

**WAVV 2009**

# **SOA . . . Give It a REST!**

SOA vs. RESTful services

Karl DeVore

# Abstract

**Could SOA be the wrong answer for you? If SOA keeps you awake at night, maybe you need some REST!**

**IT teams tasked with integrating CICS and/or other mainframe resources into a services architecture know the challenges of SOA. Now along comes REST, and with it comes a whole new set of questions. Which do you choose, SOA or REST? Or do you need a combination of both? What services approach is right for your situation? And at what cost – to your team and your budget? And how do you make the best decisions?**

**We will provide an overview of how to navigate through the SOA vs. RESTful services terrain, and provide you with a practical SOA/REST decision-making framework based on technology and business value.**

# Acknowledgements

- ❖ *The Great Debate: Should I Use Web Services or REST for My SOA?* Gartner, Roy Schulte and Nicholas Gall, APN21\_129, 12/08, AE
- ❖ *RESTful Web services: The basics* IBM: Alex Rodriguez
- ❖ *RESTful Web Services*, O'Reilly Media, Inc., Leonard Richardson; Sam Ruby
- ❖ Wikipedia

# SOA

## Service-Oriented Architecture

- ❖ ***RESTful Web Services*** *O'Reilly - Leonard Richardson; Sam Ruby*
  - software architecture based on the production and consumption of web services and the use of those services as software components
- ❖ **Gartner** *Roy Schulte and Nicholas Gall*
  - architectural style for business applications that are modular, distributable, shareable and loosely coupled
- ❖ **Wikipedia**
  - methods for systems development and integration where systems group functionality around business processes and package these as interoperable services

# SOA

Service-Oriented Architecture

## ❖ The benefits of SOA

- Sharing
- Agility
- Integration
- Reusable

# SOA

Service-Oriented Architecture

**SOA = SOAP**

# SOAP

Simple Object Access Protocol

## ❖ SOAP

- Highly structured XML-based protocol to let applications exchange information

## ❖ Web Service Description Language (WSDL)

- XML vocabulary used to describe SOAP-based services

## ❖ Universal Description Discovery and Integration (UDDI)

- platform-independent, XML-based registry to publish service listings and discover each other and define how the services or software applications interact

# SOAP

Simple Object Access Protocol

- ❖ **WS-\***
  - Quality of Service (QOS)
- ❖ **WS-Security**
  - SOAP message security
  - Username token profile
  - X.509 certificate token profile
  - Conversation
  - Policy
- ❖ **WS-Trust**

# SOAP

Not so Simple Object Access Protocol

- ❖ WS-Federation
  - Active requestor profile
  - Passive requestor profile
  - Kerberos binding
- ❖ WS-Reliable Messaging
- ❖ WS-Addressing
- ❖ WS-Enumeration
- ❖ WS-Events
- ❖ WS-Transfer
- ❖ WS-Coordination
- ❖ WS-Atomic Transaction
- ❖ WS-Business Activity
- ❖ WS-Policy
  - Assertions
  - Attachment
- ❖ WS-Discovery
- ❖ WS-Metadata Exchange

# SOAP

Not so Simple Object Access Protocol

SOAP  $\neq$  Simple

# The Revolt

- ❖ **Started with the peasants**
  - <http://a.host.com/cgi/getCustomer?id=444-44-4441>
  - Plain Old XML (POX) as the return data
- ❖ **Moved up the ranks**
  - Third party software vendors (2000)
- ❖ **Fought against by the Oligarchies**
  - IBM, Gartner (1990s-2007)
- ❖ **Adopted by freethinkers and opportunists**
  - Google and Yahoo (2005-2006)
- ❖ **Then co-opted by the Loyalist**

# SOA

Service-Oriented Architecture

**SOA ≠ SOAP**

# SOA Is ...

- ❖ **SOAP**
- ❖ **REST**
- ❖ **REST-RPC Hybrids**
  - **HTTP+POX**
  - **XML-RPC**
  - **STREST**
  - **High and low REST**
  - **Query String → XML**

# REST

Representational State Transfer

- ❖ **REST defines a set of architectural principles by which you can design Web services that focus on a system's resources, including how resource states are addressed and transferred over HTTP**
- ❖ **Applications that use this architecture are sometimes referred to as RESTful**



# Method Information

What do I do with the data?

## REST Method

- ❖ The REST methods map to the existing HTTP protocol verbs: GET, PUT, POST, DELETE
  - GET = Query
  - DELETE = Delete
  - PUT = Create
  - POST = Update

# Scope Information

What do I do with the data?

## REST Query (GET) Method

- ❖ REST is human readable

<http://demo.abcsoftware.com/MSCRFILe/customer/444-44-4441>

```
GET /MSCRFILe/customer/444-44-4441 HTTP/1.1  
Accept: */*  
Host: demo.hostbridge.com
```

# Method Information

What do I do with the data?

## REST Update (POST) Method

```
POST /MSCRFILE/customer/444-44-4441 HTTP/1.1
```

```
Accept: text/xml
```

```
Content-Type: text/xml
```

```
Host: demo.hostbridge.com
```

```
Content-Length: 54
```

```
Connection: Keep-Alive
```

```
Cache-Control: no-cache
```

```
Authorization: Basic amFtZz213XFyMXJhYg==
```

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<customer>
```

```
  <phoneNumber>BR549</phoneNumber>
```

```
  <zipCode>74074</zipCode>
```

```
</customer>
```

# What are the decision points

- ❖ **Mandated**
- ❖ **Internal, external or both**
- ❖ **Complexity**
  - **Development time**
  - **Ease of use**
- ❖ **Performance**
- ❖ **Resource consumption**
- ❖ **Client code stub generation**
- ❖ **A2A, B2B or U2A**

# SOAP/REST Comparison

## SOAP

- ❖ QOS, WS-\*
- ❖ WSDL
- ❖ Complexity is high  
For the sophisticated developer, highly trained, great tool set
- ❖ Can quickly discover and consume services
- ❖ Development more time consuming, but precise
- ❖ Higher resource consumption

## REST & Hybrids

- ❖ HTTP
- ❖ WSDL , WADL, but why
- ❖ Complexity is low  
Average developer or even a power user with an open tool set
- ❖ Consumption of services uses open tooling
- ❖ Development is fast and easy
- ❖ Higher performance

# Summary

- ❖ **Keep an open mind, SOA is the goal**
  - Sharing
  - Agility
  - Integration
  - Reusable
- ❖ **SOA is not just SOAP, it is REST & the Hybrids**
- ❖ **SOAP may be for you**
  - Depending on your A2A or B2B applications
  - Require QOS (WS-\*)
- ❖ **Rest may be the better choice**
  - Development time
  - Performance
  - Developer skill sets
  - Tooling

# Gartner Quotes

Roy Schulte and Nicholas Gall, APN21\_129, 12/08,

- ❖ It is critical to understand when complexity is inherent because of business requirements vs. when complexity is gratuitous and could be avoided because it was introduced by bad, old ways of designing applications
- ❖ If interoperability does not have the requirement for multi-transport messaging or QOS, then there is no reason to use SOAP . . . . Over time, as they are modified, the SOAP-only implementations will be phased out. WOA will replace 90% of Web SOAP-only implementations by 2012.