/

• ZSI
  – http://pywebsvcs.sourceforge.net/zsi.html

• Twisted
  – http://twistedmatrix.com/products/twisted

• 4Suite
  – http://4suite.org/index.xhtml
GT4
C WS-Core
Overview

Sam Lang
Joe Bester
December 2004
C WS Core

- Implementation of the Web Services Resource Framework (WSRF) and WS-Notification (WSN) family of specifications
  - WS-ResourceProperties support: No QueryResourceProperties provider
  - WS-Notification support is only receiver-side (NotificationConsumer)
A Closer Look

- **Provides C Binding Generation from WSDL**
  - Pure C parser/generator
  - Doc/lit support only
- **Provides C API for building services with WS-Resources**
- **Builds on Globus C Asynchronous Events**
- **Security**
  - Transport
  - SecureMessage (Integrity only)
A Closer Look (cont.)

- Built on efficient pull parser API
  - Support for XSD Wildcards
- Pluggable Message Handlers
  - WS-Addressing
  - WS-Security SecureMessage
  - Debugging
Clients

- End-to-end solution for WSRF-enabled clients in C
  - Good start-up time
  - Pure C types and interfaces (asynchronous and blocking)
  - EPR encapsulation
  - RP querying
  - Asynchronous NotificationConsumer API
Services

- WS and WSRF support, some WSRF support
  - Doc/lit operation invocations
  - WS-I Basic Profile conformance
  - WS-Addressing by default
- Services as Dynamic Modules
- Operation Providers for WSRF operations
- Pure C Container and Embeddable API
GT4
Java WS Core
Overview

Jarek Gawor, Sam Meder
December 2004
GT4 Java WS Core Implementation

- Implementations of the Web Services Resource Framework (WSRF) and Web Service Notification (WSN) family of specifications
  - Based on June 2004 OASIS schemas with minor changes
    - 2004/03 version of WS-Addressing specification
  - WS-BrokeredNotification not supported
A Closer Look

- Provides API and tools for building WS-Resources (Web services that operate on stateful resources)
- Clear separation between services and resources
  - **Services**
    - Stateless
    - Perform business logic
    - POJO
  - **Resources**
    - Stateful
    - Contain/represent the state
    - Managed by ResourceHome
- Implemented with ‘standard’ Apache software
  - Axis (the web service engine)
  - Addressing (the WS-Addressing implementation)
  - and more
Java WS Core
GT 4.0 Features

- A JNDI based registry
  - Used for discovery of ResourceHomes, etc.
- An implementation of the Work Manager and Timer (J2EE) specifications
  - Used for starting and managing background threads and periodic tasks
- A standalone and embeddable container
- Tomcat support
- Automatic service and ResourceHome activation on container startup
- Service operation providers
- Basic API for resource persistence and recovery
- Persistent subscriptions support
Differences from GT3 Java WS Core

- Resource vs. Service Instances
- No 2-level naming scheme (no more GSHs)
- Notifications are no longer tied to service data
  - Topics & Topic spaces
- No gwsdl
- No base porttype
- No pre-defined factory class/porttype
Performance

Messaging Performance

- Axis Update Branch (1/10/05)
- CVS Head (1/10/05)
- CVS Head (11/05/04)
- CVS Head (11/01/04)
Example Service

- getResourceProperty Operation Provider
- Service Specific Operation Provider
- subscribe Operation Provider
- Destroy Operation Provider

Resource Home

Service Resource

<table>
<thead>
<tr>
<th>Resource Properties</th>
<th>Persistence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topics</td>
<td>Lifetime</td>
</tr>
</tbody>
</table>
Want more?

- Current GT4 (pre-release) documentation: http://www-unix.globus.org/toolkit/docs/development
- The GT4 tutorial: http://www.casa-sotomayor.net/gt4-tutorial/