Computing in the Cloud

("Ad Hoc" Keynote, CoopIS 2007, Vilamoura, Portugal, 11/28/07)

Prof. Dr. Frank Leymann
Institute of Architecture of Application Systems
University of Stuttgart
Universitätsstr. 38
70569 Stuttgart
Germany
Phone +49-711-7816 470
Fax +49-711-7816 472
e-mail Leymann
@aas.uni-stuttgart.de

Agenda

Virtualization Technology
  ➔ Web Services and the Grid
Two Styles
  ➔ REST vs. SOAP/WSDL
Business Impact
  ➔ SaaS, On Demand, Utilities
Business Processes
  ➔ How Business Drives IT
Summary

© Frank Leymann
### Agenda

**Virtualization Technology**
- Web Services and the Grid

**Two Styles**
- REST vs. SOAP/WSDL

**Business Impact**
- SaaS, On Demand, Utilities

**Business Processes**
- How Business Drives IT

**Summary**

### Terminology

- **Service**
  - A function provided at a network address
  - Available via various transports, formats, and QoS
  - It is “always on”, i.e. you do not have to create it, you do not have to care about destruction, etc
    - Like water, power, gas, telephone,…

- **Service Oriented Computing (SOC)**
  - The compute paradigm behind services
  - E.g. “distributed computing” is such another paradigm

- **Service Oriented Architecture (SOA)**
  - An architectural style to realize SOC
  - E.g. “client/server” is an architectural style for realizing distributed computing

- **Web Service Technology (WS*)**
  - A standard- and technology stack supporting SOA
  - E.g. “stored procedures”, “Web browser/server”,… support the client/server style
...the Net is: Virtualization of Software

I want ... But I don’t care about a particular provider: Chose the one who is at this point in time „the best“ for me!

Virtual Service

Service Bus

Environmental Properties

Service Description
-Function
-QoS
-Data

Business Properties

The Grid: Virtualizing Resources

→ Service Bus !

Virtual clusters and storage
The Combination: Computing On Demand

In 3 hours I need the following environment for 45 minutes: 253 machines, 1.7 TB storage, WebSphere, DB2, and the follow SAP applications…

Remember: The Bus
From WWW…
…to BBB
(Big Basic Bus)

Web Service Standards
The Ordering and Guiding Principles

- All standard efforts should foster...
  1. an environment which is
     a. heterogeneous,
     b. distributed,
     c. interoperable,
     d. loosely coupled,
     e. multi-vendor
  2. secure, reliable, recoverable interactions
  3. a single uniform usage model for “components”
  4. business criteria to discover components needed

“The Bus”!
(Service Bus aka Enterprise Service Bus aka ESB aka... ©)

Service Bus: Service-Supporting Platform

<table>
<thead>
<tr>
<th>Discovery, Negotiation, Agreement</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orchestration</td>
<td>Interface + Bindings</td>
</tr>
<tr>
<td>Protocols</td>
<td>XML</td>
</tr>
<tr>
<td>State</td>
<td>Policy</td>
</tr>
<tr>
<td>Atomic</td>
<td>Non-XML</td>
</tr>
<tr>
<td>Reliable Messaging</td>
<td>Transport</td>
</tr>
<tr>
<td>Security</td>
<td>Messaging</td>
</tr>
<tr>
<td>Transactions</td>
<td>Description</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quality of Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Composite</td>
</tr>
<tr>
<td>Atomic</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orchestration</td>
</tr>
<tr>
<td>Reliable Messaging</td>
</tr>
</tbody>
</table>

© Frank Leymann
The Bus And Standards

<table>
<thead>
<tr>
<th>Components</th>
<th>Quality of Service</th>
<th>Description</th>
<th>Messaging</th>
<th>Transport</th>
</tr>
</thead>
<tbody>
<tr>
<td>Composite</td>
<td>WS-C, WS-N*,...</td>
<td>WS-RF</td>
<td>WS-AT, WS-BA,...</td>
<td>WS-RF</td>
</tr>
<tr>
<td>WS-RM</td>
<td>WS-Security*</td>
<td>WS-Policy*</td>
<td>JMS, RMI/IIOP,...</td>
<td>HTTP, TCP/IP, SMTP, FTP, ...</td>
</tr>
<tr>
<td>WSDL*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOAP, WS-Addressing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The First Reaction…

😊 WOW, this seems to heavy!

😊 But wait, there is tremendous benefit!
Agenda

Virtualization Technology
  ➤ Web Services and the Grid

Two Styles
  ➤ REST vs. SOAP/WSDL

Business Impact
  ➤ SaaS, On Demand, Utilities

Business Processes
  ➤ How Business Drives IT

Summary

An Ongoing Discussion 😊 😊

REST-based SOA vs SOAP/WSDL-based SOA

- …this is a “religious war” 😞

- Let’s see what’s behind it 😊

- Then you can draw your own conclusion 😊
REST

REpresentational State Transfer (REST)

- It's the abstract architecture of the Web
- Goals:
  - Speed
  - Scalability
  - Simplicity
  - Data independence

REST Ingredients

- Resources identified by URI
  - Use parameters if needed
- Representations in an open-ended set of formats such as HTML, XML, CSS, RDF,…
  - Content negotiation
- Interaction via protocols that understand URIs and representations (HTTP, FTP,…)
- Actions to be performed are generic (GET, POST, PUT, DELETE)
REST Implications

- State incorporated in message returned to caller, not maintained on server
  - = “State Transfer”
  - → no server affinity → Scalability

- Large grained interactions
  - Large grained information retrieval to get resources to caller
  - = “Fixed Set of Generic Interactions”
  - → Simplicity of API

- Messages understood by all nodes
  - = “Enablement of Caching”
  - → Scalability

But Protocols may Change Along A Path

Can not rely on any quality of services provided by “the” transport protocol, because transport protocol changes along the message path! Need a mechanism to specify QoS to be enforced by changing transport protocols!
Payload is in HTTP(S) body. Body is either encrypted or in clear text. Even if intermediary needs only parts of the body, its sees all of it! ⇒ Encryption is only between sender and a specified URL.

Based on SOAP, all (explicit or implicit) intermediaries have only access to specified payload ⇒ Payload stays encrypted along the complete message path!
SOAP Message Architecture

SOAP Envelope

SOAP Header

Header Block H1

Header Block Hn

Main mechanism for compatibility of WS standards

Payload for plumbing/middleware

Payload for application

SOAP Body

Body Sub-element B1

Body Sub-element Bk

Intended for intermediaries

May change

May access

Intended for ultimate receiver

Body contains name of function message processor should perform

Body is passed to function

SOA-Style Service

POST /accountService
Host: 194.23.124.17

<soap:Envelope>
<soap:Header>

"Host" header points to machine location of the function to invoke

Relative URI contains name of the message processor to invoke

No query string

Body contains name of function message processor should perform

Message processor inspects body to determine function

Body is passed to function

Function inspects parameters to identify resource to manipulate
REST-Style vs. SOAP/WSDL-Style

- Most problems can be solved with either style
  - There’s no simple recommendation
  - But some problems are easier to solve with one over the other
- Fundamental difference:
  - SOAP/WSDL is about custom interfaces
  - REST is about generic interfaces
  - Sometimes this frames the problem nicely
- Asynchrony or reliability or robustness or multi-protocol lean to Web Services
- Query, large grained data blobs lean to REST

Agenda

Virtualization Technology
  ➤ Web Services and the Grid
Two Styles
  ➤ REST vs. SOAP/WSDL
Business Impact
  ➤ SaaS, On Demand, Utilities
Business Processes
  ➤ How Business Drives IT
Summary
What Happened First

- **Software as a Service (SaaS)**

≈≈ “Software deployed as a hosted service and accessed over the Internet”

- More precise definition needs to include
  - Scalability of the software
  - Availability of the software
  - Multi-tenant support of the software
  - Configurability/customizability of the software
    - Addressed later
...And Do Not Forget...

- Zillions of Web Services already provided by
  - Google
  - Amazon
  - eBay
  - RightNow
  - Nortel
  - Oracle
  - ...

What Happens Currently

IT Infrastructure as a Service

(IaaS) 😊
Amazon Simple Storage Service (Amazon S3)

Amazon S3 is storage for the Internet. It is designed to make web-scale computing easier for developers.

Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. It gives any developer access to the same highly scalable, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of web sites. The service aims to maximize benefits of scale and to pass these benefits on to developers.

Amazon S3 functionality

Amazon S3 is massively scalable with a minimal feature set.
- Objects, read, and delete objects containing from 3 bytes to 5 gigabytes of data each. The number of objects you can store is unlimited.
- Each object is stored and retrieved via a unique, developer-assigned key.
- Authentication mechanisms are provided to ensure that data is kept secure from unauthorized access. Objects can be made private or public, and rights can be granted to users and groups.
- Simple text-based SQL and SOAP interfaces designed to work with any Internet development tools.
- Both is flexible so that protocol or functional layers can easily be added. Default protocol is HTTP. A SimpleDB(TM) SQL-like interface is provided to keep costs for high-scale distribution. Additional interfaces will be added in the future.

Pricing

- Pay only for what you use. There is no minimum fee, and no set-up cost.
- 0.14 per GB of storage used.
- 0.23 per GB of data transferred.

Amazon EC2 (Amazon Elastic Compute Cloud)

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers.

Amazon Elastic Compute Cloud (Amazon EC2)

Amazon EC2 makes computing in the cloud. Amazon EC2’s simple web service interface allows you to obtain and configure computer processing capacity with minimal effort. It provides you with complete control of your computing resources, and lets you run your applications on Amazon’s proven computing environment. Amazon EC2 realizes the time requested to set up and boot your server instances to minutes, allowing you to quickly scale your capacity, build up and down, and deploy your computing requirements. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use.

Amazon EC2 Functionality

Amazon EC2 provides a large virtual computing environment, allowing you to use web server instances of equivalent performance for only a few cents an hour. With complete control over your computing resources, you can manage your network access permissions, and run your image using as many or few systems as you need.

To use Amazon EC2, you simply:

- Create an Amazon Machine Image (AMI) containing your applications, libraries, data and associated configuration settings. Or use our preconfigured, template images to get started.
- Upload the AMI into Amazon S3. Amazon EC2 provides tools that make storing the AMI simple. Amazon EC2 provides a safe, reliable and fast repository to store your images.
- Use Amazon EC2 web service to configure security and network access.
- Use Amazon EC2 web service to start, terminate, and monitor as many instances as your AMI allows.
- Pay for the instance hours and bandwidth that you actually consume.
Amazon Simple Queues (Amazon SQS) offers a reliable, highly scalable service for storing and retrieving messages in a queue. Messages are stored on a queue and can be retrieved by any component in the Internet. With Amazon SQS, you can build applications that interact with external systems or trigger on-demand computing resources in a reliable, scalable way.

Amazon SQS is a fully managed service designed to enable you to create and manage message-oriented middleware. Amazon SQS handles all the details of reliably delivering messages to components, allowing you to focus on your applications.

Amazon SQS functionality:
- Developers can create and manage message-oriented middleware services.
- Messages are stored in queues and can be retrieved by any component in the Internet.
- Amazon SQS is used in various applications, such as e-commerce, social networking, and finance.

Microsoft’s Cloud Operating System – Windows Live Core

The Windows Live Core is the service that provides the back-end infrastructure for Windows Live services, including Hotmail, Live.com, and Windows Live Messenger. It is responsible for managing the infrastructure that supports these services and ensuring they are available to users.

Benefits of using Windows Live Core:
- Scalability: Windows Live Core is designed to handle large volumes of traffic efficiently.
- Reliability: It ensures that services are available 24/7.
- Integration: Components can communicate with each other to provide a seamless user experience.

The Windows Live Core platform provides a set of services that can be accessed through the Internet.

Microsoft’s Cloud Platform – Windows Live Core

Windows Live Core is integrated with the rest of Microsoft’s Cloud Platform, providing a comprehensive set of services for developers.

Features of Windows Live Core:
- Cloud services: It provides a range of cloud services, including storage, computing, and data analysis.
- Integration with other services: It integrates with other Microsoft services, such as Windows Azure and Office 365.
- Security: It offers robust security features to protect user data.

Windows Live Core is the backbone of Microsoft’s Cloud Platform, providing a secure and scalable environment for developers to build and deploy their applications.
From BBB…
…to CCC
(Cool Components Cloud)
The Long Tail… and How to Catch it

- If cost of reaching customer is reduced, more customers will buy a product
- This is how even niche markets get attractive
- This is a key aspect of the business model for SaaS from a provider point of view

Terminology

- Outsourcing
  Allocation of IT orders to external companies (e.g. running of [parts of] IT infrastructure including application functionality, development of application functions)

- Off-Shoring
  Special case of outsourcing: external company is in foreign country
Reasons For Outsourcing

- They are manifold
- They and their priority change(d) over time
- Cost-savings is mostly ranked high
  - Maintenance of infrastructure and software
    - Purchase cost often not so critical
    - Keeping it current is expensive
  - Skilled IT personnel is rare
    - Operators, systems management personnel
    - Complexity of IT is constantly raising → Training (costly) to avoid (costly) outages,…
    - Programmers
- This is a key aspect of the business model for SaaS from a customer point of view

What’s Interesting? Solutions!

- Business Process
- Middleware
  - Operating System
  - DBMS
- Components
- Hardware

© Frank Leymann
Enterprise SaaS Requires Customization

"Points-of-Variability: Application Logic"

Customizing Application Logic
Environment Element

- ...represents element of hosting environment
- Selected based on specifying SLAs
- SLAs for application result in tree of environment elements
- Application SLAs determine required environment

```
WebSphere
  └── 1.5TB
      ├── DB2
      └── zOS
        └── 7×24
```

Customizing Application Environment

```
Environment Customization Tool
I need an AppServer 7×24!
```

```
AppServer Vendor1

AppServer Vendor2
```

© Frank Leymann
Result of Customization

Provisioning in a Nutshell
…and the Result is: Computing On Demand

In 3 hours I need the following environment for 45 minutes: 253 machines, 1.7 TB storage, WebSphere, DB2, and the follow SAP applications…

Agenda

Virtualization Technology
  ➔ Web Services and the Grid

Two Styles
  ➔ REST vs. SOAP/WSDL

Business Impact
  ➔ SaaS, On Demand, Utilities

Business Processes
  ➔ How Business Drives IT

Summary
What BPEL Brings to the Table

**BPEL Value: First Glance**

- Single business process language for IT-users
  - Single runtime model
  - Single IT-level tooling
  - Single target model all business-level tools can map too
- Reuse of IT-level process models across tools and process engines

- Investment protection
- Reduced total cost of ownership: Simplification of overall infrastructure
What BPEL Brings to the Table

Did I Say “Table”? (pt1)

“BPEL is for process/workflow technology what SQL is for relational databases!” (pt2)

A Typical Situation

Often, you need more than one service to achieve your goal!

Which ones to use? How to use them? In which order? ...

© Frank Leymann
BPEL Prescribes Proper Use of Web Services

Wow! This is easy

...and Externalizes Processes as Web Services
BPEL in SOA: Abstract View

"BPEL is a recursive aggregation model for Web services"

- **Aggregation**: A set of Web services can be tied into one or more new Web service by means of a business process model.

- **Recursive**: These new Web services can again be tied into other new Web services.

BPEL in a Nutshell: Elevator Speech

- **BPEL describe in a SOA how your company performs its business processes**

- **With BPEL, it is straightforward to let your business partners and customers directly participate in your business processes**

- **With BPEL, it is straightforward to tie in Web services as activities of your business processes**
The Notion Of “Orchestration”

Application Structure
The Need for Deployment and Discovery

<table>
<thead>
<tr>
<th>Process</th>
<th>Deployment Descriptors</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Activity: Port Type, Operation, Semantics, QoS, ...</td>
</tr>
</tbody>
</table>

Service Bus

The Cloud

Programming Model

- **Programming in the Large**
  - Processes

- **Deployment**
  - Service Properties
    - Port Link Locator

- **Programming in the Small**
  - (Web) Services

- Application

  - Workflow System
  - Service Bus
  - Application Server
Thus, Business Drives IT

Agenda

Virtualization Technology
➤ Web Services and the Grid

Two Styles
➤ REST vs. SOAP/WSDL

Business Impact
➤ SaaS, On Demand, Utilities

Business Processes
➤ How Business Drives IT

Summary
Conclusion

- SOA is all about virtualization
  - REST vs SOAP/SOA is not an issue
- IT components are virtualized via SOA & Web Service middleware: They become subject to dynamic out-/insourcing
- Virtualization and provisioning enable Utility computing (provider’s view) and On Demand computing (user’s view)
- Utility/OnDemand enables new software model: Software as a Service
- Business Processes become the driver for IT, significantly enhancing the speed of reacting to new business needs (“Agile Enterprise”)
- IT might end up in the cloud 😊

The End!

Thanks
For Your Attention...