Asynchronous Peer-to-Peer Web Services and Firewalls

Aleksander Slominski¹, Alexandre di Costanzo²
Dennis Gannon¹, Denis Caromel²

¹) Indiana University, ²) INRIA Sophia Antipolis
Motivation

- SOAP: better RPC? Internet RPC?
- Messaging: WS-Addressing, WS-RM
- Future of Web Services?
  - Asynchronous
  - Peer-to-peer interactions
  - Long conversations, hours, days, ...
What is needed?

- Reliable and secure peer-to-peer interactions between Web Services peers that exchange messages
- Other features: MOMs and beyond
  - Provide load balancing,
  - Service location transparency.
  - Single sign-on,
  - ...
Asynchronous P2P WS Communication

Logical View

Service

WS-RM

Service

Services

Conversations
P2P WS? Not so simple ...

- Limited supply of IPv4 addresses
  - IPv6 address space?
- Network Address Translation Systems (NATs)
  - Routers supporting IPv6?
- Firewalls
  - Inaccessible IP address
- Clients that can not listen for messages
  - Applets ...
Related Work

- SOAP 1.2 Intermediaries
- IBM Web Service Gateway
  - WebSphere Application Server Network Deployment Version 5
  - Multi-protocol by using WSIF
- ESB ...
  - ”Enterprise Service Bus”
  - IBM, BEA, Sonic Software, ...
Public and Private Web Services

- Service Client (RPC, MSG)
- Firewall, NAT, ..
- Externally Visible Service Endpoint (Translator, Intermediary)
- Internal Service (RPC, MSG, ..)
### Connecting Client and Service: Between RPC and Message WS

<table>
<thead>
<tr>
<th></th>
<th><strong>RPC based service</strong></th>
<th><strong>Message based service</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(synchronous request-response)</td>
<td>(asynchronous one-way)</td>
</tr>
<tr>
<td><strong>RPC client</strong></td>
<td><strong>Limited</strong>: RPC connection is forwarded and client may timeout, hard to keep connection opened for very long</td>
<td><strong>Very limited</strong> (will not work – message reply may come much later)</td>
</tr>
<tr>
<td>(synchronous</td>
<td></td>
<td></td>
</tr>
<tr>
<td>request-response)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MSG client</strong></td>
<td><strong>RPC service is bottleneck</strong></td>
<td><strong>Unlimited</strong> (client and service interaction can be separated in time)</td>
</tr>
<tr>
<td>(asynchronous one-way)</td>
<td>(translating Messaging to RPC ...)</td>
<td></td>
</tr>
</tbody>
</table>
Translator/Intermediary: WSD and WSMB Combo

- Web Services Dispatcher (WSD)
  - Intermediary
  - Mediate between RPC and Messaging
  - Gatekeeper to internal services
  - *Security, Load balancing, ...*

- Web Services Mailbox (WSMB)
  - Store-and-collect messages
  - Two services that are inaccessible
    - Both behind services firewalls
    - Applet accessing service that is behind firewall
Applets And Firewalls

Client (Applet, JWS, portlet…)

Dispatcher (Mailbox and Service Registry)
Setup: hosted on portal

Service inaccessible: behind firewall/NAT/…

Service inaccessible: no endpoint, inside browser, or behind firewall (or all)

Asynchronous request

Asynchronous response

WS-RM
Two Peer Services Behind Firewalls
Design and Implementation

- Implemented in Java
  - Heterogeneous problem, run everywhere
- XSUL: Java Library
- RPC Dispatcher
  - Not multi-threaded
- MSG Dispatcher
  - Multi-threaded:
    - Concurrent Java Library (now integrated in Java 1.5): Thread-pool, FIFO queue, concurrent hash map
Web Services / XML Services Utility Library (WS/XSUL)

- Java 1.3+
- Ultra Modular (30+ modules)
- Pluggable transport:
  - Currently mini HTTP stack
- Common abstraction for SOAP 1.1/1.2
- Lightweight WSDL 1.1 API
- Pluggable XML data binding
- Handlers for security, etc.
WS-Dispatcher Design
MSG-Dispatcher Implementation
WS-MsgBox Design
Evaluation

- Message size: 483B (3864bits)
- Remotes sites:
  - INRIA Sophia France: High Connection
  - Indiana University USA: Backbone Internet
  - Bloomington IN USA: Cable Modem
RPC Communication with Low Broadband
RPC Communication with High Connectivity
Asynchronous Communication
Conclusions

- A part of solution to enable P2P WS
- Performance is good enough
  - And can be improved

- Future
  - Better registry service (WSDL)
  - Load balancing
  - Security front-end (Gatekeeper)
Thank You!

- WS-Dispatcher Home: http://www.extreme.indiana.edu/xgws/dispatcher/

- Source code under BSD-like license